Data-Efficient Generative Models For Drug Discovery and Protein Design



Leo Jannis Klarner

Brasenose College

University of Oxford

A Dissertation Submitted for the Degree of

Doctor of Philosophy

Hilary Term 2025

Declaration

This dissertation and all research contained in it are a product of my original work, except where indicated otherwise by explicit statement or reference. All ideas, quotations, and data originating from the works of others, published or otherwise, are fully acknowledged according to standard practices of the academic discipline.

Acknowledgments

Over the course of my DPhil, I have benefited from the support of many people. First and foremost, I would like to thank all those whose supervision and guidance I have received over the years. I am profoundly grateful to my supervisors—Professors Yee Whye Teh, Charlotte Deane and Garrett Morris—whose consistent support and constructive feedback have been instrumental in shaping my scientific thinking and professional growth. I would also like to thank my industrial supervisors, Torsten Schindler and Michael Reutlinger, whose advice and perspectives I have benefited from greatly. Furthermore, I would like to thank the University of Oxford's Clarendon Fund, Brasenose College and F. Hoffmann-La Roche for the financial support that has made this research possible.

I feel very lucky to have been part of a stimulating, vibrant and fun research environment and would like to extend my gratitude to all the brilliant members of the Oxford Protein Informatics Group (OPIG) and the Oxford Computational Statistics and Machine Learning Group (OxCSML) who have made my DPhil a thoroughly enjoyable experience. In particular, I would like to thank Sarah Robinson, Tobias Olsen, Brennan Abanades Kenyon, Lucy Vost, Guy Durant, Nele Quast, Nic Fishman, Michael Hutchinson, Valentin De Bortoli, Emile Mathieu, Guneet Singh Dhillon, Tim Rudner, Lino Ferreira, Carlos Outeiral Rubiera and Anum Fatima for being exceptional collaborators, officemates and friends.

Finally, I would like to express my never-ending gratitude to my parents and my sister, who have always encouraged and supported me in pursuing what I am truly passionate about.

Abstract

Developing novel and more effective therapeutics is one of the most challenging and resource-intensive endeavours in biomedical research. Despite significant advances in our understanding of biological systems and disease mechanisms, the productivity of drug discovery efforts has been declining steadily. Recent years have witnessed the resurgence of deep learning algorithms as the state-of-the-art across a range of complex modelling domains, positioning them as a powerful means of accelerating key bottlenecks throughout the drug discovery pipeline. However, their successful application to molecular design faces several fundamental challenges: data is scarce, search spaces are large, and medicinal chemists often care most about novel compounds that are meaningfully different from those that have been explored in the past.

This thesis aims to improve the performance of deep learning algorithms in this setting, presenting several methodological contributions designed to make predictive and generative models more reliable and robust in low-data, out-of-distribution regimes. We begin by focusing on classifier-guided diffusion in Chapter 3, introducing a tractable and easy-to-optimise regularisation term that improves conditional sampling and enables the more reliable generation of novel molecules with desirable functional properties. Chapter 4 considers constrained diffusion, presenting two generative modelling frameworks that facilitate the direct integration of geometric and physical constraints into standard Euclidean and Riemannian diffusion processes, resulting in samples that are guaranteed to satisfy relevant feasibility and safety criteria. Chapter 5 builds on this work to develop a novel Metropolis-based discretisation scheme, resulting in significantly faster sampling speeds, improved empirical performance and the ability to handle arbitrary non-convex constraints. Finally, Chapter 6 investigates the out-of-distribution generalisation of molecular property prediction models, introducing a semi-supervised probabilistic framework that is able to leverage relevant unlabeled data to improve predictive performance in a number of challenging evaluation settings.

Collectively, these research efforts demonstrate that integrating relevant prior knowledge and constraints into the modelling process can lead to more accurate and data-efficient deep learning methods that have the potential to accelerate key steps in the discovery of safer and more effective therapeutics. Concluding thoughts and promising directions for future work are presented in Chapter 7.

Table of Contents

1 Introduction 1

2 Background 7

2.1 Drug Discovery and Development 7

2.2 Predictive Modelling 10

2.3 Generative Modelling 19

3 Context-Guided Diffusion 29

4 Diffusion Models for Constrained Domains 47

5 Metropolis Sampling for Constrained Diffusion 67

6 Drug Discovery under Covariate Shift 83

7 Discussion and Conclusion 99

Supplementary Information 119

A Supplementary to *Context-Guided Diffusion* 121

B Supplementary to *Diffusion Models for Constrained Domains*. . . . 147

C Supplementary to *Metropolis Sampling for Constrained Diffusion* . 167

D Supplementary to *Drug Discovery under Covariate Shift* 191

x

Chapter 1

Introduction

Deep learning

The past fifteen years have witnessed the resurgence of *deep learning* algorithms as the indisputable state-of-the-art across a range of complex modelling problems. Prominent milestones include the success of deep convolutional networks in the ImageNet competition (Krizhevsky et al., 2012), the superhuman performance of reinforcement learning agents in the games of Atari (Mnih et al., 2015) and Go (Silver et al., 2016), as well as the emergence of transformer-based image generation (Saharia et al., 2022; Betker et al., 2023) and large language models (Brown et al., 2020; Ouyang et al., 2022) that are able to perform complex, multimodal reasoning over images, video and text (OpenAI et al., 2023; Gemini Team et al., 2024).

However, the impact of these advances extends beyond the traditional machine learning domains of natural language processing and computer vision, catalysing breakthrough progress across the natural and engineering sciences (Wang et al., 2023). For instance, modern deep learning methods have enabled step-change advances in modelling three-dimensional protein structures (Jumper et al., 2021) and biomolecular complexes (Abramson et al., 2024), predicting the effects of genetic variants (Cheng et al., 2023), formulating novel algorithmic approaches (Fawzi et al., 2022) and mathematical proofs (Trinh et al., 2024), and controlling plasma instabilities in fusion reactors (Degrave et al., 2022), to name a few.

Deep neural networks

Backpropagation

Universal approximation theorem

At their core, *deep neural networks* consist of multiple layers of parametric transformations, each combining learnable affine operations with non-linear activation functions to successively transform high-dimensional input data into increasingly abstract representations (LeCun et al., 2015). The parameters of these transformations are typically learned through stochastic gradient-based optimisation of a task-specific loss function, with gradients computed efficiently via the *backpropagation* algorithm (Rumelhart et al., 1986). This setup allows deep neural networks to automatically learn hierarchical features from data, obviating the need for manual feature engineering that characterised traditional machine learning methods (Bengio et al., 2013). The theoretical foundations for this representational power stem from the *universal approximation theorem* (Cybenko, 1989), which states that even a single hidden layer with suitable nonlinear activation functions can approximate any continuous function to arbitrary precision, with deeper architectures making this process more efficient (Telgarsky, 2016).

While the fundamental principles of deep neural networks have been established for decades, their remarkable progress in recent years has been sustained by a positive feedback loop of several favourable factors, including (i) the development of more flexible model architectures and efficient training regimes, (ii) the exponential growth of available training data (Sun et al., 2017), and (iii) the advent of specialised, cost-effective computing hardware, particularly graphics and tensor processing units (Epoch AI, 2024). This has catalysed a paradigm shift in model development. Driven by the discovery that performance scales predictably with increasing model size, compute, and training data (Kaplan et al., 2020), the field has witnessed the emergence of foundation models (Bommasani et al., 2021) that achieve breakthrough performance in data-rich domains like language and vision. This aligns with an important empirical observation in machine learning research: that scalable methods leveraging computation and data consistently outperform specialised approaches based on human knowledge engineering (Sutton, 2019).

Deep Learning for Early-Stage Drug Discovery and Protein Design

Small molecule drug discovery and protein design

Unfortunately, this data-driven scaling paradigm faces unique challenges in the context of *small molecule drug discovery and protein design*. The central objective in these domains is to identify novel compounds with desirable functional properties, e.g. high target affinity and favourable pharmacokinetics for small molecules, or enhanced catalytic activity and stability for proteins. The immense size of the underlying search spaces—up to 10^{60} for drug-like small molecules (Bohacek et al., 1996) and 20^N for N-length protein sequences (Maynard Smith, 1970)—renders this a challenging combinatorial optimisation problem. Considering the substantial cost of synthesising and validating candidate compounds experimentally (Scannell et al., 2012; DiMasi et al., 2016; Wouters et al., 2020), the question of how to efficiently navigate these search spaces to locate high-value subsets lies at the core of modern molecular design (Gómez-Bombarelli et al., 2018; Sanchez-Lengeling, Aspuru-Guzik, 2018; Bilodeau et al., 2022; Notin et al., 2024).

Deep learning approaches in this domain generally fall into two broad categories: *predictive models* that aim to learn parametric mappings from a set of candidate compounds $\mathbf{x} \in \mathcal{X}$ to their properties $\mathbf{y} \in \mathcal{Y}$, and *generative models* that attempt to sample novel structures or sequences $\mathbf{x} \in \mathcal{X}$ from scratch, usually conditioned on a set of desired characteristics $\mathbf{y} \in \mathcal{Y}$.

predictive models

From a probabilistic perspective, predictive models aim to learn the conditional distribution $p_\theta(\mathbf{y}|\mathbf{X} = \mathbf{x})$. They can either take the form of classification models, where $\mathbf{y} \in [0, 1]^M$ is categorical (e.g., predicting whether a compound is active or inactive against a set of M target proteins), or regression models, where $\mathbf{y} \in \mathbb{R}^M$ is continuous (e.g., predicting binding affinities or solubility values). In both cases, the output space is typically low-dimensional and well-behaved, making the learning problem relatively tractable. Depending on the application domain and choice of input representation, popular approaches include multi-layer perceptrons (Rosenblatt, 1958), convolutional (LeCun et al., 1989), recurrent (Hochreiter, Schmidhuber, 1997), and graph neural networks (Kipf, Welling, 2016), as well as transformers (Vaswani et al., 2017). These models can be deployed either as individual predictors or as components in larger screening pipelines, often running in parallel to evaluate thousands of candidates efficiently (Gentile et al., 2022).

generative models

In contrast to predictive models, generative models must learn to sample from a high-dimensional density $p_\theta(\mathbf{x}|\mathbf{Y} = \mathbf{y})$. This inverse problem is inherently more challenging, as it requires the model to learn both the complex distribution of valid molecular structures and the associated physicochemical constraints, as well as the information necessary to successfully condition the sampling process on a given set of properties $\mathbf{y}$. Important methods for deep generative modelling include variational autoencoders (Kingma, Welling, 2013; Rezende et al., 2014), generative adversarial networks (Goodfellow et al., 2020), discrete (Tabak, Vanden-Eijnden, 2010; Tabak, Turner, 2013; Rezende, Mohamed, 2015) and continuous (Chen et al., 2018) normalising flows, and, most recently, *denoising diffusion models* (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020). The latter have shown particular promise for small molecules and proteins (Jo et al., 2022; Campbell et al., 2024; Gruver et al., 2024; Huang et al., 2024) and will serve as the default modelling framework for the work presented in this thesis.

Denoising diffusion models

Data Scarcity and Out-of-Distribution Generalisation

Both predictive and generative models fundamentally rely on training data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ in the form of compound-property or sequence-property pairs $(\mathbf{x}_i, \mathbf{y}_i) \sim p$, sampled from the underlying data-generating distribution p. Unlike language and vision tasks, where generating labels takes only seconds of human effort, characterising the properties of a single drug candidate requires extensive laboratory testing that can span weeks or months. This process typically involves chemical synthesis, purification, multiple rounds of assays, and detailed analysis—often costing thousands of dollars per compound. Failed synthesis attempts, short half-lives, and inconclusive or conflicting experimental readouts can further exacerbate these challenges. The resulting scarcity of labelled data is particularly limiting when considering the large dimensionality of the underlying search spaces, where the *curse of dimensionality* (Bellman, 1958) ensures that even relatively large datasets can only cover a fraction of feasible compounds. This inherent label sparsity represents one of the key challenges when attempting to adapt existing deep learning algorithms to drug discovery applications (Durant et al., 2024).

Curse of dimensionality

This issue is compounded by the fact that, in many practical settings, researchers are primarily interested in exploring novel regions of chemical or protein sequence space that are meaningfully different from those that have been studied in the past (Reymond et al., 2010; Scannell et al., 2012). Success in such *extrapolative regimes* is only possible by understanding the fundamental physicochemical principles that drive all biomolecular interactions—e.g., protein-ligand binding energetics, allosteric interactions, membrane permeability and metabolic stability (Silverman, Holladay, 2014). The remarkable representational capacity of deep neural networks offers a promising avenue for solving this challenge: through their ability to learn complex, hierarchical representations from multiple data modalities, deep learning algorithms could, in principle, discover and encode fundamental relationships that remain invariant across much of their input space, thereby facilitating more accurate design decisions that conform to an established mechanistic understanding of biomolecular interactions (Jiménez-Luna et al., 2020).

Extrapolative regimes

Realising this potential, however, requires overcoming fundamental limitations in current approaches. While neural networks excel at pattern recognition within their training distribution, they lack explicit mechanisms to distinguish physically meaningful solutions from those that overfit to *superficial correlations* (Geirhos et al., 2020). Standard training procedures provide no explicit mechanism for favouring the former over the latter, as they optimise solely for statistical correlation between inputs and outputs without regard to the domain they are operating in. This limitation becomes all the more critical in data-scarce settings, in which limited experimental data can make models particularly prone to *overfitting and memorisation* (Wallach, Heifets, 2018). Consequently, model performance often degrades significantly when they are applied to meaningfully novel molecules and proteins (Buttenschoen et al., 2024; Durant et al., 2025), i.e., precisely when accuracy and reliability are needed the most for a successful drug discovery campaign.

Superficial correlations

Overfitting and memorisation

Rigorous validation protocols are therefore essential to assess whether models are learning to capture the fundamental biophysical principles that govern molecular interactions. While standard i.i.d. *random splits* remain the norm in many traditional machine learning domains, they have been shown to provide overly optimistic performance estimates on many molecular machine learning tasks by allowing models to exploit shared patterns between near-identical compounds from conserved chemical series (Wallach, Heifets, 2018). More stringent validation approaches are needed to evaluate a model's true predictive power: for small molecules, *clustering-based splits* that ensure chemical diversity between training and test sets (Mayr et al., 2018; Durairaj et al., 2024; Corso et al., 2024) provide a more realistic evaluation setup that tests generalisation to novel chemical matter. Likewise, sequence similarity splits that enforce a maximum allowed sequence identity between training and test sets more accurately reflect generalisation capabilities in various protein design tasks (Chinery et al., 2024; Kovtun et al., 2024).

Random splits

Clustering-based splits

Beyond measuring aggregate performance on held-out test sets, the ability to quantify *predictive uncertainty* becomes particularly important when models encounter molecules that differ substantially from their training distribution. While deep learning models can be trained to provide uncertainty estimates alongside their predictions, these estimates are often poorly calibrated—especially for out-of-distribution compounds where errors are likely to be largest (Ovadia et al., 2019; Koh et al., 2021). This *miscalibration* can have serious consequences in drug discovery applications, where unreliable confidence estimates might lead to the costly experimental validation of false positives or the oversight of promising candidates incorrectly flagged as low-confidence predictions. While recent advances have shown promise in improving uncertainty quantification (Gal, Ghahramani, 2016; Lakshminarayanan et al., 2017; Liu et al., 2020; Van Amersfoort et al., 2020), achieving consistently well-calibrated uncertainties across diverse regions of chemical and protein sequence space remains a key challenge for the practical deployment of deep learning models in real-world drug discovery pipelines.

Predictive uncertainty

Miscalibration

Structure of This Thesis

Motivated by the considerations above, the overarching aim of this thesis is to advance the development of more reliable machine learning tools for early-stage drug discovery. While deep neural networks excel at extracting patterns from large datasets, the scarcity of experimental data in many drug discovery settings means that the learnt functions often reflect superficial correlations rather than fundamental physicochemical principles. Through several complementary contributions, we demonstrate that incorporating domain-specific prior knowledge and constraints into these algorithms can improve their robustness and ability to generalise to novel regions of chemical and protein sequence space. This thesis adds to a growing body of research showing that carefully constraining the hypothesis space of deep neural networks—whether through architectural choices, training objectives, or sampling procedures—can help bridge the gap between purely data-driven approaches and a mechanistic understanding of the underlying domain. The rest of this thesis is structured as follows:

Chapter 2 outlines the foundations of machine learning for molecular design, covering fundamental topics in drug discovery and predictive modelling before progressing to deep generative models, with a particular focus on diffusion models and their applications to small molecules and proteins.

Chapter 3 presents context-guided diffusion, a method for improving the generalisation capabilities of conditional diffusion models for molecular design tasks. By incorporating relevant domain knowledge through a regularisation framework that encourages calibrated uncertainty and smooth gradients in unexplored regions, this approach enables more reliable generation of novel compounds with desirable properties. The effectiveness of this approach is demonstrated through a comprehensive evaluation across multiple molecular design tasks, including small molecules, materials and proteins, showing substantial improvements in out-of-distribution performance while maintaining the computational efficiency of standard diffusion models. This work was originally published as:

L. Klarner, T. G. Rudner, G. M. Morris, C. M. Deane, Y. W. Teh. Context-guided diffusion for out-of-distribution molecular and protein design. *International Conference on Machine Learning*, 2024.

Chapter 4 presents two principled mathematical frameworks for incorporating geometric constraints into diffusion models based on log-barrier methods and the reflected Brownian motion. The key methodological contribution is the development of rigorous time-reversal formulas for constrained diffusion processes, which enable the generation of samples that are guaranteed to satisfy physical and geometric constraints. Through applications to protein loop modelling and robotic motion planning, this work demonstrates how incorporating fundamental geometric principles into generative models can improve both the physical validity and diversity of generated samples. This work was originally published as:

N. Fishman, **L. Klarner**, V. De Bortoli, E. Mathieu, M. Hutchinson. Diffusion Models for Constrained Domains. *Trans. on Machine Learning Research, 2023.*

Chapter 5 builds on these methods to present a computationally efficient framework for sampling from constrained diffusion models with a novel, Metropolis-based discretisation scheme. The key methodological contribution is the development of a lightweight approximation to the reflected Brownian motion from Chapter 4 that achieves an order-of-magnitude speed-up over existing approaches while maintaining theoretical constraint satisfaction guarantees. This approach enables diffusion models to effectively handle both convex and non-convex constraints, demonstrating strong performance in the domains of robotics, protein design, and geospatial modelling. This work was originally published as:

N. Fishman, **L. Klarner**, E. Mathieu, M. Hutchinson, V. De Bortoli. Metropolis Sampling for Constrained Diffusion Models. *Advances in Neural Information Processing Systems*, 2023.

Chapter 6 focuses on molecular property prediction and presents a semi-supervised probabilistic framework that leverages unlabelled data to address the challenge of discriminative modelling under covariate shift. By encoding relevant knowledge of the modelled domain through carefully constructed function-space priors, this approach enables neural networks to make more reliable predictions for novel compounds that differ meaningfully from their training distribution. The effectiveness of this approach is demonstrated through a comprehensive evaluation on high-quality datasets, showing substantial improvements in both predictive accuracy and calibration compared to existing approaches, particularly in challenging extrapolative settings. This work was originally published as:

L. Klarner, T. G. Rudner, M. Reutlinger, T. Schindler, G. M. Morris, C. M. Deane, Y. W. Teh. Drug discovery under covariate shift with domain-informed prior distributions over functions. *International Conference on Machine Learning, 2023*.

Chapter 7 synthesises the key insights and common themes that emerge across these methodological contributions, highlighting both the progress towards developing more reliable machine learning methods for molecular design, as well as important limitations and opportunities for future research in data-scarce settings.

Additional Publications

In addition to the research efforts outlined above, I have been involved in a number of other projects that have resulted in the following publications:

R.-R. Griffiths, **L. Klarner**, H. Moss, A. Ravuri, S. Truong, Y. Du, S. Stanton, G. Tom, B. Rankovic, A. Jamasb, et al. GAUCHE: a library for Gaussian processes in chemistry. *Advances in Neural Information Processing Systems*, 2024.

L. Klarner, M. Reutlinger, T. Schindler, C. M. Deane, G. M. Morris. Bias in the Benchmark: Systematic experimental errors in bioactivity databases confound multi-task and meta-learning algorithms. *ICML AI for Science Workshop*, 2022.

Chapter 2

Background

2.1 Drug Discovery and Development

Developing novel and more effective therapeutics is one of the most challenging and resource-intensive endeavours in biomedical research. Despite significant advances in our understanding of biological systems and disease mechanisms, the productivity of drug discovery efforts has been declining steadily, with inflation-adjusted R&D spending per approved drug doubling approximately every 9 years (Scannell et al., 2012; DiMasi et al., 2016). This decrease in efficiency is particularly concerning given the extraordinary cost of the drug development process: a single therapeutic typically takes more than a decade to develop (Taylor, 2015), requires hundreds of millions to billions of dollars in R&D spending (Wouters et al., 2020), and faces attrition rates of up to 95% (Wong et al., 2019). A key contributing factor is the sheer difficulty of the underlying search problem. The universe of potential small-molecule drugs exceeds 10^{60} (Bohacek et al., 1996), more than 99.9% of which have never been synthesised (Reymond, Awale, 2012). Similarly, even a moderately-sized protein therapeutic of 100 amino acids has 20^{100} possible sequence variations (Maynard Smith, 1970). Researchers can only explore a tiny fraction of this search space, yet they must identify compounds that simultaneously satisfy multiple competing objectives: potent target engagement, favourable pharmacokinetic properties, minimal off-target effects, and appropriate safety profiles. This multi-dimensional optimisation problem is further complicated by our incomplete understanding of biological systems and the many unknown factors that influence a compound's behaviour in different physiological environments.

High-throughput screening

The emergence of *high-throughput screening* and multi-omics techniques (Macarron et al., 2011; Hasin et al., 2017; Karczewski, Snyder, 2018) has generated unprecedented volumes of biological and chemical data, which have created new opportunities for computational approaches to address key bottlenecks in the drug discovery pipeline (Zhang et al., 2017; Vamathevan et al., 2019). This section will provide a brief overview of the main stages of this process, followed by an in-depth overview of predictive and generative deep learning algorithms and their applications to molecular modelling and design.

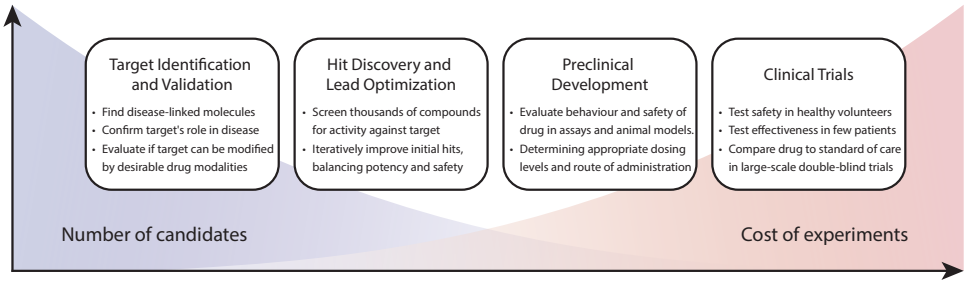


Figure 2.1. Schematic illustration of key stages in the modern drug discovery pipeline, showing the progression from target identification to clinical trials.

The Modern Drug Discovery Pipeline

Human physiology is governed by the interactions of biological macromolecules, such as nucleic acids and proteins, and their various regulators and metabolites. Disruptions in this complex system can lead to diseased states, while therapeutic interventions aim to restore normal physiological function. Modern drug discovery has evolved into a systematic, multidisciplinary approach that typically involves four main stages: (i) target identification and validation, (ii) hit discovery and lead optimisation, (iii) preclinical development, and (iv) clinical trials (Figure 2.1). Each stage presents distinct challenges in extracting meaningful signals from noisy and complex biological systems that shape both traditional approaches and emerging computational methods.

Target identification and validation

The foundation of a drug discovery campaign lies in identifying and validating biological macromolecules that play causal roles in disease pathogenesis. To this end, researchers rely on a range of mechanistic and data-driven techniques (Gibbs, 2000; Lindsay, 2003; Cong et al., 2012; Schenone et al., 2013). Genome-wide association studies, for instance, help identify genetic markers enriched in diseased subpopulations, while high-throughput transcriptomic and proteomic assays reveal altered expression patterns in disease-relevant tissues. These approaches generate rich but often heterogeneous datasets that require sophisticated analytical methods to interpret. Once a potential target has been identified, functional *in-vitro* and *in-vivo* assays are used to validate its role in the disease process. These include genetic manipulations such as gene knockdowns or overexpression, pharmacological perturbations with chemical probes, and phenotypic screens that assess the impact of target modulation on disease-relevant endpoints. Finally, researchers assess target druggability, i.e., whether the target can be effectively perturbed with a given therapeutic modality, which requires integrating structural information, cellular localisation data, and knowledge of potential binding sites.

Hit identification

Once a suitable target has been found, researchers turn to the challenge of discovering compounds that can modulate it effectively. During *hit identification*, large combinatorial libraries of drug-like molecules are tested in high-throughput screens to identify those that produce a desired response in a biological assay designed to mimic the disease pathology (Bleicher et al., 2003; Mayr, Bojanic, 2009). Alternative strategies include fragment-based and structure-based drug design, which leverage structural information to guide the search process more efficiently (Blundell, 1996; Murray, Rees, 2009).

Lead optimisation

The subsequent *lead optimisation* phase presents perhaps the most challenging multi-parameter optimisation problem in the entire pipeline (Kenakin, 2003; Jorgensen, 2009; Keserü, Makara, 2009). Medicinal chemists investigate analogues of initial hits through iterative cycles of design, synthesis, and testing. This process requires balancing multiple objectives, such as optimising binding affinity while maintaining suitable absorption, distribution, metabolism, excretion, and toxicity (ADMET) profiles. The complexity of this optimisation landscape, combined with the high cost of experimental validation, creates a natural opportunity for computational methods that can guide the exploration of chemical space and prioritise the most promising candidates for synthesis and testing (Koscher et al., 2023).

Preclinical development

Lead compounds that emerge from the optimisation process undergo extensive *preclinical development* to assess their safety and efficacy profiles in animal models (Rogge, Taft, 2010). This stage aims to characterise *in-vivo* pharmacokinetics, determine appropriate dosing regimens, and identify potential toxicity concerns. The translation of results from *in-vitro* assays and animal models to human outcomes represents a persistent challenge, with many candidates failing in clinical trials despite promising preclinical results. The final and most critical stage involves testing drug candidates in human participants through increasingly comprehensive *clinical trials* (Friedman et al., 2015). Phase I trials in healthy volunteers establish safety, tolerability, and pharmacokinetics while phase II trials test the candidate drug in patients with the targeted disease to evaluate its efficacy and determine optimal dosing. Finally, phase III trials involve large populations and provide definitive evidence of safety and efficacy compared to the standard of care.

Clinical trials

The potential benefits of integrating deep learning methods into various stages of the drug discovery pipeline are immense, offering the opportunity to accelerate the development of safe and effective therapies for a wide range of diseases. However, the successful implementation of these approaches is not without challenges. Access to large, high-quality datasets is the exception rather than the norm, and the limited interpretability and generalisability of current approaches remains a significant hurdle (Vamathevan et al., 2019; Schneider et al., 2020; Buttenschoen et al., 2024; Durant et al., 2024). Overcoming these obstacles will require the close collaboration of machine learning scientists, medicinal chemists and biologists to develop more robust methods and validate their predictions in real-world settings. The following sections will explore two key areas where machine learning is poised to make a significant impact: predictive modelling for hit identification and lead optimisation, and generative modelling for de-novo design.

2.2 Predictive Modelling

Modern drug discovery increasingly leverages computational methods to predict molecular properties, enabling researchers to prioritise promising candidate compounds for experimental validation. This approach is built on the principle that similar molecular structures exhibit comparable biological effects—effectively learning implicit approximations of the underlying physicochemical processes that govern biomolecular interactions. The practical success of these methods largely depends on their ability to discover patterns that generalise from limited experimental data to new chemical entities, since scientists are often most interested in novel compounds that are meaningfully different from those in the historical datasets used for model training. This section examines the mathematical foundations of predictive modelling in drug discovery and outlines widely used molecular representations, neural network architectures, and evaluation frameworks.

Fundamentals

Predictive models in drug discovery aim to learn meaningful patterns from molecular and biological data to estimate properties of interest. This fundamental task can be formalised through the mathematical framework of statistical learning theory (Vapnik, 1999; Vapnik, 2013), which provides a rigorous foundation for understanding how we can fit parametric models to data in order to make predictions about new, unseen molecules. This learning problem involves constructing a mapping $f \in \mathcal{F}$ from an input space of molecular or sequence representations $\mathcal{X}$ to an output space of chemical or biological properties $\mathcal{Y}$. This relationship is governed by an unknown probability distribution $p(\mathbf{x}, \mathbf{y})$, from which we observe only a finite sample of training data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$. Our goal is to find a function $f : \mathcal{X} \to \mathcal{Y}$ that minimises the expected prediction error

$$\mathcal{R}(f) = \mathbb{E}_{p(\mathbf{x},\mathbf{y})} \left[\mathcal{L}(f(\mathbf{x}), \mathbf{y}) \right] \tag{2.1}$$

defined by a task-specific loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathcal{R}$. Since we are not able to access the true data-generating distribution directly, we must instead work with the empirical risk calculated over the finite training set $\mathcal{D}$

$$\hat{\mathcal{R}}(f) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L} \left(f(\mathbf{x}_i), \mathbf{y}_i \right). \tag{2.2}$$

In standard statistical learning theory, generalisation is characterised by the gap $\epsilon = |\mathcal{R}(f) - \hat{\mathcal{R}}(f)|$ between true and empirical risk, with bounds that depend on the complexity of the function class $\mathcal{F}$ and sample size N. However, these theoretical guarantees are based on the critical assumption that training and test data are drawn from the same marginal distribution $p(\mathbf{x})$. This assumption is often at odds with the practical realities of drug discovery. For one, the vastness of the underlying search spaces means that the experimental data in $\mathcal{D}$ can only cover a small subset $\mathcal{X}' \subset \mathcal{X}$ of the input domain, where $|\mathcal{X}'| \ll |\mathcal{X}|$. In addition, this subset is often not sampled uniformly from $\mathcal{X}$, but instead follows a biased distribution $\tilde{p}(\mathbf{x}) \neq p(\mathbf{x})$ shaped by historical research trends, compound availability constraints, and the personal preferences of medicinal chemists.

This sampling bias has profound implications for model development and evaluation. Since researchers are primarily interested in predictions for novel compounds that differ from those that have been studied in the past—i.e., that have low support under $\tilde{p}(\mathbf{X})$—a key requirement is that models need to perform well in out-of-distribution settings, where the empirical risk $\hat{\mathcal{R}}(f)$ becomes an increasingly unreliable proxy for the true risk $\mathcal{R}(f)$. That is, two models with similar empirical risk might make radically different predictions on novel compounds, and standard empirical risk minimisation provides no clear mechanism of distinguishing a model that has captured fundamental physicochemical principles—and is thus expected to perform well across the broader input domain—from a model that merely reflects superficial patterns in the training data.

The challenge of generalising from a biased sample $\mathcal{D}$ to the broader input domain $\mathcal{X}$ motivates a formal distinction between two fundamental learning paradigms: *inductive* and *transductive* inference (Vapnik, 2013). The majority of predictive modelling tasks follow the inductive learning framework. Here, the objective is to use the training data $\mathcal{D}$ to learn a general-purpose function that can subsequently be applied to any new, previously unseen input $\mathbf{x}^* \in \mathcal{X}$. That is, the model is constructed without any knowledge of the specific test instances it will later be asked to evaluate. This paradigm aligns with the goal of discovering a general rule or principle that holds across the entire input domain. In contrast, transductive learning addresses a more constrained problem. Instead of learning a universal function, the goal is to predict the labels $\{y_j\}_{j=1}^M$ for a specific, predefined set of unlabelled test instances $\{\mathbf{x}_j\}_{j=1}^M$. In this setting, the learning algorithm has access to both the labelled training data $\mathcal{D}$ and additional information about the unlabelled test set at training time. By leveraging this structural information about the subset of the input domain the algorithm is applied in, a transductive model can make predictions that are specifically tailored to that particular set of instances, potentially achieving higher accuracy than a general-purpose inductive model.

Molecular Representations

In order to develop algorithms that robustly generalise to novel compounds, it is essential to establish rigorous evaluation protocols and ways to characterise distribution shifts. This challenge is fundamentally tied to how molecules are represented mathematically: the choice of input representation determines what constitutes molecular similarity and thus defines the effective distance between training and test distributions. Unlike in traditional machine learning domains such as computer vision, where input representations are largely standardised, multiple approaches for encoding molecular structures exist and are widely used in practice. This choice of featurisation can significantly impact model generalisation: a well-designed representation brings pharmacologically similar molecules closer together in the input space and simplifies the learning task, while poorly chosen representations may fail to capture important molecular features.

On a fundamental level, a molecule can be represented as an undirected, labelled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where vertices $\mathcal{V} = v_1, \ldots, v_N$ represent the atoms of an N-atom molecule and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represent covalent bonds between these atoms. These molecular graphs can be augmented with vertex and edge features encoding chemical attributes such as atom types and bond orders (see Figure 2.2).

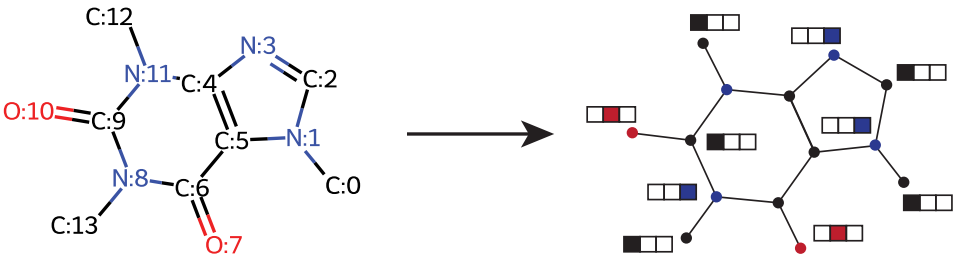


Figure 2.2. Graph-based featurisers convert molecules into permutation-invariant representations that can be consumed by specialised neural network architectures. Here, each vertext and edge can be augmented with features encoding chemical attributes such as atom types, bond orders and stereochemistry.

When three-dimensional structural information is available, this representation can be extended to describe spatial graphs by associating each vertex with its 3D coordinates $\mathbf{r}_i \in \mathbb{R}^3$, effectively featurising the molecule as a point cloud of atoms connected by bonds. This information is crucial for reasoning about chemical properties and biomolecular interactions that depend on molecular geometry, such as protein-ligand binding or conformational energies.

While graph representations capture the way medicinal chemists think about molecular structures, they require specialised neural network architectures to be processed. Molecular fingerprints are a well-established alternative approach that transforms graph-structured inputs into fixed-length vectors instead. Starting from the topological graph of a molecule, they are derived by systematically enumerating and hashing subgraphs $\mathcal{G}' = (\mathcal{V}' \subseteq \mathcal{V}, \mathcal{E}' \subseteq \mathcal{E})$. Two widely used variants are Extended Connectivity Fingerprints (ECFPs, Rogers, Hahn (2010)) and RDKit Fingerprints (RDKitFPs, Landrum et al. (2022)). Both methods enumerate all labelled subgraphs up to a specified diameter, differing primarily in their substructure space: while RDKitFPs consider subgraphs of any shape, ECFPs focus specifically on radial substructures. Starting from each atom, information is iteratively propagated to neighbouring atoms, creating unique identifiers for subgraphs of increasing size. The extracted structural features are then hashed into a d-dimensional bit vector $\mathbf{x} \in \{0,1\}^d$, where each bit indicates the presence or absence of specific substructures, as illustrated in Figure 2.3.

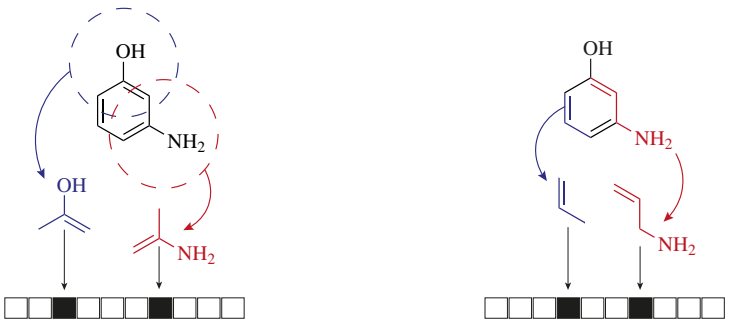


Figure 2.3. Schematic representation of extended-connectivity fingerprints **(left)** and RDKit fingerprints **(right)**. Both methods operate on the topological graph of a molecule and hash all labelled subgraphs up to a certain size into a fixed-length bit vector, differing only in the space of substructures they consider.

Text-based representations offer another approach to encoding molecular structures. The widely-used SMILES strings (Weininger, 1988) represent molecules as strings of ASCII characters, where atoms are denoted by their chemical symbols and branching and rings are indicated through parentheses and numbers (see Figure 2.4). The more recent SELFIES format (Krenn et al., 2020) establishes a bijective mapping between strings and valid molecular graphs, ensuring that any syntactically correct SELFIES string corresponds to a chemically valid structure. Both representations can be converted into a machine-readable format using standard natural language processing techniques, enabling the application of sequence-based modelling architectures to molecular property prediction tasks.

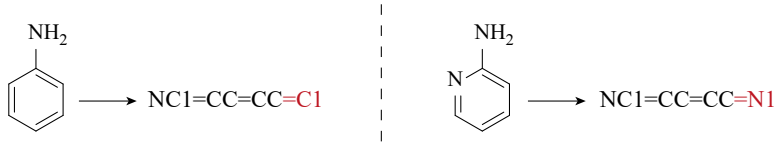


Figure 2.4. SMILES strings for structurally similar molecules. Similarity is encoded in the string through common contiguous subsequences (black). Local differences are highlighted in red.

Neural Network Architectures

Having described various approaches for encoding molecular structures as mathematical objects, this section outlines neural network architectures that are able to process these representations. While fingerprints, graphs and SMILES strings differ substantially in their format and the structural information they preserve, they can all serve as input to specialised neural networks that aim to extract meaningful patterns from chemical data. The choice of network architecture is intrinsically linked to the input representation: each representation type requires a specific architectural framework to effectively capture and transform the encoded chemical information into representations that are useful for downstream prediction tasks.

Representation learning

On a fundamental level, neural networks can be viewed as the composition of a *representation learning* trunk $h : \mathcal{X} \to \mathbb{R}^d$ and an output head $g : \mathbb{R}^d \to \mathcal{Y}$. In most cases, the former maps the input space $\mathcal{X}$ to a continuous embedding space $\mathbb{R}^d$ that captures latent structural factors relevant to the prediction problem at hand. The latter then maps this latent representation to the output space $\mathcal{Y}$. In most settings, the output head is simply a linear layer, and the expressivity of the model is primarily determined by h. This section covers different neural network architectures that can be used to construct h, examining how different designs process and extract information from their respective input domains. The subsequent section provides an overview of commonly used prediction heads and loss functions that are applicable to various drug discovery tasks.

Multi-layer perceptrons (MLPs) are fundamental building blocks for processing fixed-length vector representations such as molecular fingerprints. Given an input vector $\mathbf{x} \in \mathbb{R}^{d_0}$, an L-layer MLP is defined as a nested function

$$f_\theta(\mathbf{x}) = f_L \circ f_{L-1} \circ \cdots \circ f_1(\mathbf{x}), \tag{2.3}$$

where each layer f_l applies a transformation of the form $f_l(\mathbf{h}) = \sigma_l(\mathbf{W}^{(l)}\mathbf{h} + \mathbf{b}^{(l)})$. Here, the weight matrix $\mathbf{W}^{(l)} \in \mathbb{R}^{d_{l+1} \times d_l}$ and bias vector $\mathbf{b}^{(l)} \in \mathbb{R}^{d_{l+1}}$ specify an affine transformation, followed by a non-linear activation function σ_l. The activation functions are critical components that enable the network to model complex, non-linear relationships. Without them, the representational capacity of the entire network would be equivalent to that of a single linear transformation, regardless of its depth. Common choices include the rectified linear unit (ReLU), sigmoid, and hyperbolic tangent functions

$$\sigma_{\text{ReLU}}(x) = \max(0, x), \quad \sigma_{\text{sigmoid}}(x) = \frac{1}{1 + e^{-x}}, \quad \sigma_{\tanh}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{2.4}$$

The universal approximation theorem (Cybenko, 1989; Hornik, 1991) provides the theoretical foundation for MLPs by proving that even a network with $L = 1$ can approximate any continuous function on compact subsets of $\mathbb{R}^n$ to arbitrary precision, provided d_L is of sufficient dimensionality. Despite this theoretical guarantee, deeper architectures with $L > 1$ are nearly always more efficient, since multiple hidden layers enable the model to capture hierarchical representations and disentangle increasingly complex and abstract patterns (Telgarsky, 2016).

Graph Neural Networks (GNNs) build on MLPs to provide a specialised framework for processing variable-sized graph structures (Kipf, Welling, 2016; Veličković et al., 2018; Xu et al., 2019). Unlike standard MLPs, they operate directly on molecular graphs, preserving the topological relationships between atoms and allowing them to reason over local chemical environments. Starting from a molecular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with initial atom features $\mathbf{h}_v^{(0)}$ for each node $v \in \mathcal{V}$ and bond features $\mathbf{e}_{uv}$ for each edge $(u, v) \in \mathcal{E}$, GNNs apply a *message passing* framework that iteratively updates node representations. Each layer consists of three fundamental steps. First, every atom receives messages from its neighbours that are given by functions of the respective node and edge features

Message passing

$$\mathbf{m}_{u \to v}^{(l)} = \text{message}^{(l)}(\mathbf{h}_u^{(l-1)}, \mathbf{h}_v^{(l-1)}, \mathbf{e}_{uv}). \tag{2.5}$$

Subsequently, the messages from all neighbours $\mathcal{N}(v)$ are combined using a permutation-invariant aggregation function to ensure that the result is independent of how the neighbours are indexed

$$\mathbf{a}_v^{(l)} = \text{aggregate}^{(l)}(\mathbf{m}_{u \to v}^{(l)} | u \in \mathcal{N}(v)). \tag{2.6}$$

Finally, each node representation is updated based on its current state and the transformed and aggregated states of its neighbours

$$\mathbf{h}_v^{(l)} = \text{update}^{(l)}(\mathbf{h}_v^{(l-1)}, \mathbf{a}_v^{(l)}) \tag{2.7}$$

The message and update functions are usually parameterised through neural networks, e.g., MLPs, while the aggregate function is typically a constant operation such as sum, mean, or max. After L iterations of message passing, each atom has accumulated information from its L-hop neighbourhood, effectively capturing its local chemical environment.

To obtain a representation of the entire molecule, a readout function pools the final atom representations:

$$\mathbf{h}_{\mathcal{G}} = \text{readout}(\mathbf{h}_v^{(L)} | v \in \mathcal{V}) \tag{2.8}$$

This readout function is also permutation invariant, ensuring that the graph-level representation does not depend on the ordering of atoms. Common choices include sum, mean, or max pooling, potentially combined with skip connections (Xu et al., 2018) and followed by an additional transformation.

Transformer architectures (Vaswani et al., 2017) have emerged as the dominant approach for processing sequence-based molecular representations. Unlike MLPs, which apply the same affine transformation to every input, transformers employ a *self-attention* mechanism to dynamically compute input-dependent weights that determine how different tokens influence each other's representations. Given a d-dimensional embedding of n input tokens $\mathbf{H}^{(l)} \in \mathbb{R}^{n \times d}$, self-attention is defined as the weighted sum of values $\mathbf{V}^{(l)} \in \mathbb{R}^{n \times d_v}$ based on the similarity between keys $\mathbf{K}^{(l)} \in \mathbb{R}^{n \times d_k}$ and queries $\mathbf{Q}^{(l)} \in \mathbb{R}^{n \times d_k}$

Self-attention

$$\text{Attention}(\mathbf{Q}^{(l)}, \mathbf{K}^{(l)}, \mathbf{V}^{(l)}) = \text{softmax}\left(\frac{\mathbf{Q}^{(l)}\mathbf{K}^{(l)T}}{\sqrt{d_k}}\right)\mathbf{V}^{(l)}, \tag{2.9}$$

Here query, key, and value matrices are given by linear projections of the input

$$\mathbf{Q}^{(l)} = \mathbf{H}^{(l)}\mathbf{W}_Q^{(l)}, \quad \mathbf{K}^{(l)} = \mathbf{H}^{(l)}\mathbf{W}_K^{(l)}, \quad \mathbf{V}^{(l)} = \mathbf{H}^{(l)}\mathbf{W}_V^{(l)} \tag{2.10}$$

with $\mathbf{W}_K^{(l)}, \mathbf{W}_Q^{(l)} \in \mathbb{R}^{d_k \times d}$ and $\mathbf{W}_V^{(l)} \in \mathbb{R}^{d_v \times d}$. In practice, most transformer architectures employ multiple attention heads that operate in parallel

$$\text{MultiHead}(\mathbf{H}^{(l)}) = \text{concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}_O^{(l)}, \tag{2.11}$$

with $\mathbf{W}_O^{(l)} \in \mathbb{R}^{d_o \times (h \cdot d_k)}$. Each head can learn to focus on different aspects of molecular structure, such as specific functional groups or stereochemical features. The resulting representations are then combined with skip connections and MLPs to form a complete transformer block

$$\mathbf{Z}^{(l)} = \text{LayerNorm}(\mathbf{H}^{(l)} + \text{MultiHead}(\mathbf{H}^{(l)})) \tag{2.12}$$

$$\mathbf{H}^{(l+1)} = \text{LayerNorm}(\mathbf{Z}^{(l)} + \text{MLP}(\mathbf{Z}^{(l)})) \tag{2.13}$$

Beyond sequence-based approaches, recent research has extended transformer-based architectures to directly model three-dimensional molecular structures (Abramson et al., 2024; Hayes et al., 2025; Ellmen et al., 2025). By incorporating geometric relationships into the self-attention mechanism, these models condition token interactions on both sequence information and interatomic distances and angles. This approach combines the advantages of spatial graph-based methods with the representational capacity of dynamic attention mechanisms, enabling them to effectively reason about geometry-dependent properties.

Label Spaces and Loss Functions

While the right choice of molecular representations and neural network architectures is crucial, an equally important consideration is how the properties of interest are represented as targets. Pharmacological processes exhibit complex dynamics characterised by non-linear dose dependencies, multi-scale interactions and assay-specific measurement biases. Translating these into a well-defined numerical output amenable to predictive modelling is a non-trivial task that requires considerable domain expertise. In practice, the predictive targets are described as either regression or classification labels, which in turn influence the choice of loss function, output head, and evaluation strategy.

Classification models aim to categorise molecules according to discrete properties such as activity against a therapeutic target or off-target toxicity. These models predict the conditional label distribution $p_\theta(\mathbf{y}|\mathbf{X} = \mathbf{x})$, where $\mathbf{y} \in [0, 1]^M$ represents probabilities across M possible categories. In binary classification tasks, where $M = 1$, the output head $g : \mathbb{R}^d \to [0, 1]$ is defined as $g(\mathbf{z}) = \sigma(\mathbf{w}^T\mathbf{z} + b)$ where $\mathbf{z} \in \mathbb{R}^d$ is the latent embedding from the representation trunk h, $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are learnable parameters, and σ is the sigmoid function $\sigma(\mathbf{z}) = (1 + e^{-\mathbf{z}})^{-1}$. This transformation yields predicted probabilities $\hat{\mathbf{p}} \in [0, 1]$ that are optimised through the binary cross-entropy loss

$$\mathcal{L}_{\text{BCE}}(\mathbf{y}, \hat{\mathbf{p}}) = -\mathbf{y} \log(\hat{\mathbf{p}}) - (1 - \mathbf{y}) \log(1 - \hat{\mathbf{p}}). \tag{2.14}$$

This loss function is equivalent to the negative log-likelihood of a Bernoulli distribution, providing a principled probabilistic framework for binary decision-making tasks. For multi-class classification problems with $M > 1$ mutually exclusive categories, the output head $g : \mathbb{R}^d \to [0, 1]^M$ is defined as $g(\mathbf{z}) = \text{softmax}(\mathbf{W}^T\mathbf{z} + \mathbf{b})$ where $\mathbf{W} \in \mathbb{R}^{d\times M}$ and $\mathbf{b} \in \mathbb{R}^M$ are learnable parameters. The softmax function

$$\text{softmax}(\mathbf{z})_i = \frac{e^{\mathbf{z}_i}}{\sum_{j=1}^{M} e^{\mathbf{z}_j}} \tag{2.15}$$

transforms the resulting logits into a valid categorical distribution $\hat{\mathbf{p}} \in [0, 1]^M$ that is optimised via the corresponding multi-class cross-entropy loss

$$\mathcal{L}_{\text{MCE}}(\mathbf{y}, \hat{\mathbf{p}}) = -\sum_{i=1}^{M} \mathbf{y}_k \log(\hat{\mathbf{p}}_k). \tag{2.16}$$

Regression models predict continuous molecular properties such as binding affinity, solubility, or pharmacokinetic parameters. Here, the output head maps the latent representation $\mathbf{z} \in \mathbb{R}^d$ to the continuous label space $\mathcal{Y} = \mathbb{R}^M$. For single property prediction with $M = 1$, the output head $g : \mathbb{R}^d \to \mathbb{R}$ is a simple affine transform $g(\mathbf{z}) = \mathbf{w}^T\mathbf{z} + b$, while for multi-task predictions involving M properties it can be extended to $g(\mathbf{z}) = \mathbf{W}^T\mathbf{z}+\mathbf{b}$ with $\mathbf{W} \in \mathbb{R}^{d\times M}$ and $\mathbf{b} \in \mathbb{R}^M$. When the target variable is restricted to $\mathcal{Y} = \mathbb{R}_+$, for example when estimating predictive uncertainties, the output head is often followed by an exponential or softplus activation function.

Given a model prediction $\hat{\mathbf{y}} = g(h(\mathbf{x}))$, the most commonly used objective functions for regression tasks are the mean squared error (MSE) and mean absolute error (MAE) loss functions:

$$\mathcal{L}_{\text{MSE}}(\mathbf{y}, \hat{\mathbf{y}}) = (\mathbf{y} - \hat{\mathbf{y}})^2, \quad \mathcal{L}_{\text{MAE}}(\mathbf{y}, \hat{\mathbf{y}}) = |\mathbf{y} - \hat{\mathbf{y}}|. \tag{2.17}$$

The MSE loss is equivalent to performing maximum likelihood estimation under a homoskedastic Gaussian noise model, while the MAE loss is equivalent to using a Laplacian noise model. The latter offers increased robustness to outliers and is particularly valuable when experimental data may contain measurement artefacts.

Binned regression represents a hybrid approach that discretises continuous output spaces into ordered intervals, transforming regression problems into ordinal classification tasks. For example, binding affinities might be categorised into ranges such as "highly active" (pIC50 $<$ 8), "moderately active" (6 $<$ pIC50 $\leq$ 8), and "weakly active" (pIC50 $\leq$ 6). This approach offers several advantages, including the ability to easily model multi-modal distributions that standard regression methods struggle to capture, as well as aligning predictions with decision-making thresholds used in real-world medicinal chemistry campaigns. This comes at the cost of a more difficult optimisation problem—the standard cross-entropy loss penalises all incorrect predictions equally, regardless of their ordinal distance from the true bin—and lower model precision, as the resolution of predictions is inherently limited by the predetermined bin width.

Model Evaluation

While loss functions guide model optimisation during training, *evaluation metrics* serve the complementary purpose of assessing performance in deployment scenarios, reflecting application-specific requirements and trade-offs. Binary classification tasks typically employ several complementary metrics, such as accuracy, precision, recall, and the area under the receiver operating characteristic curve (AUC-ROC). Each metric addresses different aspects of model performance, with their relative importance depending on the specific application domain and the consequences of false positive versus false negative predictions. For continuous property prediction, standard metrics include the root mean squared error (RMSE), the mean absolute error, and the coefficient of determination (R^2). The choice of metric depends primarily on the extent of the adverse impact of larger prediction errors and the level of experimental noise in the training data.

Evaluation metrics

An additional consideration is that many drug discovery datasets feature significant *label imbalance*, which profoundly impacts the evaluation strategy. In hit identification tasks, where often only a small fraction of screened compounds are active, the area under the precision-recall curve provides more informative assessments than ROC curves. While ROC curves consider sensitivity and specificity equally, precision-recall curves emphasise the performance on the minority class, providing a more stringent and practically relevant measure (Saito, Rehmsmeier, 2015). For regression tasks, the imbalance manifests as non-uniform sampling across the response range, requiring careful stratification during validation to ensure reliable performance assessment across the entire property spectrum.

Label imbalance

Predictive uncertainty

Beyond standard metrics that measure predictive accuracy, the quantification of *predictive uncertainty* has emerged as a critical component of reliable decision-making systems (Guo et al., 2017; Ovadia et al., 2019). Uncertainty estimates help distinguish between confident predictions for compounds similar to training data and speculative outputs for molecules in unexplored regions of chemical space. For classification tasks, calibration metrics such as the expected calibration error (Naeini et al., 2015) and adaptive calibration error (Nixon et al., 2019) assess the alignment between predicted probabilities and observed frequencies, ensuring that a model's confidence accurately reflects its empirical accuracy. For regression tasks, prediction intervals and calibration curves evaluate whether the predictive uncertainties capture the true error distribution (Chung et al., 2021). Well-calibrated uncertainty estimates are crucial for informed decision-making workflows, as they highlight when model predictions should be treated with caution and when additional experimental validation is necessary.

Together, these approaches provide a comprehensive framework for evaluating the reliability of predictive models and are particularly important in settings where there is a strong distribution shift between training and deployment. By combining standard statistical metrics with domain-specific evaluation strategies and uncertainty quantification, researchers can develop a more nuanced understanding of model capabilities and limitations, ultimately enabling more effective applications in real-world drug discovery workflows.

2.3 Generative Modelling

Generative models constitute a fundamentally different paradigm, moving beyond property prediction to enable the de-novo design of molecules and proteins with desired characteristics. While predictive models map from chemical structures to properties, generative models learn to synthesise new compounds by capturing the underlying distribution of valid molecular structures. This challenging task requires models to navigate vast combinatorial search spaces while respecting complex physicochemical constraints that determine molecular validity and synthesisability. Recent advances in deep generative frameworks—particularly diffusion models—have dramatically improved our ability to sample from complex high-dimensional densities, opening up new possibilities for automated molecular design. This section outlines the theoretical foundations of generative modelling and explores how these methods have been adapted to address unique challenges in drug discovery and protein design, with a particular focus on approaches that enable the controlled generation of structures with specific properties.

Fundamentals

Generative modelling is a fundamental machine learning paradigm, concerned with approximating complex, high-dimensional probability distributions from finite observations. Unlike discriminative models that learn the mapping $p(\mathbf{y}|\mathbf{X} = \mathbf{x})$ between inputs $\mathbf{x} \in \mathcal{X}$ and outputs $\mathbf{y} \in \mathcal{Y}$, generative models capture either the inverse conditional distribution $p(\mathbf{x}|\mathbf{Y} = \mathbf{y})$ or the marginal distribution $p(\mathbf{x})$. The key challenge is to learn a generalisable approximation of the underlying data-generating distribution to enable the synthesis of new samples that maintain the statistical properties of the training set. Deep generative models have shown remarkable success in this setting, producing realistic images (Rombach et al., 2022; Saharia et al., 2022; Zhang et al., 2023), coherent text (Brown et al., 2020; OpenAI et al., 2023; Gemini Team et al., 2024), and complex structural data in various scientific domains (Ingraham et al., 2022; Abramson et al., 2024; Hayes et al., 2025).

Starting from a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ drawn from an unknown data distribution $p_{\text{data}}(x)$, the fundamental objective of generative modelling is to approximate $p_{\text{data}}(x)$ with a model distribution $p_\theta(x)$, parameterised by θ. The learning process aims to minimise the discrepancy between these two distributions, often quantified through divergence measures such as the Kullback-Leibler (KL) divergence

$$\mathcal{L}(\theta) = D_{\text{KL}}\left(p_{\text{data}}(x) \parallel p_\theta(x)\right) = \int_{\mathcal{X}} p_{\text{data}}(x) \log \frac{p_{\text{data}}(x)}{p_\theta(x)} dx \tag{2.18}$$

This learning problem presents significant challenges for a number of reasons. For one, $p_{\text{data}}(x)$ is typically accessible only through a finite set of samples, introducing estimation errors. Furthermore, estimating the integral becomes intractable for high-dimensional data and the model must balance expressiveness against computational traceability, leading to approximation errors. These fundamental difficulties have motivated a number of different approaches.

Deep Generative Models

The development of expressive and computationally efficient generative models has progressed along several distinct methodological trajectories, each offering unique tradeoffs in approaching the challenge of capturing and sampling from complex high-dimensional probability distributions (Bond-Taylor et al., 2021).

Explicit density estimation methods directly parameterise the probability density function $p_\theta(\mathbf{x})$ and enable exact likelihood evaluation at the cost of restricted model flexibility. Early approaches in this category included mixture models and kernel density estimation, but the advent of neural networks has enabled significantly more expressive models. *Energy-based models* (Hinton, Sejnowski, 1983; Hinton, 2002) emerged as a pioneering approach, defining unnormalised probability densities through an energy function $E_\theta : \mathcal{X} \to \mathbb{R}$ that associates high-density points with low energy values and vice versa. To minimise the negative log-likelihood $\mathcal{L} = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[-\log p_\theta(\mathbf{x})]$, energy-based models rely on a contrastive divergence objective that approximates the gradient of the log-likelihood as

Energy-based models

$$\nabla_\theta \mathcal{L} = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\nabla_\theta E_\theta(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim p_\theta}[\nabla_\theta E_\theta(\mathbf{x})] \tag{2.19}$$

where samples from the model distribution are obtained through Markov Chain Monte Carlo (MCMC) methods. While conceptually elegant, the reliance on MCMC sampling for training and inference has historically limited their scalability to high-dimensional input spaces. *Autoregressive models* aim to avoid these limitations by decomposing the joint distribution with the chain rule of probability

Autoregressive models

$$\mathcal{L} = -\log p_\theta(\mathbf{x}) = -\sum_{i=1}^{n} \log p_\theta(\mathbf{x}_i|\mathbf{x}_{<i}), \tag{2.20}$$

enabling tractable likelihood estimation and exact sampling. Modern implementations leverage masked self-attention mechanisms to successfully capture long-range dependencies (OpenAI et al., 2023; Gemini Team et al., 2024), however, the sequential nature of autoregressive models still limits their applicability in domains without a fixed ordering and slows down sampling for high-dimensional data. *Normalising flows* (Tabak, Vanden-Eijnden, 2010; Tabak, Turner, 2013; Rezende, Mohamed, 2015) offer an alternative approach by constructing invertible transformations $f_\theta : \mathcal{X} \to \mathcal{X}$ between a simple base distribution p_Z and the target distribution p_{data}. The density of a transformed sample is computed via the change of variables formula

Normalising flows

$$p_\theta(\mathbf{x}) = p_Z(f_\theta^{-1}(\mathbf{x})) \left| \det \left(\frac{\partial f_\theta^{-1}}{\partial \mathbf{x}} \right) \right|. \tag{2.21}$$

Here, the key challenge lies in designing transformations that are both highly expressive and computationally tractable, such as coupling layers (Dinh et al., 2014; Dinh et al., 2017) and autoregressive flows (Papamakarios et al., 2017; Huang et al., 2018). Neural ODEs (Chen et al., 2018) extend this framework by parameterising the transformation as a continuous-time flow, offering increased flexibility at the cost of additional computational complexity.

A significant shift in the field occurred with the introduction of highly performant implicit density estimation methods that can generate samples from p_{data} without explicitly learning a probability density function. *Generative Adversarial Networks* (GANs, Goodfellow et al. (2014)) pioneered this approach through an adversarial framework in which a generator $G_\theta : \mathbb{R}^{d_z} \to \mathcal{X}$ is trained to transform a latent variable $\mathbf{z} \sim p_Z(\mathbf{z})$ into a sample $\mathbf{x}$, while a discriminator $D_\phi : \mathcal{X} \to [0, 1]$ tries to distinguish between samples from G_θ and the true data-generating distribution p_{data}. Both models are trained simultaneously through a minimax objective

Generative Adversarial Networks

$$\arg\min_\theta \arg\max_\phi \; \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\log D_\phi(\mathbf{x})\right] + \mathbb{E}_{p(\mathbf{z})} \left[\log(1 - D_\phi(G_\theta(\mathbf{z})))\right]. \tag{2.22}$$

This framework has proven particularly effective for generating high-fidelity samples in domains like image synthesis (Brock et al., 2018). However, GANs often exhibit unstable training dynamics and suffer from mode collapse and vanishing gradients, which complicate model scaling (Salimans et al., 2016; Arjovsky, Bottou, 2017; Arjovsky et al., 2017; Thanh-Tung, Tran, 2020).

Variational Autoencoders (VAEs, Kingma, Welling (2013)) bridge explicit and implicit density estimation approaches by combining latent variable models with variational inference. They optimise a lower bound on the log-likelihood

Variational Autoencoders

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - D_{\text{KL}} \left(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})\right) \tag{2.23}$$

where $q_\phi(\mathbf{z}|\mathbf{x})$ represents a learned approximate posterior distribution over latent variables $\mathbf{z}$, and $p(\mathbf{z})$ is a prior distribution, typically chosen to be a standard Gaussian. In practice, VAEs often produce blurry samples when applied to complex datasets such as natural images (Dosovitskiy, Brox, 2016), which has been attributed to both the maximum likelihood objective and the restricted expressiveness of the approximate posterior (Zhao et al., 2017). Notable improvements on the standard VAE formulation include more expressive posteriors (Rezende, Mohamed, 2015), improved decoder architectures (Gulrajani et al., 2017), and alternative objectives with a better balance of reconstruction quality and distribution matching (Higgins et al., 2017). Of particular interest are hierarchical VAEs that build complex priors with multiple levels of latent variables (Vahdat, Kautz, 2020), and hybrid approaches that combine VAEs with other generative frameworks to leverage their complementary strengths (Rombach et al., 2022).

Fundamentally, each of these generative modelling paradigms addresses different aspects of the distribution learning problem. Normalising flows and autoregressive models enable exact likelihood computation and principled sampling but are constrained by invertibility requirements and scale poorly to high dimensions due to sequential generation. GANs have achieved remarkable sample quality through adversarial optimisation but lack density estimation capabilities and suffer from training instabilities. VAEs on the other hand offer a principled framework for both generation and inference but consistently produce lower fidelity samples than other, state-of-the-art methods. While continuous research efforts have proposed various improvements, the most promising development in deep generative modelling is the emergence of denoising diffusion models as a tractable and empirically effective alternative approach.

Denoising Diffusion Models

Denoising diffusion models

Denoising diffusion models have emerged as a powerful class of deep generative algorithms that can produce high-quality samples by learning to reverse a gradual noising process (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020). At their core, they consist of a forward process that systematically corrupts data samples with Gaussian noise until they converge to a tractable reference distribution, and a corresponding denoising process that maps the reference distribution back to the data-generating distribution. This approach provides a tractable framework for approximating complex, high-dimensional densities and allows for controllable generation and exact likelihood computation.

Forward process

The *forward process* is defined as a sequence of operations that gradually transform data samples into pure Gaussian noise. Given a sample from the data distribution $\mathbf{x}_0 \sim p_0(\mathbf{x})$, where $p_0 = p_{\text{data}}$, it is defined by a Markov chain that progressively adds noise according to a variance schedule $\{\beta_t\}_{t=1}^T$. At each time step $t \in \{1, \dots, T\}$, the transition kernel is given by

$$p_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right). \tag{2.24}$$

This process smoothly interpolates between the data distribution p_{data} and the reference distribution $p_T = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The choice of Gaussian transition kernels is particularly advantageous since it allows sampling from the conditional distribution of any intermediate step in closed form

$$p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right), \tag{2.25}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ represent the cumulative effect of the noise schedule. Having defined the forward process, we can construct a corresponding

Reverse process

reverse process that models the denoising transitions

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_t\right), \tag{2.26}$$

where μ_θ is parameterised by a neural network and the variance Σ_t follows directly from the forward process

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right), \quad \Sigma_t = \frac{\beta_t}{1-\bar{\alpha}_t}\mathbf{I}. \tag{2.27}$$

Here, $\epsilon_\theta(\mathbf{x}_t, t) : \mathcal{X} \times \mathbb{R} \to \mathcal{X}$ is a denoising network that learns to predict the noise component $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ that is added to the data at each time step. It is trained by minimising the simplified regression objective

$$\mathcal{L} = \mathbb{E}_{t, \mathbf{x}_0, \epsilon}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|_2^2\right]. \tag{2.28}$$

In the limit of infinitesimally small time steps, diffusion models can be elegantly recast in terms of continuous-time stochastic processes (Song et al., 2020). Here, the forward process converges to a stochastic differential equation (SDE) given by

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{B}_t, \tag{2.29}$$

where $\mathbf{x}_0 \sim p_0$ is a data sample and $\mathbf{B}_t$ represents the standard Brownian motion. The drift and diffusion coefficients $\mathbf{f}(\mathbf{x}_t, t)$ and $g(t)$ determine the nature of the forward process and correspond to $\mathbf{f}(\mathbf{x}_t, t) = -\frac{1}{2}\beta_t \mathbf{x}_t$ and $g(t) = \sqrt{\beta_t}$ for the Gaussian transition kernels and β_t variance schedule from Equation (2.24), also referred to as the Ornstein-Uhlenbeck process or Variance Preserving (VP) SDE

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta_t \mathbf{x}_t \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\mathbf{B}_t. \tag{2.30}$$

To generate new samples from p_0, continuous-time diffusion models leverage the fact that the forward noising process admits a time-reversal (Anderson, 1982; Haussmann, Pardoux, 1986). For the VP-SDE, this process is given by

$$\mathrm{d}\mathbf{x}_\tau = -\beta_\tau \left\{ \frac{1}{2}\mathbf{x}_\tau + \nabla \log p_\tau \left(\mathbf{x}_\tau\right) \right\} \mathrm{d}\tau + \sqrt{\beta_\tau}\mathrm{d}\mathbf{B}_\tau, \tag{2.31}$$

where $\tau = T - t$ flows backwards in time. Similar to the discrete case, $\mathbf{x}_T \sim p_T$ is sampled from the reference distribution and incrementally denoised until it reflects p_0. The gradient of the log-density $\nabla \log p_t(\mathbf{x_t}) = \nabla_{\mathbf{x}} \log p_t\left(\mathbf{x}\right)|_{\mathbf{x}=\mathbf{x}_t}$ in Equation (2.31) is estimated by a time-dependent score network $s_\theta(\mathbf{x}_t, t)$ using the denoising score-matching objective (Hyvärinen, 2005).

$$\mathcal{L}_{\mathrm{DSM}} = \mathbb{E}_t \left[\lambda(t) \mathbb{E}_{\mathbf{x}_0, \epsilon} \left\| \nabla \log p_t(\mathbf{x_t}) - s_\theta(\mathbf{x}_t, t) \right\|_2^2 \right] \tag{2.32}$$

where $\lambda(t)$ is a positive weighting function that can be used to improve training stability. Once fitted, $s_\theta(\mathbf{x}_t, t)$ can be used to simulate the reverse-time SDE with a range of numerical solvers. When using the same transition kernels and discretisation scheme, the predictions of the score network and denoising model are related (Song, Ermon, 2019) through

$$s_\theta(\mathbf{x}_t, t) = -\frac{1}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\bar{\theta}}(\mathbf{x}_t, t). \tag{2.33}$$

The practical implementation of diffusion models requires the careful consideration of various design decisions. Multiple choices for the variance schedule $\{\beta_t\}_{t=1}^T$ have been proposed, ranging from simple linear schedules (Ho et al., 2020) to cosine schedules that improve training stability (Nichol, Dhariwal, 2021), or learned schedules that enable faster sampling (Sahoo et al., 2024). Similarly, sample generation can be accomplished through several approaches, including direct ancestral sampling from the reverse Markov chain (Ho et al., 2020) and probability flow ordinary differential equations derived from the SDE (Song et al., 2020). These methods offer different trade-offs between computational efficiency and sample quality, with the choice often depending on the specific application requirements and computational constraints. Recent work by Karras et al. (2022) has systematically analysed these design choices, demonstrating that the combination of an editable noise schedule and a specific preconditioned network architecture can significantly improve sample quality while reducing the required number of denoising steps.

Flow Matching Models

The continuous-time formulation of diffusion models allows the denoising process to be framed as a deterministic probability flow ordinary differential equation (ODE) (Song et al., 2020), the reverse-time rollout of which is approximated with the same score estimates as the underlying SDE. This perspective naturally raises the question of whether it is possible to learn such a generative ODE directly, without relying on a predefined stochastic process and the associated score-matching objective. *Flow Matching* (FM) provides one formalism to do so by introducing a powerful simulation-free methodology for training continuous normalising flows at scale (Lipman et al., 2023; Tong et al., 2023).

Flow Matching

Instead of learning to reverse an SDE, flow matching directly approximates a vector field that deterministically transports samples from a simple prior distribution p_T to the target data distribution $p_0 = p_{\text{data}}$ via a probability path $p_t(\mathbf{x}_t)$ for $t \in [0, T]$. This transformation is parameterised by a neural network $v_\theta(\mathbf{x}_t, t)$ that defines a time-dependent vector field governing the ODE via

$$\frac{d\mathbf{x}_t}{dt} = v_\theta(\mathbf{x}, t) \text{ and } \frac{\partial p_t(\mathbf{x}_t)}{\partial t} = -\nabla_{\mathbf{x}_t} \cdot (p_t(\mathbf{x}_t) v_\theta(\mathbf{x}_t, t)). \tag{2.34}$$

Conditional Flow Matching

Conditional Flow Matching (CFM) proposes an elegant and tractable way to estimate this vector field by regressing it against a conditional vector field $u_t(\mathbf{x}|\mathbf{x}_0)$ that is defined through a simple conditional probability path. When choosing a Gaussian path based on linear interpolation $p_t(\mathbf{x}|\mathbf{x}_T, \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}; t\mathbf{x}_0 + (T-t)\mathbf{x}_T, \sigma^2\mathbf{I}\right)$, one obtains a simple corresponding conditional vector field, $u_t(\mathbf{x}|\mathbf{x}_T, \mathbf{x}_0) = \mathbf{x}_0 - \mathbf{x}_T$. CFM establishes that a model trained to predict this conditional vector field will, in expectation, learn the correct marginal vector field $u_t(\mathbf{x})$. This yields a tractable, simulation-free regression objective that samples data pairs $(\mathbf{x}_T, \mathbf{x}_0)$, a time t, and a point $\mathbf{x}_t$ along their conditional probability path, and then minimizes the squared error between the network's prediction and the target vector $\mathbf{x}_0 - \mathbf{x}_T$:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t, p_T(\mathbf{x}_T), p_{\text{data}}(\mathbf{x}_0), p_t(\mathbf{x}|\mathbf{x}_T, \mathbf{x}_0)} \left[\| v_\theta(\mathbf{x}, t) - (\mathbf{x}_0 - \mathbf{x}_T) \|_2^2 \right]. \tag{2.35}$$

While closely related, flow matching and denoising diffusion models present distinct trade-offs. The primary advantages of flow matching lie in its conceptual simplicity and training efficiency, since its direct regression objective is often more stable and computationally efficient to optimize than score-matching objectives. Diffusion models, on the other hand, benefit from being a more mature and empirically validated framework, which has fostered a rich ecosystem of proven architectural designs, advanced sampling strategies, and robust conditioning mechanisms (Karras et al., 2022). When the research for this thesis was conducted, flow matching was an emerging paradigm and the clear empirical benefits and best practices for network design and conditioning were still unclear and under active investigation. Given the importance of a robust methodological backbone for the studies presented herein, this thesis focuses on the established framework of diffusion models, while acknowledging potential applications of flow matching where appropriate.

Conditional Diffusion Models

While unconditional diffusion models excel at generating diverse samples from complex data distributions, many practical applications require control over the generation process. Conditional diffusion models extend the basic framework by incorporating additional information to guide the sampling process toward desired outcomes. That is, they modify the denoising process in Equation (2.31) to incorporate the conditioning information $\mathbf{y}$, and sample from the conditional distribution $p_0(\mathbf{x}_0|\mathbf{y})$ rather than the unconditional data distribution $p_0(\mathbf{x}_0)$. For a given $\mathbf{y}$, the conditional reverse-time SDE takes the form:

$$\mathrm{d}\mathbf{x}_t = -\beta_t \left\{ \frac{1}{2}\mathbf{x}_t + \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t|\mathbf{y}) \right\} \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\mathbf{B}_t \tag{2.36}$$

where $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t|\mathbf{y})$ represents the score of the conditional probability density. The key challenge lies in estimating this conditional score function, for which several approaches have been developed.

Classifier guidance

Classifier guidance (Ho et al., 2020; Song et al., 2020; Dhariwal, Nichol, 2021) offers a straightforward approach to conditioning a diffusion model post-hoc. Given a pre-trained classifier $p_\phi(\mathbf{y}|\mathbf{x}_t, t)$ that predicts the desired condition $\mathbf{y}$ from noised samples $\mathbf{x}_t$, we can construct conditional score estimates as

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) + \gamma \nabla_{\mathbf{x}} \log p_\phi(\mathbf{y}|\mathbf{x}_t, t) \tag{2.37}$$

using Bayes' rule. Here $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$ is provided by the score network and the gradient term $\nabla_{\mathbf{x}_t} \log p_\phi(\mathbf{y}|\mathbf{x}_t, t)$ guides the reverse process toward regions of the state space where the classifier assigns high probability to the desired condition. The parameter $\gamma > 0$ is a guidance scale that allows practitioners to control the strength of the conditioning at sampling time. While this approach has proven effective, it requires training and maintaining a separate classifier model that can handle noised inputs at different time steps.

Classifier-free guidance

Classifier-free guidance (Ho, Salimans, 2022) addresses these limitations by eliminating the need for an external classifier. Instead, it leverages a diffusion model trained with randomly masked conditioning information. During training, the model learns to handle both conditional $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y})$ and unconditional $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ generation by randomly setting the condition to a null token $\varnothing$. At inference time, the difference between conditional and unconditional score predictions provides an implicit guidance signal

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t|\mathbf{y}) = (1+\gamma) s_\theta(\mathbf{x}_t, t, \mathbf{y}) - \gamma s_\theta(\mathbf{x}_t, t, \varnothing). \tag{2.38}$$

This approach simplifies implementation while often achieving better results than classifier guidance, and has become the standard in state-of-the-art image generation models (Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022). However, the requirement to provide $\mathbf{y}$ as an explicit input to the score network prevents the application of these methods to regression-based optimisation problems, where the objective is to maximise or minimise a property for which the optimal value is often unknown and likely to have little support under the training distribution of the conditional score network.

Diffusion Posterior Sampling

Diffusion Posterior Sampling (DPS, Chung et al. (2022)) and related approaches (Song et al., 2023; Yu et al., 2023; Bansal et al., 2023) complement classifier and classifier-free guidance in settings where conditioning information is meaningfully defined for clean data samples $\mathbf{x}_0$, but not for noisy intermediate states $\mathbf{x}_t$. Since $\mathbf{y}$ only depends on $\mathbf{x}_t$ through $\mathbf{x}_0$, DPS factorises the likelihood $p(\mathbf{y}|\mathbf{x}_t)$ as

$$p(\mathbf{y}|\mathbf{x}_t) = \int p(\mathbf{y}|\mathbf{x}_0)p(\mathbf{x}_0|\mathbf{x}_t)\mathrm{d}\mathbf{x}_0 \tag{2.39}$$

and approximates the integral by estimating the expected value of $\mathbf{x}_0$ given $\mathbf{x}_t$

$$\hat{\mathbf{x}}_0 := \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = \frac{1}{\sqrt{\bar{\alpha}(t)}}(\mathbf{x}_t + (1-\bar{\alpha}(t))\nabla \log p_t(\mathbf{x}_t)). \tag{2.40}$$

Assuming a labelling oracle or reliable likelihood estimator $p(\mathbf{y}|\hat{\mathbf{x}}_0) = \mathcal{A}(\hat{\mathbf{x}}_0)$, the distance between the desired conditioning information $\mathbf{y}$ and the current prediction $\mathcal{A}(\hat{\mathbf{x}}_0)$ can then be used to guide the denoising process

$$\mathrm{d}\mathbf{x}_t = -\beta_t \left\{ \frac{1}{2}\mathbf{x}_t + s_\theta(\mathbf{x}_t, t) - \gamma \nabla_{\mathbf{x}} \|y - \mathcal{A}(\hat{\mathbf{x}}_0)\|_2^2 \right\} \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\mathbf{B}_t. \tag{2.41}$$

This makes DPS particularly well-suited for inverse problems in which physical measurement processes are only defined in the original data space and cannot be meaningfully extended to noisy intermediate states. By leveraging the pre-trained unconditional score model and the measurement operator at $t = 0$, this approach enables conditional sampling without requiring modifications to the unconditional score network or the need for a time-step-dependent classifier.

These approaches to conditional sampling with diffusion models complement one another, addressing different types of problems. Classifier guidance excels in continuous optimisation scenarios where the optimal label is unknown, making it particularly effective for tasks like hit identification or lead optimisation (Gruver et al., 2023; Lee et al., 2023; Weiss et al., 2023). Classifier-free guidance provides robust semantic control over the generation process without requiring separate classifier training, enabling conditioning on natural language descriptions and other high-level attributes (Ramesh et al., 2022; Saharia et al., 2022). DPS offers a principled framework for inverse problems where ground truth labels or measurements are only meaningfully defined for clean samples (Song et al., 2023; Yu et al., 2023), e.g., when optimising the physical plausibility of generated biomolecular complex structures. Developing new, sample-efficient conditional generation techniques remains an active area of research. For instance, Karras et al. (2025) demonstrated that conditioning can be achieved by having a stronger model follow guidance from a weaker version of itself, rather than requiring external classifiers or specialised training. Du et al. (2023) showed that diffusion models can be conditioned through explicit probability composition (e.g., products and mixtures) when using carefully designed MCMC samplers instead of standard diffusion sampling. Furthermore, Black et al. (2024) reframed the conditioning problem as a reinforcement learning task, enabling direct optimisation of the diffusion model toward arbitrary conditioning objectives through policy gradient methods.

Applications in Drug Discovery and Protein Design

In contrast to the computer vision domains for which most state-of-the-art diffusion models have been developed, applications in molecular design must contend with discrete search spaces, strict physicochemical constraints, and a scarcity of labelled data. These challenges have led to methodological innovations across two distinct but complementary paradigms, namely (i) methods operating over molecular graphs and protein sequences for de novo design and molecular optimisation, and (ii) approaches operating directly over three-dimensional structures for structure prediction and conformational modelling.

Graph and sequence-based approaches focus primarily on de novo design and lead optimisation. These methods must adapt continuous diffusion frameworks to handle discrete structures from large input domains. This creates an inherent tension between exploring novel regions of chemical or protein sequence space and maintaining molecular validity or expressibility, while at the same time respecting various other chemical and biological constraints such as desirable functional properties that are provided as conditioning variables.

For small molecules, researchers have focussed on developing diffusion models that directly operate over the space of molecular graphs to capture the dependencies between atoms and their connectivity patterns. For instance, Jo et al. (2022) introduced a system of stochastic differential equations that jointly model atomic features and molecular graph structures through interacting diffusion processes. Similarly, Vignac et al. (2023a) developed a discrete diffusion framework using Markov transition matrices that preserve chemical sparsity patterns and atom and bond type distributions. For protein design, researchers have leveraged advances in discrete diffusion models (Austin et al., 2021; Hoogeboom et al., 2021; Hoogeboom et al., 2022a) to capture the underlying distribution of natural protein sequences and generate diverse yet biologically plausible samples. For example, Alamdari et al. (2023) used an order-agnostic masked corruption process to train a diffusion model on large-scale sequence data, enabling both unconditional and conditional protein generation through sequence inpainting. Similarly, Gruver et al. (2023) incorporated gradient guidance into discrete diffusion models to optimise sequences through iterative refinement, maximising a set of desired properties while maintaining sequence naturalness. These approaches have demonstrated that diffusion models can effectively capture distributions in discrete search spaces, and have proven particularly valuable for designing molecules and proteins with specific functional properties.

The key challenge in developing robust conditional diffusion models in this setting is the mismatch between data availability and optimisation needs. While vast databases of unlabelled molecular structures (Ruddigkeit et al., 2012; Grygorenko et al., 2020; Tingle et al., 2023) and protein sequences (UniProt-Consortium, 2015; Olsen et al., 2022; UniProt-Consortium, 2023) enable the training of robust unconditional models, guiding generation toward specific properties requires experimental labels with a scope that is inherently limited by the practical constraints of wet-lab validation. This creates a particular challenge for continuous optimisation problems, where the optimal molecular property values may lie in unexplored regions of the input domain and are far from the available training examples.

Structure-based approaches focus on modelling the three-dimensional structure of biomolecular monomers and complexes given a topological graph and/or protein sequence. A fundamental limitation in this domain is the scarcity of high-quality 3D structural data, as experimental structure determination through X-ray crystallography or cryo-electron microscopy remains expensive and technically challenging, particularly for protein-protein and protein-ligand complexes.

To model small molecule conformers, Jing et al. (2022) developed a diffusion framework operating over the torsion angles of a molecule, recognising that most molecular flexibility arises from rotations around single bonds while preserving rigid local structures such as bond lengths and angles. Xu et al. (2022) approached conformer generation by treating molecules as geometric graphs with continuously evolving atomic coordinates and features, using equivariant neural networks to maintain geometric symmetries throughout the diffusion process. Similarly, Hoogeboom et al. (2022b) introduced a unified equivariant model that jointly operates on both atom coordinates and types, enabling direct generation of valid 3D structures while respecting molecular symmetries and constraints.

For protein structure prediction, Trippe et al. (2023), Yim et al. (2023a), and Yim et al. (2023b) developed diffusion models that directly operate on the SE(3) group to generate protein backbones, enabling direct modelling of rotations and translations in 3D space. Watson et al. (2023) developed a framework for generating novel protein monomer structures by repurposing a pre-trained structure prediction model, while Corso et al. (2023) and Corso et al. (2024) adapted the diffusion framework to protein-ligand docking by learning to denoise rigid body transformations and torsion angles. Most recently, Abramson et al. (2024) simplified previous approaches and proposed a state-of-the-art diffusion model that directly operates on atomic coordinates, eliminating the need for specialised handling of different macromolecules.

The continued development of diffusion models has the potential to accelerate drug discovery and protein engineering by enabling the more efficient exploration of biomolecular space. Recent work has demonstrated the capability of these models to capture protein flexibility and sample from equilibrium distributions of small molecules and protein monomers (Jing et al., 2024a; Jing et al., 2024b; Lewis et al., 2024), suggesting a path toward modelling the dynamic nature of molecular systems. Another promising direction lies in combining structure and sequence-based approaches (Vignac et al., 2023b; Campbell et al., 2024) to train diffusion models that simultaneously operate in multiple representational spaces. For proteins, this enables more refined structure-sequence co-design, while for small molecules it allows for the integration of pharmacophoric and structural features to generate compounds satisfying multiple design constraints. Key challenges remain in developing more robust conditioning strategies and evaluation metrics that capture both chemical validity and biological relevance, but the rapid progress in this field suggests that diffusion models will play an increasingly important role in understanding and engineering molecular systems.

Chapter 3

Context-Guided Diffusion

This chapter is based on the following publication:

L. Klarner, T. G. Rudner, G. M. Morris, C. M. Deane, Y. W. Teh. Context-guided diffusion for out-of-distribution molecular and protein design. *International Conference on Machine Learning*, 2024.

Personal contributions:

1. Conceptualised key research ideas and methodological contributions.
2. Identified suitable modelling frameworks and relevant datasets that allowed for a meaningful empirical evaluation of the proposed method.
3. Carried out all experiments, including data pre-processing, model training, sample generation and the analysis and visualisation of results.
4. Developed the framing and presentation of the method in collaboration with all other co-authors. Drafted the manuscript and improved it in collaboration with all other co-authors.
5. Engaged with reviewers during the peer review process.

A full statement of contributions can be found at the end of this chapter.

The supplementary information for this chapter is provided in Appendix A.

Context-Guided Diffusion for Out-of-Distribution Molecular and Protein Design

Leo Klarner [1] **Tim G. J. Rudner** [2] **Garrett M. Morris** [1] **Charlotte M. Deane** [1] **Yee Whye Teh** [1]

Abstract

Generative models have the potential to accelerate key steps in the discovery of novel molecular therapeutics and materials. Diffusion models have recently emerged as a powerful approach, excelling at unconditional sample generation and, with data-driven guidance, conditional generation within their training domain. Reliably sampling from high-value regions beyond the training data, however, remains an open challenge—with current methods predominantly focusing on modifying the diffusion process itself. In this paper, we develop context-guided diffusion (CGD), a simple plug-and-play method that leverages unlabeled data and smoothness constraints to improve the out-of-distribution generalization of guided diffusion models. We demonstrate that this approach leads to substantial performance gains across various settings, including continuous, discrete, and graph-structured diffusion processes with applications across drug discovery, materials science, and protein design.

1. Introduction

The central goal of molecular discovery is to identify novel compounds with desirable functional properties. The immense size of the underlying search spaces—up to 10^{60} for drug-like small molecules (Bohacek et al., 1996) and 20^N for N-length protein sequences (Maynard Smith, 1970)—renders this a challenging combinatorial optimization problem. Considering the substantial cost of synthesizing and validating candidate compounds experimentally, the question of how to efficiently navigate these search spaces to locate high-value subsets lies at the core of modern molecular design (Gómez-Bombarelli et al., 2018; Sanchez-Lengeling & Aspuru-Guzik, 2018; Bilodeau et al., 2022).

[1]University of Oxford, UK [2]New York University, New York, USA. Correspondence to: Leo Klarner <leo.klarner@stats.ox.ac.uk>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

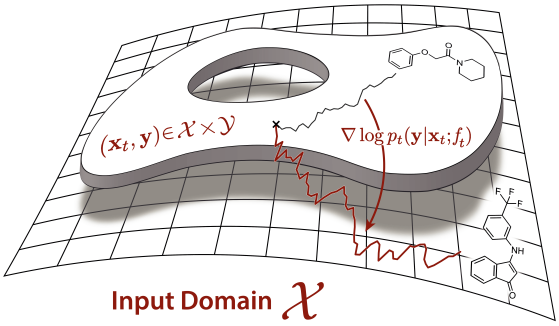


Figure 1: Guidance models that generalize poorly under distribution shifts can be a major performance bottleneck for property-guided diffusion models. We introduce a guidance model regularizer that improves generalization under distribution shifts and enables *context-guided diffusion* (CGD). We show that CGD leads to conditional sampling processes that consistently generate novel, high-value molecules (**red**).

Deep generative models have the potential to accelerate this process by capturing and abstracting key structural properties of their input domain (Sanchez-Lengeling & Aspuru-Guzik, 2018; Bilodeau et al., 2022). In recent years, denoising diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020; Karras et al., 2022) have emerged as the method of choice, displaying impressive performance in conditional and unconditional image generation tasks (Dhariwal & Nichol, 2021; Saharia et al., 2022; Rombach et al., 2022; Zhang et al., 2023), as well as applications across chemistry (Jing et al., 2022; Corso et al., 2023) and biology (Watson et al., 2023; Abramson et al., 2024).

While diffusion models excel at capturing complex, multimodal densities (Kadkhodaie et al., 2024), accurately modeling the training distribution is only the first step. To fully leverage diffusion models for applied molecular optimization, we require carefully designed guidance functions that steer the sample generation process toward compounds with desirable properties (Weiss et al., 2023; Gruver et al., 2023; Lee et al., 2023). However, when labeled data for training these guidance functions is scarce and only available for a biased and unrepresentative subset of the input domain, as is often the case in practice, overconfident guidance signals risk steering the generative process toward false-positive regions of chemical or protein sequence space (Figure 1).

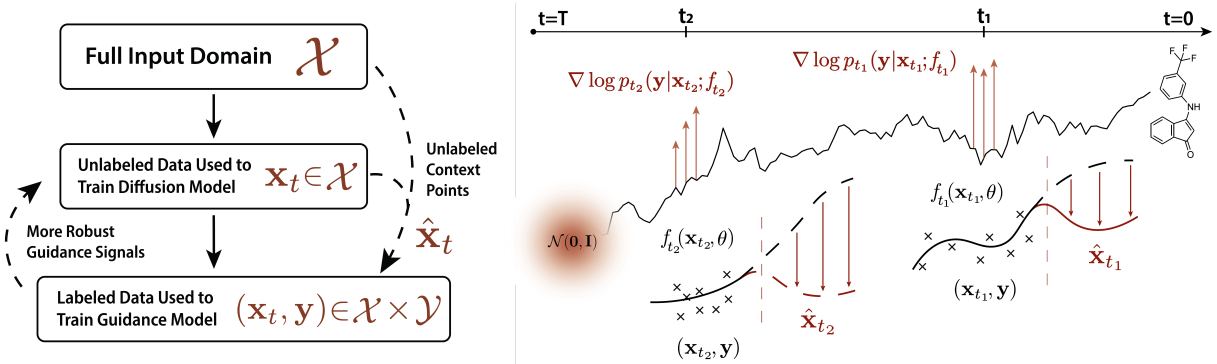


Figure 2: *Context-guided diffusion* leverages unlabeled context data to combine signals from labeled training data with structural information of the broader input domain **(left)**. Specifically, we construct a data- and noise scale-dependent guidance model regularizer that encourages smooth gradients, mean reversion, and high predictive uncertainty on out-of-distribution (OOD) inputs, allowing the conditional denoising process under a context-guided diffusion model to focus on promising near-OOD subsets of chemical and protein sequence space **(right)**.

In this paper, we propose *context-guided diffusion* (CGD)—a simple plug-and-play method that leverages unlabeled data and smoothness constraints to improve the out-of-distribution generalization of property-guided diffusion models. Specifically, we construct a guidance model regularizer that leverages unlabeled context data to train guidance models that both (i) fit the training data well and (ii) exhibit high uncertainty and smooth gradients in out-of-distribution (OOD) regions of the input domain (Figure 2). We show that using the resulting context-aware guidance model to perform context-guided diffusion leads to conditional sampling processes that consistently generate novel, near-OOD molecules with desirable properties.

CGD is straightforward to implement, requires no changes to the architecture of the guidance model, and adds no computational overhead at sampling time. Crucially, it is agnostic to the underlying generative process and thus compatible with latent (Rombach et al., 2022; Xu et al., 2023), equivariant (Hoogeboom et al., 2022), Riemannian (De Bortoli et al., 2022; Huang et al., 2022), and constrained (Lou & Ermon, 2023; Fishman et al., 2023a;b) diffusion models.

We demonstrate the versatility of our approach by applying it to the design of small molecules with graph-structured diffusion processes (Lee et al., 2023), the generation of novel materials with equivariant diffusion models (Weiss et al., 2023), and the optimization of discrete protein sequences with categorical diffusion (Gruver et al., 2023). We find that CGD consistently outperforms standard guidance models, as well as more sophisticated pre-training and domain adaptation techniques, and that it enables more reliable generation of novel samples from high-value subsets of chemical and protein sequence space.

The code for our experiments can be accessed at:
`https://github.com/leojklarner/context-guided-diffusion`

2. Guided Diffusion Models

We start by providing a brief overview of standard unconditional diffusion models, following Song et al. (2020), and then discuss conditional sampling schemes, focusing on classifier guidance as introduced in Sohl-Dickstein et al. (2015), Song et al. (2020) and Dhariwal & Nichol (2021).

Unconditional Diffusion Models. Denoising diffusion models progressively add noise to a data distribution p_0 until it approaches a tractable reference distribution p_T. This forward noising process is governed by a stochastic differential equation (SDE) (Song et al., 2020), defined as

$$\mathrm{d}\mathbf{X}_t = \mathbf{f}(\mathbf{X}_t, t) + g(t)\mathrm{d}\mathbf{B}_t, \tag{1}$$

where $\mathbf{X}_0 \sim p_0$ is sampled from the data distribution and $\mathbf{B}_t$ represents the d-dimensional Brownian motion. A common choice is the VP-SDE (Sohl-Dickstein et al., 2015; Ho et al., 2020), with drift coefficient $\mathbf{f}(\mathbf{X}_t, t) = -\frac{1}{2}\beta_t \mathbf{X}_t$ and diffusion coefficient $g(t) = \sqrt{\beta_t}$, where the noise scale $\beta_t > 0$ determines the amount of corruption. Over time steps $t \in [0, T]$, this SDE gradually noises $\mathbf{X}_t \sim p_t$ until it converges to the reference distribution $p_T = \mathcal{N}(\mathbf{0}, \mathbf{I})$.

To generate new samples from p_0, diffusion models leverage the fact that the forward noising process defined by Equation (1) admits a time-reversal (Anderson, 1982; Haussmann & Pardoux, 1986). For the VP-SDE, this process is given by

$$\mathrm{d}\mathbf{X}_t = -\beta_t \left\{ \frac{1}{2}\mathbf{X}_t + \nabla \log p_t\left(\mathbf{X}_t\right) \right\} \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\mathbf{B}_t, \tag{2}$$

which flows backward in time. $\mathbf{X}_T \sim p_T$ is sampled from the reference distribution and incrementally denoised until it reflects p_0. We have used the shorthand $\nabla \log p_t\left(\mathbf{X}_t\right)$ to denote $\nabla_{\mathbf{x}} \log p_t\left(\mathbf{x}\right)|_{\mathbf{x}=\mathbf{X}_t}$ and will continue using this shorthand in the remainder of the text.

The gradient of the log-density $\nabla \log p_t(\mathbf{x}_t)$ in Equation (2) is estimated by a time-dependent score network $s_\psi(\mathbf{x}_t, t)$ using techniques from score matching (Hyvärinen & Dayan, 2005). Once trained, s_ψ can be used to generate samples from the data distribution p_0 by simulating the reverse-time SDE with a range of numerical solvers (Song et al., 2020).

Guided Diffusion Models. In many applications, we may wish to condition the generative process of diffusion models to produce samples with specific properties $\mathbf{y}$, such as images of a certain class or molecules with a desirable pharmacological effect. That is, we want to modify the denoising process in Equation (2) to incorporate the conditioning information $\mathbf{y}$, and sample from the conditional distribution $p_0(\mathbf{x}_0 \mid \mathbf{y})$ rather than the unconditional data distribution $p_0(\mathbf{x}_0)$. The corresponding reverse-time SDE is given by

$$\mathrm{d}\mathbf{X}_t = -\beta_t \left\{ \frac{1}{2}\mathbf{X}_t + \nabla \log p_t\left(\mathbf{X}_t \mid \mathbf{y}\right) \right\} \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\mathbf{B}_t,$$

differing from Equation (2) only in the need for a conditional score function $\nabla \log p_t(\mathbf{X}_t \mid \mathbf{y})$, which can be expressed as

$$\begin{aligned} \nabla \log p_t(\mathbf{X}_t \mid \mathbf{y}) &= \nabla \log \frac{p_t(\mathbf{y} \mid \mathbf{X}_t) p_t(\mathbf{X}_t)}{Z} \\ &= \nabla \log p_t(\mathbf{y} \mid \mathbf{X}_t) + \nabla \log p_t(\mathbf{X}_t), \end{aligned} \tag{3}$$

since Z is a normalization constant independent of $\mathbf{X}_t$.

To approximate $\nabla \log p_t(\mathbf{y} \mid \mathbf{X}_t)$, a time-dependent discriminative *guidance model* $f_t(\mathbf{x}_t; \theta)$ is trained on a labeled dataset $\mathcal{D} = \cup_{t=1}^{T} \{(\mathbf{x}_t^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$ with inputs corrupted by the same forward noising process as the diffusion model. Once trained, the guidance model can be incorporated into a *guidance function* $p(\mathbf{y} \mid \mathbf{X}_t; f(\cdot; \theta))$ that is used to compute the *conditional scores* $\nabla \log p_t(\mathbf{X}_t \mid \mathbf{y}; f(\cdot; \theta))$, which in turn steer the reverse-time SDE toward samples with a high predicted likelihood of observed label $\mathbf{y}$ via

$$\mathrm{d}\mathbf{X}_t = -\beta_t \left\{ \frac{1}{2}\mathbf{X}_t + \nabla \log p_t(\mathbf{X}_t \mid \mathbf{y}; f_t(\cdot; \theta)) \right\} \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\mathbf{B}_t.$$

Alternative approaches such as classifier-free guidance (Ho & Salimans, 2022) estimate the conditional scores by passing the conditioning information to the score network

$$\nabla \log p_t(\mathbf{X}_t \mid \mathbf{y}) \approx s_\phi(\mathbf{x}_t, \mathbf{z}, t),$$

either in the form of class labels $\mathbf{z} = \mathbf{y}$ (Ho & Salimans, 2022) or pre-trained embeddings $\mathbf{z} \sim p(\mathbf{z}|\mathbf{y})$ (Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022). However, these methods' inability to provide an explicit *guidance function gradient estimate* $\nabla \log p_t(\mathbf{y} \mid \mathbf{X}_t)$ limits their applicability to regression-based optimization problems frequently encountered in molecular and protein design (Lee et al., 2023; Weiss et al., 2023; Gruver et al., 2023), where the objective is to maximize or minimize $\mathbf{y}$ without prior knowledge of its optimal value.

3. Context-Guided Diffusion Models

In this section, we present *context-guided diffusion* (CGD)—a method that produces more robust conditional gradient estimates that consistently generate novel samples with improved properties. We first introduce the prevailing approach to training guidance models and highlight its limitations. We then present a domain-informed regularization term designed to bias guidance model training toward functions that both (i) fit the training data well and (ii) revert to domain-appropriate behavior in out-of-distribution settings. Finally, we describe how to incorporate context-aware guidance models into guided diffusion to generate out-of-distribution samples with desirable properties.

3.1. Standard Guidance Model Training

We consider a standard supervised learning setting with diffusion timesteps $t = 0, ..., T$ and N i.i.d. data realizations $\mathcal{D}_t = \{(\mathbf{x}_t^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$ of the noised inputs $\mathbf{x}_t \in \mathcal{X}$ and targets $\mathbf{y} \in \mathcal{Y}$. Furthermore, we define the guidance model $f_t(\cdot\,; \theta) \doteq \theta_L h_t(\cdot\,; \theta_h)$ with $\theta \doteq \{\theta_h, \theta_L\}$ as the composition of an embedding model $h_t(\cdot\,; \theta_h) : \mathcal{X} \to \mathbb{R}^d$ and a linear output head $\theta_L : \mathbb{R}^d \to \mathcal{Y}$.

Conventional supervised learning techniques for training guidance models define the diffusion timestep-dependent negative log-likelihood objective

$$\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}, \mathcal{T}) \doteq -\sum_{n=1}^{N} \log p_\mathcal{T}(\mathbf{y}^{(n)} \mid \mathbf{x}_\mathcal{T}^{(n)}; f_\mathcal{T}(\cdot; \theta))$$

where $\mathcal{T} \sim p_\mathcal{T}(t)$ is a randomly sampled timestep. The guidance model parameters θ are optimized by minimizing the L_2-regularized objective function

$$\bar{\mathcal{L}}(\theta, \mathcal{D}_\mathcal{T}) \doteq \mathbb{E}_{p_\mathcal{T}}\left[\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}, \mathcal{T})\right] + \frac{1}{2\lambda}||\theta||_2^2, \tag{4}$$

where $\lambda \in \mathbb{R}_+$ is chosen as a hyperparameter.

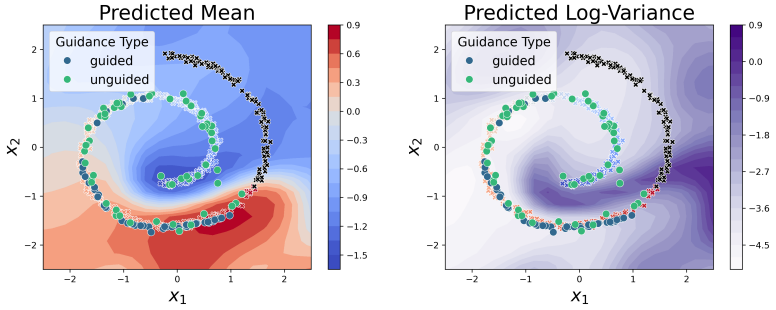
Unfortunately, this approach is only able to operate over subsets of the input domain $\mathcal{X}$ for which labeled data is available. When labels are scarce, as is the case in many practical settings, models trained with standard supervised learning objectives are prone to generating incorrect and overconfident predictions when exposed to distribution shifts (Ovadia et al., 2019; Liu et al., 2020; Van Amersfoort et al., 2020; Koh et al., 2021; Band et al., 2021; Rudner et al., 2021; Tran et al., 2022; Rudner et al., 2022; Klarner et al., 2023). This issue is exacerbated by the fact that guidance models are trained on noised data points: Since neural networks are both highly sensitive to noise perturbations (Szegedy et al., 2013; Goodfellow et al., 2014; Ho & Salimans, 2022) and able to overfit to fully corrupted inputs (Zhang et al., 2021), the errors of systematically overconfident gradient signals may accumulate over hundreds of denoising steps.

Algorithm 1 One guidance model training iteration with the context-guided regularization scheme from Equation (9).

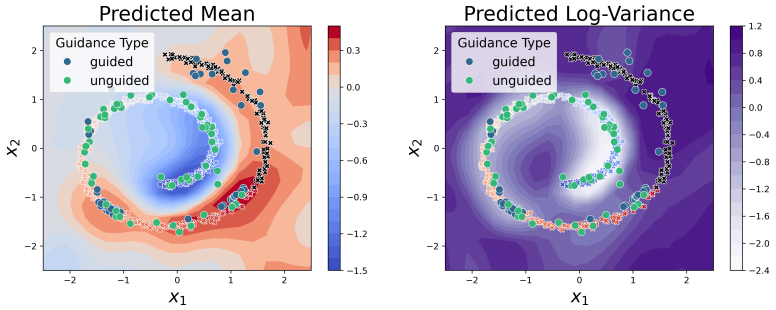
Require:
Data and context batch: $(\mathbf{x}_0, \mathbf{y}), \hat{\mathbf{x}}_0$
Diffusion time steps: $t, t' \sim \mathcal{U}(0, T)$
Guidance model: $f_t(\cdot; \theta)$
Regularizer: $D_{\mathcal{M}}(f_t(\cdot;\theta), Q_t(\cdot))^2$
Hyperparameters: $\sigma_t, \tau_t, \lambda$

$\mathbf{x}_t \leftarrow \text{NoisingProcess}(\mathbf{x}_0, t)$
$\bar{\mathbf{y}} \leftarrow f_t(\mathbf{x}_t; \theta)$
$\bar{\mathcal{L}} \leftarrow -\log p_t(\mathbf{y} \mid \mathbf{x}_t; \bar{\mathbf{y}}) + \frac{1}{2\lambda}||\theta||_2^2$
$\hat{\mathbf{x}}_{t'} \leftarrow \text{NoisingProcess}(\hat{\mathbf{x}}_0, t')$
$\hat{\mathbf{y}} \leftarrow f_{t'}(\hat{\mathbf{x}}_{t'}; \theta)$
$R \leftarrow \sum_{j=1,2} D_{\mathcal{M}}(\hat{\mathbf{y}}^j, Q^j_{t'}(\hat{\mathbf{x}}_{t'}))^2$
$\theta \leftarrow \text{Update}(\theta, \bar{\mathcal{L}} + R)$

(a) An algorithmic overview of the proposed regularization scheme.



(b) Predictions and samples from an L_2-regularized guidance model.



(c) Predictions and samples from a context-aware guidance model.

Figure 3: Panel (a) presents an algorithmic perspective on how the proposed regularization scheme fits into the training loop of a standard guidance model. Panels (b) and (c) illustrate how the regularization scheme affects the functions a guidance model learns. Two regression models with different regularizers are trained on a low-label subset of an illustrative dataset (color-coded crosses). However, without additional information about the input domain, the predictions of the L_2-regularized model (b) exhibit poor generalization and miscalibrated uncertainty estimates when evaluated in new, unseen regions. In contrast, the context-aware model (c) is able to leverage unlabeled data sampled uniformly from $[-2.5, 2.5]^2 \subset \mathbb{R}^2$ to generate more accurate and better-calibrated predictions. The corresponding context-guided denoising process generates samples that recover a held-out, high-label test set (black crosses). Full experimental details are presented in Appendix B.1.

3.2. Context-Aware Guidance Models

To improve the performance and reliability of guidance models in this setting, we propose a family of diffusion timestep-dependent regularizers designed to favor parameters θ that encode desirable model behaviors on out-of-distribution data. While ideal model behaviors under distribution shifts are often application-dependent, we design a general-purpose regularizer—applicable across domains—that encourages uncertain predictions in areas of the input domain with little or no signal from the labeled data. This regularizer maximizes sensitivity at the expense of specificity and enables the conditional denoising process to focus on regions in data space that are near the training data and have the highest likelihood of containing molecules with improved properties.

To encode this behavior, we construct a data- and timestep-dependent distribution $Q_t(\cdot)$ that is explicitly designed to exhibit high predictive uncertainty on a problem-specific, out-of-distribution context set $\hat{\mathbf{x}}_{\mathcal{C}} \doteq \{\mathbf{x}_{\mathcal{C}}^{(i)}\}_{i=1}^{N_{\mathcal{C}}}$.

More specifically, we consider regularizers of the form

$$R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t}) \doteq \mathbb{E}_{p_{\hat{\mathbf{X}}_t}}[D_{\mathcal{M}}(f_t(\hat{\mathbf{x}}_t;\theta), Q_t(\hat{\mathbf{x}}_t))^2], \quad (5)$$

where $p_{\hat{\mathbf{X}}_t}$ is a uniform distribution over noised context batches $\hat{\mathbf{x}}_t$ of size M (with $M \ll N_{\mathcal{C}}$).

$$D_{\mathcal{M}}(f_t(\hat{\mathbf{x}}_t;\theta), Q_t(\hat{\mathbf{x}}_t))^2 \quad (6)$$
$$\doteq (f_t(\hat{\mathbf{x}}_t;\theta) - m_t(\hat{\mathbf{x}}_t))^\top K_t(\hat{\mathbf{x}}_t)^{-1}(f_t(\hat{\mathbf{x}}_t;\theta) - m_t(\hat{\mathbf{x}}_t))$$

is the squared Mahalanobis distance between noised model predictions $f_t(\hat{\mathbf{x}}_t;\theta)$ and a data- and timestep-dependent distribution $Q_t(\hat{\mathbf{x}}_t)$ with mean $m_t(\hat{\mathbf{x}}_t)$ and covariance $K_t(\hat{\mathbf{x}}_t)$. Here, $m_t(\hat{\mathbf{x}}_t)$ specifies the desired out-of-distribution behavior, and $K_t(\hat{\mathbf{x}}_t) \in \mathbb{R}^{M\times M}$ allows us to encode additional information about the structure of the input domain. Specifically, $K_t(\hat{\mathbf{x}}_t) \neq \mathbf{I}$ controls how strongly the guidance model predictions $f_t(\hat{\mathbf{x}}_t;\theta)$ can vary between similar context points, allowing us to enforce smoother gradient estimates and more stable guidance signals in regions of data space with limited label signal.

This covariance function can be defined in terms of any valid, domain-appropriate similarity metric. In the following, we present a general approach to constructing K_t from model embeddings. Specifically, we consider the neural network embeddings $h_t(\hat{\mathbf{x}}_t, \phi) \in \mathbb{R}^{M \times d}$ of all molecules in the context batch $\hat{\mathbf{x}}_t$ and use their inner product to construct

$$K_t(\hat{\mathbf{x}}_t) \doteq \sigma_t h_t(\hat{\mathbf{x}}_t, \phi) h_t(\hat{\mathbf{x}}_t, \phi)^T + \tau_t \mathbf{I}, \tag{7}$$

where σ_t is a covariance scale parameter and τ_t is a diagonal offset parameter. These hyperparameters allow us to calibrate the behavior of our regularizer, with τ_t determining how closely the predictions have to match the mean function $m_t(\hat{\mathbf{x}}_t)$ and σ_t determining the strength of the smoothness constraints placed on $f_t(\hat{\mathbf{x}}_t; \theta)$.

This approach is agnostic to how the embeddings $h_t(\hat{\mathbf{x}}_t, \phi)$ are derived, so long as they capture relevant structural properties of the input domain. We find that a relatively simple approach is already sufficient to produce sizeable performance gains in the applications we consider here. Specifically, we use a fixed set of randomly initialized parameters ϕ to obtain $h_t(\hat{\mathbf{x}}_t, \phi)$ from the model's embedding trunk. This is motivated by the observation that neural networks with suitable inductive biases, such as permutation- or roto-translational equivariance in Sections 5.1 and 5.2, can already produce robust and informative representations in a randomly initialized state (Kipf & Welling, 2016).

Having constructed the covariance matrix $K_t(\hat{\mathbf{x}}_t)$, we specify the mean function $m_t(\hat{\mathbf{x}}_t)$ as follows. Letting $f_t(\cdot; \theta)$ be a standard multi-output regression model consisting of a learned mean function $f_t^1(\hat{\mathbf{x}}_t; \theta)$ and, optionally, a learned variance function $f_t^2(\hat{\mathbf{x}}_t; \theta)$, we wish for the mean predictions to revert to the mean of the training labels and for the variance to be high when the model is evaluated on points that are meaningfully distinct from the training data. To encode this desired behavior, we define the regularizer

$$R(\theta, f_t, t, p_{\hat{\mathbf{x}}_t}) \doteq \mathbb{E}_{p_{\hat{\mathbf{x}}_t}}\left[\sum\nolimits_{j=1}^{2} D_{\mathcal{M}}(f_t^j(\hat{\mathbf{x}}_t; \theta), Q_t^j(\hat{\mathbf{x}}_t))^2\right], \tag{8}$$

where $m_t^1(\hat{\mathbf{x}}_t) \doteq \frac{1}{N}\sum_{n=1}^{N} y^{(n)}$ for the training data labels $y^{(n)}$, $m_t^2(\hat{\mathbf{x}}_t) \doteq \boldsymbol{\sigma}_0^2$, for a target variance parameter $\boldsymbol{\sigma}_0^2$, and $p_{\hat{\mathbf{x}}_t}$ is defined as in Equation (5).

Adding this regularizer to the supervised learning loss from Equation (4), we arrive at the modified training objective

$$\mathcal{L}^*(\theta, \mathcal{D}_{\mathcal{T}}) = \mathbb{E}_{p_{\mathcal{T}}}[\mathcal{L}(\theta, \mathcal{D}_{\mathcal{T}}, \mathcal{T}) + R(\theta, f_{\mathcal{T}}, \mathcal{T}, p_{\hat{\mathbf{X}}_{\mathcal{T}}})], \tag{9}$$

where both expectations (over $p_{\mathcal{T}}$ and $p_{\hat{\mathbf{X}}_t}$) can be estimated via simple Monte Carlo estimation.

Implementation. The proposed regularizer R does not require any changes to the guidance model itself and can be easily integrated into standard guidance model training pipelines: At each iteration, we sample a mini-batch $\hat{\mathbf{x}}_t$ of size M from the context set $\hat{\mathbf{x}}_{\mathcal{C}}$ and perturb it with the same noising process as the unconditional diffusion model in Equation (1). We then perform a gradient-free forward pass of the embedding model to compute $h_t(\hat{\mathbf{x}}_t, \phi)$ and construct the covariance matrix $K_t(\hat{\mathbf{x}}_t)$. Finally, we get the predictions of the guidance model $f_t(\hat{\mathbf{x}}_t; \theta)$ on the context batch to compute the full regularization term in Equation (8). An algorithmic view of this process is shown in Figure 3.

As the risk of miscalibrated gradient estimates is particularly high in the earlier stages of the sample generation process—where highly diffused input points are associated with the labels of the corresponding un-noised points—it is important to ensure that our regularizer is able to take the noise level β_t into account. Specifically, we increase τ_t with the same schedule as the noise levels β_t to penalize overconfident predictions as context points approach the reference distribution $p_T = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Similarly, as the distinction between in-distribution and out-of-distribution data points becomes less meaningful at larger noise levels, we decrease σ_t with an inverted schedule.

3.3. Generation via Context-Guided Diffusion

Once the context-aware guidance model $f_t(\cdot; \theta)$ has been trained, we can use it to steer the denoising process towards out-of-distribution samples that are most likely to exhibit desired properties. To do so, we use the context-aware guidance model to specify a context-aware guidance function $p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot; \theta))$. This function defines a distribution—conditioned on both the training data and the context set—that reflects the probability of obtaining the desired properties given $\mathbf{x}_t$ and allows integrating signals from labeled data points along with structural knowledge about the broader input domain into sampling. Details about guidance function specification are provided in Appendix A, and an illustrative example that showcases the effect of the proposed regularizer on the guidance model (and the resulting guided diffusion process) is shown in Figure 3.

Finally, the context-aware guidance function can be seamlessly integrated into existing guided diffusion model frameworks by replacing the standard guidance function at sampling time. This modular, plug-and-play approach ensures that our method is independent of the specific diffusion model architecture and generative process and requires minimal implementation overhead. Importantly, like conventional guided diffusion, our method does not necessitate any (re-)training or fine-tuning of the diffusion model itself, making it a convenient out-of-the-box tool for guiding the generation process of a pre-trained model.

4. Related Work

Guided Diffusion for Molecular Design. Guiding the denoising process of diffusion models with gradients from a classifier or conditional score network is an effective approach that has been applied to a broad range of scientific problems, such as the generation of desirable protein folds (Watson et al., 2023; Ingraham et al., 2023) and molecular conformations (Hoogeboom et al., 2022; Peng et al., 2023; Guan et al., 2023). Similarly to our approach, recent work has explored property-guided diffusion models for molecular optimization with a particular focus on out-of-distribution sample generation. For instance, Lee et al. (2023) proposed a modification of the unconditional denoising process that, in combination with standard supervised guidance, generated more diverse small molecules with improved properties. Likewise, Weiss et al. (2023) used a guided diffusion model to optimize the electronic properties of polycyclic aromatic systems, demonstrating generalization to unseen heterocycles. Furthermore, Gruver et al. (2023) explored property-guided discrete diffusion for multi-objective protein sequence optimization, balancing naturalness with improved properties through a grid of sampling hyperparameters. However, all of these approaches primarily focus on modifying the unconditional sampling process itself and rely on standard supervised learning techniques for guidance model training. Our method is orthogonal to these contributions and aims to further improve the performance of property-guided diffusion models across application domains and generative processes.

Regularization with Unlabeled Data. Using unlabeled data to improve the performance of predictive models is a well-established paradigm that underlies many successful semi-supervised (Kingma et al., 2014; Laine & Aila, 2016; Berthelot et al., 2019) and self-supervised (Brown et al., 2020; Bommasani et al., 2021) methods. This includes a range of unsupervised domain adaptation and generalization techniques that use unlabeled data to mitigate the adverse effect of distribution shifts (Tzeng et al., 2015; Sun & Saenko, 2016; Ganin et al., 2016; Kang et al., 2019; Li et al., 2020). However, these methods predominantly focus on aligning model embeddings and are, unlike our approach, unable to directly influence a model's predictions. The method most closely related to our approach is the regularization scheme proposed in Rudner et al. (2023). However, in contrast to Rudner et al. (2023), who only consider supervised image classification tasks, our approach is specifically designed for the conditional generation of molecular structures. Key methodological developments in this regard include the introduction of a factorized regularization term that is suitable for use with regression models, as well as an explicit timestep-dependence that accounts for the noising process of the underlying diffusion model.

5. Empirical Evaluation

We compare our method to both standard guidance models and sophisticated pre-training and domain adaptation techniques across a range of experimental settings. Specifically, we demonstrate the versatility of context-guided diffusion by applying it to the design of small molecules with graph-structured diffusion processes (Section 5.1), to the generation of novel materials with equivariant diffusion models (Section 5.2), and to the optimization of discrete protein sequences with categorical diffusion models (Section 5.3). We observe that context-guided diffusion leads to improved conditional sampling processes that consistently and substantially outperform existing methods across diffusion model types and application domains.

5.1. Graph-Structured Diffusion for Small Molecules

To evaluate our method in the context of graph-structured diffusion processes for small molecule generation, we follow the framework of Lee et al. (2023) and consider a discrete diffusion process over the space of molecular graphs. Defining a graph $\mathbf{g}_t$ as a tuple of node feature and adjacency matrices $\mathbf{g}_t = (\mathbf{x}_t, \mathbf{a}_t)$, the forward and reverse processes are given by a joint system of SDEs (Jo et al., 2022). To encourage the exploration of novel chemical space, Lee et al. (2023) condition the denoising process on a hyperparameter $\gamma \in [0, 1)$ that shrinks the unconditional gradients

$$\begin{aligned} \nabla \log p_t(\mathbf{g}_t|\gamma) &= \nabla \log p_t(\mathbf{g}_t) + \nabla \log p_t(\gamma|\mathbf{g}_t) \\ &= (1 - \sqrt{\gamma})\, \nabla \log p_t(\mathbf{g}_t). \end{aligned} \tag{10}$$

In addition, the denoising process is conditioned on desirable molecular properties via a standard supervised guidance model as outlined in Sections 2 and 3.1. This model is trained to predict the labels $\mathbf{y} \in [0, 1]$, defined as a composite of the synthetic accessibility (SA; Ertl & Schuffenhauer, 2009), drug-likeness (QED; Bickerton et al., 2012), and QuickVina docking scores (Alhossary et al., 2015), described in further detail in Appendix B.2. In this setting, larger labels denote more desirable properties.

As the docking scores are protein pocket-dependent, we repeat all experiments across the five different targets chosen by Lee et al. (2023), namely PARP1, FA7, 5HT1B, BRAF and JAK2. We closely follow their experimental setup and train guidance models on a subset of 250 000 small molecules from the ZINC database (Irwin et al., 2012). Specifically, we pre-compute the labels $\mathbf{y}$ for every target and use them to split the data into a low-property training and a high-property validation set. This label split allows us to select regularization hyperparameters that maximize the ability of the guidance models to generalize well to novel, high-value regions of chemical space, serving as a proxy for the desired behavior at sampling time.

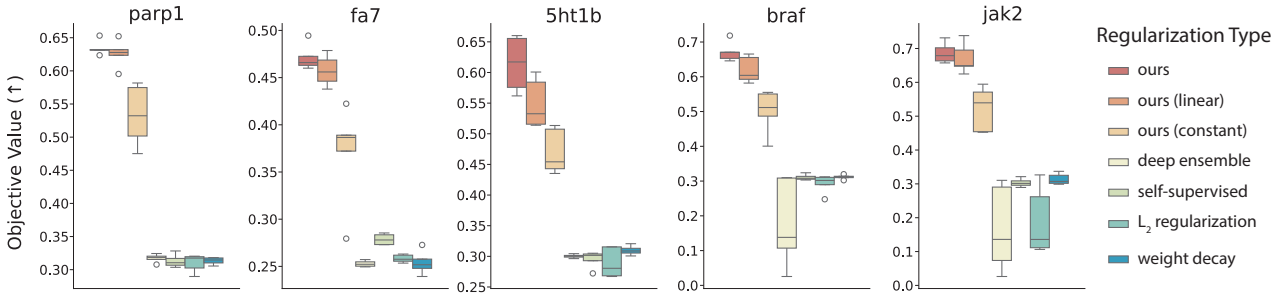


Figure 4: Comparison of the small molecules generated with different guided diffusion models across five distinct protein targets. Objective values (↑) are normalized with respect to the highest score of the held-out high-property validation set and averaged across five independent training and sampling runs with different random seeds.

To construct an appropriate context set for use with our regularizer, we sample an additional 500 000 unlabeled small molecules from the ZINC database. Following Section 3.2, we specify a predictive distribution $Q_t(\hat{\mathbf{x}}_t)$ that reverts to the training set mean and high uncertainty on out-of-distribution context points. We compare this approach to a range of alternative regularization schemes, including weight decay (used in Lee et al. (2023)), explicit L_2 regularization, and an ensemble of independently trained networks (Lakshminarayanan et al., 2017). To facilitate a fair comparison of these methods, we ensure that all training parameters, except regularization type and strength, are kept constant across experiments. After selecting the optimal hyperparameters using the high-property validation set, all models are independently retrained with five different random seeds. We refer to Appendix B.2 for full implementational details, hyperparameter ranges, additional results, and sanity checks.

We additionally evaluate the performance of other approaches that are able to make use of unlabeled data. Specifically, we pre-train a guidance model on all molecules from our context set with a self-supervised denoising objective (Zaidi et al., 2023) and then fine-tune it on the labeled data, using weight decay. Furthermore, we consider a range of unsupervised domain adaptation and generalization techniques, namely DeepCoral (Sun & Saenko, 2016), domain-adversarial neural networks (Ganin et al., 2016), and domain confusion (Tzeng et al., 2015), and present the results for these methods in Appendix B.2.

Adhering to the evaluation protocol of Lee et al. (2023), we use the retrained guidance models to generate 3000 molecules each and derive the corresponding labels with the provided docking and evaluation pipeline. As in Lee et al. (2023), we only retain compounds that pass synthetic accessibility and drug-likeness thresholds and display a maximum ECFP4-based Tanimoto similarity (Rogers & Hahn, 2010) of less than 0.4 to any molecule in the training set, before reporting the top 5% of the remaining docking scores.

To put these absolute values into perspective, we normalize them so that 1 corresponds to the highest docking score in the high-property validation set. The results are reported in Figure 4 and demonstrate that our method consistently improves the performance of property-guided diffusion models across different protein targets. The full distributions of all components of the objective function, the results of models trained with label noise, as well as versions normalized by heavy atom count, are presented in Appendix B.2 and display identical trends. While ensembling and pre-training techniques perform best on in-distribution tasks (see Appendix B.2), we find that they alone are not sufficient to overcome the limitations of standard supervised training/fine-tuning techniques. However, we note that context guidance is an orthogonal approach that is straightforward to use in combination with these methods.

To investigate how important the timestep-dependence of our method is to its empirical performance, we carry out a series of ablations in which the regularization hyperparameters σ_t and τ_t either (i) mimic the β_t schedule of the noising process, (ii) change linearly over time, or (iii) stay constant across all noise scales. We observe a strict performance drop across these settings, indicating that adapting the regularizer to the noising process is highly beneficial (Figure 4).

Furthermore, we investigate the impact that the size and composition of our context set have on the performance of our method by comparing models trained with the full context set to models trained with randomly sampled subsets of size 10% and 1%, respectively. As expected, we observe a strong decrease in performance, with the results of models trained with the smallest context set (1%) reverting to that of standard weight decay. Additionally, we select the 10% of compounds in the original context set that are either most or least similar to the labeled training data and find that using a much smaller set of more similar, near-OOD molecules is able to match the performance of the full context set. Full details and additional results are presented in Table 6.

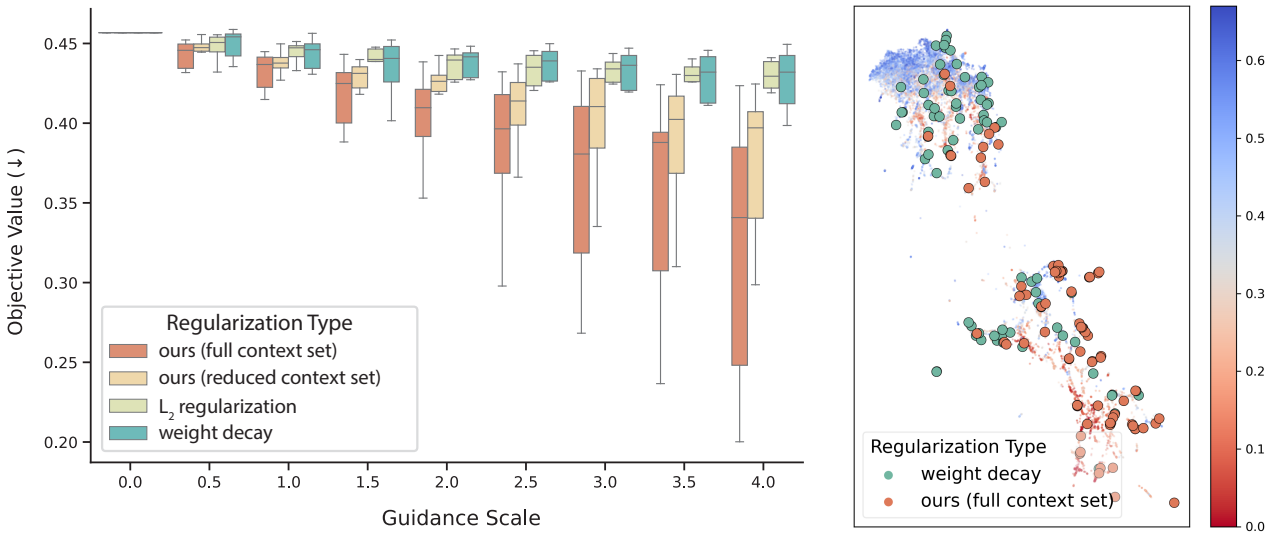


Figure 5: Comparison of polycyclic aromatic systems generated with different guidance models across ten independent training and sampling runs. **Left:** Full distribution of generated objective values (↓) ablated over different context sets and guidance scales. **Right:** UMAP plot (McInnes et al., 2018) of training (upper left) and test set (lower right), as well as samples from guided diffusion models. Validity and novelty are analyzed in Appendix B.3 and show similar trends.

5.2. Equivariant Diffusion For Materials

We additionally evaluate our method by applying it to equivariant diffusion models for materials design. Specifically, we follow the experimental setup of Weiss et al. (2023) and train an E(3)-equivariant diffusion model (Hoogeboom et al., 2022) on all polycyclic aromatic systems consisting of up to 11 cata-condensed benzene rings (Wahab et al., 2022). As in Weiss et al. (2023), we guide the sampling process towards molecules with desirable electronic properties by leveraging the gradients of an E(3)-equivariant graph neural network (Satorras et al., 2021) to minimize a composite objective of adiabatic ionization potential, electron affinity, and HOMO-LUMO gap.

In contrast to the experimental setting in Section 5.1, this application considers an exhaustively enumerated search space, allowing us to construct clustered data splits in which maximal generalization is required to reach held-out regions with optimal objective values. Using this setup, we train a diffusion model and different guidance networks on a low-value training set and examine how well they are able to recover molecules from held-out regions with desirable electronic properties. We present a comparison of different methods in Figure 5, confirming that our approach performs significantly better than standard regularization techniques across guidance scales. We additionally ablate the information content of the context set by removing the most relevant entries, observing that these are a key determinant of our method's success. Full experimental details and additional results are provided in Appendix B.3.

5.3. Discrete Diffusion for Protein Sequences

Furthermore, we explore how well our method extends to categorical diffusion models in the context of protein sequence optimization. In particular, we follow the experimental framework of Gruver et al. (2023) to apply context-guided diffusion to the property-conditioned infilling of antibody complementarity-determining regions. In contrast to the applications in Sections 5.1 and 5.2, the guidance model of Gruver et al. (2023) is not a separate network, but rather a regression head of the score model.Specifically, a masked language model (Austin et al., 2021; Bhargava et al., 2021) is used to learn sequence embeddings from which a linear output head and a feed-forward network estimate the scores and properties, respectively.

At sampling time, Gruver et al. (2023) generate sequences with a grid of denoising hyperparameters and examine the resulting Pareto front of objective values versus "naturalness" i.e. how likely a sequence is to be synthesizable, estimated by its likelihood under a protein language model (Ferruz et al., 2022). Following this approach, we train different guidance models and use them to generate samples with the same hyperparameter grid as Gruver et al. (2023). The corresponding Pareto fronts are presented in Figure 6. While standard methods fare well in in-distribution settings, we observe that models trained with our regularizer outperform them as samples progress into an out-of-distribution regime, consistently producing better properties for a given level of naturalness. Full experimental details and additional results are provided in Appendix B.4.

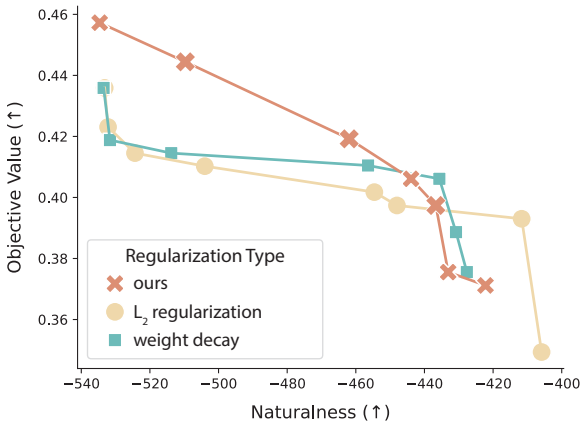


Figure 6: Pareto fronts of samples generated with different regularization schemes, highlighting the trade-off between objective value (↑) and naturalness (↑). As samples move away from the training data and enter an out-of-distribution regime, our method consistently generates sequences with better properties for any given level of naturalness.

Table 1: Ablation over the context batch size used by our regularizer. Larger context batches generally yield better results but incur higher computational costs. We find that the context batch size can be reduced substantially with only a marginal deterioration of model performance. Performance metrics and runtimes are averaged across 5 independent training and sampling runs with different random seeds.

Graph-Structured Diffusion for Small Molecules (Section 5.1)

Batch Size	Batch Time (s)	PARP1 (↑)	FA7 (↑)	5HT1B (↑)
128	0.09±0.01	**0.66**±0.01	**0.48**±0.01	**0.57**±0.01
64	0.08±0.00	**0.66**±0.01	**0.48**±0.01	**0.57**±0.01
32	**0.07**±0.00	**0.66**±0.01	**0.48**±0.01	**0.57**±0.02
16	**0.07**±0.00	0.65±0.02	0.47±0.02	0.56±0.03
8	**0.07**±0.00	0.62±0.03	0.45±0.03	0.53±0.03

Equivariant Diffusion For Materials (Section 5.2)

Batch Size	Batch Time (s)	scale=0 (↓)	scale=2 (↓)	scale=4 (↓)
128	0.28±0.02	0.45±0.00	**0.39**±0.01	**0.34**±0.01
64	0.25±0.03	0.45±0.00	**0.40**±0.01	**0.34**±0.02
32	0.26±0.01	0.45±0.00	0.42±0.01	0.36±0.01
16	0.24±0.02	0.45±0.00	0.41±0.01	0.38±0.03
8	**0.23**±0.01	0.45±0.00	0.42±0.01	0.39±0.02

6. Discussion and Limitations

We conducted a comprehensive empirical evaluation of our method across three distinct molecular and protein design tasks, observing consistent and significant improvements over existing state-of-the-art guided diffusion models. By leveraging unlabeled data to train guidance functions that exhibit high uncertainty and well-behaved gradients on out-of-distribution context points, our context-guided models enabled the more reliable generation of novel molecules with improved properties. These gains were most pronounced when generalization from scarce and biased training data was required to generate compounds from unseen, high-value subsets of chemical and protein sequence space. Our results highlight that even relatively simple regularization techniques can yield substantial empirical benefits when combined with relevant context sets.

The proposed method is best suited for domains in which unlabeled data is abundant or can be easily generated. While it does not incur any computational overhead at sampling time, where hundreds of model evaluations are required to generate a single molecule, it does come with an increased cost during training. Specifically, computing the regularization term requires two additional forward passes per training iteration to generate the context set embeddings and predictions. Fortunately, we found this added cost to be moderate in absolute terms (see Appendix B.5), as guidance models tend to be relatively lightweight. Additionally, we provide a sensitivity study over context batch sizes in Table 1, which suggests that the computational overhead can be further reduced while maintaining favorable empirical performance.

Beyond choosing the context batch size, our approach requires the optimization of two additional hyperparameters: the covariance scale σ_t and the diagonal offset parameter τ_t. Additionally, we find that the choice of the context set significantly impacts model performance, requiring sufficient domain expertise to ensure the inclusion of informative and relevant unlabeled context points.

7. Conclusions

In this paper, we introduced context-guided diffusion, a simple plug-and-play method that leverages unlabeled data and smoothness constraints to improve the out-of-distribution generalization of guided diffusion models. We demonstrated the versatility of this approach by applying it to the design of small molecules, materials, and proteins, achieving substantial performance gains in each domain. Several promising directions for future work exist, such as encoding more complex out-of-distribution behavior into the guidance model, for instance by reverting to the outputs of physics-based methods rather than high predictive uncertainty. Another promising avenue is the construction of maximally informative context sets, potentially through active learning strategies that iteratively select unlabeled data points based on their relevance and potential to improve guidance model performance. Exploring the integration of context-guided diffusion with techniques such as multi-task learning or meta-learning could also enable the development of more versatile guidance functions that leverage knowledge from related domains or tasks. We believe that exploring this line of research has the potential to lead to more adaptable and robust diffusion models that are able to solve a wider range of challenging real-world problems.

Impact Statement

This work introduces a regularization technique to improve the performance of guided diffusion models for inverse molecular design. These methods have the potential to accelerate the discovery and development of new medicines, functional materials, and sustainable alternatives that could benefit society (Wang et al., 2023). However, like any powerful technology, there is also the risk of dual-use or misuse, for example for the generation of hazardous agents that could endanger public safety (Urbina et al., 2022).

Acknowledgements

We thank the anonymous reviewers for their useful feedback. LK acknowledges financial support from the University of Oxford's Clarendon Fund and F. Hoffmann-La Roche AG. We gratefully acknowledge the Oxford Advanced Research Computing service for providing computing resources and infrastructure.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024.

Alhossary, A., Handoko, S. D., Mu, Y., and Kwoh, C.-K. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics*, 31(13):2214–2216, 2015.

Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Baek, J., Kang, M., and Hwang, S. J. Accurate learning of graph representations with graph multiset pooling. *arXiv preprint arXiv:2102.11533*, 2021.

Bajusz, D., Rácz, A., and Héberger, K. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1):1–13, 2015.

Band, N., Rudner, T. G. J., Feng, Q., Filos, A., Nado, Z., Dusenberry, M. W., Jerfel, G., Tran, D., and Gal, Y. Benchmarking Bayesian deep learning on diabetic retinopathy detection tasks. In *Advances in Neural Information Processing Systems 34*, 2021.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

Bhargava, P., Drozd, A., and Rogers, A. Generalization in nli: Ways (not) to go beyond simple heuristics. *arXiv preprint arXiv:2110.01518*, 2021.

Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, 2012.

Bilodeau, C., Jin, W., Jaakkola, T., Barzilay, R., and Jensen, K. F. Generative models for molecular discovery: Recent advances and challenges. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(5):e1608, 2022.

Bohacek, R. S., McMartin, C., and Guida, W. C. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal Research Reviews*, 16 (1):3–50, 1996.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422, 2009.

Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. S. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kKF8_K-mBbS.

De Bortoli, V., Mathieu, E., Hutchinson, M., Thornton, J., Teh, Y. W., and Doucet, A. Riemannian score-based generative modelling. *Advances in Neural Information Processing Systems*, 35:2406–2422, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Dunbar, J. and Deane, C. M. Anarci: antigen receptor numbering and receptor classification. *Bioinformatics*, 32(2):298–300, 2016.

Dunbar, J., Krawczyk, K., Leem, J., Baker, T., Fuchs, A., Georges, G., Shi, J., and Deane, C. M. Sabdab: the structural antibody database. *Nucleic Acids Research*, 42 (D1):D1140–D1146, 2014.

Ertl, P. Cheminformatics analysis of organic substituents: identification of the most common substituents, calculation of substituent properties, and automatic identification of drug-like bioisosteric groups. *Journal of Chemical Information and Computer Sciences*, 43(2):374–380, 2003.

Ertl, P. and Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1:1–11, 2009.

Ferruz, N., Schmidt, S., and Höcker, B. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

Fishman, N., Klarner, L., Bortoli, V. D., Mathieu, E., and Hutchinson, M. J. Diffusion models for constrained domains. *Transactions on Machine Learning Research*, 2023a. ISSN 2835-8856. URL https://openreview.net/forum?id=xuWTFQ4VGO. Expert Certification.

Fishman, N., Klarner, L., Mathieu, E., Hutchinson, M. J., and Bortoli, V. D. Metropolis sampling for constrained diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=jzseUq55eP.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E., et al. The chembl database in 2017. *Nucleic acids research*, 45(D1):D945–D954, 2017.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gruver, N., Stanton, S., Frey, N. C., Rudner, T. G., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., and Wilson, A. G. Protein design with guided discrete diffusion. *arXiv preprint arXiv:2305.20009*, 2023.

Guan, J., Zhou, X., Yang, Y., Bao, Y., Peng, J., Ma, J., Liu, Q., Wang, L., and Gu, Q. DecompDiff: Diffusion models with decomposed priors for structure-based drug design. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11827–11846. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/guan23a.html.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Haussmann, U. G. and Pardoux, E. Time reversal of diffusions. *The Annals of Probability*, pp. 1188–1205, 1986.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

Huang, C.-W., Aghajohari, M., Bose, J., Panangaden, P., and Courville, A. C. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35: 2750–2761, 2022.

Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Hyvärinen, A. and Dayan, P. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Ingraham, J. B., Baranov, M., Costello, Z., Barber, K. W., Wang, W., Ismail, A., Frappier, V., Lord, D. M., Ng-Thow-Hing, C., Van Vlack, E. R., et al. Illuminating protein space with a programmable generative model. *Nature*, pp. 1–9, 2023.

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. ZINC: a free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012.

Isert, C., Atz, K., Jiménez-Luna, J., and Schneider, G. Qmugs, quantum mechanical properties of drug-like molecules. *Scientific Data*, 9(1):273, 2022.

Ji, Y., Zhang, L., Wu, J., Wu, B., Li, L., Huang, L.-K., Xu, T., Rong, Y., Ren, J., Xue, D., et al. Drugood: Out-of-distribution dataset curator and benchmark for ai-aided drug discovery–a focus on affinity prediction problems with noise annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8023–8031, 2023.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. S. Torsional diffusion for molecular conformer generation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=w6fj2r62r_H.

Jo, J., Lee, S., and Hwang, S. J. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*, pp. 10362–10383. PMLR, 2022.

Kadkhodaie, Z., Guth, F., Simoncelli, E. P., and Mallat, S. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *The Twelfth International Conference on Learning Representations*, 2024.

Kang, G., Jiang, L., Yang, Y., and Hauptmann, A. G. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4893–4902, 2019.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Klarner, L., Rudner, T. G., Reutlinger, M., Schindler, T., Morris, G. M., Deane, C., and Teh, Y. W. Drug discovery under covariate shift with domain-informed prior distributions over functions. In *International Conference on Machine Learning*, pp. 17176–17197. PMLR, 2023.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

Landrum, G. et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8:31, 2013.

Lee, S., Jo, J., and Hwang, S. J. Exploring chemical space with score-based out-of-distribution generation. In *International Conference on Machine Learning*, pp. 18872–18892. PMLR, 2023.

Li, R., Jiao, Q., Cao, W., Wong, H.-S., and Wu, S. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9641–9650, 2020.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in neural information processing systems*, 33:7498–7512, 2020.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lou, A. and Ermon, S. Reflected diffusion models. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22675–22701. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/lou23a.html.

Marsland, S. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2011.

Maynard Smith, J. Natural selection and the concept of a protein space. *Nature*, 225(5232):563–564, 1970.

McInnes, L., Healy, J., and Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Mysinger, M. M., Carchia, M., Irwin, J. J., and Shoichet, B. K. Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking. *Journal of Medicinal Chemistry*, 55(14):6582–6594, 2012.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814, 2010.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Olsen, T. H., Boyles, F., and Deane, C. M. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science*, 31(1):141–146, 2022.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

pandas development team, T. pandas-dev/pandas: Pandas, February 2020. URL https://doi.org/10.5281/zenodo.3509134.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Peng, X., Guan, J., Liu, Q., and Ma, J. MolDiff: Addressing the atom-bond inconsistency problem in 3D molecule diffusion generation. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27611–27629. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/peng23b.html.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50 (5):742–754, 2010.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Rudner, T. G., Chen, Z., Teh, Y. W., and Gal, Y. Tractable function-space variational inference in Bayesian neural networks. *Advances in Neural Information Processing Systems*, 35:22686–22698, 2022.

Rudner, T. G., Kapoor, S., Qiu, S., and Wilson, A. G. Function-space regularization in neural networks: A probabilistic perspective. In *International Conference on Machine Learning*, pp. 29275–29290. PMLR, 2023.

Rudner, T. G. J., Lu, C., Osborne, M. A., Gal, Y., and Teh, Y. W. On pathologies in KL-regularized reinforcement learning from expert demonstrations. In *Advances in Neural Information Processing Systems 34*, 2021.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022.

Sanchez-Lengeling, B. and Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, pp. 9323–9332. PMLR, 2021.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Stiefl, N., Watson, I. A., Baumann, K., and Zaliani, A. Erg: 2d pharmacophore descriptions for scaffold hopping. *Journal of Chemical Information and Modeling*, 46(1): 208–220, 2006.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer, 2016.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tingle, B. I., Tang, K. G., Castanon, M., Gutierrez, J. J., Khurelbaatar, M., Dandarchuluun, C., Moroz, Y. S., and Irwin, J. J. ZINC-22 - a free multi-billion-scale database of tangible compounds for ligand discovery. *Journal of Chemical Information and Modeling*, 63(4):1166–1176, 2023.

Tran, D., Liu, J., Dusenberry, M. W., Phan, D., Collier, M., Ren, J., Han, K., Wang, Z., Mariet, Z., Hu, H., Band, N., Rudner, T. G. J., Singhal, K., Nado, Z., van Amersfoort, J., Kirsch, A., Jenatton, R., Thain, N., Yuan, H., Buchanan, K., Murphy, K., Sculley, D., Gal, Y., Ghahramani, Z., Snoek, J., and Lakshminarayanan, B. Plex: Towards reliability using pretrained large model extensions. In *ICML Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward*, 2022.

Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pp. 4068–4076, 2015.

Urbina, F., Lentzos, F., Invernizzi, C., and Ekins, S. Dual use of artificial-intelligence-powered drug discovery. *Nature Machine Intelligence*, 4(3):189–191, 2022.

Van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pp. 9690–9700. PMLR, 2020.

Van Rossum, G. and Drake, F. L. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Wahab, A., Pfuderer, L., Paenurk, E., and Gershoni-Poranne, R. The compas project: A computational database of polycyclic aromatic systems. phase 1: cata-condensed polybenzenoid hydrocarbons. *Journal of Chemical Information and Modeling*, 62(16):3704–3713, 2022.

Wang, H., Fu, T., Du, Y., Gao, W., Huang, K., Liu, Z., Chandak, P., Liu, S., Van Katwyk, P., Deac, A., et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

Waskom, M. L. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976): 1089–1100, 2023.

Weiss, T., Mayo Yanes, E., Chakraborty, S., Cosmo, L., Bronstein, A. M., and Gershoni-Poranne, R. Guided diffusion for inverse molecular design. *Nature Computational Science*, 3(10):873–882, 2023.

Xu, M., Powers, A. S., Dror, R. O., Ermon, S., and Leskovec, J. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*, pp. 38592–38610. PMLR, 2023.

Zaidi, S., Schaarschmidt, M., Martens, J., Kim, H., Teh, Y. W., Sanchez-Gonzalez, A., Battaglia, P., Pascanu, R., and Godwin, J. Pre-training via denoising for molecular property prediction. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.

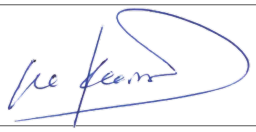
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Context-Guided Diffusion for Out-of-Distribution Molecular and Protein Design
Publication Status	Published
Publication Details	Klarner, L., Rudner, T.G., Morris, G.M., Deane, C. and Teh, Y.W., 2024, July. Context-Guided Diffusion for Out-of-Distribution Molecular and Protein Design. In *International Conference on Machine Learning* (pp. 24770-24807). PMLR.

Student Confirmation

Student Name	Leo Klarner		
Contribution	• Conceptualized key research ideas and methodological contributions. • Identified suitable modelling frameworks and relevant datasets that allowed for a meaningful empirical evaluation of the proposed method. • Carried out all experiments, including data pre-processing, model training, sample generation and the analysis and visualization of results. • Developed the framing and presentation of the method in collaboration with all other co-authors. Drafted the manuscript and improved it in collaboration with all other co-authors. Engaged with reviewers during the peer review process.		
Signature		Date	Lausanne, 14. April 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Yee-Whye Teh		
Supervisor comments: I confirm.		
Signature 	Date	23 April 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 4

Diffusion Models for Constrained Domains

This chapter is based on the following publication:

N. Fishman, **L. Klarner**, V. De Bortoli, E. Mathieu, M. Hutchinson. Diffusion Models for Constrained Domains. *Trans. on Machine Learning Research, 2023.*

Personal contributions:

1. Conceptualised key research ideas in collaboration with all other co-authors.
2. Identified relevant datasets and baselines that allowed for a meaningful empirical evaluation of the proposed method. Collaborated with NF in developing practical implementations and code.
3. Collaborated with NF, VDB and MH in the development of specialised, problem-specific adaptations of the proposed approaches.
4. Collaborated with all other co-authors in developing the framing of the method and drafting of the manuscript.
5. Engaged with reviewers during the peer review process in collaboration with all other co-authors.

A full statement of contributions can be found at the end of this chapter.

The supplementary information for this chapter is provided in Appendix B.

Diffusion Models for Constrained Domains

Nic Fishman *njwfish@gmail.com*
Department of Statistics, University of Oxford

Leo Klarner *leo.klarner@stats.ox.ac.uk*
Department of Statistics, University of Oxford

Valentin De Bortoli *valentin.debortoli@gmail.com*
CNRS, ENS Ulm

Emile Mathieu *ebm32@cam.ac.uk*
Department of Engineering, University of Cambridge

Michael Hutchinson *michael.hutchinson@stats.ox.ac.uk*
Department of Statistics, University of Oxford

Reviewed on OpenReview: *https://openreview.net/forum?id=xuWTFQ4VGO*

Abstract

Denoising diffusion models are a novel class of generative algorithms that achieve state-of-the-art performance across a range of domains, including image generation and text-to-image tasks. Building on this success, diffusion models have recently been extended to the Riemannian manifold setting, broadening their applicability to a range of problems from the natural and engineering sciences. However, these Riemannian diffusion models are built on the assumption that their forward and backward processes are well-defined for all times, preventing them from being applied to an important set of tasks that consider manifolds defined via a set of inequality constraints. In this work, we introduce a principled framework to bridge this gap. We present two distinct noising processes based on (i) the *logarithmic barrier* metric and (ii) the *reflected* Brownian motion induced by the constraints. As existing diffusion model techniques cannot be applied in this setting, we derive new tools to define such models in our framework. We then demonstrate the practical utility of our methods on a number of synthetic and real-world tasks, including applications from robotics and protein design.

1 Introduction

Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Song et al., 2021; Ho et al., 2020) have recently been introduced as a powerful new paradigm for generative modelling. They work as follows: noise is progressively added to data following a Stochastic Differential Equation (SDE)—the forward *noising* process—until it is approximately Gaussian. The generative model is given by an approximation of the associated *time-reversed* process called the backward *denoising* process. This is also an SDE whose drift depends on the gradient of the logarithmic densities of the forward process, referred to as the Stein score. This score is approximated by leveraging techniques from deep learning and score matching (Hyvärinen, 2005; Vincent, 2011). Building on the success of diffusion models in domains such as images and text, this framework has recently been extended to a wide range of Riemannian manifolds (De Bortoli et al., 2022; Huang et al., 2022), broadening their applicability to various important modelling domains from the natural and engineering sciences—including Lie groups such as the group of rotations SO(3), the group of rigid body motions SE(3), and many others (see e.g. Trippe et al., 2022; Corso et al., 2022; Watson et al., 2022; Leach et al., 2022; Urain et al., 2022; Yim et al., 2023).

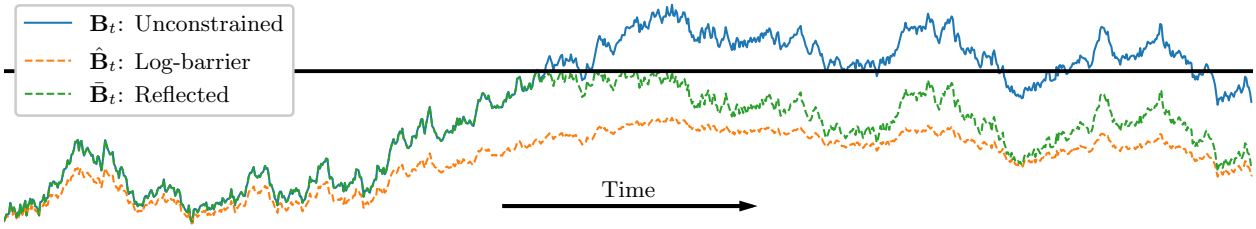


Figure 1: The behaviour of different types of noising processes considered in this work defined on the unit interval. $\mathbf{B}_t$: Unconstrained (Euclidean) Brownian motion. $\hat{\mathbf{B}}_t$: Log-barrier forward noising process. $\bar{\mathbf{B}}_t$: Reflected Brownian motion. All sampled with the same initial point and driving noise. Black line indicates the boundary.

However, a key assumption of the Riemannian diffusion models introduced in De Bortoli et al. (2022) and Huang et al. (2022) is that the stochastic processes they consider are defined *for all times*. While this holds for a large class of stochastic processes, it is not the case for most manifolds defined via a set of inequality constraints. For instance, in the case of the hypercube $(-1, 1)^d$ equipped with the Euclidean metric, the Riemannian Brownian motion coincides with the Euclidean d-dimensional Brownian motion $(\mathbf{B}_t)_{t\in[0,T]}$ as long as $\mathbf{B}_t \in (-1, 1)$. With probability one, $(\mathbf{B}_t)_{t\geq 0}$ escapes from $(-1, 1)$, meaning that the Riemannian Brownian motion is not defined for all times and the frameworks introduced in De Bortoli et al. (2022) and Huang et al. (2022) do not apply. Such constrained manifolds comprise a wide variety of settings—including polytopes and convex sets of Euclidean spaces—and are studied across a large number of disciplines, ranging from computational statistics (Morris, 2002), over robotics (Han & Rudolph, 2006) and quantum physics (Lukens et al., 2020), to computational biology (Thiele et al., 2013). Deriving principled diffusion models that are able to operate directly on these manifolds is thus of significant practical importance, as they enable generative modelling in data-scarce and safety-critical settings in which constraints on the modelled domain may reduce the number of degrees of freedom or prevent unwanted behaviour.

As sampling problems on such manifolds are important (Kook et al., 2022; Heirendt et al., 2019), a flurry of Markov chain based methods have been developed to sample from unnormalised densities. Successful algorithms include the reflected Brownian motion (Williams, 1987; Petit, 1997; Shkolnikov & Karatzas, 2013), log-barrier methods (Kannan & Narayanan, 2009; Lee & Vempala, 2017; Noble et al., 2022; Kook et al., 2022; Gatmiry & Vempala, 2022; Lee & Vempala, 2018) and hit-and-run approaches in the case of polytopes (Smith, 1984; Lovász & Vempala, 2006). In this work, we study the generative modelling counterparts of these algorithms through the lens of diffusion models. Among existing methods for statistical sampling on constrained manifolds, the geodesic Brownian motion (Lee & Vempala, 2017) and the reflected Brownian motion (Williams, 1987) are continuous stochastic processes, and thus well suited for extending the continuous Riemannian diffusion framework developed by De Bortoli et al. (2022) and Huang et al. (2022). In particular, we introduce two principled diffusion models for generative modelling on constrained domains based on (i) the geodesic Brownian motion, leveraging tools from the log-barrier methods, and (ii) the reflected Brownian motion. In both cases, we show how one can extend the ideas of time-reversal and score matching to these settings. We demonstrate the practical utility of these methods on a range of tasks defined on convex polytopes and the space of symmetric positive definite matrices, including the constrained conformational modelling of proteins and robotic arms. The code for all of our experiments is available here.

2 Background

Riemannian manifolds. A Riemannian manifold is a tuple $(\mathcal{M}, \mathfrak{g})$ with $\mathcal{M}$ a smooth manifold and $\mathfrak{g}$ a metric which defines an inner product on tangent spaces. The metric $\mathfrak{g}$ induces key quantities on the manifold, such as an exponential map $\exp_x : \mathrm{T}_x\mathcal{M} \to \mathcal{M}$, defining the notion of following straight lines on manifolds, a gradient operator ∇[1] and a divergence operator div[2]. It also induces the Laplace-Beltrami operator Δ and

[1]The (Riemannian) gradient ∇ is defined s.t. for any smooth $f \in \mathrm{C}^\infty(\mathcal{M})$, $x \in \mathcal{M}, v \in \mathrm{T}_x\mathcal{M}$, $\mathfrak{g}(x)(f(x), v) = \mathrm{d}f(x)(v)$.
[2]The Riemannian divergence acts vector fields and can be defined using the volume form of $\mathcal{M}$.

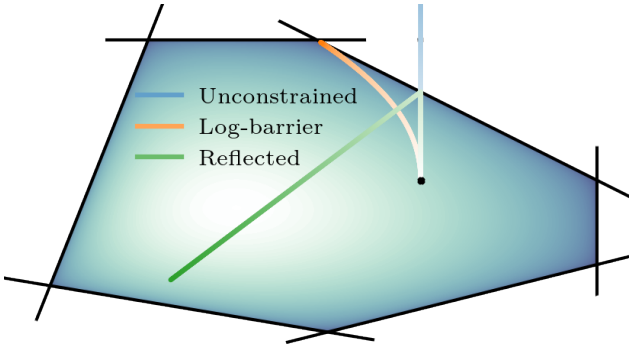


Figure 2: A convex polytope defined by six constraints $\{f_i\}_{i\in\mathcal{I}}$, along with the log barrier potential, and 'straight trajectories' under the log-barrier metric and under the Euclidean metric with and without reflection at the boundary.

consequently a Brownian motion with density (w.r.t. the volume form[3]), whose density is given by the heat equation $\partial_t p_t = \Delta p_t$. We refer the reader to Appendix A for a brief introduction to differential geometry, to Lee (2013) for a thorough treatment and to Hsu (2002) for details on stochastic analysis on manifolds.

Constrained manifolds. In this work, we are concerned with *constrained* manifolds. More precisely, given a Riemannian manifold $(\mathcal{N}, \mathfrak{h})$, we consider a family of real functions $\{f_i : \mathcal{N} \to \mathbb{R}\}_{i\in\mathcal{I}}$ indexed by $\mathcal{I}$. We then define

$$\mathcal{M} = \{x \in \mathcal{N} \,:\, f_i(x) < 0,\ i \in \mathcal{I}\}. \tag{1}$$

In this scenario, $\{f_i\}_{i\in\mathcal{I}}$ is interpreted as a set of constraints on $\mathcal{N}$. For example, choosing $\mathcal{N} = \mathbb{R}^d$ and affine constraints $f_i(x) = \langle a_i, x\rangle - b_i$, $x \in \mathbb{R}^d$, we get that $\mathcal{M}$ is an open polytope as illustrated in Figure 2. This setting naturally appears in many areas of engineering, biology, and physics (Boyd et al., 2004; Han & Rudolph, 2006; Lukens et al., 2020).

While the two methods we introduce in Section 3 can be applied to the general framework (1), in our applications, we focus on two specific settings: (a) Polytopes—$\mathcal{N} = \mathbb{R}^d$ and (b) symmetric positive-definite (SPD) matrices under trace constraints—$\mathcal{N} = \mathcal{S}^d_{++}$.

Continuous diffusion models. We briefly recall the framework for constructing continuous diffusion processes introduced by Song et al. (2021) in the context of generative modelling over $\mathbb{R}^d$. At minimum diffusion models need four things: (i) A forward noising process converging to an invariant distribution. (ii) A time reversal for the reverse process. (iii) A discretization of the continuous-time process for the forward/reverse process. (iv) A score matching loss — in this paper we will focus on the implicit score matching loss. Song et al. (2021) consider a forward *noising process* $(\mathbf{X}_t)_{t\in[0,T]}$ which progressively noises a data distribution p_0 into a Gaussian $\mathrm{N}(0, \mathrm{Id})$. More precisely $(\mathbf{X}_t)_{t\in[0,T]}$ is an Ornstein–Uhlenbeck (OU) process which is given by the following stochastic differential equation (SDE)

$$\mathrm{d}\mathbf{X}_t = -\tfrac{1}{2}\mathbf{X}_t \mathrm{d}t + \mathrm{d}\mathbf{B}_t, \qquad \mathbf{X}_0 \sim p_0.$$

Under mild conditions on p_0, the time-reversed process $(\overleftarrow{\mathbf{X}}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ also satisfies an SDE (Cattiaux et al., 2021; Haussmann & Pardoux, 1986) given by

$$\mathrm{d}\overleftarrow{\mathbf{X}}_t = \{\tfrac{1}{2}\overleftarrow{\mathbf{X}}_t + \nabla \log p_{T-t}(\overleftarrow{\mathbf{X}}_t)\}\mathrm{d}t + \mathrm{d}\mathbf{B}_t, \qquad \overleftarrow{\mathbf{X}}_0 \sim p_T, \tag{2}$$

where p_t denotes the density of $\mathbf{X}_t$. This construction allows direct sampling of the forward process and leverages the Euler-Maruyama discretisation to facilitate sampling of the reverse process. Finally, the quantity $\nabla \log p_t$ is referred as the Stein score and is unavailable in practice. It can be approximated with a score network $s_\theta(t, \cdot)$ trained by minimising a denoising score matching (dsm) loss

$$\mathcal{L}(\theta) = \mathbb{E}[\lambda_t \left\| \nabla \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0) - s_\theta(t, \mathbf{X}_t)\right\|^2], \tag{3}$$

[3]We assume that $\mathcal{M}$ is orientable and therefore that a volume form exists.

or an equivalent implicit score matching (ism) loss

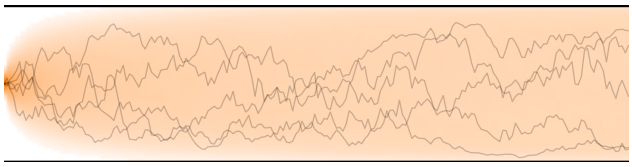
$$\mathcal{L}(\theta) = \mathbb{E}[\lambda_t\{\tfrac{1}{2}\|s_\theta(t, \mathbf{X}_t)\|^2 + \mathrm{div}(s_\theta)(t, \mathbf{X}_t)\}] + C, \tag{4}$$

where $C \geq 0$ and $\lambda_t > 0$ is a weighting function, and the expectation is taken over $t \sim \mathcal{U}([0, T])$ and $(\mathbf{X}_0, \mathbf{X}_t)$. For an arbitrarily flexible score network, the global minimiser $\theta^\star = \mathrm{argmin}_\theta \mathcal{L}(\theta)$ satisfies $s_{\theta^\star}(t, \cdot) = \nabla \log p_t$.

3 Inequality-Constrained Diffusion Models

We are now ready to introduce our methodology to deal with manifolds defined via *inequality constraints* (1). In Section 3.1, we propose a Riemannian diffusion model endowed with a metric induced by a log-barrier potential. In Section 3.2, we introduce a *reflected* diffusion model. While both models extend classical diffusion models to inequality-constrained settings, they exhibit very different behaviours. We discuss their practical differences in Section 5.1.

3.1 Log-barrier diffusion models



$$\mathrm{d}\mathbf{X}_t = \tfrac{1}{2}\mathrm{div}(\mathfrak{g}^{-1})(\mathbf{X}_t)\mathrm{d}t + \mathfrak{g}^{-1/2}(\mathbf{X}_t)\mathrm{d}\mathbf{B}_t$$

Figure 3: Convergence of the Barrier Langevin dynamics on the unit interval to the uniform distribution.

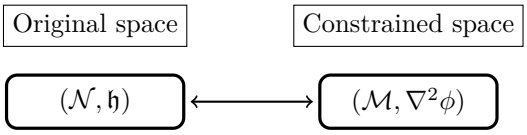


Figure 4: Illustrative diagram of the barrier method and the change of metric.

Barrier Langevin dynamics. Barrier methods work by constructing a smooth potential $\phi : \mathcal{M} \to \mathbb{R}$ such that it blows up on the boundary of a desired set, see Nesterov et al. (2018). Such potentials form the basis of interior point methods in convex optimisation (Boyd et al., 2004). Of these functions, the *logarithmic barrier* is the most popular among practitioners (Lee & Vempala, 2017). For a convex polytope $\mathcal{M}$ defined by the constraints $\mathrm{A}x < b$, with $\mathrm{A} \in \mathbb{R}^{m \times d}$ and $b \in \mathbb{R}^m$, the logarithmic barrier $\phi : \mathcal{M} \to \mathbb{R}_+$ is given for any $x \in \mathcal{M}$ by

$$\phi(x) = -\textstyle\sum_{i=1}^m \log(\langle \mathrm{A}_i, x\rangle - b_i). \tag{5}$$

Assuming that $\|\mathrm{A}_i\| = 1$, we have that for any $x \in \mathcal{M}$, $\phi(x) = -\sum_{i=1}^m \log(d(x, \partial\mathcal{M}_i))$, where $\partial\mathcal{M}_i = \{x \in \mathbb{R}^d : \langle \mathrm{A}_i, x\rangle = b\}$. More generally we can define for any $x \in \mathcal{M}$

$$\phi(x) = -\textstyle\sum_{i=1}^m \log(d(x, \partial\mathcal{M}_i))$$

where $d(x, \partial\mathcal{M}_i)$ computes the minimum geodesic distance from x to the boundary $\partial\mathcal{M}$. In general this is a highly non-trivial optimization problem, contrary to the polytope case which admits a simple closed form.

While developed and most commonly used in optimisation, barrier methods can also be used for sampling (Lee & Vempala, 2017). The core idea of barrier methods is to 'warp the geometry' of the constrained space, stretching it as the process approaches the boundary so that it never hits it, hence bypassing the need to explicitly deal with the boundary. Assuming ϕ to be strictly convex and smooth, its Hessian $\nabla^2\phi$ is positive definite and thus defines a valid Riemannian metric on $\mathcal{M}$. The formal approach to 'warping the geometry' of the convex space with the boundary is to endow $\mathcal{M}$ with the Hessian as a Riemannian metric $\mathfrak{g} = \nabla^2\phi$, making it into a Hessian Manifold, see Shima & Yagi (1997). In the special case where the barrier is given by (5), we get for any $x \in \mathcal{M}$

$$\mathfrak{g}(x) = \mathrm{A}^\top S^{-2}(x)\mathrm{A} \text{ with } S(x) = \mathrm{diag}(b_i - \langle \mathrm{A}_i, x\rangle)_i.$$

Equipped with this Riemannian metric, we consider the following Langevin dynamics as a forward process

$$\mathrm{d}\mathbf{X}_t = \tfrac{1}{2}\mathrm{div}(\mathfrak{g}^{-1})(\mathbf{X}_t)\mathrm{d}t + \mathfrak{g}(\mathbf{X}_t)^{-\frac{1}{2}}\mathrm{d}\mathbf{B}_t, \tag{6}$$

with $\mathrm{div}(F)(x) \triangleq (\mathrm{div}(F_1)(x), \dots, \mathrm{div}(F_d)(x))^\top$, for any smooth $F : \mathbb{R}^d \to \mathbb{R}^d$. Under mild assumptions on $\mathcal{M}$, $\mathrm{div}(\mathfrak{g}^{-1})$ and $\mathfrak{g}^{-1}$ we get that $(\mathbf{X}_t)_{t\geq 0}$ is well-defined and for any $t \geq 0$, $\mathbf{X}_t \in \mathcal{M}$. In particular, for any $t \geq 0$, $\mathbf{X}_t$ does not reach the boundary. This stochastic process was first proposed by Lee & Vempala (2017) in the context of efficient sampling from the uniform distribution over a polytope. Under similar conditions, $\mathbf{X}_t$ admits a density p_t w.r.t. the Lebesgue measure and we have that $\partial_t p_t = \frac{1}{2}\mathrm{Tr}(\mathfrak{g}^{-1}\nabla^2 p_t)$. In addition, $(\mathbf{X}_t)_{t\geq 0}$ is irreducible. Hence, assuming that $\mathcal{M}$ is compact, the uniform distribution on $\mathcal{M}$ is the unique invariant measure of the process $(\mathbf{X}_t)_{t\geq 0}$ and $(\mathbf{X}_t)_{t\geq 0}$ converges to the uniform distribution in some sense. We refer the reader to Appendix C for a proof of these results.

Time-reversal. Assuming that $\mathfrak{g}^{-1}$ and its derivative are bounded on $\mathcal{M}$, the time-reversal of (6) is given by Cattiaux et al. (2021), in particular we have

$$\begin{aligned}\mathrm{d}\overleftarrow{\mathbf{X}}_t &= [-\tfrac{1}{2}\mathrm{div}(\mathfrak{g}^{-1}) + \mathrm{div}(\mathfrak{g}^{-1}) + \mathfrak{g}^{-1}\nabla \log p_{T-t}](\overleftarrow{\mathbf{X}}_t)\mathrm{d}t + \mathfrak{g}(\overleftarrow{\mathbf{X}}_t)^{-\frac{1}{2}}\mathrm{d}\mathbf{B}_t, \\ &= \left[\tfrac{1}{2}\mathrm{div}(\mathfrak{g}^{-1}) + \mathfrak{g}^{-1}\nabla \log p_{T-t}\right](\overleftarrow{\mathbf{X}}_t)\mathrm{d}t + \mathfrak{g}(\overleftarrow{\mathbf{X}}_t)^{-\frac{1}{2}}\mathrm{d}\mathbf{B}_t. \end{aligned} \tag{7}$$

$\overleftarrow{\mathbf{X}}_0$ is initialised with the uniform distribution on $\mathcal{M}$ (which is close to the one of $\mathbf{X}_T$ for large T).

The estimation of the score term $\nabla \log p_t$ is done by minimising the ism loss function (4). We refer to Section 3.3 for details on the training and parameterisation.

Sampling. Sampling from the forward (6) and backward (7) processes, once the score is learnt, requires a discretisation scheme. We use Geodesic Random Walks (GRW) (Jørgensen, 1975) for this purpose, see Algorithm 1. This discretisation is a generalisation of the Euler-Maruyama discretisation of SDE in Euclidean spaces, where the $+$ operator is replaced by the exponential mapping on the manifold, computing the geodesics.

Algorithm 1 *Geodesic Random Walk.* Discretisation of the SDE $\mathrm{d}\mathbf{X}_t = d(t, \mathbf{X}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t$.

Require: T (simulation time), N (number of steps), X_0^γ (initial point), d (drift function)
$\gamma = T/N$
for $k \in \{0, \dots, N-1\}$ **do**
  $Z_{k+1} \sim \mathrm{N}(0, \mathrm{Id})$
  $W_{k+1} = \gamma d(k\gamma, X_k) + \sqrt{\gamma} Z_{k+1}$
  $X_{k+1}^\gamma = \exp_{X_k}[W_{k+1}] \approx \mathrm{proj}_{\mathcal{M}}(X_k + W_{k+1})$
return $\{X_k\}_{k=0}^N$

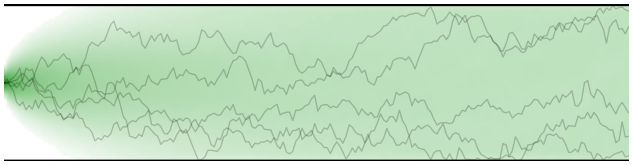
However, contrary to De Bortoli et al. (2022), we will not have access explicitly to the exponential mapping of the Hessian manifold. Instead, we rely on an approximation, using a *retraction* (see Absil & Malick (2012); Boumal (2023) for a definition and alternative schemes).

3.2 Reflected diffusion models

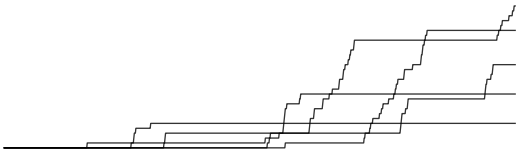
Another approach to deal with the geometry of $\mathcal{M}$ is to use the standard metric $\mathfrak{h}$ and forward dynamics of $\mathcal{N}$ and constraining it to $\mathcal{M}$ by *reflecting* the process whenever it would encounter a boundary. We will first assume that $\mathcal{M}$ is compact and convex. To simplify the presentation, we focus on the Euclidean case $\mathcal{N} = \mathbb{R}^d$ with smooth boundary ∂M[4].

The key difference between this approach and the barrier approach is that in the reflected case we leave the geometry unchanged, so all we need to do is show that the dynamics induced by reflecting the forward process whenever it hits the boundary leads to an invariant distribution and admits a time-reversal.

[4]We refer to Appendix G for a definition of smooth boundary.



$\mathrm{d}\bar{\mathbf{B}}_t = \mathrm{d}\mathbf{B}_t - \mathrm{d}\mathbf{k}_t$



$|\mathbf{k}|_t$

Figure 5: *Left:* Convergence of the reflected Brownian motion on the unit interval to the uniform distribution. *Right:* Value of $|\mathbf{k}|_t$ for the trajectory samples on the left through time.

It is worth noting that while barrier approaches have received considerable theoretical attention in the sampling literature (Lee & Vempala, 2017; Noble et al., 2022), reflected methods have remained comparatively undeveloped from the methodological and practical point of view. In the next section, we recall the basics of reflected stochastic processes.

Skorokhod problem. The reflected Brownian motion is defined as the solution to the Skorokhod problem. Roughly speaking a solution to the Skorokhod problem consists of two coupled processes, $(\bar{\mathbf{B}}_t, \mathbf{k}_t)_{t\geq 0}$, such that $(\bar{\mathbf{B}}_t)_{t\geq 0}$ acts *locally* as a Euclidean Brownian motion $(\mathbf{B}_t)_{t\geq 0}$ and $\mathbf{k}_t$ compensates for the excursion of $(\mathbf{B}_t)_{t\geq 0}$ so that $(\bar{\mathbf{B}}_t)_{t\geq 0}$ remains in $\mathcal{M}$. We say that $(\bar{\mathbf{B}}_t, \mathbf{k}_t)_{t\geq 0}$ is a solution to the *Skorokhod problem* (Skorokhod, 1961) if $(\mathbf{k}_t)_{t\geq 0}$ and $(\bar{\mathbf{B}}_t)_{t\geq 0}$ are two processes satisfying mild conditions, see Appendix D for a rigorous introduction, such that for any $t \geq 0$,

$$\bar{\mathbf{B}}_t = \bar{\mathbf{B}}_0 + \mathbf{B}_t - \mathbf{k}_t \in \mathcal{M}, \tag{8}$$

and $|\mathbf{k}|_t = \int_0^t \mathbf{1}_{\bar{\mathbf{B}}_s \in \partial\mathcal{M}} \mathrm{d}|\mathbf{k}|_s$, $\mathbf{k}_t = \int_0^t \mathbf{n}(\bar{\mathbf{B}}_s)\mathrm{d}|\mathbf{k}|_s$, where $(|\mathbf{k}|_t)_{t\geq 0}$ is the total variation of $(\mathbf{k}_t)_{t\geq 0}$ and we recall that $\mathbf{n}$ is the outward normal to $\mathcal{M}$[5]. When $(\bar{\mathbf{B}}_t)_{t\geq 0}$ hits the boundary, the condition $\mathbf{k}_t = \int_0^t \mathbf{n}(\bar{\mathbf{B}}_s)\mathrm{d}|\mathbf{k}|_s$, tells us that $-\mathbf{k}_t$ "compensates" for $\bar{\mathbf{B}}_t$ by pushing the process back into $\mathcal{M}$ along the inward normal $-\mathbf{n}$, while the condition $|\mathbf{k}|_t = \int_0^t \mathbf{1}_{\bar{\mathbf{B}}_s \in \partial\mathcal{M}} \mathrm{d}|\mathbf{k}|_s$ can be interpreted as $\mathbf{k}_t$ being constant when $(\bar{\mathbf{B}}_t)_{t\geq 0}$ does not hit the boundary. As a result $(\bar{\mathbf{B}}_t)_{t\geq 0}$ can be understood as the continuous-time counterpart to the reflected Gaussian random walk. The process $(\mathbf{k}_t)_{t\geq 0}$ can be related to the notion of *local time* (Revuz & Yor, 2013) and quantifies the amount of time $(\bar{\mathbf{B}}_t)_{t\geq 0}$ spends at the boundary $\partial\mathcal{M}$. Lions & Sznitman (1984, Theorem 2.1) ensure the existence and uniqueness of a solution to the Skorokhod problem. One key observation is that the event $\{\bar{\mathbf{B}}_t \in \partial\mathcal{M}\}$ has probability zero (Harrison & Williams, 1987, Section 7, Lemma 7). As in the *unconstrained* setting, one can describe the dynamics of the density of $\bar{\mathbf{B}}_t$.

Proposition 3.1 (Burdzy et al. (2004))**.** *For any $t > 0$, the distribution of $\bar{\mathbf{B}}_t$ admits a density w.r.t. the Lebesgue measure denoted p_t. In addition, we have for any $x \in \mathrm{int}(\mathcal{M})$ and $x_0 \in \partial\mathcal{M}$*

$$\partial_t p_t(x) = \tfrac{1}{2}\Delta p_t(x), \quad \partial_{\mathbf{n}} p_t(x_0) = 0, \tag{9}$$

where we recall that $\mathbf{n}$ is the outward normal to $\mathcal{M}$.

Note that contrary to the unconstrained setting, the heat equation has *Neumann* boundary conditions. Similarly to the compact Riemannian setting (Saloff-Coste, 1994) it can be shown that the reflected Brownian motion converges to the uniform distribution on $\mathcal{M}$ exponentially fast (Loper, 2020; Burdzy et al., 2006), see Section 3.2. Hence, $(\bar{\mathbf{B}}_t)_{t\geq 0}$ is a candidate for a forward noising process in the context of diffusion models.

Time-reversal. In order to extend the diffusion model approach to the reflected setting, we need to derive a *time-reversal* for $(\bar{\mathbf{B}}_t)_{t\in[0,T]}$. Namely, we need to characterise the evolution of $(\overleftarrow{\mathbf{X}}_t)_{t\in[0,T]} = (\bar{\mathbf{B}}_{T-t})_{t\in[0,T]}$. It can be shown that the time-reversal of $(\bar{\mathbf{B}}_t)_{t\in[0,T]}$ is also the solution to a Skorokhod problem.

[5]We extend the normal $\mathbf{n}$ to the whole space by letting $\mathbf{n}(x) = 0$ if $x \notin \partial\mathcal{M}$.

Algorithm 2 *Reflected step algorithm.* The algorithm operates by repeatedly taking geodesic steps until one of the constraints is violated, or the step is fully taken. Upon hitting the boundary we parallel transport the tangent vector to the boundary and then reflect it against the boundary. We then start a new geodesic from this point in the new direction. The $\arg\text{intersect}_t$ function computes the distance one must travel along a geodesic in direction $\boldsymbol{s}$ til constraint f_i is intersected. For a discussion of paralleltransport, $\exp_{\mathfrak{g}}$ and reflect please see Appendix A.

Input: $x \in \mathcal{M}$, $\boldsymbol{v} \in \mathrm{T}_x\mathcal{M}$, $\{f_i\}_{i\in\mathcal{I}}$
  $\ell \leftarrow \|\boldsymbol{v}\|_{\mathfrak{g}}$
  $\boldsymbol{s} \leftarrow \boldsymbol{v}/\|\boldsymbol{v}\|_{\mathfrak{g}}$
  while $\ell \geq 0$ **do**
    $d_i = \arg\text{intersect}_t\big[\exp_{\mathfrak{g}}(x, t\boldsymbol{s}), f_i\big]$
    $i \leftarrow \arg\min_i \ d_i \ s.t. \ d_i > 0$
    $\alpha \leftarrow \min(d_i, \ell)$
    $x' \leftarrow \exp_{\mathfrak{g}}(x, \alpha\boldsymbol{s})$
    $\boldsymbol{s} \leftarrow \text{paralleltransport}_{\mathfrak{g}}(x, \boldsymbol{s}, x')$
    $\boldsymbol{s} \leftarrow \text{reflect}(\boldsymbol{s}, f_i)$
    $\ell \leftarrow \ell - \alpha$
    $x \leftarrow x'$
  return x

Algorithm 3 *Reflected Random Walk.* Discretisation of the SDE $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t - \mathrm{d}\mathbf{k}_t$.

  Require: T (simulation time), N (number of steps), X_0^γ (initial point), $\{f_i\}_{i\in\mathcal{I}}$ (boundary functions)
  $\gamma = T/N$
  for $k \in \{0, \dots, N-1\}$ **do**
    $Z_{k+1} \sim \mathrm{N}(0, \mathrm{Id})$
    $X_{k+1}^\gamma = \text{ReflectedStep}[X_k^\gamma, \sqrt{\gamma} Z_{k+1}, \{f_i\}_{i\in\mathcal{I}}]$
  return $\{X_k^\gamma\}_{k=0}^N$

Theorem 3.2. *There exist* $(\overleftarrow{\mathbf{k}}_t)_{t\geq 0}$ *a bounded variation process and a Brownian motion* $(\mathbf{B}_t)_{t\geq 0}$ *such that*

$$\overleftarrow{\mathbf{X}}_t = \overleftarrow{\mathbf{X}}_0 + \mathbf{B}_t + \int_0^t \nabla \log p_{T-s}(\overleftarrow{\mathbf{X}}_s)\mathrm{d}s - \overleftarrow{\mathbf{k}}_t.$$

In addition, for any $t \in [0, T]$ *we have*

$$|\overleftarrow{\mathbf{k}}|_t = \int_0^t \mathbf{1}_{\overleftarrow{\mathbf{X}}_s \in \partial\mathcal{M}} \mathrm{d}|\overleftarrow{\mathbf{k}}|_s, \quad \overleftarrow{\mathbf{k}}_t = \int_0^t \mathbf{n}(\overleftarrow{\mathbf{X}}_s)\mathrm{d}|\overleftarrow{\mathbf{k}}|_s.$$

The proof, see Appendix G, follows Petit (1997) which provides a time-reversal in the case where $\mathcal{M}$ is the positive orthant. It is based on an extension of Haussmann & Pardoux (1986) to the reflected setting, with a careful handling of the boundary conditions. In particular, contrary to Petit (1997), we do not rely on an explicit expression of p_t but instead use the intrinsic properties of $(\mathbf{k}_t)_{t\geq 0}$. Informally, Theorem 3.2 means that the process $(\overleftarrow{\mathbf{X}}_t)_{t\in[0,T]}$ satisfies

$$\mathrm{d}\overleftarrow{\mathbf{X}}_t = \nabla \log p_{T-t}(\overleftarrow{\mathbf{X}}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t - \mathrm{d}\overleftarrow{\mathbf{k}}_t, \tag{10}$$

which echoes the usual time-reversal formula (2). In practice, in order to sample from $(\overleftarrow{\mathbf{X}}_t)_{t\in[0,T]}$, one needs to consider the *reflected* version of the *unconstrained* dynamics $\mathrm{d}\overleftarrow{\mathbf{X}}_t = \nabla \log p_{T-t}(\overleftarrow{\mathbf{X}}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t$.

Sampling. In practice, we approximately sample the reflected dynamics by considering the Markov chain given by Algorithm 3. We refer to Pacchiarotti et al. (1998) and Bossy et al. (2004) for weak convergence results on this numerical scheme in the Euclidean setting.

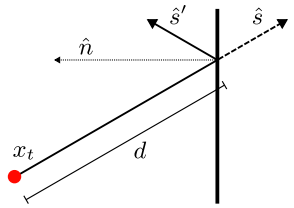


Figure 6: Reflection against a linear boundary. For a step s with magnitude $|s|$ and direction $\hat{s}$ the distance to the boundary described by the normal $\hat{n}$ and offset b is $d = \frac{\langle \hat{s}, x_t \rangle - b}{\langle \hat{s}, \hat{n} \rangle}$. The reflected direction is given by $\hat{s}' = \hat{s} - 2\langle \hat{s}, \hat{n} \rangle \hat{n}$.

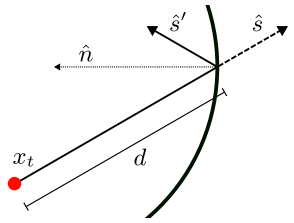


Figure 7: Reflection against a spherical boundary. For a sphere of radius r, the distance to the boundary from x_t in direction s is given by $d = \frac{1}{2}(\langle \hat{s}, x_t \rangle^2 + 4(r^2 - \|x_t\|^2))^{1/2} - \frac{1}{2}\langle \hat{s}, x_t \rangle$. The normal at the intersection can be computed as the unit vector in the direction $-2(d\hat{s} + x_t)$, and then $\hat{s}'$ as above.

Likelihood evaluation. In the case of a reflected Brownian motion, it is possible to compute an equivalent ODE in order to perform likelihood evaluation. The associated ODE was first derived in Lou & Ermon (2023). The form of the ODE and the proof that it remains in $\mathcal{M}$ are postponed to Appendix F.

3.3 Loss and score network parameterisation

In order to train log-barrier and reflected diffusion models we prove that we can use a *tractable* score matching loss in constrained manifolds. We will prove that the implicit score-matching loss leads to the recovery of the correct score when we have a boundary, so long as we enforce that the score is zero on the boundary (see Appendix E.1). This proof holds for both the log-barrier and the reflected process.

Proposition 3.3. *Let $s \in \mathrm{C}^\infty([0,T] \times \mathbb{R}^d, \mathbb{R}^d)$ such that for any $x \in \partial\mathcal{M}$ and $t \geq 0$, $s_t(x) = 0$. Then, there exists $C > 0$ such that*

$$\mathbb{E}[\|\nabla \log p_t - s_t\|^2] = \mathbb{E}[\|s_t\|^2 + 2 \operatorname{div}(s_t)] + C,$$

where $\mathbb{E}$ is taken over $\mathbf{X}_t \sim p_t$ and $t \sim \mathcal{U}([0,T])$.

This result immediately implies we can optimise the score network using the ism loss function so long as we enforce a Neumann boundary condition. The estimation of the score term $\nabla \log p_t$ is also done by minimising the ism loss function (4).

The score term $\nabla \log p_{T-t}$ appearing in both time-reversal processes (7) and (10) is intractable. It is thus approximated with a *score network* $s_\theta(t, \cdot) \approx \nabla \log p_t$. We use a multi-layer perceptron architecture with sin activation functions, see Appendix J for more details on the experimental setup. Due to boundary condition (9), the normal component of the score is zero at the boundary in the reflected case. A similar result holds in the log-barrier setting. This is additionally required for our proof of the ism loss.

Following Liu et al. (2022), we can accommodate this in our score parameterisation by additionally scaling the score output of the neural network by a monotone function $h(d(x, \partial\mathcal{M}))$ where d is the distance from x to the boundary and $h(0) = 0$. In particular, we use a clipped ReLU function: $s_\theta(t,x) = \min(1, \mathrm{ReLU}(d(x, \partial\mathcal{M}) - \delta)) \cdot \mathrm{NN}_\theta(t,x)$ with $\delta > 0$ a threshold so the model is forced to be zero "close" to the boundary as well as exactly on the boundary, see Appendix J for an illustration. The inclusion of this scaling function is necessary to produce reasonable results as we show in Appendix E.2. The weighting function in (4) is set to $\lambda_t = (t+1)$.

The forward processes (6) and (8) for the barrier and reflected methods cannot be sampled in closed form, so at training time samples from the conditional marginals $p(\mathbf{X}_t | \mathbf{X}_0)$ are obtained by discretising these processes. As to take the most of this computational overhead, we use several samples from the discretised forward trajectory $(\mathbf{X}_{t_1}, \cdots, \mathbf{X}_{t_k} | \mathbf{X}_0)$ instead of only using the last sample.

4 Related Work

Sampling on constrained manifolds. Sampling from a distribution on a space defined by a set of constraints is an important ingredient in several computational tasks, such as computing the volume of a polytope (Lee & Vempala, 2017). Incorporating such constraints within MCMC algorithms while preserving fast convergence properties is an active field of research (Kook et al., 2022; Lee & Vempala, 2017; Noble et al., 2022). In this work, we are interested in sampling from the uniform distribution defined on the constrained set in order to define a proper *forward process* for our diffusion model. Log-barrier methods such as the Dikin walk or Riemannian Hamiltonian Monte Carlo (Kannan & Narayanan, 2009; Lee & Vempala, 2017; Noble et al., 2022) change the geometry of the underlying space and define stochastic processes which never violate the constraints, see Kannan & Narayanan (2009) and Noble et al. (2022) for instance. If we keep the Euclidean metric, then *unconstrained* stochastic processes might not be well-defined for all times. To counter this effect, it has been proposed to *reflect* the Brownian motion (Williams, 1987; Petit, 1997; Shkolnikov & Karatzas, 2013). Finally, we also highlight hit-and-run approaches (Smith, 1984; Lovász & Vempala, 2006), which generalise Gibbs' algorithm and enjoy fast convergence properties provided that one knows how to sample from the one-dimensional marginals.

Diffusion models on manifolds. De Bortoli et al. (2022) extended the work of Song et al. (2021) to Riemannian manifolds by defining forward and backward stochastic processes in this setting. Concurrently, a similar framework was introduced by Huang et al. (2022), extending the maximum likelihood approach of Huang et al. (2021). Existing applications of denoising diffusion models on Riemannian manifolds have been focused on well-known manifolds for which one can find metrics so that the framework of De Bortoli et al. (2022) applies. In particular, on compact Lie groups, geodesics and Brownian motions can be defined in a canonical manner. Their specific structure can be leveraged to define efficient diffusion models (Yim et al., 2023). Leach et al. (2022) define diffusion models on $\mathrm{SO}(3)$ for rotational alignment, while Jing et al. (2022) consider the product of tori for molecular conformer generation. Corso et al. (2022) use diffusion models on $\mathbb{R}^3 \times \mathrm{SO}(3) \times \mathrm{SO}(2)$ for protein docking applications. RFDiffusion (Watson et al., 2022) and FrameDiff also incorporate $\mathrm{SE}(3)$ diffusion models. Finally, Urain et al. (2022) introduce a methodology for $\mathrm{SE}(3)$ diffusion models with applications to robotics.

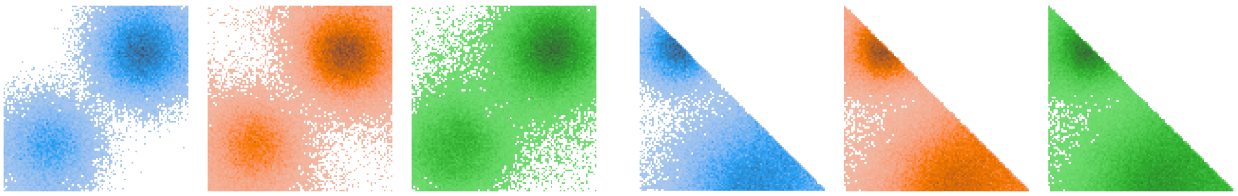
Comparison with Lou & Ermon (2023). We now discuss a few key differences between our work and the reflected diffusion models presented in Lou & Ermon (2023). First, in the hypercube setting, our methodologies are identical from a theoretical viewpoint. The main difference is that Lou & Ermon (2023) use an approxiamted version of the DSM loss (3), whereas we rely on the ISM loss (4). Lou & Ermon (2023) exploit the specific factorised structure of the hypercube to make the DSM loss tractable, leading to significant practical advantages. These can however be directly employed in our framework for reflected models, and also in the log-barrier setting. Second, in our work, we target scientific applications where the underlying geometry is not Euclidean, whereas Lou & Ermon (2023) focus on the case where the constrained domain of interest is the hypercube (a subset of Euclidean space) with image applications, or where the domain can be easily projected into the hypercube, such as the simplex. Our approach is designed to handle a wider range of settings, such as non-convex polytopes in Euclidean space, or non-Euclidean geometries.

5 Experimental results

To demonstrate the practical utility of the constrained diffusion models introduced in Section 3, we evaluate them on a series of increasingly difficult synthetic tasks on convex polytopes, including the hypercube and the simplex, in Section 5.1. We then highlight their applicability to real-world settings by considering two problems from robotics and protein design. In particular, we show that our models are able to learn distributions over the space of $d \times d$ symmetric positive definite (SPD) matrices S^d_{++} under trace constraints in Section 5.2—a setting that is essential to describing and controlling the motions and exerted forces of robotic platforms (Jaquier et al., 2021). In Section 5.3, we use the parametrisation introduced in Han & Rudolph (2006) to map the problem of modelling the conformational ensembles of proteins under positional constraints on their endpoints to the product manifold of a convex polytope and a torus. The code is available here. We refer the reader to Appendix J for more details on the experimental setup.

Table 1: MMD metrics between samples from synthetic distributions and trained constrained and unconstrained (Euclidean) diffusion models. Means and confidence intervals are computed over 5 different runs.

Space	d	Log-barrier		Reflected		Euclidean	
		MMD	% in $\mathcal{M}$	MMD	% in $\mathcal{M}$	MMD	% in $\mathcal{M}$
$[-1,1]^d$	2	.066±.006	100.0	.055±.015	100.0	.062±.011	98.8
	3	.209±.077	100.0	.080±.004	100.0	.076±.004	98.5
	10	.330±.004	100.0	.313±.048	100.0	.081±.005	96.4
Δ^d	2	.050±.012	100.0	.043±.002	100.0	.055±.013	96.4
	3	.238±.009	100.0	.181±.007	100.0	.068±.014	96.3
	10	.275±.015	100.0	.290±.009	100.0	.060±.003	92.6



(a) 2D square data. (b) 2D Dirichlet data.

Figure 8: Histograms of samples from the data distribution and from trained constrained diffusion models.

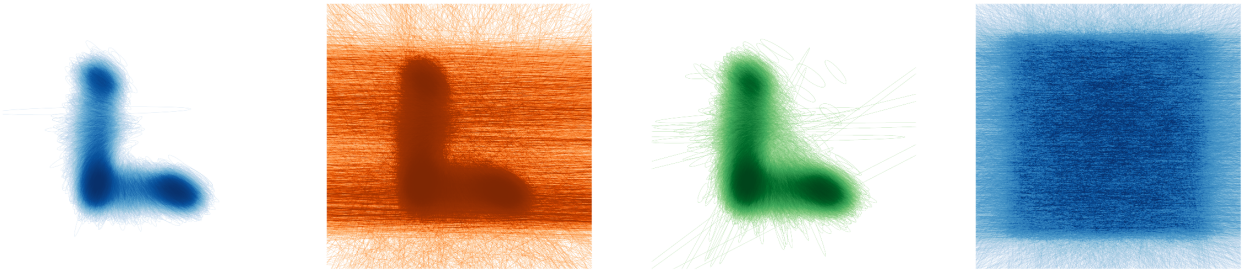
5.1 Method characterisation on convex polytopes

First, we aim to assess the empirical performance of our methods on constrained manifolds of increasing dimensionality. For this, we focus on polytopes and construct synthetic datasets that represent bimodal distributions. In particular, we investigate two specific instances of polytopes: hypercubes and simplices. In Appendix J.1 we also present results on the Birkhoff polytope. We quantify the performance of each model via the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012), which is a kernel-based metric between distributions. We present a qualitative comparison of the logarithmic barrier and reflected Brownian motion models in Figure 8, and observe that both methods recover the data distribution on the two-dimensional hypercube and simplex, although the reflected method seems to produce a better fit. In Table 1, we report the MMD between the data distribution and the learnt diffusion models. Here, we similarly observe that the reflected method consistently yields better results than the log-barrier one.

We additionally compare both of these models to a set of unconstrained Euclidean diffusion models, noting that they are outperformed by the constrained models in lower dimensions, but generate better results in higher dimensions. There are a number of potential explanations for this: First, we note that the mixture of Normal distributions we use as a synthetic data-generating process places significantly less probability mass near the boundary as its dimensionality increases, more closely resembling an unconstrained mixture distribution that is easier for the Euclidean diffusion models to learn, while posing a challenge to the log-barrier and reflected diffusion models that initialise at the uniform distribution within the constraints. Additionally, we note that the design space and hyperparameters used for all experiments were informed by best practices for Euclidean models that may be suboptimal for the more complex dynamics of constrained diffusion models.

5.2 Modelling robotic arms under force and velocity constraints

Accurately determining and controlling the movement and exerted forces of robotic platforms is a fundamental problem in many real-world robotics applications. A kinetostatic descriptor that is commonly used to quantify the ability of a robotic arm to move and apply forces along certain dimensions is the so-called manipulability ellipsoid $E \in \mathbb{R}^d$ (Yoshikawa, 1985), which is naturally described as a symmetric positive definite (SPD)



(a) Samples from the data distribution.

(b) Samples from our log-barrier diffusion model.

(c) Samples from our reflected diffusion model.

(d) Samples from the uniform distribution.

Figure 9: Samples in $S^2_{++} \times \mathbb{R}^2$ from (a) the data distribution, (b) our log-barrier diffusion model, (c) our reflected diffusion model and (d) the uniform distribution. Each sample is visualised as the manipulability ellipsoid encoded by the SPD matrix $M \in S^2_{++}$ placed at the corresponding location in $\mathbb{R}^2$. Additional results and full correlation plots are postponed to Appendix J.2.

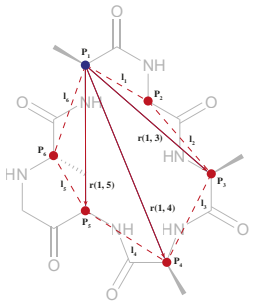
matrix $\mathrm{M} \in \mathbb{R}^{d \times d}$ (Jaquier et al., 2021). The manifold of such $d \times d$ SPD matrices, denoted as S^d_{++}, is defined as the set of matrices $\{x^\top \mathrm{M} x \geq 0,\ x \in \mathbb{R}^d : \mathrm{M} \in \mathbb{R}^{d \times d}\}$. In many practical settings, it may be desirable to constrain the volume of E to retain flexibility or limit the magnitude of an exerted force, which can be expressed as an upper bound on the trace of M, i.e. as the inequality constraint $\sum_{i=1}^d \mathrm{M}_{ii} < C$ with $C > 0$. Constraining the rest of the entries of the matrix to ensure it is SPD is non-trivial. Alternatively, we can parameterise the SPD matrices via their Cholesky decomposition. Each SPD matrix has a unique decomposition of the form $\mathrm{M} = \mathrm{L}\mathrm{L}^\top$, where L is a lower triangular matrix with strictly positive diagonal (Golub & Van Loan, 2013, p.143). Constraining the entries of this matrix simply requires ensuring the diagonal is positive. The trace of the SPD matrix is given by $\mathrm{Tr}(\mathrm{M}) = \mathrm{Tr}(\mathrm{L}\mathrm{L}^\top) = \sum_{ij} \mathrm{L}^2_{ij}$, and results in the constraint on the entries of L to live in a ball of radius C. We additionally model the two-dimensional position of the arm. In summary, the space over which we parameterise the diffusion models is defined as $\{\mathrm{L} \in \mathbb{R}^{d(d+1)/2} : \mathrm{L}_{i,i} > 0, \sum_{i,j} \mathrm{L}^2_{i,j} < C\} \times \mathbb{R}^2$. Under the Euclidean metric, we can apply both our log-barrier and reflected approaches. The positive diagonal (linear) constraint is handled similarly to the polytope setting. The reflection on the ball boundary is defined and illustrated on Figures 6 and 7.

Using this framework, we model the datasets presented in Jaquier et al. (2021) (see Appendix H for full experimental details). The joint distribution over the SPD matrices (represented as ellipsoids) and their positions is presented in Figure 9. We qualitatively observe that the reflected method is able to model the joint data distribution better than the log-barrier one. This is reflected by an MMD of 0.161 and 0.247, respectively.

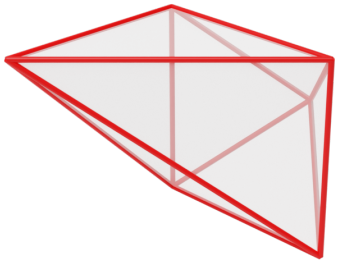
5.3 Modelling protein loops with anchored endpoints

Modelling the conformational ensembles of proteins is an important task in the field of molecular biology, particularly in the context of bioengineering and drug discovery. In many data-scarce practical settings such as antibody or enzyme design, it is often unnecessary or even undesirable to model the structure of an entire protein, as researchers are primarily interested in specific functional sites with distinct biochemical properties. However, generating conformational ensembles for such substructural elements necessitates positional constraints on their endpoints to ensure that they can be accommodated by the remaining scaffold.

This problem can be reformulated as modelling a spatial chain with spherical joints and fixed end points. Following the framework outlined in Han & Rudolph (2006), we parametrise the conformations of such chains with d fixed-length links and arbitrary end-point constraints as the product of a convex polytope $\mathbb{P}$ and torus $\mathbb{T}$: $\mathbb{P}^{(d-3)} \times \mathbb{T}^{(d-2)}$. The essential idea of this parameterisation is to fix one end point as an "anchor" and model the chain as the series of triangles formed by the anchor and each pair of adjacent joints in the chain. A point in the polytope corresponds to the lengths of the diagonals of these triangles, and a point in the torus to the angles between each pair of subsequent triangles. See Figure 10 for an illustration and Appendix I.1 for a full description.

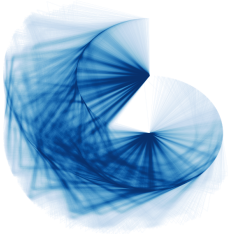


(a) An illustrative diagram of the parameterisation used for the conformational modelling of the C_α trace of a cyclic peptide, introduced in Han & Rudolph (2006).

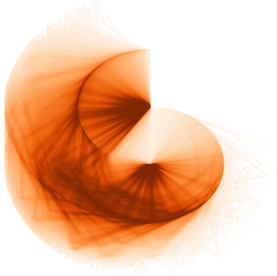


(b) The convex polytope constraining the diagonals of the triangles for the given bond lengths in the illustrated molecule. The total design space is the product of this polytope with the 4D flat torus.

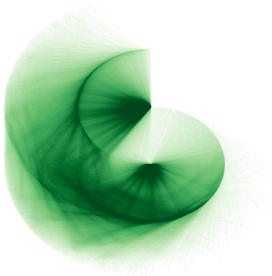
Figure 10: Illustrations of the parametrisation used to model distribution of polypeptide backbone conformations under anchor point distance constraints as the product of a convex polytope $\mathbb{P}$ and torus $\mathbb{T}$: $\mathbb{P}^3 \times \mathbb{T}^4$.



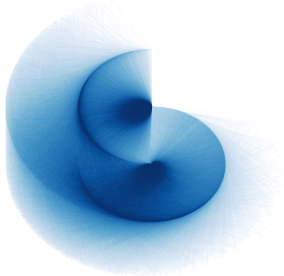
(a) Samples from the data distribution.



(b) Samples from our log-barrier diffusion model.



(c) Samples from our reflected diffusion model.



(d) Samples from the uniform distribution.

Figure 11: Planar projection of the modelled C_α chains from (a) the training dataset, (b) our log-barrier diffusion model, (c) our reflected diffusion model and (d) the uniform distribution. Additional results and full correlation plots are postponed to Appendix J.3.

Using this framework, we model the conformational landscape of the cyclic peptide c-AAGAGG, consisting of a circular polypeptide chain with coinciding endpoints. We generate 10^6 backbone conformations using tools from molecular dynamics (Eastman et al., 2017; Hornak et al., 2006) and divide them into training and evaluation datasets, (see Appendix I.2 for full experimental details). Drawing on the definition above, the space on which we effectively parameterise our constrained diffusion models for a circular polypeptide chain of length $d = 6$ is given by the product manifold $\mathbb{P}^3 \times \mathbb{T}^4$.

To learn a distribution over this space, we leverage the methodology introduced in Section 3 for the polytope component $\mathbb{P}^3$ and in De Bortoli et al. (2022) for the torus component $\mathbb{T}^4$. A qualitative comparison of samples from the data distribution, our log-barrier and reflected diffusion models, and the uniform distribution is presented in Figures 11a to 11d. For enhanced visual clarity, we project the modelled spatial chain onto the 2D plane by removing the (unconstrained) torus component of the product manifold and only plotting the planar chains encoded by the (constrained) polytope component (a correlation plot of the full product manifold is presented in Figure 25).

It is apparent that the data distribution is highly multimodal, encompassing a large number of locally optimal conformational clusters. Nevertheless, our reflected and log-barrier diffusion models are able to robustly approximate this energetic landscape, producing samples that reflect key conformational states and producing comparable MMD metrics of $0.032_{\pm 0.021}$ and $0.032_{\pm 0.001}$, respectively. As a point of comparison, the uniform distribution on the polytope-torus product has an MMD of $0.112_{\pm 0.001}$.

6 Discussion

Learning complex distributions whose support is confined to constrained spaces is a crucial task with many applications in the natural and engineering sciences, including computational statistics (Morris, 2002), robotics (Han & Rudolph, 2006), quantum physics (Lukens et al., 2020) and computational biology (Thiele et al., 2013). In this work we extend continuous diffusion models to this setting, proposing two complementary approaches—one based on log-barrier methods and the other on the reflected Brownian motion. For both methods, we derive the time-reversal formula, propose discretisation schemes and extend the score-matching toolbox. We demonstrate the utility of our methods on a range of synthetic and real-world tasks, including the constrained conformational modelling of proteins and robotic arms, and find that reflected methods, while enjoying fewer theoretical guarantees than their log-barrier counterparts, often yield preferable results.

We conclude by highlighting important directions of future research. First, the computational cost of performing the reflection when discretising the reflected Brownian motion is high. Finding numerically efficient approximations of the reflected process is therefore necessary to extend this methodology to very high dimensional settings. Second, the retraction used in place of the exponential map for the barrier method leads to a high number of discretisation steps to ensure a good approximation. Designing a faster forward process for the log-barrier method is key to targeting more complex distributions.

References

P-A Absil and Jérôme Malick. Projection-like retractions on matrix manifolds. *SIAM Journal on Optimization*, 22(1):135–158, 2012.

RN Bhattacharya. Criteria for recurrence and existence of invariant measures for multidimensional diffusions. *The Annals of Probability*, pp. 541–553, 1978.

Mireille Bossy, Emmanuel Gobet, and Denis Talay. A symmetrized euler scheme for an efficient approximation of reflected diffusions. *Journal of applied probability*, 41(3):877–889, 2004.

Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Krzysztof Burdzy, Zhen-Qing Chen, and John Sylvester. The heat equation and reflected brownian motion in time-dependent domains. *The Annals of Probability*, 32(1B):775–804, 2004.

Krzysztof Burdzy, Zhen-Qing Chen, and Donald E Marshall. Traps for reflected brownian motion. *Mathematische Zeitschrift*, 252:103–132, 2006.

Patrick Cattiaux, Giovanni Conforti, Ivan Gentil, and Christian Léonard. Time reversal of diffusion processes under a finite entropy condition. *arXiv preprint arXiv:2104.07708*, 2021.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking, October 2022. URL http://arxiv.org/abs/2210.01776.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modeling, 2022.

Patrick G Dougherty, Ashweta Sahni, and Dehua Pei. Understanding cell penetration of cyclic peptides. *Chemical Reviews*, 119(17):10241–10287, 2019.

Peter Eastman, Jason Swails, John D Chodera, Robert T McGibbon, Yutong Zhao, Kyle A Beauchamp, Lee-Ping Wang, Andrew C Simmonett, Matthew P Harrigan, Chaya D Stern, et al. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology*, 13(7): e1005659, 2017.

Lawrence Craig Evans and Ronald F Gariepy. *Measure theory and fine properties of functions*. CRC press, 2015.

Khashayar Gatmiry and Santosh S Vempala. Convergence of the riemannian langevin algorithm. *arXiv preprint arXiv:2204.10818*, 2022.

Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(null):723–773, March 2012. ISSN 1532-4435.

Li Han and Lee Rudolph. Inverse kinematics for a serial chain with joints under distance constraints. In *Robotics: Science and systems*, 2006.

J Michael Harrison and Ruth J Williams. Brownian models of open queueing networks with homogeneous customer populations. *Stochastics: An International Journal of Probability and Stochastic Processes*, 22(2): 77–115, 1987.

Ulrich G Haussmann and Etienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, pp. 1188–1205, 1986.

Laurent Heirendt, Sylvain Arreckx, Thomas Pfau, Sebastián N Mendoza, Anne Richelle, Almut Heinken, Hulda S Haraldsdóttir, Jacek Wachowiak, Sarah M Keating, Vanja Vlasov, et al. Creation and analysis of biochemical constraint-based models using the cobra toolbox v. 3.0. *Nature protocols*, 14(3):639–702, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020.

Viktor Hornak, Robert Abel, Asim Okur, Bentley Strockbine, Adrian Roitberg, and Carlos Simmerling. Comparison of multiple amber force fields and development of improved protein backbone parameters. *Proteins: Structure, Function, and Bioinformatics*, 65(3):712–725, 2006.

Elton P Hsu. *Stochastic analysis on manifolds*. Number 38. American Mathematical Soc., 2002.

Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based generative models and score matching. *Advances in Neural Information Processing Systems*, 34:22863–22876, 2021.

Chin-Wei Huang, Milad Aghajohari, Avishek Joey Bose, Prakash Panangaden, and Aaron Courville. Riemannian Diffusion Models, August 2022.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Nobuyuki Ikeda and Shinzo Watanabe. *Stochastic differential equations and diffusion processes*. Elsevier, 2014.

Noémie Jaquier, Leonel Rozo, Darwin G Caldwell, and Sylvain Calinon. Geometry-aware manipulability learning, tracking, and transfer. *The International Journal of Robotics Research*, 40(2-3):624–650, 2021.

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional Diffusion for Molecular Conformer Generation, June 2022. URL http://arxiv.org/abs/2206.01729.

Erik Jørgensen. The central limit problem for geodesic random walks. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 32(1-2):1–64, 1975.

Weining Kang and Kavita Ramanan. On the submartingale problem for reflected diffusions in domains with piecewise smooth boundaries. 2017.

Ravi Kannan and Hariharan Narayanan. Random walks on polytopes and an affine interior point method for linear programming. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pp. 561–570, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585062. doi: 10.1145/1536414.1536491. URL https://doi.org/10.1145/1536414.1536491.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL http://arxiv.org/abs/1412.6980.

Yunbum Kook, Yin Tat Lee, Ruoqi Shen, and Santosh S Vempala. Sampling with riemannian hamiltonian monte carlo in a constrained space. *arXiv preprint arXiv:2202.01908*, 2022.

Thomas J Lane. Protein structure prediction has reached the single-structure frontier. *Nature Methods*, pp. 1–4, 2023.

Adam Leach, Sebastian M Schmon, Matteo T Degiacomi, and Chris G Willcocks. Denoising Diffusion Probabilistic Models on SO(3) for Rotational Alignment. *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, pp. 8, 2022.

John M Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

John M Lee. Smooth manifolds. In *Introduction to Smooth Manifolds*, pp. 1–31. Springer, 2013.

Yin Tat Lee and Santosh S. Vempala. Geodesic walks in polytopes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 927–940, Montreal Canada, June 2017. ACM. ISBN 978-1-4503-4528-6. doi: 10.1145/3055399.3055416.

Yin Tat Lee and Santosh S Vempala. Convergence rate of riemannian hamiltonian monte carlo and faster polytope volume computation. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1115–1121, 2018.

Pierre-Louis Lions and Alain-Sol Sznitman. Stochastic differential equations with reflecting boundary conditions. *Communications on pure and applied Mathematics*, 37(4):511–537, 1984.

Song Liu, Takafumi Kanamori, and Daniel J Williams. Estimating density models with truncation boundaries using score matching. *Journal of Machine Learning Research*, 23(186):1–38, 2022.

Jackson Loper. Uniform ergodicity for brownian motion in a bounded convex set. *Journal of Theoretical Probability*, 33(1):22–35, 2020.

Aaron Lou and Stefano Ermon. Reflected diffusion models. *arXiv preprint arXiv:2304.04740*, 2023.

László Lovász and Santosh Vempala. Hit-and-run from a corner. *SIAM Journal on Computing*, 35(4): 985–1005, 2006. doi: 10.1137/S009753970544727X.

Joseph M Lukens, Kody JH Law, Ajay Jasra, and Pavel Lougovski. A practical and efficient approach for bayesian quantum state estimation. *New Journal of Physics*, 22(6):063038, 2020.

Sean P Meyn and Richard L Tweedie. Stability of markovian processes ii: Continuous-time processes and sampled chains. *Advances in Applied Probability*, 25(3):487–517, 1993.

Nina Miolane, Nicolas Guigui, Alice Le Brigant, Johan Mathe, Benjamin Hou, Yann Thanwerdas, Stefan Heyder, Olivier Peltre, Niklas Koep, Hadi Zaatiti, Hatem Hajri, Yann Cabanes, Thomas Gerald, Paul Chauchat, Christian Shewmake, Daniel Brooks, Bernhard Kainz, Claire Donnat, Susan Holmes, and Xavier Pennec. Geomstats: A python package for riemannian geometry in machine learning. *Journal of Machine Learning Research*, 21(223):1–9, 2020. URL http://jmlr.org/papers/v21/19-027.html.

Ben J. Morris. Improved bounds for sampling contingency tables. *Random Structures & Algorithms*, 21(2): 135–146, 2002. doi: https://doi.org/10.1002/rsa.10049. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rsa.10049.

Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

Maxence Noble, Valentin De Bortoli, and Alain Durmus. Barrier Hamiltonian Monte Carlo, October 2022.

David Nualart. *The Malliavin calculus and related topics*, volume 1995. Springer, 2006.

Barbara Pacchiarotti, Cristina Costantini, and Flavio Sartoretto. Numerical approximation for functionals of reflecting diffusion processes. *SIAM Journal on Applied Mathematics*, 58(1):73–102, 1998.

Frédérique Petit. Time reversal and reflected diffusions. *Stochastic Processes and their Applications*, 69(1): 25–53, July 1997. ISSN 03044149. doi: 10.1016/S0304-4149(97)00035-5.

Daniel Revuz and Marc Yor. *Continuous martingales and Brownian motion*, volume 293. Springer Science & Business Media, 2013.

Piotr Rotkiewicz and Jeffrey Skolnick. Fast procedure for reconstruction of full-atom protein models from reduced representations. *Journal of computational chemistry*, 29(9):1460–1465, 2008.

Laurent Saloff-Coste. Precise estimates on the rate at which certain diffusions tend to equilibrium. *Mathematische Zeitschrift*, 217(1):641–677, 1994.

Hirohiko Shima and Katsumi Yagi. Geometry of hessian manifolds. *Differential geometry and its applications*, 7(3):277–290, 1997.

Mykhaylo Shkolnikov and Ioannis Karatzas. Time-reversal of reflected Brownian motions in the orthant, July 2013. URL http://arxiv.org/abs/1307.4422.

Anatoliy V Skorokhod. Stochastic equations for diffusion processes in a bounded region. *Theory of Probability & Its Applications*, 6(3):264–274, 1961.

Robert L. Smith. Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed over Bounded Regions. *Operations Research*, 32(6):1296–1308, December 1984. doi: 10.1287/opre.32.6.1296. URL https://ideas.repec.org/a/inm/oropre/v32y1984i6p1296-1308.html.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Ines Thiele, Neil Swainston, Ronan M T Fleming, Andreas Hoppe, Swagatika Sahoo, Maike K Aurich, Hulda Haraldsdottir, Monica L Mo, Ottar Rolfsson, Miranda D Stobbe, Stefan G Thorleifsson, Rasmus Agren, Christian Bölling, Sergio Bordel, Arvind K Chavali, Paul Dobson, Warwick B Dunn, Lukas Endler, David Hala, Michael Hucka, Duncan Hull, Daniel Jameson, Neema Jamshidi, Jon J Jonsson, Nick Juty, Sarah Keating, Intawat Nookaew, Nicolas Le Novère, Naglis Malys, Alexander Mazein, Jason A Papin, Nathan D Price, Evgeni Selkov, Martin I Sigurdsson, Evangelos Simeonidis, Nikolaus Sonnenschein, Kieran Smallbone, Anatoly Sorokin, Johannes H G M van Beek, Dieter Weichart, Igor Goryanin, Jens Nielsen, Hans V Westerhoff, Douglas B Kell, Pedro Mendes, and Bernhard Ø Palsson. A community-driven global reconstruction of human metabolism. *Nature Biotechnology*, March 2013. ISSN 1087-0156. doi: 10.1038/nbt.2488.

Brian L. Trippe, Jason Yim, Doug Tischer, Tamara Broderick, David Baker, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem, June 2022.

Julen Urain, Niklas Funk, Georgia Chalvatzaki, and Jan Peters. Se (3)-diffusionfields: Learning cost functions for joint grasp and motion optimization through diffusion. *arXiv preprint arXiv:2209.03855*, 2022.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23 (7):1661–1674, 2011.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models, December 2022.

R. J. Williams. Reflected Brownian motion with skew symmetric data in a polyhedral domain. *Probability Theory and Related Fields*, 75(4):459–485, August 1987. ISSN 0178-8051, 1432-2064. doi: 10.1007/BF00320328.

Kevin E Wu, Kevin K Yang, Rianne van den Berg, James Y Zou, Alex X Lu, and Ava P Amini. Protein structure generation via folding diffusion. *arXiv preprint arXiv:2209.15611*, 2022.

Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023.

Tsuneo Yoshikawa. Manipulability of robotic mechanisms. *The international journal of Robotics Research*, 4 (2):3–9, 1985.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Diffusion Models for Constrained Domains
Publication Status	Published
Publication Details	Fishman, N., Klarner, L., De Bortoli, V., Mathieu, E. and Hutchinson, M.J., Diffusion Models for Constrained Domains. *Transactions on Machine Learning Research, 2023.*

Student Confirmation

Student Name	Leo Klarner		
Contribution	• Conceptualized key research ideas in collaboration with all other co-authors. • Identified relevant datasets and baselines that allowed for a meaningful empirical evaluation of the proposed method. Collaborated with NF in developing practical implementations and code. • Collaborated with NF, VDB and MH in the development of specialized, problem-specific adaptations of the proposed approaches. • Collaborated with all other co-authors in developing the framing of the method and drafting of the manuscript. Engaged with reviewers during the peer review process in collaboration with all other co-authors.		
Signature		Date	Lausanne, 14. April 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Yee-Whye Teh		
Supervisor comments: I confirm		
Signature 	Date	23 April 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 5

Metropolis Sampling for Constrained Diffusion

This chapter is based on the following publication:

N. Fishman, **L. Klarner**, E. Mathieu, M. Hutchinson, V. De Bortoli. Metropolis Sampling for Constrained Diffusion Models. *Advances in Neural Information Processing Systems*, 2023.

Personal contributions:

1. Conceptualised key research ideas in collaboration with all other co-authors
2. Identified relevant datasets and baselines that allowed for a meaningful empirical evaluation of the proposed method. Collaborated with NF in developing practical implementations and code.
3. Collaborated with NF, VDB and MH in the development of specialised, problem-specific adaptations of the proposed approaches.
4. Collaborated with all other co-authors in developing the framing of the method and drafting of the manuscript.
5. Engaged with reviewers during the peer review process in collaboration with all other co-authors.

A full statement of contributions can be found at the end of this chapter.

The supplementary information for this chapter is provided in Appendix C.

Metropolis Sampling for Constrained Diffusion Models

Nic Fishman
University of Oxford

Leo Klarner
University of Oxford

Emile Mathieu
University of Cambridge

Michael Hutchinson
University of Oxford

Valentin De Bortoli
ENS Ulm

{njwfish,leojklarner}@gmail.com

Abstract

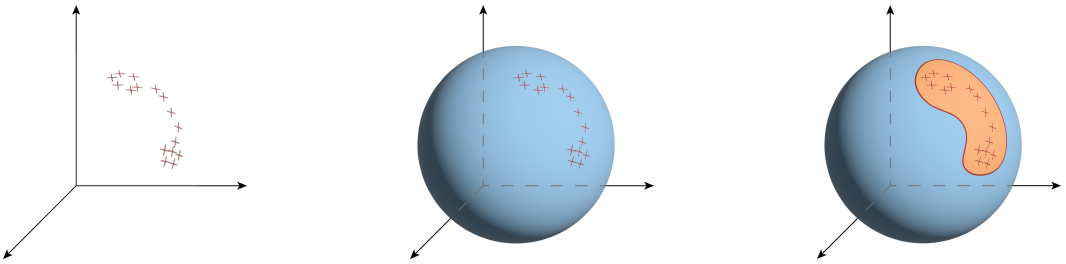
Denoising diffusion models have recently emerged as the predominant paradigm for generative modelling on image domains. In addition, their extension to Riemannian manifolds has facilitated a range of applications across the natural sciences. While many of these problems stand to benefit from the ability to specify arbitrary, domain-informed constraints, this setting is not covered by the existing (Riemannian) diffusion model methodology. Recent work has attempted to address this issue by constructing novel noising processes based on the reflected Brownian motion and logarithmic barrier methods. However, the associated samplers are either computationally burdensome or only apply to convex subsets of Euclidean space. In this paper, we introduce an alternative, simple noising scheme based on Metropolis sampling that affords substantial gains in computational efficiency and empirical performance compared to the earlier samplers. Of independent interest, we prove that this new process corresponds to a valid discretisation of the reflected Brownian motion. We demonstrate the scalability and flexibility of our approach on a range of problem settings with convex and non-convex constraints, including applications from geospatial modelling, robotics and protein design.

1 Introduction

In recent years, denoising diffusion models (Sohl-Dickstein et al., 2015; Song et al., 2019; Song et al., 2021; Ho et al., 2020) have emerged as a powerful paradigm for generative modelling, achieving state-of-the-art performance across a range of domains. They work by progressively adding noise to data following a Stochastic Differential Equation (SDE)—the forward *noising* process—until it is close to the invariant distribution of the SDE. The generative model is then given by an approximation of the associated time-reversed *denoising* process, which is also an SDE whose drift depends on the gradient of the logarithmic densities of the forward process, referred to as the *Stein score*. Building on the success of diffusion models for image generation tasks, De Bortoli et al. (2022) and Huang et al. (2022) have recently extended this framework to a wide range of Riemannian manifolds, enabling the specification of inherent structural properties of the modelled domain. This has broadened the applicability of diffusion models to problems in the natural and engineering sciences, including the conformational modelling of small molecules (Jing et al., 2022; Corso et al., 2022), proteins (Trippe et al., 2022; Watson et al., 2022; Yim et al., 2023) and robotic platforms (Urain et al., 2022).

However, in many data-scarce or safety-critical settings, researchers may want to restrict the modelled domain even further by specifying problem-informed constraints to make maximal use of limited experimental data or prevent unwanted behaviour (Morris, 2002; Han et al., 2006; Thiele et al., 2013; Lukens et al., 2020). As illustrated in Figure 1, such domain-informed constraints can be naturally represented as a *Riemannian manifold with boundary*. Training diffusion models on such constrained manifolds is thus an important problem that requires principled noising processes—and corresponding discretisations—that stay within the constrained set.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).



(a) $\mathbb{R}^3$ (b) $\mathcal{S}^2 \subset \mathbb{R}^3$ (c) $\mathcal{M} \subset \mathcal{S}^2 \subset \mathbb{R}^3$

Figure 1: When operating in data-scarce settings, it may often be beneficial to specify as much prior knowledge of the modelled domain as possible. Consider a distribution over a subset $\mathcal{M}$ of the unit sphere $\mathcal{S}^2 \subset \mathbb{R}^3$. While a Euclidean diffusion model can approximate the distribution in $\mathbb{R}^3$ (a), directly modelling it on $\mathcal{S}^2$ can make learning significantly easier (b). Restricting the problem space even further by only constructing a diffusion model on $\mathcal{M}$ can lead to even better performance (c).

Recent work by Fishman et al. (2023) has attempted to derive such processes and extend the applicability of diffusion models to inequality-constrained manifolds by investigating the generative modelling applications of classic sampling schemes based on log-barrier methods (Kannan et al., 2009; Lee et al., 2017; Noble et al., 2022; Kook et al., 2022; Gatmiry et al., 2022; Lee et al., 2018) and the reflected Brownian motion (Williams, 1987; Petit, 1997; Shkolnikov et al., 2013). While empirically promising, the proposed algorithms can be computationally and numerically burdensome, and require bespoke implementations for different manifolds and constraints. Concurrently, Lou et al. (2023) have investigated the use of reflected diffusion models for image applications. They focus on the high-dimensional hypercube, as this setting admits a theoretically grounded treatment of the *static thresholding* method which is widely used in image models such as Saharia et al. (2022). More recently, Liu et al. (2023) have investigated the use of log-barrier-based mirror maps to transform a constrained domain into an unconstrained dual space for applications in image watermarking. While both methods exhibit robust scaling properties and impressive results, they only consider convex subsets of Euclidean space and do not extend to more general manifolds.

Here, we propose a new method for generative modelling on constrained manifolds based on a Metropolis-based discretisation of the reflected Brownian motion. The Metropolised process' chief advantage is that it is lightweight: the only additional requirement over those outlined in De Bortoli et al. (2022) that is needed to implement a constrained diffusion model is an efficient binary function that indicates whether any given point is within the constrained set. This Metropolised approximation of the reflected Brownian motion is substantially easier to implement, faster to compute and more numerically stable than the previously considered sampling schemes. Our core theoretical contribution is to show that this new discretisation converges to the reflected SDE by using the invariance principle for SDEs with boundary (Stroock et al., 1971). To the best of our knowledge, this is the first time that such a process has been investigated. We demonstrate that our method attains improved empirical results on diverse manifolds with convex and non-convex constraints by applying it to a range of problems from geospatial modelling, robotics and protein design.

2 Background

Riemannian manifolds. A Riemannian manifold is defined as a tuple $(\mathcal{M}, \mathfrak{g})$ with $\mathcal{M}$ a smooth manifold and $\mathfrak{g}$ a metric defining an inner product on tangent spaces. In this work, we will use the exponential map $\exp_x : \mathrm{T}_x\mathcal{M} \to \mathcal{M}$, as well as the extension of the gradient ∇, divergence div and Laplace Δ operators to $\mathcal{M}$. All of these quantities can be defined in local coordinates in terms of the metric. The extension of the Laplace operator to $\mathcal{M}$ is called the Laplace-Beltrami operator, also denoted Δ when there is no ambiguity. Using Δ, we can define a Brownian motion on $\mathcal{M}$, denoted $(\mathbf{B}_t)_{t\geq 0}$ and with density w.r.t. the volume form of $\mathcal{M}$ denoted p_t for any $t > 0$. We refer to Appendix B for a more detailed exposition, to Lee (2013) for a thorough treatment of Riemannian manifolds and to Hsu (2002) for details on stochastic analysis on manifolds. In the following, we consider a constrained manifold $\mathcal{M}$ defined by

$$\mathcal{M} = \{x \in \mathcal{N} : f_i(x) < 0, \ i \in \mathcal{I}\}, \tag{1}$$

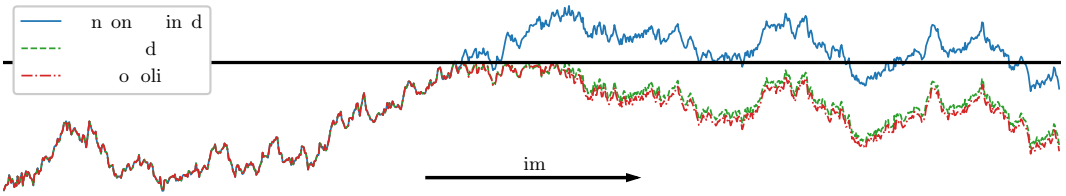


Figure 2: Visual comparison of a discretisation of the unconstrained Brownian motion (blue) and two discretisations of the reflected Brownian motion: one based on a reflection scheme (green) and the other based on our Metropolis sampler (red). The Metropolised trajectory is very close to that of the reflected one while being significantly easier to implement and cheaper to compute.

where $(\mathcal{N}, \mathfrak{g})$ is a Riemannian manifold, $\mathcal{I}$ is an arbitrary finite indexing family and for any $i \in \mathcal{I}$, $f_i \in \mathrm{C}(\mathcal{N}, \mathbb{R})$. Since $\mathcal{I}$ is finite and f_i continuous for any $i \in \mathcal{I}$, $\mathcal{M}$ is an open set of $\mathcal{N}$ and inherits its metric $\mathfrak{g}$. This captures simple Euclidean polytopes and complex constrained spaces like Figure 1.

Denoising diffusion models. Denoising diffusion models (Song et al., 2019; Ho et al., 2020; Song et al., 2021) work as follows: let $(\mathbf{X}_t)_{t\in[0,T]}$ be a *noising process* that corrupts the original data distribution p_0. We assume that $(\mathbf{X}_t)_{t\geq 0}$ converges to $\mathrm{N}(0, \sigma^2 \mathrm{Id})$, with $\sigma > 0$. Several such processes exist, but in practice we consider the Ornstein-Uhlenbeck (OU) process, also referred to as VP-SDE, which is defined by the following Stochastic Differential Equation (SDE)

$$\mathrm{d}\mathbf{X}_t = -\tfrac{1}{2}\mathbf{X}_t \mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t, \qquad \mathbf{X}_0 \sim p_0. \tag{2}$$

Under conditions on p_0, for any $T > 0$, $(\mathbf{Y}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ is also the (weak) solution to a SDE (Anderson, 1982; Haussmann et al., 1986; Cattiaux et al., 2021)

$$\mathrm{d}\mathbf{Y}_t = \{\tfrac{1}{2}\mathbf{Y}_t + \sigma^2 \nabla \log p_{T-t}(\mathbf{Y}_t)\}\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t, \ \mathbf{Y}_0 \sim p_T, \tag{3}$$

where p_t denotes the density of $\mathbf{X}_t$. In practice, $\nabla \log p_t$ is approximated with a score network $(t, x) \mapsto s_\theta(t, x)$ trained by minimising either a denoising score matching (dsm) loss or an implicit score matching (ism) loss (Vincent, 2011)

$$\ell(\theta) = \mathbb{E}_{t\sim\mathcal{U}([0,T]),(\mathbf{X}_0,\mathbf{X}_t)}[\lambda_t(\tfrac{1}{2}\|s_\theta(t, \mathbf{X}_t)\|^2 + \mathrm{div}(s_\theta)(t, \mathbf{X}_t))], \tag{4}$$

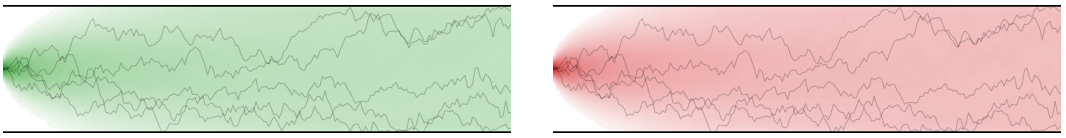
where $\lambda_t > 0$. For a flexible score network, the global minimiser $\theta^\star = \mathrm{argmin}_\theta \mathcal{L}(\theta)$ satisfies $s_{\theta^\star}(t, \cdot) = \nabla \log p_t$. De Bortoli et al. (2022) and Huang et al. (2022) have extended denoising diffusion models to the Riemannian setting. The time-reversal formula (3) remains the same, replacing the Euclidean gradient with its Riemannian equivalent. The ism loss can still be computed in that setting. However, the samplers used in the Riemannian setting differ from the classical Euler-Maruyama discretisation used in the Euclidean framework. De Bortoli et al. (2022) use Geodesic Random Walks (Jørgensen, 1975), which ensure that the samples remain on the manifold at every step. In this paper, we propose a sampler with similar properties in the case of *constrained* manifolds.

Reflected SDE. We conclude this section by recalling the framework for studying reflected SDEs, which is introduced via the notion of the *Skorokhod problem*. For simplicity, we focus on Euclidean space $\mathbb{R}^d$ here, but note that reflected processes can be defined on arbitrary smooth manifolds $\mathcal{N}$. In the case of the Brownian motion, a solution to the Skorokhod problem is a process of the form $(\bar{\mathbf{B}}_t, \mathbf{k}_t)_{t\geq 0}$. Locally, $(\bar{\mathbf{B}}_t)_{t\geq 0}$ can be seen as a regular Brownian motion $(\mathbf{B}_t)_{t\geq 0}$ while $(\mathbf{k}_t)_{t\geq 0}$ forces $(\bar{\mathbf{B}}_t)_{t\geq 0}$ to remain in $\mathcal{M}$. Under mild additional regularity conditions on $\mathcal{M}$ and $(\bar{\mathbf{B}}_t, \mathbf{k}_t)_{t\geq 0}$, see Skorokhod (1961), $(\bar{\mathbf{B}}_t, \mathbf{k}_t)_{t\geq 0}$ is a solution to the *Skorokhod problem* if for any $t \geq 0$

$$\bar{\mathbf{B}}_t = \bar{\mathbf{B}}_0 + \mathbf{B}_t - \mathbf{k}_t \in \mathcal{M}, \tag{5}$$

$|\mathbf{k}|_t = \int_0^t \mathbf{1}_{\bar{\mathbf{B}}_s \in \partial\mathcal{M}} \mathrm{d}|\mathbf{k}|_s$ and $\mathbf{k}_t = \int_0^t \mathbf{n}(\bar{\mathbf{B}}_s)\mathrm{d}|\mathbf{k}|_s$, where $(|\mathbf{k}|_t)_{t\geq 0}$ is the total variation of $(\mathbf{k}_t)_{t\geq 0}$[1]. Let us provide some intuition on this definition. When $(\bar{\mathbf{B}}_t)_{t\geq 0}$ hits the boundary $\partial\mathcal{M}$, $-\mathbf{k}_t$ pushes the process back into $\mathcal{M}$ along the inward normal $-\mathbf{n}(\bar{\mathbf{B}}_t)$, according to $\mathbf{k}_t = \int_0^t \mathbf{n}(\bar{\mathbf{B}}_s)\mathrm{d}|\mathbf{k}|_s$.

[1] In this case $(\mathbf{k}_t)_{t\geq 0}$ is not regular enough, but if it were in the class C^1, its total variation would be given by $\int_0^t |\partial_t \mathbf{k}_t| \mathrm{d}s$ in the one-dimensional case.



(a) Reflected Brownian motion (b) Metropolised approximation of Brownian motion

Figure 3: Evolution of the density of the reflected Brownian motion and its Metropolis sampling-based approximation on the unit interval starting from a delta mass.

The condition $|\mathbf{k}|_t = \int_0^t \mathbf{1}_{\bar{\mathbf{B}}_s \in \partial\mathcal{M}} \mathrm{d}|\mathbf{k}|_s$ is more technical and can be seen as imposing that $\mathbf{k}_t$ remains constant so long as $(\bar{\mathbf{B}}_t)_{t\geq 0}$ does not hit $\partial\mathcal{M}$. We refer to Fishman et al. (2023) and Lou et al. (2023) for a more thorough introduction of these concepts in the context of diffusion models.

3 Diffusion models for constrained manifolds via Metropolis sampling

In Section 3.1, we highlight the practical limitations of existing constrained diffusion models and propose an alternative Metropolis sampling-based approach. In Section 3.2, we outline our proof that this process corresponds to a valid discretisation of the reflected Brownian motion, justifying its use in diffusion models. An overview of the samplers we cover in this section is presented in Table 1.

3.1 Practical limitations of existing constrained diffusion models

Barrier methods. In the barrier approach, a constrained manifold is transformed into an unconstrained space via a barrier metric. This metric is defined by $\nabla^2\phi(x)$ with $\phi(x) = \sum_{i\in\mathcal{I}} \phi_i(d(x, f_i))$ where $d(x, f_i)$ is the minimum distance from the point x to the set defined by $f_i(x) = 0$ and ϕ_i is a monotonically decreasing function such that $\lim_{z\to 0} \phi_i(z) = \infty$. Under additional regularity assumptions, ϕ is called a *barrier function* (see Nesterov et al. (1994)). This definition ensures that the barrier function induces a well-defined exponential map on the manifold, making the Riemannian diffusion model frameworks of De Bortoli et al. (2022) and Huang et al. (2022) applicable. In the log-barrier method of Fishman et al. (2023), evaluating ϕ requires computing $d(x, \partial\mathcal{M})$ (and its derivatives), which can be prohibitively expensive. Furthermore, since the exponential map under the induced manifold is difficult to compute, it is approximated by projecting the exponential map on the original manifold back onto the constraint set, incurring an additional bias. Liu et al. (2023) propose a more tractable method by constructing a mirror map that transforms a constrained domain into an unconstrained dual space, in which one can train a standard Euclidean diffusion model. However, this approach is only applicable to convex subsets of $\mathbb{R}^d$ and does not extend to arbitrary Riemannian manifolds. More generally, warping the geometry of the modelled domain can adversely impact the interpolative performance of log-barrier-based diffusion models, as the space between data points expands rapidly when approaching the boundary.

Reflected stochastic processes. Fishman et al. (2023) and Lou et al. (2023) introduce diffusion models based on the *reflected Brownian motion* (RBM). In Fishman et al. (2023), the reflected SDE is discretised by (i) considering a classical step of the Euler-Maruyama discretization (or the Geodesic Random Walk in the Riemannian setting) and (ii) reflecting this step according to the boundary defined by $\partial\mathcal{M}$. To compute the reflection, one must check whether the step crosses the boundary. If it does, the point of intersection needs to be calculated in order to reflect the ray and continue the step in the reflected direction. This can require an arbitrarily large number of reflections depending on the step size, the geodesic on the manifold, and the geometry of the bounded region within the manifold. We refer to Appendix C for the pseudocode of the reflection step and additional comments. An alternative approach to discretising a reflected SDE is to replace the reflection with a projection (Słomiński, 1994). However, the projection requires the most expensive part of the reflection algorithm: computing the intersection of the geodesic with the boundary. Lou et al. (2023) propose a more tractable approach that exploits the product structure of the unit hypercube to afford simulation-free sampling but does not extend to arbitrary Riemannian manifolds. Additionally, specifying convex constraints in their framework requires a bijection onto the hypercube, distorting the modelled geometry and incurring the same issues as outlined above.

Table 1: Comparison of the advantages and disadvantages of the different constrained (Riemannian) diffusion models covered in Section 3.1.

Diffusion Model	Tractable conditional score	Simulation-free forward sampling	Modelling domain	Preserves metric of $\mathcal{M}$
	Both required for fast DSM loss			
Reflected Diffusions				
Lou et al. (2023)	✓	✓	convex $\subset \mathbb{R}^d$	✗
Fishman et al. (2023)	✗	$\mathcal{O}(d^2)$	Any $\mathcal{M}$	✓
Metropolis (ours)	✗	$\mathcal{O}(d)$	Any $\mathcal{M}$	✓
Barrier Diffusions				
Fishman et al. (2023)	✗	✗	convex $\subset$ any $\mathcal{M}$	✗
Liu et al. (2023)	✓	✓	convex $\subset \mathbb{R}^d$	✗

Metropolis approximation. Existing approaches to constrained (Riemannian) diffusion models either only apply to convex subsets of $\mathbb{R}^d$ or require manifold- and constraint-specific implementations that become computationally intractable as the complexity of the modelled geometry increases. This limits their practicality even for relatively simple manifolds with well-defined exponential maps and linear inequality constraints such as for example polytopes. In the following, we introduce a method that aims to solve both of these problems. The sampler we propose is similar to a classical Euler-Maruyama discretisation of the Brownian motion, except that, whenever a step would carry the Brownian motion out of the constrained region, we reject it (see Algorithm 1). This is a *Metropolised* version of the usual discretisation and is trivial to implement compared to the existing barrier, reflection and projection methods. Hence, this method enables the principled extension of diffusion models to arbitrarily constrained manifolds at virtually *no added implementational complexity or computational expense*.

Algorithm 1 *Metropolis approx. of RBM*

Require: $p \in \mathcal{M}$, $\{f_i\}_{i \in \mathcal{I}}$
  Sample $\boldsymbol{v} \sim \mathrm{N}(0, \mathrm{Id}) \in \mathrm{T}_p\mathcal{M}$
  $p' \leftarrow \exp_p(\boldsymbol{v})$
  if $f_i(p') < 0 \; \forall \, i$ **then**
    $p \leftarrow p'$
  end if
  return p

3.2 Relating the Metropolis sampler to the reflected Brownian motion

In this section, we prove that the proposed Metropolis sampling-based process (Algorithm 1) corresponds to a valid discretisation of the reflected process, justifying its use in diffusion models. Here we focus on a concise presentation of the core concepts and the main results. A full proof can be found in Appendix D. For simplicity, we restrict ourselves to the Euclidean setting. All of our results require particular assumptions on $\mathcal{M}$, which we discuss at the end of this section. We begin with a definition of the Metropolis approximation of RBM.

Definition 1. *For any $\gamma > 0$ and $k \in \mathbb{N}$, let $X_0^\gamma \in \mathcal{M}$ and $X_{k+1}^\gamma = X_k^\gamma + \sqrt{\gamma} Z_k^\gamma$ if $X_k^\gamma + \sqrt{\gamma} Z_k^\gamma \in \mathcal{M}$ and X_k^γ otherwise. The sequence $(X_k^\gamma)_{k \in \mathbb{N}}$ is called the Metropolis approximation of RBM.*

For any $\gamma > 0$, we consider $(\mathbf{X}_t^\gamma)_{t \geq 0}$, the linear interpolation of $(X_k^\gamma)_{k \in \mathbb{N}}$ such that for any $k \in \mathbb{N}$, $\mathbf{X}_{k\gamma}^\gamma = X_k^\gamma$. The following result is the main theoretical contribution of our paper.

Theorem 2. *Under assumptions on $\mathcal{M}$, for any $T \geq 0$, $(\mathbf{X}_t^\gamma)_{t \in [0,T]}$ weakly converges to the RBM $(\bar{\mathbf{B}}_t)_{t \in [0,T]}$ as $\gamma \to 0$.*

The rest of the section is devoted to a high level presentation of the proof of Theorem 2. It is theoretically impractical to work directly with the Metropolis approximation of RBM. Instead, we introduce an auxiliary process, show this converges to the RBM, and finally prove that the convergence of the auxiliary process implies the convergence of our Metropolis discretisation.

Definition 3. *For any $\gamma > 0$ and $k \in \mathbb{N}$, let $\hat{X}_0^\gamma = x \in \mathcal{M}$ and $\hat{X}_{k+1}^\gamma = \hat{X}_k^\gamma + \sqrt{\gamma} Z_k^\gamma$ with Z_k^γ a Gaussian random variable conditioned on $\hat{X}_k^\gamma + \sqrt{\gamma} Z_k^\gamma \in \mathcal{M}$. The sequence $(\hat{X}_k^\gamma)_{k \in \mathbb{N}}$ is called the Rejection approximation of RBM.*

We call this process *Rejection approximation of RBM* since in practice, Z_k^γ is sampled using rejection sampling, see Algorithm 2. For any $\gamma > 0$, we also consider $(\hat{\mathbf{X}}_t^\gamma)_{t\geq 0}$, the linear interpolation of $(\hat{X}_k^\gamma)_{k\in\mathbb{N}}$ such that for any $k \in \mathbb{N}$, $\hat{\mathbf{X}}_{k\gamma}^\gamma = \hat{X}_k^\gamma$. In Appendix D, we prove the following result.

Theorem 4. *Under assumptions on $\mathcal{M}$, for any $T \geq 0$, $(\hat{\mathbf{X}}_t^\gamma)_{t\in[0,T]}$ weakly converges to the Reflected Brownian Motion $(\bar{\mathbf{B}}_t)_{t\in[0,T]}$ as $\gamma \to 0$.*

Algorithm 2 *Rejection approx. of RBM*

Require: $p \in \mathcal{M}$, $\{f_i\}_{i\in\mathcal{I}}$
  Sample $\boldsymbol{v} \sim \mathrm{N}(0, \mathrm{Id}) \in \mathrm{T}_p\mathcal{M}$
  $p' \leftarrow \exp_p(\boldsymbol{v})$
  while $f_i(p') \geq 0$ for any i **do**
    Sample $\boldsymbol{v} \sim \mathrm{Id}(0, 1) \in \mathrm{T}_p\mathcal{M}$
    $p' \leftarrow \exp_p(\boldsymbol{v})$
  end while
  return p'

Proof. Here we give some elements of the proof. Details and full derivations are postponed to Appendix D. Our approach is based on the invariance principle of Stroock et al. (1971). More precisely, we show that we can compute an equivalent 'drift' and 'diffusion matrix' for the discretised process and that, as $\gamma \to 0$, the drift converges to zero and the diffusion matrix converges to Id. In the Euclidean setting, this result, accompanied by mild regularity and growth assumptions, ensures that the discretization weakly converges to the original SDE. However, the case with boundary is much more complicated, primarily because the approximate drift might explode near the boundary, thus we need to verify exactly how the drift behaves as $\gamma \to 0$ and as the process approaches the boundary. We show that the *normalised* drift converges to the inward normal near the boundary. This ensures that (a) in the interior of $\mathcal{M}$ the drift converges to zero, i.e. locally in the interior of $\mathcal{M}$ the Brownian motion and the Reflected Brownian Motion coincide, (b) on the boundary, the drift pushes the samples inside the manifold according to the inward normal, mimicking $(\mathbf{k}_t)_{t\geq 0}$ in (5). Finally, with results from Stroock et al. (1971) and Kang et al. (2017), we show the convergence to the RBM. □

Our next step is to show that the approximate drift and diffusion matrix of the Metropolised process are upper and lower bounded by their counterparts in the rejection process. While the upper-bound is easy to derive, the lower-bound requires the following result.

Proposition 5. *Under assumptions on $\mathcal{M}$, $\forall\ \varepsilon > 0$, $\exists\ \bar{\gamma} > 0$ such that for any $\gamma \in (0, \bar{\gamma})$ and for any $x \in \mathcal{M}$, $\gamma \in (0, \bar{\gamma})$ and $Z \sim \mathrm{N}(0, \mathrm{Id})$ we have $\mathbb{P}(x + \sqrt{\gamma}Z \in \mathcal{M}) \geq 1/2 - \varepsilon$, with $Z \sim \mathrm{N}(0, \mathrm{Id})$.*

Proposition 5 tells us that *locally* the boundary looks like a half-space when integrating w.r.t. a Gaussian measure. A corollary is that, for $\gamma > 0$ small enough and for any $k \in \mathbb{N}$, the probability that $X_{k+1}^\gamma = X_k^\gamma$ is upper bounded *uniformly* w.r.t. $X_k^\gamma \in \mathcal{M}$. The proof of Proposition 5 uses Theorem 7 in Appendix D, whose proof relies on the concept of *tubular neighborhoods* (Lee et al., 2012).

Having established the lower and upper bound, we can conclude the proof by noting that the approximate drift and the diffusion matrix in the rejection and Metropolis case coincide as $\gamma \to 0$. This is enough to apply the same results as before, giving the desired convergence.

Assumptions on $\mathcal{M}$. Before concluding this section, we detail the assumptions we make on $\mathcal{M}$. For Theorem 2 to hold, we assume that $\mathcal{M} = \{x \in \mathbb{R}^d : \Phi(x) > 0\}$ is bounded, with $\Phi \in \mathrm{C}^2(\mathbb{R}^d, \mathbb{R})$ concave. We have that $\partial\mathcal{M} = \{x \in \mathbb{R}^d : \Phi(x) = 0\}$. In addition, we assume that for any $x \in \partial\mathcal{M}$, $\|\nabla\Phi(x)\| = 1$. These assumptions match those Stroock et al. (1971) use for their study of the existence of solutions to the RBM. While it seems possible to relax the *global* existence of Φ to a *local* one, the regularity assumption of the domain is key. This regularity is essential to establish Proposition 5 and the associated geometrical result on tubular neighbourhoods. We also emphasize that the smoothness of the domain is central in the results of Kang et al. (2017) on the equivalence of two definitions of RBMs which we rely on.

4 Related work on approximations of reflected SDEs

Several schemes have been introduced to approximately sample from reflected Stochastic Differential Equations. They can be interpreted as modifications of classical Euler-Maruyama schemes used to discretise SDEs without boundary. One of the most common approaches is to use the Euler-Maruyama discretisation and project the solution onto the boundary if it escapes from the domain $\mathcal{M}$.

Table 2: Log-likelihood (↑) of a held-out test set from a synthetic bimodal distribution over convex subsets of $\mathbb{R}^d$ bounded by the hypercube $[-1,1]^d$ and unit simplex Δ^d. Means and standard deviations are computed over 3 different runs. Average training time is provided in hours.

MANIFOLD	DIMENSION	REFLECTED		METROPOLIS	
		log-likelihood	runtime	log-likelihood	runtime
$[-1,1]^d$	2	$2.25_{\pm.01}$	8.95	$\mathbf{2.32}_{\pm.05}$	**0.72**
	3	$3.77_{\pm.13}$	8.97	$\mathbf{4.15}_{\pm.15}$	**0.71**
	10	$7.42_{\pm.77}$	10.1	$\mathbf{10.80}_{\pm.34}$	**0.90**
Δ^d	2	$1.01_{\pm.01}$	9.17	$\mathbf{1.06}_{\pm.02}$	**0.82**
	3	$2.64_{\pm.01}$	9.43	$\mathbf{3.23}_{\pm.17}$	**0.78**
	10	$7.00_{\pm.13}$	10.5	$\mathbf{7.81}_{\pm.20}$	**0.97**

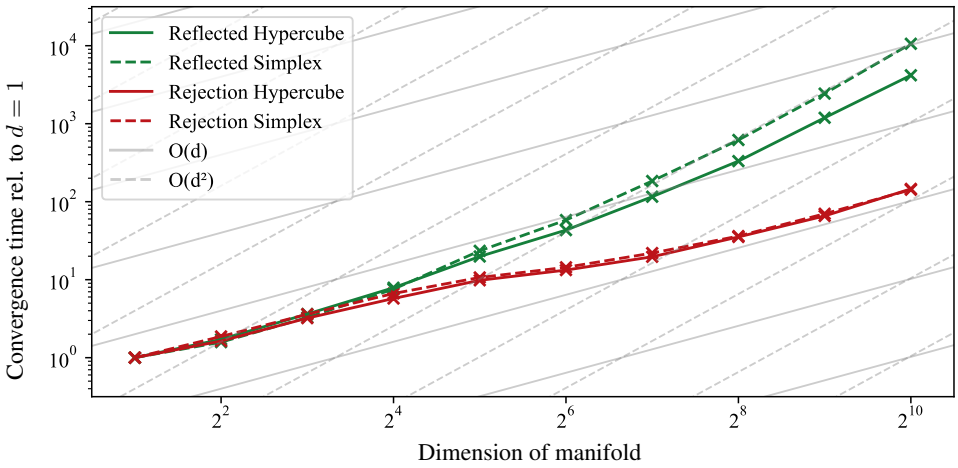
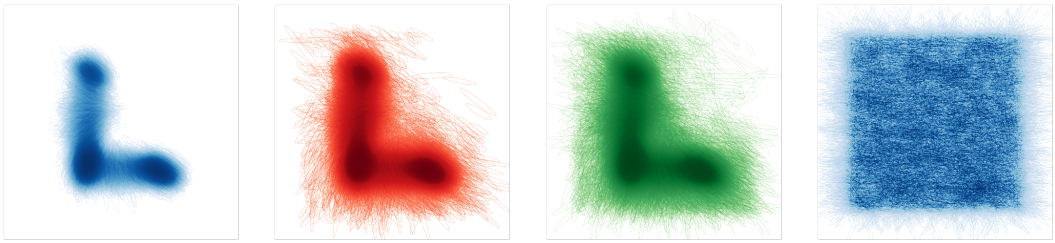


Figure 4: Convergence time of the Reflected (green) and Metropolis (red) forward noising processes to the uniform distribution on the hypercube $[-1,1]^d$ and unit simplex Δ^d. The lines indicate functions fit with the PySR symbolic regression package (Cranmer, 2023) and correspond to empirical runtime complexities of $\mathcal{O}(d^2)$ and $\mathcal{O}(d)$, respectively, matching the superimposed scaling law isocontours.

In this case, mean-square error rates of order *almost* $1/2$ have been proven under various conditions (Liu, 1995; Chitashvili et al., 1981; Pettersson, 1995; Słomiński, 1994). Concretely this means that $\mathbb{E}[\|\bar{\mathbf{B}}_t - X_n^{t/n}\|^2] = O(n^{-1+\varepsilon})$ with $\varepsilon > 0$ arbitrary small and where $(X_k^\gamma)_{k\in\mathbb{N}}$ is the projection scheme. The rate $1/2$ is tight (Pacchiarotti et al., 1998). It is possible to use the Euler-Peano method to get slight improvements for a mean-square error rate of order of $1/2$, but this is impractical as it assumes that one can solve a (simplified) Skorokhod problem, which is usually intractable. Liu (1993) introduced a *penalised* method which pushes the solution away from the boundary and shows a mean-square error of order $1/4$, see also Pettersson (1997). Weak errors of order 1 have been obtained in Bossy et al. (2004) and Gobet (2001) by introducing a reflection component in the discretisation or using some local approximation of the domain to a half-space. We refer to Pilipenko (2014) for an introduction to the discretisation of reflected SDEs. Closer to our work, Burdzy et al. (2008) consider three different methods to approximate reflected Brownian motions on general domains (two based on discrete methods and one based on killed diffusions). Only qualitative results are provided. To the best of our knowledge, no previous work in the probability literature has investigated the *Metropolised* scheme we propose. Our Metropolis scheme is also related to the ball walk (Applegate et al., 1991), which replaces the Gaussian random variable with a uniform over the ball (or the Dikin ellipsoid). Applegate et al. (1991) and Lovász et al. (2007) have studied the asymptotic convergence rate of the ball walk, but, to the best of our knowledge, its limiting behaviour when the step size goes to zero has not been investigated.



(a) Data distribution. (b) Metropolis samples. (c) Reflected samples. (d) Uniform distribution.

Figure 5: A qualitative visual comparison of 10^6 samples from the data distribution, our Metropolis diffusion model, a reflected diffusion model and the uniform distribution for the constrained configurational modelling of robotic arms on $\mathcal{S}^2_{++} \times \mathbb{R}^2$.

5 Experimental results

To demonstrate the practical utility and empirical performance of the proposed Metropolis diffusion models, we conduct a comprehensive evaluation on a range of synthetic and real-world tasks. In Section 5.1, we assess the scalability of our method by applying it to synthetic distributions on hypercubes and simplices of increasing dimensionality. In Section 5.2, we extend the evaluation to real-world tasks on manifolds with convex constraints by applying our method to the robotics and protein design datasets presented in Fishman et al. (2023). In Section 5.3, we additionally demonstrate that our method extends to constrained manifolds with highly *non-convex* boundaries—a setting that is intractable with existing approaches.

As we found—in line with Fishman et al. (2023)—that log-barrier diffusion models perform strictly worse than reflected approaches across all experimental settings, we focus on a more detailed comparison with the latter here and postpone additional empirical results to Appendix F.1. These include additional performance metrics and a comparison to an unconstrained Euclidean diffusion model on the synthetic datasets presented in Section 5.1.

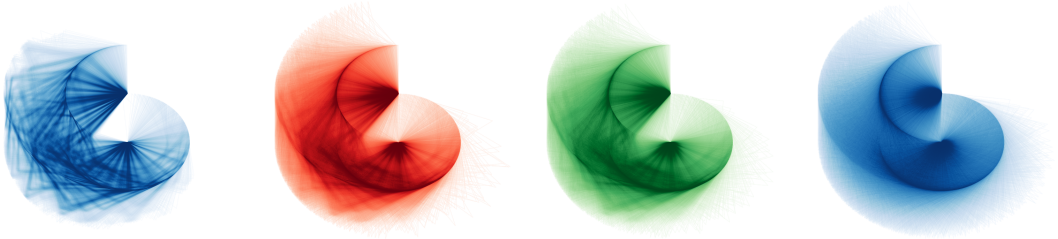
For all experiments, we use a simple 6-layer MLP with sine activations and a score rescaling function to ensure that the score reaches zero at the boundary, scaling linearly into the interior of the constrained set as in Liu et al. (2022) and Fishman et al. (2023). We set $T = 1$, $\beta_0 = 1 \times 10^{-3}$ and tune β_1 to ensure that the forward process reaches the invariant distribution with a linear β-schedule. We use a learning rate of 2×10^{-4} with a cosine learning rate schedule and an ism loss with a modified loss weighting function of $(1 + t)$, a batch size of 1024 and 8 repeats per batch. All models were trained on a single NVIDIA GeForce GTX 1080 GPU. Additional details are provided in Appendix F.2.

All source code that is needed to reproduce the results presented below is made available under https://github.com/oxcsml/score-sde/tree/metropolis, which requires a supporting package to handle the different geometries that is available under https://github.com/oxcsml/geomstats/tree/polytope.

5.1 Synthetic distributions on simple polytopes

In this section, we investigate the scalability of the proposed Metropolis diffusion models by applying them to synthetic bimodal distributions over the d-dimensional hypercube $[-1, 1]^d$ and unit simplex Δ^d. A quantitative comparison of the log-likelihood of a held-out test set is presented in Table 2, while a visual comparison is postponed to Appendix F.3. We find that our Metropolis models outperform reflected approaches across all dimensions and constraint geometries by a substantial and statistically significant margin while training in one tenth of the time. The degree of improvement seems to scale with the dimensionality of the problem: the larger the dimension of the experiment, the larger the gain in performance from using our proposed Metropolis scheme.

We observe a similar difference in the scaling properties of reflected and Metropolis models when measuring the convergence times of the respective forward noising processes to the uniform distribution on hypercubes $[-1, 1]^d$ and simplices Δ^d of increasing dimensionality. The results are presented in Section 4 and show that the convergence time of the Metropolis process scales linearly in the dimension, while the reflected process scales quadratically.



(a) Data distribution. (b) Metropolis samples. (c) Reflected samples. (d) Uniform distribution.

Figure 6: A qualitative comparison of 10^5 samples from the data distribution, our Metropolis diffusion model, a reflected diffusion model and the uniform distribution for the constrained conformational modelling of cyclic peptide backbones. For visual clarity, the figures only show the constrained planar projections encoded by $\mathbb{P} \subset \mathbb{R}^3$.

5.2 Modelling proteins and robotic arms under convex constraints

In addition to illustrating our method's scalability on high-dimensional synthetic tasks, we follow the experimental setup from Fishman et al. (2023) to additionally demonstrate its practical utility and favourable empirical performance on two real-world problems from robotics and protein design.

Constrained configurational modelling of robotic arms. The problem of modelling the configurations and trajectories of a robotic arm can be formulated as learning a distribution over the locations and manipulability ellipsoids of its joints, parameterised on $\mathbb{R}^d \times \mathcal{S}^d_{++}$, where $\mathcal{S}^d_{++}$ is the manifold of symmetric positive-definite (SPD) $d \times d$ matrices (Yoshikawa, 1985; Jaquier et al., 2021). For practical robotics applications, it may be desirable to restrict the maximal velocity with which a robotic arm can move or the maximum force it can exert. This manifests in a trace constraint $C > 0$ on $\mathcal{S}^d_{++}$, resulting in a constrained manifold $\{M \in \mathcal{S}^d_{++} : \sum_{i=1}^d M_{ii} < C\}$. Following Fishman et al. (2023), we parametrise this constraint via the Cholesky decomposition (Lin, 2019) and use the resulting setup to model the dataset presented in Jaquier et al. (2021).

Conformational modelling of protein backbones. Modelling the conformational ensembles of proteins is a data-scarce problem with a range of important applications in biotechnology and drug discovery (Lane, 2023). In many practical settings, it may often be unnecessary to model the structural ensembles of an entire protein, as researchers are primarily interested in specific functional sites that are embedded in a structurally conserved scaffold (Huang et al., 2016). Modelling the conformational ensembles of such substructural elements requires positional constraints on their endpoints to ensure that they can be accommodated by the remaining protein. Using the parametrisation and dataset presented in Fishman et al. (2023), we formulate the problem of modelling the backbone conformations of a cyclic peptide of length $N = 6$ as learning a distribution over the product of a polytope $\mathbb{P} \subset \mathbb{R}^3$ and the hypertorus $\mathbb{T}^4$.

Table 3: Log-likelihood (↑) of a held-out test set for the robotics and protein applications. Means and standard deviations are computed over 3 different runs. Average training time is provided in hours.

DATASET	DOMAIN	REFLECTED		METROPOLIS	
		log-likelihood	runtime	log-likelihood	runtime
Robotics	$\mathcal{S}^2_{++} \times \mathbb{R}^2$	$8.39_{\pm.06}$	9.52	$\mathbf{9.13}_{\pm.03}$	**1.36**
Proteins	$\mathbb{P} \subset \mathbb{R}^3 \times \mathbb{T}^4$	$15.20_{\pm.06}$	24.80	$\mathbf{15.33}_{\pm.02}$	**3.12**

We quantify the empirical performance of different methods by evaluating the log-likelihood of a held-out test set and present the resulting performance metrics in Table 3. Again, we find that our Metropolis model outperforms the reflected approach by a statistically significant margin while training 7-8 times as fast. Qualitative visual comparisons of samples from the true distribution, the trained diffusion models and the uniform distribution are presented in Figures 5 and 6, with full univariate marginal and pairwise bivariate correlation plots postponed to Appendices F.4 and F.5.



(a) Data distribution. (b) Metropolis samples. (c) Uniform distribution.

Figure 7: Orthographic projection of 10^5 samples from (a) the data distribution, (b) our Metropolis diffusion model, and (c) the uniform distribution, for geospatial data (wildfire incidence rates) within a non-convex boundary (the continental United States). The projections are aligned with the geometric centre of the boundary and zoomed in ten-fold for visual clarity.

5.3 Modelling geospatial data within non-convex country borders

Motivated by the strong empirical performance of our approach on tasks with challenging convex constraints, we investigated its ability to model distributions whose support is restricted to manifolds with highly non-convex boundaries—a setting that is intractable with existing approaches. To this end, we derived a geospatial dataset based on wildfire incidence rates within the continental United States (see Appendix E for full details) and trained a Metropolis diffusion model constrained by the corresponding country borders on the sphere $\mathcal{S}^2$. A qualitative visual comparison of samples from the true distribution, our model, and the uniform distribution is presented in Figures 7a to 7c and a quantitative comparison to a Riemannian diffusion model on $\mathcal{S}^2$ (De Bortoli et al., 2022) is given in Table 4. Both demonstrate that our approach is able to successfully capture challenging multimodal and sparse distributions on constrained manifolds with highly non-convex boundaries.

Table 4: MMD (↓) of a held-out test set for the geospatial modelling dataset. Means and standard deviations are computed over 3 different runs. Average training time is provided in hours.

MODEL	DOMAIN	MMD	RUNTIME	% IN BOUNDARY
Unconstrained	$\mathcal{S}^2$	$0.1567_{\pm 0.013}$	**0.81**	63.3
Metropolis	$\mathbb{P} \subset \mathcal{S}^2$	$\mathbf{0.1388_{\pm 0.015}}$	3.86	**100.0**

6 Conclusion

Accurately modelling distributions on constrained Riemannian manifolds is a challenging problem with a range of impactful practical applications. In this work, we have proposed a mathematically principled and computationally tractable extension of existing diffusion model methodology to this setting. Based on a *Metropolisation* of random walks in Euclidean spaces and on Riemannian manifolds, we have shown that our approach corresponds to a valid discretisation of the reflected Brownian motion, justifying its use in diffusion models. To demonstrate the practical utility of our method, we have performed an extensive empirical evaluation, showing that it outperforms existing constrained diffusion models on a range of synthetic and real-world tasks defined on manifolds with convex boundaries, including applications from robotics and protein design. Leveraging the flexibility and simplicity of our method, we have also demonstrated that it extends beyond convex constraints and is able to successfully model distributions on manifolds with highly non-convex boundaries. While we found our method to perform well across the synthetic and real-world applications we considered, we expect it to perform poorly on certain constraint geometries. For instance, the current implementation relies on an isotropic noise distribution which could impede its performance on exceedingly narrow constraint geometries, even with correspondingly small step sizes. In this context, an important direction of future research would be to investigate whether we can instead sample from more suitable distributions, e.g. a Dikin ellipsoid, while maintaining the simplicity and efficiency of the Metropolis approach.

Acknowledgements

NF thanks the Rhodes Trust for supporting their studies at Oxford and this work. LK acknowledges support from the University of Oxford's Clarendon Fund.

References

B. D. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. Cited on page 3.

D. Applegate and R. Kannan. Sampling and integration of near log-concave functions. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 156–163, 1991. Cited on page 7.

J. A. Bednar, J. Crail, J. Crist-Harif, P. Rudiger, G. Brener, I. Thomas, C. B, J. Mease, J. Signell, M. Liquet, J.-L. Stevens, B. Collins, A. Thorve, thuydotm, S. H. Hansen, esc, kbowen, N. Abdennur, O. Smirnov, maihde, A. Hawley, A. Oriekhov, A. Ahmadia, B. A. B. Jr, C. H. Brandt, C. Tolboom, E. G., E. Welch, J. Bourbeau, and J. J. Schmidt. Holoviz/datashader: version 0.14.4, version v0.14.4, Feb. 2023. DOI: 10.5281/zenodo.7599872. URL: https://doi.org/10.5281/zenodo.7599872. Cited on page 16.

M. Bevis and J.-L. Chatelain. Locating a point on a spherical surface relative to a spherical polygon of arbitrary shape. *Mathematical geology*, 21:811–828, 1989. Cited on pages 15, 16.

M. Bossy, E. Gobet, and D. Talay. A symmetrized euler scheme for an efficient approximation of reflected diffusions. *Journal of applied probability*, 41(3):877–889, 2004. Cited on page 7.

K. Burdzy and Z.-Q. Chen. Discrete approximations to reflected brownian motion, 2008. Cited on page 7.

P. Cattiaux, G. Conforti, I. Gentil, and C. Léonard. Time reversal of diffusion processes under a finite entropy condition. *arXiv preprint arXiv:2104.07708*, 2021. Cited on page 3.

R. Chitashvili and N. Lazrieva. Strong solutions of stochastic differential equations with boundary conditions. *Stochastics: an international journal of probability and stochastic processes*, 5(4):255–309, 1981. Cited on page 7.

G. Corso, H. Stärk, B. Jing, R. Barzilay, and T. Jaakkola. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking, Oct. 2022. URL: http://arxiv.org/abs/2210.01776. Cited on page 1.

M. Cranmer. Interpretable machine learning for science with pysr and symbolicregression. jl. *arXiv preprint arXiv:2305.01582*, 2023. Cited on page 7.

V. De Bortoli, E. Mathieu, M. Hutchinson, J. Thornton, Y. W. Teh, and A. Doucet. Riemannian score-based generative modeling, 2022. arXiv: 2202.02763 [cs.LG]. Cited on pages 1–4, 10, 18.

N. Fishman, L. Klarner, V. De Bortoli, E. Mathieu, and M. Hutchinson. Diffusion models for constrained domains. *arXiv preprint arXiv:2304.05364*, 2023. Cited on pages 2, 4, 5, 8, 9, 1, 16–18, 20.

K. Gatmiry and S. S. Vempala. Convergence of the riemannian langevin algorithm. *arXiv preprint arXiv:2204.10818*, 2022. Cited on page 2.

E. Gobet. Euler schemes and half-space approximation for the simulation of diffusion in a domain. *ESAIM: Probability and Statistics*, 5:261–297, 2001. Cited on page 7.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(null):723–773, Mar. 2012. ISSN: 1532-4435. Cited on page 16.

L. Han and L. Rudolph. Inverse kinematics for a serial chain with joints under distance constraints. In *Robotics: Science and systems*, 2006. Cited on pages 1, 20.

U. G. Haussmann and E. Pardoux. Time reversal of diffusions. *The Annals of Probability*:1188–1205, 1986. Cited on page 3.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020. Cited on pages 1, 3.

E. P. Hsu. *Stochastic analysis on manifolds*, number 38. American Mathematical Soc., 2002. Cited on page 2.

C.-W. Huang, M. Aghajohari, A. J. Bose, P. Panangaden, and A. Courville. Riemannian Diffusion Models, Aug. 2022. DOI: 10.48550/arXiv.2208.07949. Cited on pages 1, 3, 4.

P.-S. Huang, S. E. Boyken, and D. Baker. The coming of age of de novo protein design. *Nature*, 537(7620):320–327, 2016. Cited on page 9.

N. Jaquier, L. Rozo, D. G. Caldwell, and S. Calinon. Geometry-aware manipulability learning, tracking, and transfer. *The International Journal of Robotics Research*, 40(2-3):624–650, 2021. Cited on page 9.

B. Jing, G. Corso, J. Chang, R. Barzilay, and T. Jaakkola. Torsional Diffusion for Molecular Conformer Generation, June 2022. URL: http://arxiv.org/abs/2206.01729. Cited on page 1.

E. Jørgensen. The central limit problem for geodesic random walks. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 32(1-2):1–64, 1975. Cited on page 3.

W. Kang and K. Ramanan. On the submartingale problem for reflected diffusions in domains with piecewise smooth boundaries, 2017. Cited on pages 6, 4, 14.

R. Kannan and H. Narayanan. Random walks on polytopes and an affine interior point method for linear programming. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pages 561–570, Bethesda, MD, USA. Association for Computing Machinery, 2009. ISBN: 9781605585062. DOI: 10.1145/1536414.1536491. URL: https://doi.org/10.1145/1536414.1536491. Cited on page 2.

R. Ketzner, V. Ravindra, and M. Bramble. A robust, fast, and accurate algorithm for point in spherical polygon classification with applications in geoscience and remote sensing. *Computers & Geosciences*, 167:105185, 2022. Cited on pages 15, 16.

Y. Kook, Y. T. Lee, R. Shen, and S. S. Vempala. Sampling with riemannian hamiltonian monte carlo in a constrained space. *arXiv preprint arXiv:2202.01908*, 2022. Cited on page 2.

T. J. Lane. Protein structure prediction has reached the single-structure frontier. *Nature Methods*:1–4, 2023. Cited on page 9.

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*:1302–1338, 2000. Cited on page 6.

J. M. Lee. Smooth manifolds. In *Introduction to Smooth Manifolds*, pages 1–31. Springer, 2013. Cited on pages 2, 1.

J. M. Lee and J. M. Lee. *Smooth manifolds*. Springer, 2012. Cited on pages 6, 4, 5.

Y. T. Lee and S. S. Vempala. Convergence rate of riemannian hamiltonian monte carlo and faster polytope volume computation. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1115–1121, 2018. Cited on page 2.

Y. T. Lee and S. S. Vempala. Geodesic walks in polytopes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 927–940, Montreal Canada. ACM, June 2017. ISBN: 978-1-4503-4528-6. DOI: 10.1145/3055399.3055416. Cited on page 2.

Z. Lin. Riemannian geometry of symmetric positive definite matrices via cholesky decomposition. *SIAM Journal on Matrix Analysis and Applications*, 40(4):1353–1370, 2019. Cited on pages 9, 2.

G.-H. Liu, T. Chen, E. A. Theodorou, and M. Tao. Mirror diffusion models for constrained and watermarked generation. *arXiv preprint arXiv:2310.01236*, 2023. Cited on pages 2, 4, 5.

S. Liu, T. Kanamori, and D. J. Williams. Estimating density models with truncation boundaries using score matching. *Journal of Machine Learning Research*, 23(186):1–38, 2022. Cited on page 8.

Y. Liu. Discretization of a class of reflected diffusion processes. *Mathematics and computers in simulation*, 38(1-3):103–108, 1995. Cited on page 7.

Y. Liu. *Numerical approaches to stochastic differential equations with boundary conditions*. Purdue University, 1993. Cited on page 7.

A. Lou and S. Ermon. Reflected diffusion models. *arXiv preprint arXiv:2304.04740*, 2023. Cited on pages 2, 4, 5.

L. Lovász and S. Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms*, 30(3):307–358, 2007. Cited on page 7.

J. M. Lukens, K. J. Law, A. Jasra, and P. Lougovski. A practical and efficient approach for bayesian quantum state estimation. *New Journal of Physics*, 22(6):063038, 2020. Cited on page 1.

Met Office. *Cartopy: a cartographic python library with a Matplotlib interface*. Exeter, Devon, 2010 - 2015. URL: https://scitools.org.uk/cartopy. Cited on page 15.

B. J. Morris. Improved bounds for sampling contingency tables. *Random Structures & Algorithms*, 21(2):135–146, 2002. DOI: https://doi.org/10.1002/rsa.10049. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rsa.10049. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rsa.10049. Cited on page 1.

Natural Earth. Natural earth, 2023. URL: https://www.naturalearthdata.com/. [Online; accessed 31-Jan-2023]. Cited on page 15.

Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994. Cited on page 4.

M. Noble, V. De Bortoli, and A. Durmus. Barrier Hamiltonian Monte Carlo, Oct. 2022. DOI: 10.48550/arXiv.2210.11925. Cited on page 2.

B. Pacchiarotti, C. Costantini, and F. Sartoretto. Numerical approximation for functionals of reflecting diffusion processes. *SIAM Journal on Applied Mathematics*, 58(1):73–102, 1998. Cited on page 7.

F. Petit. Time reversal and reflected diffusions. *Stochastic Processes and their Applications*, 69(1):25–53, July 1997. ISSN: 03044149. DOI: 10.1016/S0304-4149(97)00035-5. Cited on page 2.

R. Pettersson. Approximations for stochastic differential equations with reflecting convex boundaries. *Stochastic processes and their applications*, 59(2):295–308, 1995. Cited on page 7.

R. Pettersson. Penalization schemes for reflecting stochastic differential equations. *Bernoulli*:403–414, 1997. Cited on page 7.

A. Pilipenko. *An introduction to stochastic differential equations with reflection*, volume 1. Universitätsverlag Potsdam, 2014. Cited on page 7.

K. Ramanan. Reflected diffusions defined via the extended skorokhod map, 2006. Cited on pages 4, 14.

P. Rudiger, J. A. Bednar, J.-L. Stevens, M. Liquet, C. B, S. H. Hansen, B. Little, Andrew, J. Signell, M. Hedley, C. Bosley, R. Hattersley, G. Brener, ea42gh, kbowen, A. Hilboll, I. Lustig, J. deWerd, N. Hand, R. Signell, J. Bampton, pmav99, scaine1, and zassa. Holoviz/geoviews: version 1.9.6, version v1.9.6, Jan. 2023. DOI: 10.5281/zenodo.7543863. URL: https://doi.org/10.5281/zenodo.7543863. Cited on page 16.

C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. Cited on page 2.

M. Shkolnikov and I. Karatzas. Time-reversal of reflected Brownian motions in the orthant, July 2013. URL: http://arxiv.org/abs/1307.4422. Cited on page 2.

A. V. Skorokhod. Stochastic equations for diffusion processes in a bounded region. *Theory of Probability & Its Applications*, 6(3):264–274, 1961. Cited on page 3.

L. Słomiński. On approximation of solutions of multidimensional sde's with reflecting boundary conditions. *Stochastic processes and their Applications*, 50(2):197–219, 1994. Cited on pages 4, 7.

J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. Cited on page 1.

Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019. Cited on pages 1, 3.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. Cited on pages 1, 3.

D. W. Stroock and S. S. Varadhan. Diffusion processes with boundary conditions. *Communications on Pure and Applied Mathematics*, 24(2):147–225, 1971. Cited on pages 2, 6, 4, 14, 15.

I. Thiele, N. Swainston, R. Fleming, A. Hoppe, S. Sahoo, M. Aurich, H. Haraldsdottir, M. Mo, O. Rolfsson, M. Stobbe, S. Thorleifsson, R. Agren, C. Bölling, S. Bordel, A. Chavali, P. Dobson, W. Dunn, L. Endler, D. Hala, M. Hucka, D. Hull, D. Jameson, N. Jamshidi, J. Jonsson, N. Juty, S. Keating, I. Nookaew, N. Le Novère, N. Malys, A. Mazein, J. Papin, N. Price, E. Selkov, M. Sigurdsson, E. Simeonidis, N. Sonnenschein, K. Smallbone, A. Sorokin, J. van Beek, D. Weichart, I. Goryanin, J. Nielsen, H. Westerhoff, D. Kell, P. Mendes, and B. Palsson. A community-driven global reconstruction of human metabolism. English. *Nature Biotechnology*, Mar. 2013. ISSN: 1087-0156. DOI: 10.1038/nbt.2488. Cited on page 1.

B. L. Trippe, J. Yim, D. Tischer, T. Broderick, D. Baker, R. Barzilay, and T. Jaakkola. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem, June 2022. DOI: 10.48550/arXiv.2206.04119. Cited on page 1.

J. Urain, N. Funk, G. Chalvatzaki, and J. Peters. Se (3)-diffusionfields: learning cost functions for joint grasp and motion optimization through diffusion. *arXiv preprint arXiv:2209.03855*, 2022. Cited on page 1.

P. Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. Cited on page 3.

J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S. V. Torres, A. Lauko, V. D. Bortoli, E. Mathieu, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, and D. Baker. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models, Dec. 2022. DOI: 10.1101/2022.12.09.519842. Cited on page 1.

J. Welty and M. Jeffries. Combined wildfire datasets for the united states and certain territories, 1878-2019: us geological survey data release, 2020. Cited on page 15.

R. J. Williams. Reflected Brownian motion with skew symmetric data in a polyhedral domain. *Probability Theory and Related Fields*, 75(4):459–485, Aug. 1987. ISSN: 0178-8051, 1432-2064. DOI: 10.1007/BF00320328. Cited on page 2.

J. Yim, B. L. Trippe, V. De Bortoli, E. Mathieu, A. Doucet, R. Barzilay, and T. Jaakkola. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023. Cited on page 1.

T. Yoshikawa. Manipulability of robotic mechanisms. *The international journal of Robotics Research*, 4(2):3–9, 1985. Cited on page 9.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Metropolis sampling for constrained diffusion models
Publication Status	Published
Publication Details	Fishman, N., Klarner, L., Mathieu, E., Hutchinson, M. and De Bortoli, V., 2023. Metropolis sampling for constrained diffusion models. *Advances in Neural Information Processing Systems*, *36*, pp.62296-62331.

Student Confirmation

Student Name	Leo Klarner		
Contribution	• Conceptualized key research ideas in collaboration with all other co-authors. • Identified relevant datasets and baselines that allowed for a meaningful empirical evaluation of the proposed method. Collaborated with NF in developing practical implementations and code. • Collaborated with NF, VDB and MH in the development of specialized, problem-specific adaptations of the proposed approaches. • Collaborated with all other co-authors in developing the framing of the method and drafting of the manuscript. Engaged with reviewers during the peer review process in collaboration with all other co-authors.		
Signature		Date	Lausanne, 14. April 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Yee-Whye Teh		
Supervisor comments: I confirm.		
Signature 	Date	23 April 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 6

Drug Discovery under Covariate Shift

This chapter is based on the following publication:

L. Klarner, T. G. Rudner, M. Reutlinger, T. Schindler, G. M. Morris, C. M. Deane, Y. W. Teh. Drug discovery under covariate shift with domain-informed prior distributions over functions. *International Conference on Machine Learning, 2023.*

Personal contributions:

1. Conceptualised key research ideas and methodological contributions in collaboration with TGR.
2. Identified relevant datasets and baselines that allowed for a meaningful empirical evaluation of the proposed method in collaboration with all other co-authors.
3. Carried out all experiments, including data pre-processing, model training, sample generation and the analysis and visualisation of results.
4. Developed the framing and presentation of the method in collaboration with all other co-authors. Drafted the manuscript and improved it in collaboration with all other co-authors.
5. Engaged with reviewers during the peer review process in collaboration with TGR.

A full statement of contributions can be found at the end of this chapter.

The supplementary information for this chapter is provided in Appendix D.

Drug Discovery under Covariate Shift with Domain-Informed Prior Distributions over Functions

Leo Klarner [1] **Tim G. J. Rudner** [1] **Michael Reutlinger** [2] **Torsten Schindler** [2]
Garrett M. Morris [1] **Charlotte M. Deane** [1] **Yee Whye Teh** [1]

Abstract

Accelerating the discovery of novel and more effective therapeutics is an important pharmaceutical problem in which deep learning is playing an increasingly significant role. However, real-world drug discovery tasks are often characterized by a scarcity of labeled data and significant covariate shift—a setting that poses a challenge to standard deep learning methods. In this paper, we present Q-SAVI, a probabilistic model able to address these challenges by encoding explicit prior knowledge of the data-generating process into a prior distribution over functions, presenting researchers with a transparent and probabilistically principled way to encode data-driven modeling preferences. Building on a novel, gold-standard bioactivity dataset that facilitates a meaningful comparison of models in an extrapolative regime, we explore different approaches to induce data shift and construct a challenging evaluation setup. We then demonstrate that using Q-SAVI to integrate contextualized prior knowledge of drug-like chemical space into the modeling process affords substantial gains in predictive accuracy and calibration, outperforming a broad range of state-of-the-art self-supervised pre-training and domain adaptation techniques.

1. Introduction

Discovering novel drug candidates that are able to safely and effectively treat neglected diseases or combat multidrug-resistant pathogens is a challenging biomedical research problem of considerable scientific and societal importance. Leveraging modern deep learning algorithms to accurately predict clinically relevant molecular properties and reduce the need for time- and resource-intensive experiments has the potential to significantly accelerate the development of promising and innovative chemical leads in drug discovery.

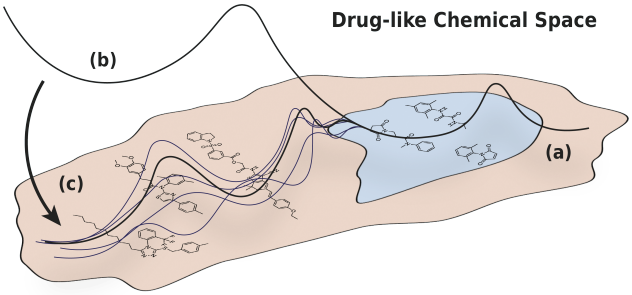


Figure 1. In early-stage drug discovery, bioactivity labels are usually only available for a small and biased subset of compounds (a). However, predictions are often most useful for novel molecules that are dissimilar to the ones already explored—an evaluative setting in which deep learning algorithms perform unreliably (b). We demonstrate that using a regularizing distribution over functions to encode prior knowledge of drug-like chemical space into the modeling process improves the predictive performance and calibration of neural networks in this extrapolative regime (c).

A key feature of practical early-stage drug discovery research is the application of predictive models to novel compounds that are *structurally or functionally dissimilar* to molecules that have already been explored (see Figure 1). In such an *extrapolative regime*, the practical utility of machine learning systems hinges on their ability to (a) robustly generalize to unexplored areas of chemical space and (b) reliably indicate when they fail to do so by generating well-calibrated predictive uncertainty estimates. However, standard deep learning algorithms often perform poorly under covariate shift, generating both incorrect and highly miscalibrated predictions (Ovadia et al., 2019; Koh et al., 2021). This is particularly problematic in the context of early-stage drug discovery, where experimental labels are expensive to acquire and therefore only available for a small and often highly biased subset of compounds.

[1]Department of Statistics, University of Oxford, United Kingdom [2]Pharma Research & Early Development, Roche, Switzerland. Correspondence to: Leo Klarner <leo.klarner@stats.ox.ac.uk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

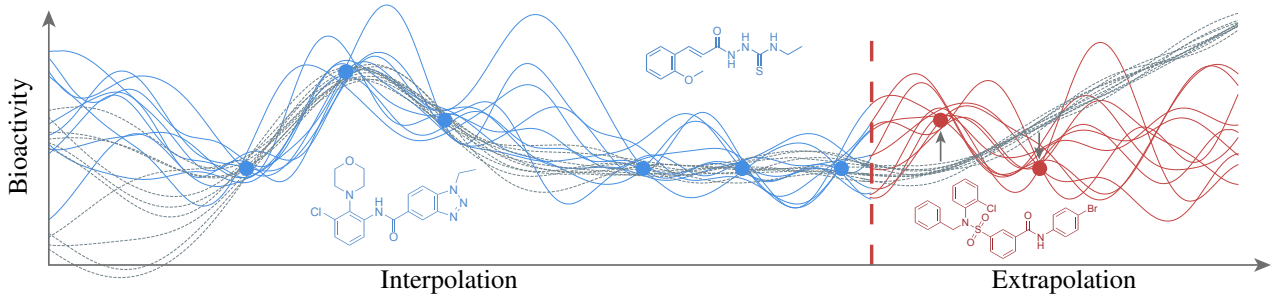


Figure 2. When trained on a small and highly biased subset of chemical space, standard neural networks (gray) rarely generalize well in the extrapolative regime. Our approach enables the construction of a problem-informed regularizing prior distribution over functions to place soft constraints on a neural network's hypothesis space, enabling better generalization and uncertainty quantification under covariate shift. In-distribution training points are shown in (blue) and out-of-distribution test points are shown in (red).

To improve the predictive performance of deep learning algorithms in such resource-constrained, low-data settings, we may wish to use relevant prior knowledge about the problem domain to specify inductive biases that make some predictive functions more likely than others. Common approaches to imbuing neural networks with useful inductive biases include (a) pre-training them on larger, potentially unlabeled datasets (Finn et al., 2017; Tan et al., 2018; Bommasani et al., 2021) and (b) adjusting their architectures to mirror appropriate invariances of their input domain (Bronstein et al., 2017; Satorras et al., 2021). However, these approaches are only an indirect—and often insufficiently precise—way of translating explicit modeling preferences into constraints over a neural network's hypothesis space.

In this paper, we present an alternative approach. To encode domain-informed prior knowledge of the data-generating process into neural network training, we specify a prior distribution over the space of **Q**uantitative **S**tructure-**A**ctivity mappings evaluated at a carefully selected set of context points, and perform **V**ariational **I**nference in the resulting probabilistic model (see Figure 2). We will refer to this method as **Q-SAVI**.

To demonstrate the practical utility of this approach, we construct a robust evaluation setup based on a carefully pre-processed bioactivity dataset. We then apply several different techniques to induce strong covariate and label shifts, resulting in challenging and practically meaningful train-test splits. Finally, we use Q-SAVI to specify explicit and problem-informed prior knowledge of drug-like chemical space and show that this substantially improves the predictive accuracy and calibration of deep learning algorithms in an out-of-distribution setting, outperforming a range of strong self-supervised pre-training, domain adaptation, and ensembling techniques.

Code and datasets are provided at:
https://github.com/leojklarner/Q-SAVI.

2. Predicting Properties to Discover Drugs

The overarching objective of small molecule drug discovery is to identify compounds that modulate a biological target of interest and elicit a therapeutically beneficial response. Unfortunately, the process of discovering a promising candidate to take into clinical trials is difficult and often unsuccessful, as the search space of viable drug-like molecules $\mathcal{X} = \{m_1, m_2, ...\}$ is vast, with estimates of $|\mathcal{X}|$ ranging from 10^{20} to 10^{60} (Bohacek et al., 1996; Ertl, 2003; Polishchuk et al., 2013). This is compounded by the inherent experimental limitations of medicinal chemistry, meaning that labels can only be acquired for a vanishingly small subset of compounds $\mathcal{X}' \subset \mathcal{X}$, with $|\mathcal{X}'| \ll |\mathcal{X}|$. Naturally, this has generated substantial interest in training supervised machine learning algorithms on available data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) | \mathbf{x}_i \in \mathcal{X}', \mathbf{y}_i \in \mathcal{Y}\}_{i=1}^{N}$ to predict the properties of compounds in $\mathcal{X} \setminus \mathcal{X}'$.

As the purpose of such models is to accelerate the discovery of novel and more effective therapeutics, predictions are usually desired most on compounds that are meaningfully dissimilar to molecules in $\mathcal{X}'$. Were $\mathcal{X}'$ sampled uniformly from $\mathcal{X}$, that is, $\mathcal{X}' \sim \mathcal{U}(\mathcal{X})$, these predictions would be made in an interpolative regime, in which standard regularization techniques such as weight decay, dropout (Srivastava et al., 2014), and batch normalization (Ioffe and Szegedy, 2015) constitute effective approaches to minimizing the expected loss on new samples from $\mathcal{U}(\mathcal{X})$.

In practice, however, the composition of $\mathcal{X}'$ is largely determined by empirical considerations such as compound availability and the preferences and intuitions of medicinal chemists, resulting in a highly biased subsample $\mathcal{X}' \sim \tilde{p}_{\mathcal{X}}$. This means that, in order to reliably predict the properties of novel and scientifically interesting compounds, it is essential for machine learning algorithms to perform well in an extrapolative regime. As this requirement is distinct from in-distribution generalization, standard approaches to regularization are unlikely to be effective.

Instead, we propose an alternative regularization scheme—Q-SAVI—that builds on the fact that we are able to approximately sample from $\mathcal{U}(\mathcal{X})$ through large chemical databases such as ZINC (Irwin et al., 2020) or GDB (Polishchuk et al., 2013) to specify arbitrary modeling preferences on $\mathcal{X} \setminus \mathcal{X}'$. Specifically, we construct a probabilistic model of neural network *functions* and define a tractable prior distribution over parametric function mappings evaluated at points in $\mathcal{U}(\mathcal{X})$. We then extend this probabilistic model to include a label-space prior over $\mathcal{X}$, which encodes contextualized information on $\tilde{p}_{\mathcal{X}}$ and $\mathcal{Y}$, and demonstrate empirically that variational inference in this probabilistic model results in neural networks that make accurate predictions in regions of chemical space that they can reliably extrapolate to while generating well-calibrated predictive uncertainty estimates that indicate when correct predictions are unlikely.

3. Related Work

Starting with foundational attempts to link the electronic properties of different substituents to the reactivity (Hammett, 1937) and bioactivity (Hansch et al., 1962) of benzoic acid derivatives, the problem of predicting the properties of a molecule from its structure has long received considerable attention (Cherkasov et al., 2014). While simpler algorithms such as support vector machines (Cortes and Vapnik, 1995) and random forests (Breiman, 2001) remain a popular choice for such quantitative structure-activity relationship (QSAR) models, recent years have seen substantial interest in applying modern deep learning algorithms to this task (Ma et al., 2015; Gawehn et al., 2016; Zhang et al., 2017), including important attempts to improve their performance in low-data and out-of-distribution regimes.

Self-supervised pre-training techniques. To this end, Hu et al. (2019) and Rong et al. (2020) have introduced a range of self-supervised objectives to pre-train graph neural networks and graph transformers on a set of unlabeled molecular structures to generate initializations that can be efficiently fine-tuned on downstream tasks. However, the out-of-distribution generalization of their approaches was only assessed on scaffold splits—a setting that may underestimate of covariate and label shift encountered in many practical applications (Wallach and Heifets, 2018).

Domain adaptation techniques. Building on the fact that biases in the data collection process are often known at training time, domain adaptation and generalization techniques (Ganin et al., 2016; Sun and Saenko, 2016; Sagawa et al., 2019; Arjovsky et al., 2019) aim to improve the performance of deep learning algorithms in out-of-distribution settings by leveraging pre-specified domain indicators. However, these methods—originally developed for image data—have been found to provide limited benefits in the context of molecular property prediction (Ji et al., 2022).

Bayesian inference-based techniques. Bayesian Neural Networks (BNNs; Neal (1996)) provide a principled probabilistic framework for posterior inference over neural networks parameters and have long been explored in the context of drug discovery (Burden and Winkler, 1999; Burden et al., 2000). Even though they conceptually guarantee robustness in low-data regimes, their empirical performance often falls short of ensembling techniques or even standard stochastic gradient descent (Ovadia et al., 2019; Foong et al., 2019; Farquhar et al., 2020), including in the context of molecular property prediction (Ryu et al., 2019; Zhang et al., 2019).

While these approaches may improve the robustness of deep learning algorithms in some settings, they are limited in the extent to which they can encode problem-specific modeling preferences that, for example, encourage high predictive uncertainty away from the training data or specify prior knowledge of synthetic accessibility and patentability. For instance, the standard parameter-space formulation of BNNs precludes the specification of semantically meaningful prior information due to the highly non-linear and complex relationship between a neural network's parameters and the functions they encode.

Building on recent work that aims to address the shortcomings of BNNs (e.g., in specifying meaningful prior distributions and providing reliable uncertainty quantification) via function-space variational inference (Sun et al., 2019; Rudner et al., 2021; 2022b), we reframe QSAR modeling as inferring a posterior distribution over functions. We do so by specifying a prior distribution over function mappings along with a prior distribution over function evaluation points and performing variational inference in this probabilistic model, which allows us to explicitly encode prior beliefs about the distribution over functions as well as about the structure of the input space into neural network training.

4. Quantitative Structure-Activity VI

Consider the supervised learning setup outlined in Section 2, with the objective of training a machine learning model on the experimental labels of N independent and identically distributed samples drawn from a biased subset of chemical space, resulting in the data realizations $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N = (\mathbf{x}_\mathcal{D}, \mathbf{y}_\mathcal{D})$ of inputs $\mathbf{x}_i \in \mathcal{X}' \subset \mathcal{X}$ and labels $\mathbf{y}_i \in \mathcal{Y}$, where $\mathcal{Y} \subseteq \mathbb{R}^K$ for regression and $\mathcal{Y} \subseteq \{0, 1\}^K$ for classification tasks with K labels.

Let $p_{\mathbf{Y}|f(\mathbf{X};\mathbf{\Theta})}$ be an observation model of the labels $\mathbf{Y}$ given a latent stochastic function $f(\mathbf{X}; \mathbf{\Theta}) : \mathcal{X} \times \mathbb{R}^P \to \mathcal{Y}$ induced by a set of stochastic parameters $\mathbf{\Theta} \in \mathbb{R}^P$ and evaluated at a set of input points $\mathbf{X} \in \mathcal{X}$. Additionally, let $p_{f(\mathbf{X};\mathbf{\Theta})}$ be a prior distribution over such latent stochastic functions. $p_{\mathbf{Y}|f(\mathbf{X};\mathbf{\Theta})}(\mathbf{y}_\mathcal{D} \mid f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}))$ is then the likelihood of observing labels $\mathbf{y}_\mathcal{D}$ under $f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta})$ — a realization of the stochastic function evaluated at inputs $\mathbf{x}_\mathcal{D}$.

Instead of formulating the posterior inference problem as finding the posterior distribution over stochastic parameters $\boldsymbol{\Theta}$, we follow Rudner et al. (2021) and reframe variational inference in stochastic neural networks as finding a posterior distribution over the *latent stochastic functions* $f(\mathbf{x}_\mathcal{D}; \boldsymbol{\Theta})$ at the training points $\mathbf{x}_\mathcal{D}$. In particular, while the parameter-space Bayesian inference problem is given by

$$p_{\boldsymbol{\Theta}|\mathcal{D}}(\boldsymbol{\Theta} \mid \mathcal{D}) = \frac{p_{\mathbf{Y}|\boldsymbol{\Theta},\mathbf{X}}(\mathbf{y}_\mathcal{D} \mid \boldsymbol{\theta}, \mathbf{x}_\mathcal{D})\, p_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}_\mathcal{D} \mid \mathbf{x}_\mathcal{D})}, \tag{1}$$

the inference problem over $f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta})$ is expressed by

$$\begin{aligned} &p_{f(\mathbf{X};\boldsymbol{\Theta})|\mathcal{D}}(f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}) \mid \mathcal{D}) \\ &= \frac{p_{\mathbf{Y}|f(\mathbf{X};\boldsymbol{\Theta})}(\mathbf{y}_\mathcal{D} \mid f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}))\, p_{f(\mathbf{X};\boldsymbol{\Theta})}(f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}))}{p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}_\mathcal{D} \mid \mathbf{x}_\mathcal{D})}, \end{aligned} \tag{2}$$

which includes an explicit dependence on the function-space prior evaluated at the training points $p_{f(\mathbf{X};\boldsymbol{\Theta})}(f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}))$, which allows us to specify arbitrary preferences for suitable parametric function mappings f.

To show how the inference problem in Equation (1) and Equation (2) are related, note that for a prior distribution over parameters $p_{\boldsymbol{\Theta}}$, the prior distribution $p_{f(\mathbf{X};\boldsymbol{\theta})}$ over $f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta})$ induced by $p_{\boldsymbol{\Theta}}$ is given by

$$\begin{aligned} &p_{f(\mathbf{X};\boldsymbol{\Theta})}(f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta})) \\ &= \int_{\mathbb{R}^P} p_{\boldsymbol{\Theta}}(\boldsymbol{\theta}')\, \delta(f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}) - f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}'))\, \mathrm{d}\boldsymbol{\theta}', \end{aligned} \tag{3}$$

and, similarly, the posterior distribution $p_{f(\mathbf{X};\boldsymbol{\Theta})|\mathcal{D}}$ over $f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta})$ induced by the posterior distribution over parameters $p_{\boldsymbol{\Theta}|\mathcal{D}}$ is given by

$$\begin{aligned} &p_{f(\mathbf{X};\boldsymbol{\Theta})|\mathcal{D}}(f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}) \mid \mathcal{D}) \\ &= \int_{\mathbb{R}^P} p_{\boldsymbol{\Theta}|\mathcal{D}}(\boldsymbol{\theta}' \mid \mathcal{D})\, \delta(f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}) - f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}'))\, \mathrm{d}\boldsymbol{\theta}', \end{aligned} \tag{4}$$

where $\delta(\cdot)$ is the Dirac delta function (Wolpert, 1993; Rudner et al., 2022a). In the remainder of this section, subscripts will be dropped from probability density functions when the dependence is clear from context.

We will now extend this function-space formulation of Bayesian inference to define a probabilistic model that is able to integrate prior knowledge of the full input space $\mathcal{X}$ beyond a biased subset of training points $\mathbf{x}_\mathcal{D} \subseteq \mathcal{X}'$. Specifically, we extend the probabilistic model above to the random variables $f(\{\mathbf{X}, \mathbf{X}_\mathcal{C}\}; \boldsymbol{\Theta})$ and $\mathbf{X}_\mathcal{C}$, where $\mathbf{X}_\mathcal{C} \subseteq \mathcal{X} \setminus \mathcal{X}'$ is a set of *context points*, yielding the posterior distribution

$$\begin{aligned} &p(f(\{\mathbf{x}_\mathcal{D}, \mathbf{x}_\mathcal{C}\}; \boldsymbol{\theta}), \mathbf{x}_\mathcal{C} \mid \mathcal{D}) \\ &= \frac{p(\mathbf{y}_\mathcal{D} \mid f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}))\, p(f(\{\mathbf{x}_\mathcal{D}, \mathbf{x}_\mathcal{C}\}; \boldsymbol{\theta}) \mid \mathbf{x}_\mathcal{D}, \mathbf{x}_\mathcal{C})\, p(\mathbf{x}_\mathcal{C})}{p(\mathbf{y}_\mathcal{D} \mid \mathbf{x}_\mathcal{D})} \end{aligned} \tag{5}$$

where, for a stochastic function evaluation $f(\{\mathbf{x}_\mathcal{D}, \mathbf{x}_\mathcal{C}\}; \boldsymbol{\theta})$ defined by a valid stochastic process over $f(\cdot\,; \boldsymbol{\Theta})$, the likelihood $p(\mathbf{y}_\mathcal{D} \mid f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}))$ and marginal likelihood $p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}_\mathcal{D} \mid \mathbf{x}_\mathcal{D})$ are independent of $f(\mathbf{X}_\mathcal{C}; \boldsymbol{\Theta})$ and $\mathbf{X}_\mathcal{C}$ by marginal consistency.

For non-linear function mappings $f : \mathcal{X} \times \mathbb{R}^P \to \mathcal{Y}$ parameterized by high-dimensional $\boldsymbol{\Theta} \in \mathbb{R}^P$, the inference problem specified in Equation (5) is analytically intractable. Instead, we may frame it variationally as

$$\min_{q_{\boldsymbol{\Theta}} \in \mathcal{Q}_{q_{\boldsymbol{\Theta}}}} \mathbb{D}_{\mathrm{KL}}(q_{f(\{\mathbf{X},\mathbf{X}_\mathcal{C}\};\boldsymbol{\Theta}),\mathbf{X}_\mathcal{C}} \,\|\, p_{f(\{\mathbf{X},\mathbf{X}_\mathcal{C}\};\boldsymbol{\Theta}),\mathbf{X}_\mathcal{C}|\mathcal{D}}) \tag{6}$$

for some variational distribution over parameters $q_{\boldsymbol{\Theta}}$ in a variational family $\mathcal{Q}_{q_{\boldsymbol{\Theta}}}$ (Wainwright and Jordan, 2008). Letting the variational distribution factorize as

$$q_{f(\{\mathbf{X},\mathbf{X}_\mathcal{C}\};\boldsymbol{\Theta}),\mathbf{X}_\mathcal{C}} \doteq q_{f(\{\mathbf{X},\mathbf{X}_\mathcal{C}\};\boldsymbol{\Theta})} q_{\mathbf{X}_\mathcal{C}} = q_{f(\{\mathbf{X},\mathbf{X}_\mathcal{C}\};\boldsymbol{\Theta})} p_{\mathbf{X}_\mathcal{C}}, \tag{7}$$

and assuming that $q_{\mathbf{X}_\mathcal{C}} = p_{\mathbf{X}_\mathcal{C}}$, we can reformulate the inference problem above in a simplified form as

$$\min_{q_{\boldsymbol{\Theta}} \in \mathcal{Q}_{q_{\boldsymbol{\Theta}}}} \mathbb{E}_{p_{\mathbf{X}_\mathcal{C}}} \left[\mathbb{D}_{\mathrm{KL}}(q_{f(\mathbf{X};\boldsymbol{\Theta})|\mathbf{X}_\mathcal{C}} \,\|\, p_{f(\mathbf{X};\boldsymbol{\Theta})|\mathbf{X}_\mathcal{C},\mathcal{D}})\right], \tag{8}$$

which can in turn be equivalently expressed as

$$\begin{aligned} \max_{q_{\boldsymbol{\Theta}} \in \mathcal{Q}_{q_{\boldsymbol{\Theta}}}} \Big\{ &\mathbb{E}_{q_{\boldsymbol{\Theta}} p_{\mathbf{X}_\mathcal{C}}} \left[\log p(\mathbf{y}_\mathcal{D} \mid f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}))\right] \\ &- \mathbb{D}_{\mathrm{KL}}(q_{f(\{\mathbf{X},\mathbf{X}_\mathcal{C}\};\boldsymbol{\Theta})} \,\|\, p_{f(\{\mathbf{X},\mathbf{X}_\mathcal{C}\};\boldsymbol{\Theta})}) \Big\}. \end{aligned} \tag{9}$$

If $p_{\boldsymbol{\Theta}}$ is chosen to be an isotropic Gaussian distribution and $\mathcal{Q}_{q_{\boldsymbol{\Theta}}}$ is the family of mean-field Gaussian distributions, the prior and variational distributions in Equation (9) can be approximated using the local linearization scheme introduced in Rudner et al. (2022a). These approximations result in a factorized variational objective, making stochastic variational inference and stochastic gradient-based optimization techniques applicable (Hinton and van Camp, 1993; Graves, 2011; Hoffman et al., 2013; Blundell et al., 2015).

By enabling the specification of the context point distribution $p_{\mathbf{X}_\mathcal{C}}$ and the prior distribution over functions $p_{f(\{\mathbf{X},\mathbf{X}_\mathcal{C}\};\boldsymbol{\Theta})}$, this framework enables us to explicitly encode arbitrary modeling preferences as distributions that place high probability mass on relevant regions of the input domain and specify prior knowledge of preferred parametric function mappings on unlabelled data points.

After optimizing the variational objective with respect to the parameters of $q_{\boldsymbol{\Theta}}$, we obtain samples from approximate posterior predictive distribution through

$$q(\mathbf{y}_* \mid \mathbf{x}_*) \approx \frac{1}{M_*} \sum_{j=1}^{M_*} p(\mathbf{y}_* \mid f(\mathbf{x}_*; \boldsymbol{\theta}^{(j)})), \tag{10}$$

with $\boldsymbol{\theta}^{(j)} \sim q_{\boldsymbol{\Theta}}$ and M_* being the number of Monte Carlo samples used to estimate the predictive distribution.

5. Empirical Evaluation

To demonstrate the practical utility of Q-SAVI, we establish a robust evaluation setup: In Section 5.1, we argue that many commonly-used bioactivity datasets may not be able to meaningfully assess the extrapolative power of supervised machine learning algorithms and present a carefully cleaned and pre-processed alternative dataset and in Section 5.2, we define an appropriate set of statistics to quantify covariate and label shifts in chemical space and use them to investigate the extent to which different splitting techniques induce data shift. In Section 5.3, we then use this experimental setup to demonstrate that employing Q-SAVI to incorporate domain-informed prior knowledge into the modeling process leads to significant gains in predictive accuracy and calibration, outperforming a range of strong self-supervised pre-training, domain adaptation, and ensembling techniques. Finally, in Section 5.4, we show that these strong empirical results extend to real-world production settings by evaluating our method on the time-split data presented in Ma et al. (2015).

5.1. Curating an Appropriate Dataset

A fundamental obstacle to training and evaluating QSAR models in the public domain is the scarcity of sufficiently large datasets with high-quality labels (Schneider et al., 2020). Even though collections of publicly available bioactivity data exist (Wu et al., 2018; Huang et al., 2021), they are often sourced directly from repositories of high-throughput screening (HTS) data such as PUBCHEM (Kim et al., 2019), CHEMBL (Mendez et al., 2019) or TOXCAST (Richard et al., 2016) without significant filtering or pre-processing. While this approach maximizes the number of available data points, it may reduce the discriminative power of model performance comparisons. For instance, a well-known problem of confirmatory dose-response screens—which make up the bulk of measurements in the above repositories—is that they usually contain a large number of reproducible false positive readouts (in many cases up to 95% of hits (Thorne et al., 2010)) caused by molecular substructures that interfere with an assay's readout system (Baell and Holloway, 2010; Dahlin et al., 2015). Using such data without further processing runs the risk of simply testing for the ability of algorithms to memorize these substructures instead of assessing meaningful extrapolative performance (Klarner et al., 2022).

To curate a dataset of sufficient quality to enable an informative comparison of predictive models, we used the measurement meta-data of bioactivity and toxicity screens to prioritize certain data points for further inspection. After surveying the publications associated with the most promising datasets, we selected a high-quality screening campaign for inhibitors of the development of liver-stage malaria parasites for further processing (Antonova-Koch et al., 2018).

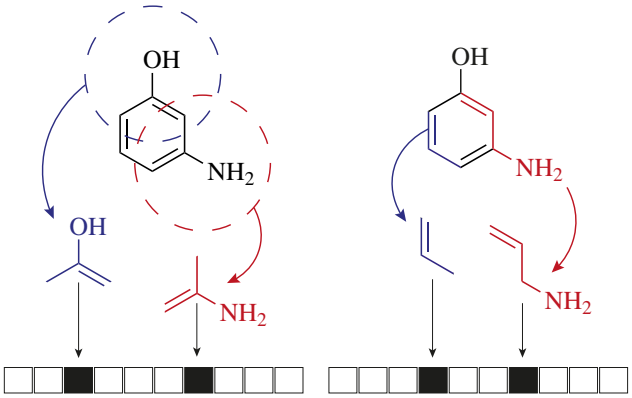


Figure 3. Schematic representation of extended-connectivity fingerprints (**left**) and RDKIT fingerprints (**right**). Both methods operate on the topological graph of a molecule and enumerate all labeled subgraphs up to a certain diameter, differing only in the space of substructures they consider. While RDKITFPS enumerate subgraphs of any shape, ECFPS are restricted to radial substructures. Once extracted, this set of subgraphs is hashed into a fixed-length bit vector.

Specifically, we retrieved and reprocessed the raw measurement data to remove likely false positives and other experimental artifacts, yielding a binary classification dataset with 7,301 inactive and 849 active molecules, each measured in biological duplicate and confirmed as a true positive or negative through a set of quality-assuring counter-screens (see Appendix A for full details).

5.2. Inducing and Quantifying Data Shift

Featurization. Commonly used techniques to numerically represent the structural properties of a molecule include strings, graphs, and topological fingerprints. For the following experiments, each molecule was featurized as both an extended-connectivity fingerprint (ECFP; Rogers and Hahn (2010)) and an RDKIT fingerprint (RDKITFPS), using the respective implementations in the open-source cheminformatics package RDKIT (Landrum et al., 2022). An illustration of this process is presented in Figure 3.

Statistics for covariate and label shift. To evaluate the extent to which different train-test splits induce covariate and label shift, we identified a set of suitable two-sample test statistics and used it to quantify the dissimilarity of the marginal covariate and label distributions of the respective training and test sets $\mathcal{D}_{\text{tr}} = (\mathbf{X}_{\text{tr}}, \mathbf{y}_{\text{tr}})$ and $\mathcal{D}_{\text{te}} = (\mathbf{X}_{\text{te}}, \mathbf{y}_{\text{te}})$.

Since $\mathbf{y}_{\text{tr}}$ and $\mathbf{y}_{\text{te}}$ consist of binary indicators of antimalarial activity, well-established categorical statistics such as Fisher's exact test (Upton, 1992) are applicable. In the following, its negative logarithmic p-value is used as a scalar indicator of label shift.

Defining a corresponding statistic to quantify covariate shift between two sets of molecules is more challenging, as they constitute disjoint sets of discrete objects. For this purpose, we used the maximum mean discrepancy (MMD) metric (Gretton et al., 2012) to quantify the difference between two samples of molecules as the distance between the embeddings of their expectations in a reproducing kernel Hilbert space (RKHS) defined by some mapping $\phi : \mathcal{X} \to \mathcal{H}$ and an associated kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$. An empirical estimator of this statistic is obtained by

$$\begin{aligned}
&\mathrm{MMD}^2(\mathbf{X}_{\text{tr}}, \mathbf{X}_{\text{te}}) \\
&= \left\| \mathbb{E}_{\mathbf{x} \in \mathbf{X}_{\text{tr}}} [\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \in \mathbf{X}_{\text{te}}} [\phi(\mathbf{x})] \right\|_{\mathcal{H}}^2 \\
&= \mathbb{E}_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_{\text{tr}}} [k(\mathbf{x}_i, \mathbf{x}_j)] + \mathbb{E}_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_{\text{te}}} [k(\mathbf{x}_i, \mathbf{x}_j)] \\
&\quad - 2\mathbb{E}_{\mathbf{x}_i \in \mathbf{X}_{\text{tr}}, \mathbf{x}_j \in \mathbf{X}_{\text{te}}} [k(\mathbf{x}_i, \mathbf{x}_j)] ,
\end{aligned}$$

using the Jaccard/Tanimoto similarity coefficient

$$k_{\text{jac}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cap \mathbf{x}_j}{\mathbf{x}_i \cup \mathbf{x}_j} = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - \langle \mathbf{x}_i, \mathbf{x}_j \rangle}$$

as an appropriate similarity metric, both due to its established use in the cheminformatics community (Bajusz et al., 2015) and the favorable properties of the RKHS that it induces. The MMD statistic is only valid if the mean embedding $\mathbb{E}_{\mathbf{x} \in \mathbf{X}} [\phi(\mathbf{x})]$ is injective, which is the case for strictly positive definite kernels operating in discrete domains (Borgwardt et al., 2006), such as k_{jac} (Bouchard et al., 2013).

Random and scaffold splits. Equipped with the appropriate statistical tools to quantify distributional similarities, we investigated the extent to which different train-test splits are able to emulate practically relevant covariate and label shifts, beginning with the two most popular approaches of splitting data either randomly or by scaffold. While *random splits* are commonly used in many domains, they are known to produce unrealistically optimistic performance estimates in the context of molecular property prediction. This is a consequence of the biased composition of many experimental datasets, which often contain structurally similar compounds from so-called chemical series. As these often exhibit very similar properties, distributing them evenly across data splits leads to a de-facto overlap between training and test sets that incentivizes overfitting and memorization (Wallach and Heifets, 2018). *Scaffold splits* attempt to mitigate this shortcoming by mapping each molecule to an overarching compound class—usually its Bemis-Murcko scaffold (Bemis and Murcko, 1996; 1999)—and splitting the data so that all molecules of a given scaffold are assigned to the same partition. However, this approach often results in a similar pathology, as even molecules with nominally different scaffolds can exhibit a high degree of structural and functional similarity, as illustrated in Figure 5.

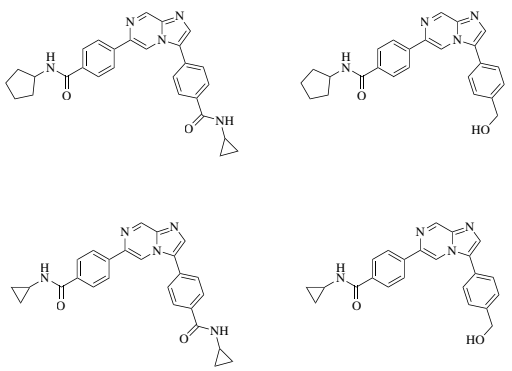


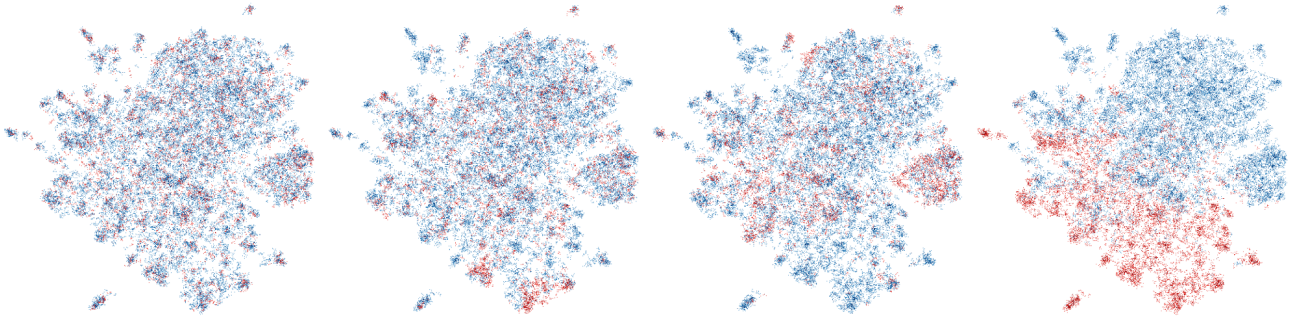
Figure 5. Even molecules with nominally different Bemis-Murcko scaffolds can exhibit a high degree of structural and functional similarity. Depicted are four structurally similar molecules from our antimalarial dataset that are assigned to four different scaffolds.

Molecular weight and spectral splits. To facilitate the comparison of models in an extrapolative regime, we explored two alternative approaches. A straightforward *molecular weight split* was used to induce data shift by assigning molecules into training and test sets based on a molecular weight cut-off, relying on the correlation of molecular size and binding strength to also induce strong label shift (Hopkins et al., 2014). More rigorously, we developed a clustering-based *spectral split* to generate data splits that are guaranteed to exhibit maximal covariate shift under the MMD statistic. By interpreting the Jaccard kernel Gram matrix $\mathbf{W}^{\text{jac}} \in [0, 1]^{|\mathbf{X}_{\mathcal{D}}| \times |\mathbf{X}_{\mathcal{D}}|}$ of a given set of molecules $\mathbf{X}_{\mathcal{D}}$ as the weighted adjacency matrix of a fully-connected similarity graph S, well-established spectral clustering algorithms (Von Luxburg, 2007) can be employed to identify an optimal partitioning of S that maximizes the similarity within and minimizes the similarity between partitions.

We present a comparison of the resulting covariate and label shift statistics in Table 1, which shows that molecular weight and spectral clustering-based splits generate a significantly more extrapolative evaluation setup than random and scaffold splits. This is substantiated by the qualitative visualization presented in Figure 4.

Table 1. A summary of the covariate and label shifts induced by the different train-test splits presented in Section 5.2, using rdkit and extended-connectivity (EC) fingerprints. Covariate shift is quantified as the Jaccard kernel-based MMD statistic, while label shift is quantified as the negative $\log p$-value of Fisher's exact test.

Split	Covariate Shift *(rdkit, EC)*	Label Shift *(rdkit, EC)*
Random	0.00, 0.00	0.00
Scaffold	0.08, 0.07	4.23
Weight	0.14, 0.10	**61.96**
Spectral	**0.34**, **0.25**	17.49, 50.05



Random split. Scaffold split. Molecular weight split. Spectral split.

Figure 4. A visual comparison of the covariate shift induced by different train-test splits using rdkit fingerprints, colored in **blue** and **red** respectively. While random and scaffold splits lead to relatively similar training and test sets, molecular weight and spectral splits induce significantly stronger covariate shifts. The plots were generated using UMAP dimensionality reduction (McInnes et al., 2018).

5.3. Model Construction, Baselines & Results

Model construction. Using the increasingly data-shifted splits constructed in Sections 5.1 and 5.2, we assessed Q-SAVI with respect to its ability to improve the predictive accuracy and calibration of deep learning algorithms under covariate and label shifts. By leveraging the option to specify both an arbitrary context point distribution $p_{\mathbf{X}_c}$ and a prior distribution over parametric function mappings $p_{f(\{\mathbf{X},\mathbf{X}_c\};\boldsymbol{\Theta})}$, we used Q-SAVI to encode relevant information about both the input domain and the label space of the problem setup into the model. Specifically, we precomputed the featurizations of a uniform subsample of the ZINC database of commercially available compounds (Irwin et al., 2020) and used them to construct a uniform context point distribution $p_{\mathbf{X}_c} = \mathcal{U}(\bar{\mathcal{X}})$ over a set of 2×10^6 synthetically accessible drug-like molecules $\bar{\mathcal{X}}$. Additionally, we used the prior distribution $p_{f(\{\mathbf{X},\mathbf{X}_c\};\boldsymbol{\Theta})}$ over parametric mappings to encode an informative function-space prior that encourages high predictive uncertainty in unexplored regions of chemical space, counteracting the likelihood term in Equation (9) to generate better predictive uncertainty estimates.

Baselines. We compared the performance of the resulting probabilistic model to a range of standard baselines and state-of-the-art pre-training and domain adaptation techniques. The simplest of these models is regularized **logistic regression**, which is expected to underperform in an extrapolative regime due to the linearity of its logit function. While **random forest classifiers** (Breiman, 2001) represent a more flexible baseline with strong in-distribution generalization guarantees, they generally exhibit coarser decision boundaries at the fringes of the training distribution that are unlikely to perform well on covariate-shifted inputs. Standard deep learning methods such as multi-layer perceptrons **(MLPs)** have an even higher representational capacity, yet also generally underperform under data shift, yielding both incorrect and highly overconfident predictions (Ovadia et al., 2019; Koh et al., 2021). **Deep ensembles** are an effective technique to improve the predictive performance of MLPs by averaging the predictive distributions of a set of independently trained neural networks (Lakshminarayanan et al., 2017). To investigate the extent to which existing self-supervised pre-training techniques and more expressive model architectures impact the performance of deep learning algorithms in this setting, we fine-tuned graph isomorphism networks (**GINs**; Xu et al. (2018)) provided by Hu et al. (2019) both from scratch and from initializations that were pre-trained on compounds from the ZINC database using *context prediction* and *attribute masking* objectives. Additionally, we fine-tuned the graph transformer (**GROVER**) proposed by Rong et al. (2020) from a pre-trained initialization that was optimized on molecules from the ZINC and ChEMBL databases using self-supervised contextual property and graph-level motif prediction techniques. Finally, we adapted a range of domain adaptation and generalization techniques, including invariant risk minimization (**IRM**; Arjovsky et al. (2019)), group-distributionally robust training (**GroupDRO**; Sagawa et al. (2019)), domain-adversarial networks (**DANN**; Ganin et al. (2016)), and deep correlation alignment (**DeepCoral**; Sun and Saenko (2016)) from Ji et al. (2022) who provided them with data split-specific domain indicators.

Training and evaluation. To facilitate a fair comparison, we carried out an extensive hyperparameter search for every model, data split, and featurization. After an initial division of the data into training and test sets, the same data-splitting technique was applied again to derive a representative validation set. The hyperparameter setting with the lowest negative log-likelihood on that validation set was then used to train ten independent models using different random seeds. Full implementation details and hyperparameter ranges are provided in Appendix B.

Table 2. An overview of the test set performance of each model for each data split and featurization technique, quantified by the AUC-ROC (↑) and the BRIER SCORE (↓). All entries indicate the mean and standard errors computed over 10 independent training runs with different random seeds. The best models within a margin of statistical significance are highlighted in bold.

	Model & Featurization	Spectral Split ECFP	Spectral Split rdkitFP	Weight Split ECFP	Weight Split rdkitFP	Scaffold Split ECFP	Scaffold Split rdkitFP	Random Split ECFP	Random Split rdkitFP
AUC-ROC (↑)	Logistic Regression	.583±.000	.551±.000	.626±.000	.632±.000	.684±.000	.698±.000	.704±.000	.687±.000
	Random Forest	.576±.009	.552±.006	.592±.006	.567±.004	.605±.004	.642±.003	.696±.002	.690±.002
	MLP	.574±.006	.571±.003	.614±.004	.577±.005	.625±.010	.631±.014	**.720±.002**	.692±.004
	Deep Ensemble	.589±.006	.571±.002	.644±.001	.594±.002	.679±.001	**.697±.003**	**.720±.001**	**.710±.003**
	GIN	.549±.009	.551±.007	.582±.007		.664±.005		.685±.004	
	GIN (attr masking)	.588±.004	.559±.010	.625±.004		**.700±.002**		.705±.002	
	GIN (context pred)	.541±.005	.566±.009	.621±.003		.674±.003		**.713±.003**	
	Grover	.574±.002	.544±.006	.623±.003		.689±.003		.701±.001	
	Q-SAVI	**.606±.003**	**.603±.006**	**.650±.002**	**.643±.003**	.657±.004	**.701±.002**	.708±.001	.681±.002
BRIER SCORE (↓)	Logistic Regression	.131±.000	.111±.000	.051±.000	.049±.000	.101±.000	.100±.000	.087±.000	.088±.000
	Random Forest	.133±.000	**.110±.000**	.055±.000	.058±.000	.104±.000	.102±.000	**.085±.000**	**.086±.000**
	MLP	.133±.001	.111±.000	.050±.000	.055±.002	.103±.000	.108±.003	.087±.000	.088±.000
	Deep Ensemble	.133±.001	**.110±.000**	.048±.000	.052±.001	.101±.000	.100±.000	.086±.000	.086±.000
	GIN	.132±.001	.112±.001	.050±.000		.103±.000		.090±.001	
	GIN (attr masking)	**.130±.000**	.114±.002	.049±.000		**.100±.000**		.087±.000	
	GIN (context pred)	.134±.000	.113±.001	.050±.000		.101±.000		.087±.000	
	Grover	.134±.001	.111±.001	.049±.000		.101±.000		.088±.000	
	Q-SAVI	**.130±.000**	.112±.003	**.047±.000**	**.048±.000**	.102±.000	**.099±.000**	.088±.000	.090±.000

Following model training and hyperparameter selection, the predictive accuracy and calibration of the estimated test-set label probabilities were characterized by the area under the ROC curve (AUC-ROC) and the BRIER SCORE, as these enable the direct comparison of models across test sets with different label distributions (see Table 2). Additionally, each algorithm's performance was characterized by the area under the precision-recall curve (AUC-PRC) and the adaptive calibration error (ACE; Nixon et al. (2019)), which closely mirror the AUC-ROC and BRIER SCORE (see Table 5).

Results. The predictive accuracy and calibration metrics presented in Tables 2 and 5 (see Appendix B.1) demonstrate that Q-SAVI achieves significant performance gains in an out-of-distribution setting. On the spectral and molecular weight splits—the evaluation settings with the strongest covariate and label shift—Q-SAVI outperformed all other algorithms by a substantial and statistically significant margin in terms of predictive accuracy. Similarly, its predictive uncertainty estimates were significantly better calibrated than all other algorithms on the molecular weight split and most other algorithms on the ECFP-based spectral split.

On the substantially less data-shifted scaffold and random splits, relatively simple machine learning algorithms (e.g., random forests and deep ensembles) as well as more sophisticated self-supervised pre-training-based approaches consistently achieved the best predictive performance.

In line with the empirical observations of Ji et al. (2022), IRM, GroupDRO, DANN, and DeepCoral—domain adaptation and generalization techniques originally developed for images—were found to perform worse than most other techniques across most splits and featurizations (see Table 5).

5.4. Merck Molecular Activity Challenge

As a complementary assessment of the practical utility of Q-SAVI, we evaluated the method on the Merck Molecular Activity Challenge (Ma et al., 2015). Consisting of 15 datasets from real-world production settings, it provides time-split training and test sets that represent the data shift encountered throughout a molecular optimization campaign (Sheridan, 2013). As the compound structures are only provided in the form of anonymized atom-pair descriptors in count and bit vector form, using a uniform subsample of a large chemical database as a context point distribution is not possible.

Table 3. Covariate and label shift of time-split data from the Merck Molecular Activity Challenge. Covariate shift is quantified as the (multi-)set Jaccard kernel-based MMD statistic, while label shift is quantified as the two-sample Kolmogorov–Smirnov test statistic.

Dataset	Label Shift	Covariate Shift Count Vector	Covariate Shift Bit Vector
HIVPROT	0.579	0.132	0.162
DPP4	0.375	0.112	0.125
NK1	0.419	0.071	0.062

Table 4. A summary of the test set performance of each model for each of the datasets from the Merck Molecular Activity Challenge, quantified by the mean squared error (↓). All entries indicate the mean and standard error computed over 10 independent training runs with different random seeds. The best models within a margin of statistical significance are highlighted in bold.

Model	HIVPROT		DPP4		NK1	
	count vector	bit vector	count vector	bit vector	count vector	bit vector
L_1-Regression	$1.137_{\pm.000}$	$0.714_{\pm.000}$	$1.611_{\pm.000}$	$1.130_{\pm.000}$	$0.482_{\pm.000}$	$0.442_{\pm.000}$
L_2-Regression	$0.999_{\pm.000}$	$0.723_{\pm.000}$	$1.495_{\pm.000}$	$1.143_{\pm.000}$	$0.498_{\pm.000}$	$0.436_{\pm.000}$
Random Forest	$0.815_{\pm.009}$	$0.834_{\pm.010}$	$1.473_{\pm.008}$	$1.461_{\pm.012}$	$0.458_{\pm.002}$	$0.438_{\pm.002}$
MLP	$0.768_{\pm.014}$	$2.118_{\pm.015}$	$1.393_{\pm.024}$	$1.094_{\pm.029}$	$0.443_{\pm.007}$	$0.399_{\pm.006}$
Q-SAVI	$\mathbf{0.682}_{\pm.019}$	$\mathbf{0.664}_{\pm.028}$	$\mathbf{1.332}_{\pm.017}$	$\mathbf{1.028}_{\pm.027}$	$\mathbf{0.436}_{\pm.007}$	$\mathbf{0.387}_{\pm.012}$

Instead, our evaluation focused on the three most covariate- and label-shifted datasets (see Table 3), repurposing the remaining data as an anonymized context point distribution. All methods were evaluated following the protocol outlined in Section 5, with full details presented in Appendix C. The performance metrics for our method and the baseline algorithms investigated in Ma et al. (2015) are presented in Table 4, demonstrating that Q-SAVI performs favorably across every setting and outperforms all other models on the strongly data-shifted HIVPROT, DPP4, and NK1 datasets by a substantial and statistically significant margin.

6. Summary and Conclusions

The objective of early-stage drug discovery is to identify lead compounds that exhibit sufficient evidence of modulating a given disease phenotype—as well as suitable safety profiles—to qualify them for further investigation in in-vivo studies. Computational techniques that reliably predict the properties of novel molecules in unexplored regions of chemical space have the potential to substantially accelerate this time- and resource-intensive process. Motivated by the practical importance of developing such methods, we derived Q-SAVI, a probabilistic model that allows encoding explicit, problem-informed prior knowledge about the prediction domain into neural network training.

To construct a robust experimental setup and facilitate a practically meaningful evaluation of the proposed method, we carefully pre-processed a high-quality bioactivity dataset and explored different domain-specific statistics to quantify distribution shifts in this setting. Using these statistics to highlight the limited extent to which commonly used random and scaffold splits are able to induce meaningful covariate and label shifts, we built on two alternative molecular weight- and spectral clustering-based approaches to construct challenging train-test splits. Leveraging this extrapolative evaluation setup, we demonstrated that using Q-SAVI to provide neural networks with relevant and contextualized information on drug-like chemical space significantly improves both the predictive accuracy and calibration of neural network models, outperforming a range of state-of-the-art self-supervised pre-training, ensembling, and domain adaptation techniques.

The main limitation of the proposed method compared to standard training regimes is its increased computational cost, due to the amortized cost of having to pre-process a suitable context point distribution and the direct cost of having to perform each forward pass over both a mini-batch and a sample of context points. However, by keeping the size of each context set sample to be roughly comparable to the size of each mini-batch, we found this increase in computational cost to be manageable—especially in comparison to the computational cost of pre-training and fine-tuning related self-supervised methods or deep ensembles.

Promising avenues for future work include an investigation into how using Q-SAVI to specify problem-informed modeling preferences may improve the performance of deep learning algorithms for drug discovery applications that heavily rely on out-of-distribution generalization. For instance, the approach could be used to construct an acquisition function for an active learning loop to propose structural modifications that optimize the therapeutic properties of an existing lead compound (Nicolaou et al., 2007; Gómez-Bombarelli et al., 2018), as Q-SAVI generates robust predictions and additionally enables researchers to explicitly specify desirable exit vectors. It may also accelerate the discovery of novel compound classes that exhibit similar pharmacological properties to already explored molecules (Böhm et al., 2004; Hu et al., 2017), enabling the optimization of certain pharmacokinetic properties or the circumvention of patent restrictions. More broadly, we hope that this work encourages further research into the utility of probabilistic inference and domain-informed prior distributions over functions for drug discovery and beyond.

Acknowledgments

We thank anonymous reviewers for useful feedback. LK is funded by a Clarendon Scholarship. TGJR is funded by a Qualcomm Innovation Fellowship. We gratefully acknowledge the Oxford Advanced Research Computing service for providing computing resources and infrastructure.

References

Antonova-Koch, Y., Meister, S., Abraham, M., Luth, M. R., Ottilie, S., Lukens, A. K., Sakata-Kato, T., Vanaerschot, M., Owen, E., Jado, J. C., et al. (2018). Open-source discovery of chemical leads for next-generation chemoprotective antimalarials. *Science*, 362(6419):eaat9446.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Baell, J. B. and Holloway, G. A. (2010). New substructure filters for removal of pan assay interference compounds (pains) from screening libraries and for their exclusion in bioassays. *Journal of Medicinal Chemistry*, 53(7):2719–2740.

Bajusz, D., Racz, A., and Heberger, K. (2015). Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1):1–13.

Bemis, G. W. and Murcko, M. A. (1996). The properties of known drugs. 1. molecular frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893.

Bemis, G. W. and Murcko, M. A. (1999). Properties of known drugs. 2. side chains. *Journal of Medicinal Chemistry*, 42(25):5095–5099.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural networks. volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France. PMLR.

Bohacek, R. S., McMartin, C., and Guida, W. C. (1996). The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal Research Reviews*, 16(1):3–50.

Böhm, H.-J., Flohr, A., and Stahl, M. (2004). Scaffold hopping. *Drug Discovery Today: Technologies*, 1(3):217–224.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., and Smola, A. J. (2006). Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57.

Bouchard, M., Jousselme, A.-L., and Doré, P.-E. (2013). A proof for the positive definiteness of the jaccard index matrix. *International Journal of Approximate Reasoning*, 54(5):615–626.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

Burden, F. R., Ford, M. G., Whitley, D. C., and Winkler, D. A. (2000). Use of automatic relevance determination in qsar studies using bayesian neural networks. *Journal of Chemical Information and Computer Sciences*, 40(6):1423–1430.

Burden, F. R. and Winkler, D. A. (1999). Robust qsar models using bayesian regularized neural networks. *Journal of Medicinal Chemistry*, 42(16):3183–3187.

Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., Dearden, J., Gramatica, P., Martin, Y. C., Todeschini, R., et al. (2014). Qsar modeling: where have you been? where are you going to? *Journal of Medicinal Chemistry*, 57(12):4977–5010.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

Dahlin, J. L., Nissink, J. W. M., Strasser, J. M., Francis, S., Higgins, L., Zhou, H., Zhang, Z., and Walters, M. A. (2015). Pains in the assay: chemical mechanisms of assay interference and promiscuous enzymatic inhibition observed during a sulfhydryl-scavenging hts. *Journal of Medicinal Chemistry*, 58(5):2091–2113.

Ertl, P. (2003). Cheminformatics analysis of organic substituents: identification of the most common substituents, calculation of substituent properties, and automatic identification of drug-like bioisosteric groups. *Journal of Chemical Information and Computer Sciences*, 43(2):374–380.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Farquhar, S., Osborne, M. A., and Gal, Y. (2020). Radial Bayesian neural networks: Beyond discrete support in large-scale Bayesian deep learning. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1352–1362. PMLR.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

Foong, A. Y. K., Li, Y., Hernández-Lobato, J. M., and Turner, R. E. (2019). 'in-between' uncertainty in Bayesian neural networks.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Gawehn, E., Hiss, J. A., and Schneider, G. (2016). Deep learning in drug discovery. *Molecular Informatics*, 35(1):3–14.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276.

Graves, A. (2011). Practical variational inference for neural networks. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 2348–2356, Red Hook, NY, USA. Curran Associates Inc.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Hammett, L. P. (1937). The effect of structure upon the reactions of organic compounds. benzene derivatives. *Journal of the American Chemical Society*, 59(1):96–103.

Hansch, C., Maloney, P. P., Fujita, T., and Muir, R. M. (1962). Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients. *Nature*, 194(4824):178–180.

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., et al. (2020). Array programming with numpy. *Nature*, 585(7825):357–362.

Hinton, G. E. and van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, page 5–13, New York, NY, USA. Association for Computing Machinery.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347.

Hopkins, A. L., Keserü, G. M., Leeson, P. D., Rees, D. C., and Reynolds, C. H. (2014). The role of ligand efficiency metrics in drug discovery. *Nature Reviews Drug Discovery*, 13(2):105–121.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. (2019). Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*.

Hu, Y., Stumpfe, D., and Bajorath, J. (2017). Recent advances in scaffold hopping: miniperspective. *Journal of Medicinal Chemistry*, 60(4):1238–1246.

Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. (2021). Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv preprint arXiv:2102.09548*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR.

Irwin, J. J., Tang, K. G., Young, J., Dandarchuluun, C., Wong, B. R., Khurelbaatar, M., Moroz, Y. S., Mayfield, J., and Sayle, R. A. (2020). Zinc20—a free ultralarge-scale chemical database for ligand discovery. *Journal of Chemical Information and Modeling*, 60(12):6065–6073.

Ji, Y., Zhang, L., Wu, J., Wu, B., Huang, L.-K., Xu, T., Rong, Y., Li, L., Ren, J., Xue, D., et al. (2022). Drugood: Out-of-distribution (ood) dataset curator and benchmark for ai-aided drug discovery–a focus on affinity prediction problems with noise annotations. *arXiv preprint arXiv:2201.09637*.

Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., et al. (2019). Pubchem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47(D1):D1102–D1109.

Klarner, L., Reutlinger, M., Schindler, T., Deane, C., and Morris, G. (2022). Bias in the benchmark: Systematic experimental errors in bioactivity databases confound multi-task and meta-learning algorithms. In *ICML 2022 2nd AI for Science Workshop*. PMLR.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. (2021). Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan,

S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6402–6413.

Landrum, G. et al. (2022). Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., and Svetnik, V. (2015). Deep neural nets as a method for quantitative structure–activity relationships. *Journal of Chemical Information and Modeling*, 55(2):263–274.

McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Mendez, D., Gaulton, A., Bento, A. P., Chambers, J., De Veij, M., Félix, E., Magariños, M. P., Mosquera, J. F., Mutowo, P., Nowotka, M., et al. (2019). Chembl: towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(D1):D930–D940.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*.

Neal, R. M. (1996). Bayesian Learning for Neural Networks.

Nicolaou, C. A., Brown, N., and Pattichis, C. S. (2007). Molecular optimization using computational multiobjective methods. *Current Opinion in Drug Discovery and Development*, 10(3):316.

Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., and Tran, D. (2019). Measuring calibration in deep learning. In *CVPR Workshops*, volume 2.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems 32*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Polishchuk, P. G., Madzhidov, T. I., and Varnek, A. (2013). Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of Computer-aided Molecular Design*, 27(8):675–679.

Richard, A. M., Judson, R. S., Houck, K. A., Grulke, C. M., Volarath, P., Thillainadarajah, I., Yang, C., Rathman, J., Martin, M. T., Wambaugh, J. F., et al. (2016). Toxcast chemical landscape: paving the road to 21st century toxicology. *Chemical Research in Toxicology*, 29(8):1225–1251.

Rogers, D. and Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754.

Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., and Huang, J. (2020). Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571.

Rudner, T. G., Chen, Z., Teh, Y. W., and Gal, Y. (2021). Tractable function-space variational inference in bayesian neural networks. In *Advances in Neural Information Processing Systems*.

Rudner, T. G. J., Chen, Z., Teh, Y. W., and Gal, Y. (2022a). Tractabe Function-Space Variational Inference in Bayesian Neural Networks.

Rudner, T. G. J., Smith, F. B., Feng, Q., Teh, Y. W., and Gal, Y. (2022b). Continual Learning via Sequential Function-Space Variational Inference. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR.

Ryu, S., Kwon, Y., and Kim, W. Y. (2019). A bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification. *Chemical Science*, 10(36):8438–8446.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Satorras, V. G., Hoogeboom, E., and Welling, M. (2021). E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, pages 9323–9332. PMLR.

Schneider, P., Walters, W. P., Plowright, A. T., Sieroka, N., Listgarten, J., Goodnow, R. A., Fisher, J., Jansen, J. M., Duca, J. S., Rush, T. S., et al. (2020). Rethinking drug design in the artificial intelligence era. *Nature Reviews Drug Discovery*, 19(5):353–364.

Sheridan, R. P. (2013). Time-split cross-validation as a method for estimating the goodness of prospective prediction. *Journal of Chemical Information and Modeling*, 53(4):783–790.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Sun, B. and Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 443–450. Springer.

Sun, S., Zhang, G., Shi, J., and Grosse, R. B. (2019). Functional variational Bayesian neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer.

Thorne, N., Auld, D. S., and Inglese, J. (2010). Apparent activity in high-throughput screening: origins of compound-dependent assay interference. *Current Opinion in Chemical Biology*, 14(3):315–324.

Upton, G. J. (1992). Fisher's exact test. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 155(3):395–402.

Van Rossum, G. and Drake Jr, F. L. (1995). *Python tutorial*, volume 620. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Wainwright, M. J. and Jordan, M. I. (2008). *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc., Hanover, MA, USA.

Wallach, I. and Heifets, A. (2018). Most ligand-based classification benchmarks reward memorization rather than generalization. *Journal of Chemical Information and Modeling*, 58(5):916–932.

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Wes McKinney (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Wolpert, D. H. (1993). Bayesian backpropagation over i-o functions rather than weights. In Cowan, J., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. (2018). Moleculenet: a benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Zhang, L., Tan, J., Han, D., and Zhu, H. (2017). From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug Discovery Today*, 22(11):1680–1685.

Zhang, Y. et al. (2019). Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. *Chemical Science*, 10(35):8154–8163.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Drug discovery under covariate shift with domain-informed prior distributions over functions
Publication Status	Published
Publication Details	Klarner, L., Rudner, T.G., Reutlinger, M., Schindler, T., Morris, G.M., Deane, C. and Teh, Y.W., 2023, July. Drug discovery under covariate shift with domain-informed prior distributions over functions. In *International Conference on Machine Learning* (pp. 17176-17197). PMLR.

Student Confirmation

Student Name	Leo Klarner		
Contribution	• Conceptualized key research ideas and methodological contributions in collaboration with TGR. • Identified relevant datasets and baselines that allowed for a meaningful empirical evaluation of the proposed method in collaboration with all other co-authors. • Carried out all experiments, including data pre-processing, model training, sample generation and the analysis and visualization of results. • Developed the framing and presentation of the method in collaboration with all other co-authors. Drafted the manuscript and improved it in collaboration with all other co-authors. Engaged with reviewers during the peer review process in collaboration with TGR.		
Signature		Date	Lausanne, 14. April 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Yee-Whye Teh		
Supervisor comments: I confirm		
Signature 	Date	23 April 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 7

Discussion and Conclusion

This thesis has explored several complementary approaches to addressing key challenges in machine learning for drug discovery. Through a series of methodological contributions spanning both predictive and generative modelling, we have demonstrated that incorporating domain-specific prior knowledge and constraints into deep learning algorithms can substantially enhance their reliability in data-scarce settings and allow them to more robustly and effectively generalise to novel regions of chemical and protein sequence space. This chapter synthesises key insights from our work and outlines promising directions for future research.

Summary

The methodological advances presented in this thesis aim to leverage readily available hard and soft constraints to bias the hypothesis space of deep neural networks towards solutions that reflect fundamental physicochemical principles rather than superficial correlations in their training data. This approach offers a promising path toward developing more reliable machine learning tools for early-stage drug discovery, where the scarcity of labelled experimental data often limits the applicability of purely data-driven methods.

The context-guided diffusion framework introduced in Chapter 3 addresses a fundamental challenge in conditional diffusion models, namely their tendency to overfit to the limited subset of chemical space represented in their training data. By introducing a regularisation scheme that promotes smooth gradients and calibrated uncertainty estimates in unexplored regions, this method enables the more reliable generation of novel compounds with desired functional properties. The effectiveness of this approach across multiple molecular design tasks—including small molecules, materials, and proteins—demonstrates that even relatively simple modifications to standard training procedures can yield substantial improvements. It also highlights the importance of uncertainty quantification in conditional generative models, particularly in settings where the conditioning attributes may lie in regions with limited training data support: by encouraging models to maintain appropriate levels of uncertainty in these regions, our approach mitigates overconfident extrapolation and produces more reliable guidance signals.

In addition, the constrained diffusion frameworks developed in Chapters 4 and 5 provide principled mathematical solutions to the challenge of incorporating hard geometric constraints into generative models. By adapting concepts from traditional reflected Brownian motion and log-barrier methods, these approaches ensure that generated samples satisfy physical constraints by construction, rather than relying on post-processing or guidance, and demonstrate that incorporating fundamental geometric principles into generative models can improve both the quality and physical validity of their samples. The Metropolis-based discretisation introduced in Chapter 5 highlights the value of adapting other well-established ideas from statistical physics to this setting. The proposed sampling scheme produces better samples and leads to an order-of-magnitude speed-up of training and inference, while maintaining the theoretical constraint satisfaction guarantees of the previous approaches.

Finally, the semi-supervised probabilistic framework presented in Chapter 6 tackles the challenge of discriminative modelling under covariate shift by leveraging unlabelled data and carefully constructed function-space priors. This approach encodes relevant domain knowledge into the learning process and leads to more robust predictions for novel compounds that differ meaningfully from the training data. The comprehensive evaluation on high-quality datasets demonstrates substantial improvements in both predictive accuracy and calibration compared to existing approaches, particularly in challenging extrapolative settings. This work underscores the viability of probabilistic approaches that can effectively quantify uncertainty in their predictions and leverage unlabelled data to improve generalisation.

Key Contributions and Insights

Several overarching themes emerge from the research presented in this thesis, offering insights that extend beyond the specific techniques and applications considered in Chapters 3-6. A consistent finding across our work is that incorporating prior knowledge of the modelled domain into the training and inference algorithms of neural networks can significantly improve generalisation in low-data regimes. Whether through geometric constraints, uncertainty-aware regularisation, or function-space priors, these inductive biases guide models toward solutions that are less likely to overfit to superficial statistical patterns in their training data. This approach represents a middle ground between the high variance of purely data-driven deep learning algorithms and the potential bias of traditional physics-based methods. Such a hybrid approach proves particularly valuable in scientific domains like drug discovery, where mechanistic understanding and empirical data provide important complementary insights.

The methods developed in this thesis span both predictive and generative modelling, highlighting the complementary nature of these paradigms in molecular design. While predictive models help evaluate and prioritise candidate compounds, generative models enable efficient exploration of vast chemical and protein sequence spaces. Our work demonstrates that similar principles—particularly uncertainty quantification and physical constraints—can improve both types of models. We believe that the integration of predictive and generative approaches, as exemplified by the conditional diffusion methods in Chapter 3, offers a powerful framework for targeted molecular design.

Relevant Recent Work

Perhaps the most relevant methodological developments since the completion of the research presented in this thesis pertain to the constrained generation problems in Chapters 4 and 5. While the proposed constrained diffusion frameworks provide mathematically rigorous solutions for incorporating relevant physical constraints into the modelling process, their reliance on modified stochastic differential equations introduces significant implementational complexity and computational overhead. The emergence of robust and performant flow matching models offers a simpler and conceptually appealing alternative. In particular, the ability to construct probability paths based on linear optimal transport maps allows the simulation-free handling of convex constraints: if an initial data point and a target noise point both lie within a convex set, the learned vector field would be guaranteed to produce sampling trajectories that satisfy these constraints by construction. While it remains an open question whether this approach is strictly superior to the constrained diffusion methods presented here, its relative conceptual and practical simplicity are compelling. If approaching these constrained generation problems again today, flow matching models would represent a compelling starting point.

Similarly, the emergence of powerful inference-time search algorithms for gradient-free conditioning of pre-trained diffusion models presents a compelling alternative starting point for addressing the problems outlined in Chapter 3. By leveraging Sequential Monte Carlo (Wu et al., 2023; Cardoso et al., 2024) and Monte Carlo Tree Search (Li et al., 2024; Ma et al., 2025) techniques, these methods replace the classifier guidance component with a search procedure that steers generation using a black-box reward function that typically evaluates only the final, denoised samples. This has a significant methodological advantage: since it obviates the need to train a parametric guidance model on noisy inputs, the complex, timestep-dependent regularization at the core of the context-guided diffusion framework is no longer necessary, greatly simplifying the training process. While this paradigm offers greater flexibility and modularity by readily incorporating non-differentiable objectives, it does so at the cost of a substantially increased computational burden at sampling time. The two approaches may therefore be viewed as complementary; the robust models produced by our framework could provide an efficient and robust predictive model to power these more expensive search-based techniques.

Limitations and Future Work

While we believe that the contributions presented in this thesis represent meaningful advances in machine learning for molecular design, the important limitations outlined in each of the chapters remain. Most importantly, all methods require access to appropriate domain knowledge or unlabelled data to function as intended, and are only applicable in settings in which these additional sources of information are abundant. Furthermore, translating this information into a format that is compatible with the proposed methods could require substantial efforts from domain experts and should be considered when deciding on the most appropriate modelling approach. Finally, the computational complexity of the proposed algorithms—in particular the constrained diffusion models—increases rapidly with system size, limiting their applicability to larger biomolecular complexes like protein-protein interactions and multi-component protein-ligand systems.

While the experiments presented in this thesis necessarily focus on retrospective evaluations, the most promising application of the developed methods lies in their ability to guide experimental efforts and accelerate the discovery of novel therapeutics. Future research should explore tighter integration between computational models and experimental workflows, enabling iterative refinement of predictions based on experimental feedback. Active learning frameworks that strategically select compounds for experimental validation could help address the data scarcity challenges highlighted throughout this thesis, by focusing limited experimental resources on the most informative regions of the input domain. Furthermore, real-world drug discovery typically requires the simultaneous optimisation of multiple competing properties, ranging from target affinity and selectivity to pharmacokinetics and safety profiles. Our context-guided diffusion framework provides a foundation for this type of multi-objective optimisation, but further work is needed to explore the most effective approaches to navigating these complex trade-offs under experimental uncertainty. Promising directions include Pareto-optimal sampling methods that explicitly model the frontier of non-dominated solutions across multiple objectives, as well as the application of state-of-the-art active learning and Bayesian optimisation methods to this setting.

Conclusion

The development of reliable machine learning methods for molecular design presents a critical challenge at the intersection of artificial intelligence and drug discovery. The context-guided diffusion framework, constrained diffusion methods, and semi-supervised predictive models presented in this thesis offer complementary approaches for addressing the fundamental challenges of generalisation and uncertainty quantification in low-data regimes. Together, they demonstrate that incorporating domain-specific prior knowledge and constraints into modern deep learning algorithms can enhance their robustness, physical validity, and reliability in practically relevant settings. Looking forward, the continued integration of deep learning with domain expertise and novel experimental techniques holds tremendous promise in aiding experimental scientists to navigate vast chemical and protein spaces, to ultimately accelerate the development of safer and more effective medicines.

Bibliography

J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*:1–3, 2024. Cited on pages 1, 15, 19, 28.

S. Alamdari, N. Thakkar, R. van den Berg, N. Tenenholtz, B. Strome, A. Moses, A. X. Lu, N. Fusi, A. P. Amini, K. K. Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*:2023–09, 2023. Cited on page 27.

B. D. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. Cited on page 23.

M. Arjovsky, L. Bottou. Towards principled methods for training generative adversarial networks. *arXiv:1701.04862*, 2017. Cited on page 21.

M. Arjovsky, S. Chintala, L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017. Cited on page 21.

J. Austin, D. D. Johnson, J. Ho, D. Tarlow, R. Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021. Cited on page 27.

A. Bansal, H.-M. Chu, A. Schwarzschild, S. Sengupta, M. Goldblum, J. Geiping, T. Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023. Cited on page 26.

R. Bellman. Dynamic programming. *Chapter IX, Princeton University Press, Princeton, New Jersey*, 1958. Cited on page 3.

Y. Bengio, A. Courville, P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. Cited on page 1.

J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo, et al. Improving image generation with better captions. *Computer Science*, 2(3):8, 2023. Cited on page 1.

C. Bilodeau, W. Jin, T. Jaakkola, R. Barzilay, K. F. Jensen. Generative models for molecular discovery: Recent advances and challenges. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(5):e1608, 2022. Cited on page 2.

K. Black, M. Janner, Y. Du, I. Kostrikov, S. Levine. Training Diffusion Models with Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*, 2024. Cited on page 26.

K. H. Bleicher, H.-J. Böhm, K. Müller, A. I. Alanine. Hit and lead generation: beyond high-throughput screening. *Nature Reviews Drug Discovery*, 2(5):369–378, 2003. Cited on page 8.

T. L. Blundell. Structure-based drug design. *Nature*, 384(6604):23, 1996. Cited on page 8.

R. S. Bohacek, C. McMartin, W. C. Guida. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal Research Reviews*, 16(1):3–50, 1996. Cited on pages 2, 7.

R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv:2108.07258*, 2021. Cited on page 2.

S. Bond-Taylor, A. Leach, Y. Long, C. G. Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7327–7347, 2021. Cited on page 20.

A. Brock, J. Donahue, K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv:1809.11096*, 2018. Cited on page 21.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language Models are Few-Shot Learners. *arXiv:2005.14165*, 2020. Cited on pages 1, 19.

M. Buttenschoen, G. M. Morris, C. M. Deane. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024. Cited on pages 4, 9.

A. Campbell, J. Yim, R. Barzilay, T. Rainforth, T. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv:2402.04997*, 2024. Cited on pages 3, 28.

G. V. Cardoso, Y. J. El Idrissi, S. Le Corff, E. Moulines. Monte Carlo guided Diffusion for Bayesian linear inverse problems. In *ICLR International Conference on Learning Representations*, 2024. Cited on page 101.

R. T. Chen, Y. Rubanova, J. Bettencourt, D. K. Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018. Cited on pages 3, 20.

J. Cheng, G. Novati, J. Pan, C. Bycroft, A. Žemgulytė, T. Applebaum, A. Pritzel, L. H. Wong, M. Zielinski, T. Sargeant, et al. Accurate proteome-wide missense variant effect prediction with AlphaMissense. *Science*, 381(6664):eadg7492, 2023. Cited on page 1.

L. Chinery, A. M. Hummer, B. B. Mehta, R. Akbar, P. Rawat, A. Slabodkin, K. L. Quy, F. Lund-Johansen, V. Greiff, J. R. Jeliazkov, et al. Baselining the buzz Trastuzumab-HER2 affinity, and beyond. *bioRxiv*:2024–03, 2024. Cited on page 4.

H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, J. C. Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv:2209.14687*, 2022. Cited on page 26.

Y. Chung, I. Char, H. Guo, J. Schneider, W. Neiswanger. Uncertainty Toolbox: an Open-Source Library for Assessing, Visualizing, and Improving Uncertainty Quantification. *arXiv:2109.10254*, 2021. Cited on page 18.

F. Cong, A. K. Cheung, S.-M. A. Huang. Chemical genetics–based target identification in drug discovery. *Annual Review of Pharmacology and Toxicology*, 52(1):57–78, 2012. Cited on page 8.

G. Corso, A. Deng, N. Polizzi, R. Barzilay, T. S. Jaakkola. Deep Confident Steps to New Pockets: Strategies for Docking Generalization. In *The Twelfth International Conference on Learning Representations*, 2024. Cited on pages 4, 28.

G. Corso, B. Jing, R. Barzilay, T. Jaakkola, et al. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. In *International Conference on Learning Representations (ICLR 2023)*, 2023. Cited on page 28.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989. Cited on pages 1, 14.

J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022. Cited on page 1.

P. Dhariwal, A. Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. Cited on page 25.

J. A. DiMasi, H. G. Grabowski, R. W. Hansen. Innovation in the pharmaceutical industry: new estimates of R&D costs. *Journal of Health Economics*, 47:20–33, 2016. Cited on pages 2, 7.

L. Dinh, D. Krueger, Y. Bengio. Nice: Non-linear independent components estimation. *arXiv:1410.8516*, 2014. Cited on page 20.

L. Dinh, J. Sohl-Dickstein, S. Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017. Cited on page 20.

A. Dosovitskiy, T. Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in Neural Information Processing Systems*, 29, 2016. Cited on page 21.

Y. Du, C. Durkan, R. Strudel, J. B. Tenenbaum, S. Dieleman, R. Fergus, J. Sohl-Dickstein, A. Doucet, W. S. Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International Conference on Machine Learning*, pages 8489–8510. PMLR, 2023. Cited on page 26.

J. Durairaj, Y. Adeshina, Z. Cao, X. Zhang, V. Oleinikovas, T. Duignan, Z. McClure, X. Robin, D. Kovtun, E. Rossi, et al. PLINDER: The protein-ligand interactions dataset and evaluation resource. *bioRxiv*:2024–07, 2024. Cited on page 4.

G. Durant, F. Boyles, K. Birchall, C. M. Deane. The future of machine learning for small-molecule drug discovery will be driven by data. *Nature Computational Science*:1–9, 2024. Cited on pages 3, 9.

G. Durant, F. Boyles, K. Birchall, B. Marsden, C. M. Deane. Robustly interrogating machine learning-based scoring functions: what are they learning? *Bioinformatics*, 41(2):btaf040, Jan. 2025. Cited on page 4.

I. Ellmen, C. Schneider, M. I. Raybould, C. M. Deane. Transformers trained on proteins can learn to attend to Euclidean distance. *arXiv:2502.01533*, 2025. Cited on page 15.

Epoch AI. Data on Machine Learning Hardware, 2024. Accessed: 2025-01-29. Cited on page 2.

A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. J. R Ruiz, J. Schrittwieser, G. Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022. Cited on page 1.

N. Fishman, **L. Klarner**, V. De Bortoli, E. Mathieu, M. Hutchinson. Diffusion Models for Constrained Domains. *Trans. on Machine Learning Research, 2023*.

N. Fishman, **L. Klarner**, E. Mathieu, M. Hutchinson, V. De Bortoli. Metropolis Sampling for Constrained Diffusion Models. *Advances in Neural Information Processing Systems*, 2023.

L. M. Friedman, C. D. Furberg, D. L. DeMets, D. M. Reboussin, C. B. Granger. *Fundamentals of Clinical Trials*. Springer, 2015. Cited on page 9.

Y. Gal, Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016. Cited on page 4.

R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. Cited on page 4.

Gemini Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*, 2024. Cited on pages 1, 19, 20.

F. Gentile, J. C. Yaacoub, J. Gleave, M. Fernandez, A.-T. Ton, F. Ban, A. Stern, A. Cherkasov. Artificial intelligence–enabled virtual screening of ultra-large chemical libraries with deep docking. *Nature Protocols*, 17(3):672–697, 2022. Cited on page 2.

J. B. Gibbs. Mechanism-based target identification and drug discovery in cancer research. *Science*, 287(5460):1969–1973, 2000. Cited on page 8.

R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. Cited on page 2.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. Cited on page 3.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial networks. *arXiv:1406.2661*, 2014. Cited on page 21.

R.-R. Griffiths, **L. Klarner**, H. Moss, A. Ravuri, S. Truong, Y. Du, S. Stanton, G. Tom, B. Rankovic, A. Jamasb, et al. GAUCHE: a library for Gaussian processes in chemistry. *Advances in Neural Information Processing Systems*, 2024.

N. Gruver, S. Stanton, N. Frey, T. G. Rudner, I. Hotzel, J. Lafrance-Vanasse, A. Rajpal, K. Cho, A. G. Wilson. Protein design with guided discrete diffusion. *Advances in Neural Information Processing Systems*, 36, 2024. Cited on page 3.

N. Gruver, S. Stanton, N. C. Frey, T. G. Rudner, I. Hotzel, J. Lafrance-Vanasse, A. Rajpal, K. Cho, A. G. Wilson. Protein Design with Guided Discrete Diffusion. *arXiv:2305.20009*, 2023. Cited on pages 26, 27.

O. O. Grygorenko, D. S. Radchenko, I. Dziuba, A. Chuprina, K. E. Gubina, Y. S. Moroz. Generating multibillion chemical space of readily accessible screening compounds. *Iscience*, 23(11), 2020. Cited on page 27.

I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, A. Courville. PixelVAE: A Latent Variable Model for Natural Images. In *International Conference on Learning Representations*, 2017. Cited on page 21.

C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017. Cited on page 18.

Y. Hasin, M. Seldin, A. Lusis. Multi-omics approaches to disease. *Genome Biology*, 18(1):1–15, 2017. Cited on page 7.

U. G. Haussmann, E. Pardoux. Time reversal of diffusions. *The Annals of Probability*, 14(4):1188–1205, 1986. Cited on page 23.

T. Hayes, R. Rao, H. Akin, N. J. Sofroniew, D. Oktay, Z. Lin, R. Verkuil, V. Q. Tran, J. Deaton, M. Wiggert, R. Badkundri, I. Shafkat, J. Gong, A. Derry, R. S. Molina, N. Thomas, Y. A. Khan, C. Mishra, C. Kim, L. J. Bartie, M. Nemeth, P. D. Hsu, T. Sercu, S. Candido, A. Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. Cited on pages 15, 19.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International Conference on Learning Representations*, 2017. Cited on page 21.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. Cited on page 20.

G. E. Hinton, T. J. Sejnowski. Optimal perceptual inference. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, volume 448, pages 448–453. Citeseer, 1983. Cited on page 20.

J. Ho, A. Jain, P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020. Cited on pages 3, 22, 23, 25.

J. Ho, T. Salimans. Classifier-Free Diffusion Guidance, 2022. arXiv:2207.12598: 2207.12598. Cited on page 25.

S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. Cited on page 2.

E. Hoogeboom, A. A. Gritsenko, J. Bastings, B. Poole, R. van den Berg, T. Salimans. Autoregressive Diffusion Models. In *International Conference on Learning Representations*, 2022. Cited on page 27.

E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, M. Welling. Argmax Flows and Multinomial Diffusion: Towards Non-Autoregressive Language Models. *arXiv:2102.05379*, 2021. Cited on page 27.

E. Hoogeboom, V. G. Satorras, C. Vignac, M. Welling. Equivariant Diffusion for Molecule Generation in 3D. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8867–8887. PMLR, 17–23 Jul 2022. Cited on page 28.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. Cited on page 14.

C.-W. Huang, D. Krueger, A. Lacoste, A. Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087. PMLR, 2018. Cited on page 20.

L. Huang, T. Xu, Y. Yu, P. Zhao, X. Chen, J. Han, Z. Xie, H. Li, W. Zhong, K.-C. Wong, et al. A dual diffusion model enables 3D molecule generation and lead optimization based on target pockets. *Nature Communications*, 15(1):2657, 2024. Cited on page 3.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. Cited on page 23.

J. Ingraham, M. Baranov, Z. Costello, V. Frappier, A. Ismail, S. Tie, W. Wang, V. Xue, F. Obermeyer, A. Beam, et al. Illuminating protein space with a programmable generative model. *bioRxiv*:2022–12, 2022. Cited on page 19.

J. Jiménez-Luna, F. Grisoni, G. Schneider. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10):573–584, 2020. Cited on page 3.

B. Jing, B. Berger, T. Jaakkola. AlphaFold Meets Flow Matching for Generating Protein Ensembles. In *Forty-first International Conference on Machine Learning*, 2024. Cited on page 28.

B. Jing, G. Corso, J. Chang, R. Barzilay, T. Jaakkola. Torsional diffusion for molecular conformer generation. *Advances in Neural Information Processing Systems*, 35:24240–24253, 2022. Cited on page 28.

B. Jing, H. Stark, T. Jaakkola, B. Berger. Generative Modeling of Molecular Dynamics Trajectories. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. Cited on page 28.

J. Jo, S. Lee, S. J. Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*, pages 10362–10383. PMLR, 2022. Cited on pages 3, 27.

W. L. Jorgensen. Efficient drug lead discovery and optimization. *Accounts of Chemical Research*, 42(6):724–733, 2009. Cited on page 9.

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. Cited on page 1.

J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei. Scaling laws for neural language models. *arXiv:2001.08361*, 2020. Cited on page 2.

K. J. Karczewski, M. P. Snyder. Integrative omics for health and disease. *Nature Reviews Genetics*, 19(5):299–310, 2018. Cited on page 7.

T. Karras, M. Aittala, T. Aila, S. Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022. Cited on pages 23, 24.

T. Karras, M. Aittala, T. Kynkäänniemi, J. Lehtinen, T. Aila, S. Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2025. Cited on page 26.

T. Kenakin. Predicting therapeutic value in the lead optimization phase of drug discovery. *Nature Reviews Drug Discovery*, 2(6):429–438, 2003. Cited on page 9.

G. M. Keserü, G. M. Makara. The influence of lead discovery strategies on the properties of drug candidates. *Nature Reviews Drug Discovery*, 8(3):203–212, 2009. Cited on page 9.

D. P. Kingma, M. Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013. Cited on pages 3, 21.

T. N. Kipf, M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907*, 2016. Cited on pages 2, 14.

P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021. Cited on page 4.

B. A. Koscher, R. B. Canty, M. A. McDonald, K. P. Greenman, C. J. McGill, C. L. Bilodeau, W. Jin, H. Wu, F. H. Vermeire, B. Jin, et al. Autonomous, multiproperty-driven molecular discovery: From predictions to measurements and back. *Science*, 382(6677):eadi1407, 2023. Cited on page 9.

D. Kovtun, M. Akdel, A. Goncearenco, G. Zhou, G. Holt, D. Baugher, D. Lin, Y. Adeshina, T. Castiglione, X. Wang, et al. PINDER: The protein interaction dataset and evaluation resource. *bioRxiv*:2024–07, 2024. Cited on page 4.

M. Krenn, F. Häse, A. Nigam, P. Friederich, A. Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020. Cited on page 13.

A. Krizhevsky, I. Sutskever, G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012. Cited on page 1.

L. Klarner, M. Reutlinger, T. Schindler, C. M. Deane, G. M. Morris. Bias in the Benchmark: Systematic experimental errors in bioactivity databases confound multi-task and meta-learning algorithms. *ICML AI for Science Workshop*, 2022.

L. Klarner, T. G. Rudner, G. M. Morris, C. M. Deane, Y. W. Teh. Context-guided diffusion for out-of-distribution molecular and protein design. *International Conference on Machine Learning*, 2024.

L. Klarner, T. G. Rudner, M. Reutlinger, T. Schindler, G. M. Morris, C. M. Deane, Y. W. Teh. Drug discovery under covariate shift with domain-informed prior distributions over functions. *International Conference on Machine Learning, 2023*.

B. Lakshminarayanan, A. Pritzel, C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017. Cited on page 4.

G. Landrum et al. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 2022. Cited on page 12.

Y. LeCun, Y. Bengio, G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. Cited on page 1.

Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel. Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 2, 1989. Cited on page 2.

S. Lee, J. Jo, S. J. Hwang. Exploring chemical space with score-based out-of-distribution generation. In *International Conference on Machine Learning*, pages 18872–18892. PMLR, 2023. Cited on page 26.

S. Lewis, T. Hempel, J. Jiménez Luna, M. Gastegger, Y. Xie, A. Y. Foong, V. García Satorras, O. Abdin, B. S. Veeling, I. Zaporozhets, et al. Scalable emulation of protein equilibrium ensembles with generative deep learning. *bioRxiv*:2024–12, 2024. Cited on page 28.

X. Li, Y. Zhao, C. Wang, G. Scalia, G. Eraslan, S. Nair, T. Biancalani, S. Ji, A. Regev, S. Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024. Cited on page 101.

M. A. Lindsay. Target discovery. *Nature Reviews Drug Discovery*, 2(10):831–838, 2003. Cited on page 8.

Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, M. Le. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*, 2023. Cited on page 24.

J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, B. Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020. Cited on page 4.

N. Ma, S. Tong, H. Jia, H. Hu, Y.-C. Su, M. Zhang, X. Yang, Y. Li, T. Jaakkola, X. Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025. Cited on page 101.

R. Macarron, M. N. Banks, D. Bojanic, D. J. Burns, D. A. Cirovic, T. Garyantes, D. V. Green, R. P. Hertzberg, W. P. Janzen, J. W. Paslay, et al. Impact of high-throughput screening in biomedical research. *Nature Reviews Drug Discovery*, 10(3):188–195, 2011. Cited on page 7.

J. Maynard Smith. Natural selection and the concept of a protein space. *Nature*, 225(5232):563–564, 1970. Cited on pages 2, 7.

A. Mayr, G. Klambauer, T. Unterthiner, M. Steijaert, J. K. Wegner, H. Ceulemans, D.-A. Clevert, S. Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical science*, 9(24):5441–5451, 2018. Cited on page 4.

L. M. Mayr, D. Bojanic. Novel trends in high-throughput screening. *Current Opinion in Pharmacology*, 9(5):580–588, 2009. Cited on page 8.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. Cited on page 1.

C. W. Murray, D. C. Rees. The rise of fragment-based drug discovery. *Nature Chemistry*, 1(3):187–192, 2009. Cited on page 8.

M. P. Naeini, G. Cooper, M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29 of number 1, 2015. Cited on page 18.

A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv:2112.10741*, 2021. Cited on page 25.

A. Q. Nichol, P. Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. Cited on page 23.

J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, D. Tran. Measuring Calibration in Deep Learning. In *CVPR Workshops*, volume 2 of number 7, 2019. Cited on page 18.

P. Notin, N. Rollins, Y. Gal, C. Sander, D. Marks. Machine learning for functional protein design. *Nature Biotechnology*, 42(2):216–228, 2024. Cited on page 2.

T. H. Olsen, F. Boyles, C. M. Deane. Observed Antibody Space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science*, 31(1):141–146, 2022. Cited on page 27.

OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. GPT-4 Technical Report. *arXiv:2303.08774*, 2023. Cited on pages 1, 19, 20.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. Cited on page 1.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019. Cited on pages 4, 18.

G. Papamakarios, T. Pavlakou, I. Murray. Masked autoregressive flow for density estimation. *Advances in Neural Information Processing Systems*, 30, 2017. Cited on page 20.

A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv:2204.06125*, 1(2):3, 2022. Cited on pages 25, 26.

J.-L. Reymond, M. Awale. Exploring chemical space for drug discovery using the chemical universe database. *ACS Chemical Neuroscience*, 3(9):649–657, 2012. Cited on page 7.

J.-L. Reymond, R. Van Deursen, L. C. Blum, L. Ruddigkeit. Chemical space as a source for new drugs. *MedChemComm*, 1(1):30–38, 2010. Cited on page 3.

D. Rezende, S. Mohamed. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538. PMLR, 2015. Cited on pages 3, 20, 21.

D. J. Rezende, S. Mohamed, D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286. PMLR, 2014. Cited on page 3.

D. Rogers, M. Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. Cited on page 12.

M. C. Rogge, D. R. Taft. Preclinical drug development. *Drugs and the Pharmaceutical Sciences*, 187, 2010. Cited on page 9.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. Cited on pages 19, 21.

F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958. Cited on page 2.

L. Ruddigkeit, R. van Deursen, L. C. Blum, J.-L. Reymond. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *J. Chem. Inf. Model.*, 52(11):2864–2875, Nov. 2012. Publisher: American Chemical Society. Cited on page 27.

T. G. Rudner, Z. Chen, Y. Gal. Rethinking Function-Space Variational Inference in Bayesian Neural Networks. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2020. Cited on page 191.

D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition. *Biometrika*, 71(599-607):6, 1986. Cited on page 1.

C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35, 2022. Cited on pages 1, 19, 25, 26.

S. S. Sahoo, A. Gokaslan, C. D. Sa, V. Kuleshov. Diffusion Models With Learned Adaptive Noise. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. Cited on page 23.

T. Saito, M. Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS One*, 10(3):e0118432, 2015. Cited on page 17.

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen. Improved techniques for training gans. *arXiv:1606.03498*, 2016. Cited on page 21.

B. Sanchez-Lengeling, A. Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018. Cited on page 2.

J. W. Scannell, A. Blanckley, H. Boldon, B. Warrington. Diagnosing the decline in pharmaceutical R&D efficiency. *Nature Reviews Drug Discovery*, 11(3):191–200, 2012. Cited on pages 2, 3, 7.

M. Schenone, V. Dančík, B. K. Wagner, P. A. Clemons. Target identification and mechanism of action in chemical biology and drug discovery. *Nature Chemical Biology*, 9(4):232–240, 2013. Cited on page 8.

P. Schneider, W. P. Walters, A. T. Plowright, N. Sieroka, J. Listgarten, R. A. Goodnow Jr, J. Fisher, J. M. Jansen, J. S. Duca, T. S. Rush, et al. Rethinking drug design in the artificial intelligence era. *Nature Reviews Drug Discovery*, 19(5):353–364, 2020. Cited on page 9.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. Cited on page 1.

R. B. Silverman, M. W. Holladay. *The organic chemistry of drug design and drug action*. Academic press, 2014. Cited on page 3.

J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. PMLR, 2015. Cited on pages 3, 22.

J. Song, Q. Zhang, H. Yin, M. Mardani, M.-Y. Liu, J. Kautz, Y. Chen, A. Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pages 32483–32498. PMLR, 2023. Cited on page 26.

Y. Song, S. Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019. Cited on page 23.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, B. Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2020. Cited on pages 3, 22–25.

C. Sun, A. Shrivastava, S. Singh, A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 843–852, 2017. Cited on page 2.

R. Sutton. The Bitter Lesson. 2019. URL: http://www.incompleteideas.net/IncIdeas/BitterLesson.html (visited on 02/15/2025). Cited on page 2.

E. G. Tabak, C. V. Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013. Cited on pages 3, 20.

E. G. Tabak, E. Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010. Cited on pages 3, 20.

D. Taylor. The pharmaceutical industry and the future of drug development, 2015. Cited on page 7.

M. Telgarsky. Benefits of depth in neural networks. In *Conference on Learning Theory*, pages 1517–1539. PMLR, 2016. Cited on pages 1, 14.

H. Thanh-Tung, T. Tran. Catastrophic forgetting and mode collapse in GANs. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2020. Cited on page 21.

B. I. Tingle, K. G. Tang, M. Castanon, J. J. Gutierrez, M. Khurelbaatar, C. Dandarchuluun, Y. S. Moroz, J. J. Irwin. ZINC-22 A free multi-billion-scale database of tangible compounds for ligand discovery. *Journal of Chemical Information and Modeling*, 63(4):1166–1176, 2023. Cited on page 27.

A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, Y. Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023. Cited on page 24.

T. H. Trinh, Y. Wu, Q. V. Le, H. He, T. Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024. Cited on page 1.

B. L. Trippe, J. Yim, D. Tischer, D. Baker, T. Broderick, R. Barzilay, T. S. Jaakkola. Diffusion Probabilistic Modeling of Protein Backbones in 3D for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations*, 2023. Cited on page 28.

UniProt-Consortium. UniProt: a hub for protein information. *Nucleic Acids Research*, 43(D1):D204–D212, 2015. Cited on page 27.

UniProt-Consortium. UniProt: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2023. Cited on page 27.

A. Vahdat, J. Kautz. NVAE: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020. Cited on page 21.

J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18(6):463–477, 2019. Cited on pages 7, 9.

J. Van Amersfoort, L. Smith, Y. W. Teh, Y. Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pages 9690–9700. PMLR, 2020. Cited on page 4.

V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013. Cited on pages 10, 11.

V. N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999. Cited on page 10.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. Cited on pages 2, 15.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, 2018. Cited on page 14.

C. Vignac, I. Krawczuk, A. Siraudin, B. Wang, V. Cevher, P. Frossard. DiGress: Discrete Denoising diffusion for graph generation. In *The Eleventh International Conference on Learning Representations*, 2023. Cited on page 27.

C. Vignac, N. Osman, L. Toni, P. Frossard. Midi: Mixed graph and 3d denoising diffusion for molecule generation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 560–576. Springer, 2023. Cited on page 28.

I. Wallach, A. Heifets. Most ligand-based classification benchmarks reward memorization rather than generalization. *Journal of Chemical Information and Modeling*, 58(5):916–932, 2018. Cited on page 4.

H. Wang, T. Fu, Y. Du, W. Gao, K. Huang, Z. Liu, P. Chandak, S. Liu, P. Van Katwyk, A. Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023. Cited on page 1.

J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. Cited on page 28.

D. Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. Cited on page 13.

T. Weiss, E. Mayo Yanes, S. Chakraborty, L. Cosmo, A. M. Bronstein, R. Gershoni-Poranne. Guided diffusion for inverse molecular design. *Nature Computational Science*, 3(10):873–882, 2023. Cited on page 26.

C. H. Wong, K. W. Siah, A. W. Lo. Estimation of clinical trial success rates and related parameters. *Biostatistics*, 20(2):273–286, 2019. Cited on page 7.

O. J. Wouters, M. McKee, J. Luyten. Estimated research and development investment needed to bring a new medicine to market, 2009-2018. *Jama*, 323(9):844–853, 2020. Cited on pages 2, 7.

L. Wu, B. Trippe, C. Naesseth, D. Blei, J. P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023. Cited on page 101.

K. Xu, W. Hu, J. Leskovec, S. Jegelka. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*, 2019. Cited on page 14.

K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, S. Jegelka. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pages 5453–5462. PMLR, 2018. Cited on page 15.

M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon, J. Tang. GeoDiff: A Geometric Diffusion Model for Molecular Conformation Generation. In *International Conference on Learning Representations*, 2022. Cited on page 28.

J. Yim, A. Campbell, A. Y. Foong, M. Gastegger, J. Jiménez-Luna, S. Lewis, V. G. Satorras, B. S. Veeling, R. Barzilay, T. Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv:2310.05297*, 2023. Cited on page 28.

J. Yim, B. L. Trippe, V. De Bortoli, E. Mathieu, A. Doucet, R. Barzilay, T. Jaakkola. SE (3) diffusion model with application to protein backbone generation. In *International Conference on Machine Learning*, pages 40001–40039. PMLR, 2023. Cited on page 28.

J. Yu, Y. Wang, C. Zhao, B. Ghanem, J. Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23174–23184, 2023. Cited on page 26.

L. Zhang, J. Tan, D. Han, H. Zhu. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug Discovery Today*, 22(11):1680–1685, 2017. Cited on page 7.

L. Zhang, A. Rao, M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. Cited on page 19.

S. Zhao, J. Song, S. Ermon. Towards deeper understanding of variational autoencoding models. *arXiv:1702.08658*, 2017. Cited on page 21.

Supplementary Information

Appendix A

Supplementary to *Context-Guided Diffusion*

Appendix

Table of Contents

A Guidance Function Parameterization **17**

A.1 Probabilistic Interpretation 17

B Experimental Details **18**

B.1 Illustrating the Behaviour of Guided Diffusion Models on the Swiss Roll Dataset 18

B.2 Graph-Structured Diffusion for Small Molecules 21

B.3 Equivariant Diffusion for Materials. 31

B.4 Discrete Diffusion for Protein Sequences. 35

B.5 Runtime Comparison. 37

B.6 Similarity of the Generated Samples to the Training and Validation Sets. 38

Reproducibility

All data and code needed to reproduce our experiments can be accessed at:
https://github.com/leojklarner/context-guided-diffusion

A. Guidance Function Parameterization

To guide sampling in a diffusion model toward certain desired properties, a *guidance model* $f_t(\cdot;\theta)$ is trained on a dataset $\mathcal{D}$, a parametric *guidance function* $p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta))$ is defined using the learned guidance model, and then the *conditional gradient/score function*

$$\nabla \log p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta)) + \nabla \log p(\mathbf{x}_t), \tag{A.1}$$

are used to guide generation towards high-density regions $p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta))$. To provide a general framework that accommodates both classification and regression settings, we let $\mathbf{Y}$ denote an "optimality" random variable indicating the optimality (under some domain-specific optimality function) of a given input $\mathbf{x}_t$.

For discrete desirable properties, the guidance model is trained on a classification task, and the guidance function $p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta))$ is defined as a categorical likelihood function, making $-\log p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta))$ a cross-entropy loss. To guide a diffusion model towards generating samples consistent with a specific class c, the guidance function is computed under $\mathbf{y} = \text{one-hot}(c)$, denoting that the desired class c is represented by a one-hot encoding. The score function then becomes the class probability of the desired class under the learned model:

$$\nabla \log p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta)) = \nabla \log \text{softmax}(f_t(\mathbf{x} \mid \theta))_c. \tag{A.2}$$

For continuous desirable properties, the guidance model is trained on a regression task, and the guidance function $p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta))$ is defined as a Bernoulli likelihood function with

$$p(\mathbf{y} = 1 \,|\, \mathbf{x}_t, C; f_t(\cdot;\theta)) = \exp\left(f_t(\mathbf{x}_t;\theta) - C\right), \tag{A.3}$$

denoting the probability of optimality, and C representing a domain-specific "goal value". Importantly, this formulation requires that $f_t(\mathbf{x}_t;\theta) < C$ so that increases in $f_t(\mathbf{x}_t;\theta))$ indicate desirable properties.

A.1. Probabilistic Interpretation

Letting $\mathcal{O}$ be an "optimality" random variable indicating whether a prediction $f_t(\hat{\mathbf{x}}_t;\theta)$ is optimal for achieving a desired behavior (e.g., high predictive uncertainty on data points far away from the training data), we can define a Bernoulli observation model where

$$p(\mathcal{O} = 1 \,|\, \theta, t; f_t, p_{\hat{\mathbf{X}}_t}) = \exp\left(-R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})\right), \tag{A.4}$$

denotes the probability of optimality as a function of the regularizer. That is, the smaller $R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})$, the higher the probability that a prediction $f_t(\hat{\mathbf{x}}_t;\theta)$ is optimal.

Following the result in Rudner et al. (2023), we note that specifying a data-dependent regularizer of the form shown in Equation (8) corresponds to specifying a data-driven prior distribution over neural network parameters

$$p(\theta \,|\, t, \mathcal{O} = 1; f_t, p_{\hat{\mathbf{X}}_t}) = \frac{p(\mathcal{O} = 1 \,|\, \theta, t; f_t, p_{\hat{\mathbf{X}}_t})\, p(\theta)}{p(\mathcal{O} = 1 \,|\, t; f_t, p_{\hat{\mathbf{X}}_t})}.$$

Minimizing the objective function in Equation (9) corresponds to finding a variational approximation to the posterior implied by this data-driven prior and the observed data. That is, minimizing the objective function in Equation (9) (approximately) corresponds to finding a variational distribution $q_\Theta = \mathcal{N}(\theta; \theta', \sigma^2)$ (with fixed and very small σ^2) that solves the variational problem

$$\min_{q_\Theta} \mathbb{E}_{p_\mathcal{T}} \left[\mathbb{D}_{\text{KL}}(q_\Theta \,||\, p_{\Theta|\mathcal{D}_\mathcal{T},\mathcal{O},\mathcal{T}})\right], \tag{A.5}$$

allowing us to interpret the learned guidance model through the lens of Bayesian inference.

B. Experimental Details

This section provides additional information and experimental results complementing the main text. Appendix B.1 details the regression experiments from Section 3 and contrasts our approach with standard L_2 regularization, illustrated in Figure 3. Appendix B.2 presents our extension of Lee et al. (2023)'s work on graph-structured diffusion models for small molecule generation (Section 5.1). Appendix B.3 presents our extension of Weiss et al. (2023)'s experiments on equivariant diffusion models for generating novel polycyclic aromatic systems (Section 5.2). Finally, Appendix B.4 presents our extension of Gruver et al. (2023)'s approach to optimizing discrete protein sequences using categorical diffusion models (Section 5.3).

All code was written in (Van Rossum & Drake, 2009) and can be accessed at https://github.com/leojklarner/context-guided-diffusion. A range of core scientific computing libraries were used for data preparation and analysis, including NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020), Pandas (pandas development team, 2020), Matplotlib (Hunter, 2007), Seaborn (Waskom, 2021), scikit-learn (Pedregosa et al., 2011) and rdkit (Landrum et al., 2013). All deep learning models were implemented in PyTorch (Paszke et al., 2019).

B.1. Illustrating the Behaviour of Guided Diffusion Models on the Swiss Roll Dataset

Dataset. In order to illustrate the impact of our regularizer on the learning dynamics of a guidance model, particularly when only trained on a constrained and less diverse subset of the input domain, we conduct a comparison with standard parameter-space regularization schemes, namely explicit L_2 regularization. This comparison is performed on the Swiss roll dataset, as described by Marsland (2011) and implemented in the scikit-learn library (Pedregosa et al., 2011). The input data $\mathbf{X}_t \in \mathbb{R}^2$ and regression labels $\mathbf{y} \in \mathbb{R}$ are depicted in Figure 7. A threshold of $\mathbf{y} = 1$ is used to split the data into training and validation sets. The training set encompasses data points with lower label values, while the validation set consists of data points with higher labels. This split is designed to evaluate the capability of a guided diffusion model trained on a subset of lower-value data to extrapolate to and generate data points from higher-value, out-of-distribution regions.

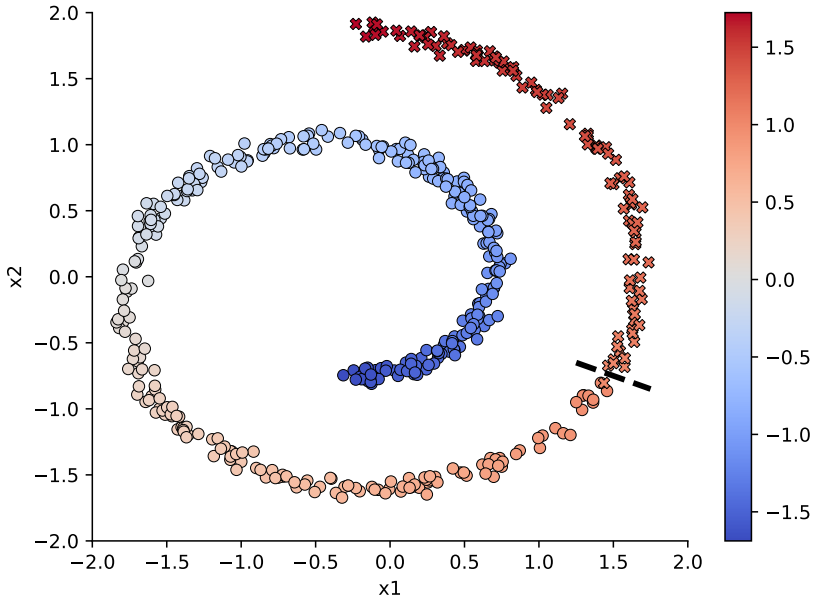


Figure 7: A visualization of the Swiss roll dataset used to train different guidance models. It consists of 500 data points generated with the corresponding function in the scikit-learn library (Pedregosa et al., 2011) using a noise parameter of 0.3. Both covariates $\mathbf{X}_t \in \mathbb{R}^2$ and labels $\mathbf{y} \in \mathbb{R}$ have been normalized to a mean of zero and a standard deviation of one. The color-coded regression labels are derived from the data points' sequential ordering along the manifold's arc length. A label threshold of $\mathbf{y} = 1$, denoted by a dashed line, shows the training-validation split: the training data (circles) contains data points with lower label values, while the validation data (crosses) consists of data points with higher labels. This split is designed to evaluate the ability of a guided diffusion model trained on a subset of lower-value data to extrapolate to and generate data points from higher-value, out-of-distribution regions.

Diffusion Model Training. To train unconditional diffusion models, we adapt an existing implementation of the denoising diffusion probabilistic models from Ho et al. (2020) to train diffusion models on datasets in $\mathbb{R}^2$ (https://github.com/albarji/toy-diffusion). Specifically, we generate 100 000 data points from the same data-generating process as our labeled training set, i.e. data points from the Swiss roll generator with standardized labels $\mathbf{y} < 1$. Using a standard cosine beta schedule (Nichol & Dhariwal, 2021)

$$\beta_t = 1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, \quad \text{with} \quad \bar{\alpha}_t = \frac{f(t)}{f(0)} \quad \text{and} \quad f(t) = \cos\left(\frac{t/T + \epsilon}{1+\epsilon} \cdot \frac{\pi}{2}\right)^2 \tag{B.6}$$

where $\epsilon = 0.008$ and $T = 40$, we ensure that the noising process converges to the target distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The score network $s_\theta : \mathbb{R}^2 \times [1, \dots, T] \to \mathbb{R}^2$ is constructed as a multi-layer perceptron with 5 hidden layers of dimension 64 and ReLU activation functions (Nair & Hinton, 2010). It is trained with the L_{simple} noise prediction objective from Ho et al. (2020) over 100 epochs of stochastic gradient descent. Specifically, we use the ADAM optimizer (Kingma & Ba, 2014) with a batch size of 2048 and a learning rate of 1×10^{-3} that drops off to 1×10^{-5} following a linear learning rate decay schedule. Samples from this unconditional model are shown in Figure 8.

Guidance Function Training. For the training of guidance models $f_t(\cdot; \theta) : \mathbb{R}^2 \times [1, \dots, T] \to \mathbb{R}^2$, we use smaller multi-layer perceptrons with 3 hidden layers of dimension 32, applying $\sin$ activation functions and dropout (Srivastava et al., 2014) with $p = 0.2$ after each layer. These networks are trained to output both the most likely regression label $\boldsymbol{\mu}_t(\mathbf{x}_t; \theta) = f_t^1(\mathbf{x}_t; \theta)$, as well as log-variances $\log \boldsymbol{\sigma}_t^2(\mathbf{x}_t; \theta) = f_t^2(\mathbf{x}_t; \theta)$ that serve as an estimator of their predictive uncertainty. Given the regression labels $\mathbf{y}$, the models are optimized with respect to a negative log-likelihood loss $\mathcal{L}_t = -\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2 \mathbf{I})$ over 100 epochs of stochastic gradient descent, using the ADAM optimizer with a batch size of 128 and a learning rate of 1×10^{-2}. The corresponding objective $\mathcal{L}$ with an explicit L_2 regularization term is given by

$$\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}) \doteq \mathbb{E}_{p_\mathcal{T}} \left[-\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_\mathcal{T}, \boldsymbol{\sigma}_\mathcal{T}^2 \mathbf{I}) \right] + \frac{1}{2\lambda} ||\theta||_2^2, \tag{B.7}$$

where the regularization strength λ is optimized as a hyperparameter. Similarly, we train context-guided models by adding the additional regularization term $R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})$ as introduced in Section 3 and given by

$$R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t}) = \mathbb{E}_{p_{\hat{\mathbf{X}}_t}} \left[\sum_{j=1}^{2} \left(f_t^{(j)}(\hat{\mathbf{x}}_t; \theta) - m_t(\hat{\mathbf{x}}_t) \right)^\top K_t(\hat{\mathbf{x}}_t)^{-1} \left(f_t^{(j)}(\hat{\mathbf{x}}_t; \theta) - m_t(\hat{\mathbf{x}}_t) \right) \right], \tag{B.8}$$

Here, $\hat{\mathbf{x}}_t \sim p_{\hat{\mathbf{X}}_t}(\hat{\mathbf{x}}_t)$ represents a context batch sampled from an out-of-distribution context set comprising 10 000 points distributed uniformly over $[-2.5, 2.5]^2 \subset \mathbb{R}^2$. As detailed in Section 3, we use a randomly initialized parameter set ϕ to generate 32-dimensional context set embeddings $h_t(\hat{\mathbf{x}}_t, \phi) \in \mathbb{R}^{M \times 32}$ and use them to construct the covariance matrix

$$K_t(\hat{\mathbf{x}}_t) = \sigma_t h_t(\hat{\mathbf{x}}_t, \phi) h_t(\hat{\mathbf{x}}_t, \phi)^T + \tau_t \mathbf{I}.$$

As the model returns mean and log-variance estimates $f_t^1(\hat{\mathbf{x}}_t; \theta)$ and $f_t^2(\hat{\mathbf{x}}_t; \theta)$, the target function values $m_t(\hat{\mathbf{x}}_t)$ are given by the training set mean $m_t^1(\hat{\mathbf{x}}_t) = -0.38$ and a log-variance hyperparameter $m_t^2(\hat{\mathbf{x}}_t) = \boldsymbol{\sigma}_0^2$, respectively. The latter is set to $\boldsymbol{\sigma}_0 = 0.7$, chosen to induce high predictive uncertainty estimates of $\exp(\boldsymbol{\sigma}^2) \approx 2$ in out-of-distribution regions of the input domain. The values of the covariance scale σ_t and the diagonal offset τ_t are optimized as hyperparameters. While empirical performance tends to improve when σ_t and τ_t are made time-dependent, we found that in this setting we can already achieve good performance when keeping them constant. The resulting augmented training objective is given as

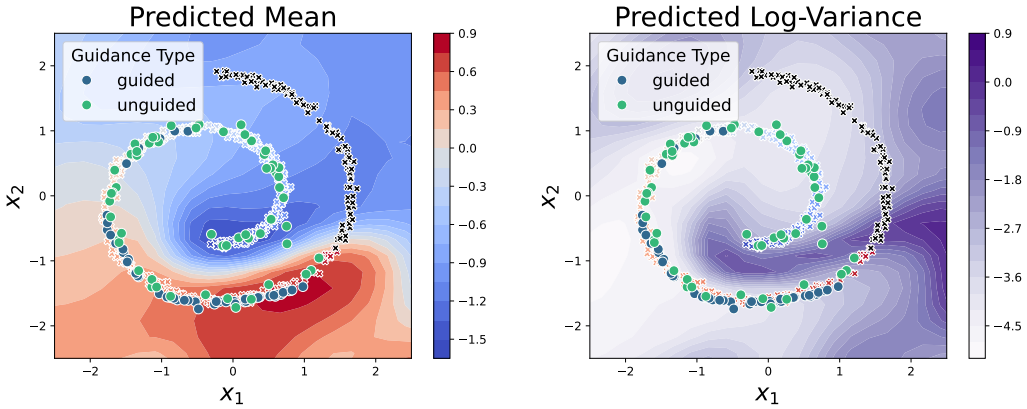
$$\mathcal{L}^*(\theta, \mathcal{D}) = \mathbb{E}_{p_\mathcal{T}} \left[\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}, \mathcal{T}) + R(\theta, f_t, \mathcal{T}, p_{\hat{\mathbf{X}}_\mathcal{T}}) \right]. \tag{B.9}$$

The optimal regularization hyperparameters were selected by performing grid search over the hyperparameter grid presented in Table 2 with respect to the supervised loss $\mathcal{L}$ evaluated on the held-out validation set.

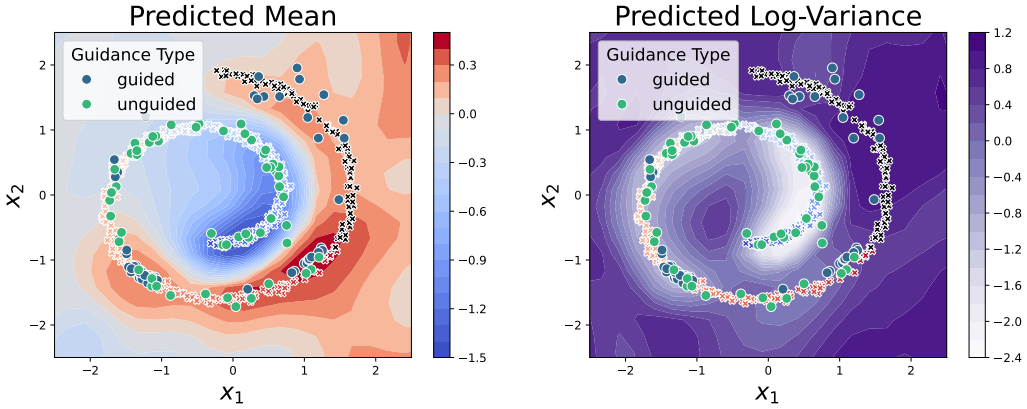
Table 2: Hyperparameter search space for regularization schemes of regression models detailed in Appendix B.1.

Hyperparameter	Description	Search Space
λ	L_2 regularization strength	$10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3, 10^4$
σ_t	covariance scale	$10^{-5}, 10^{-3}, 10^{-1}, 1, 10^1, 10^3, 10^5$
τ_t	diagonal offset	$10^{-5}, 10^{-3}, 10^{-1}, 1, 10^1, 10^3, 10^5$

Results A comparative analysis of model performances is provided in Figure 8. We observe that both L_2-regularized and context-guided models fit the training data well. However, without additional information about the input domain, the predictions of the L_2-regularized model exhibit poor generalization and miscalibrated confidence estimates when evaluated in new, unseen regions. In contrast, the context-guded regularizer is able to use this additional information to steer guidance model training toward functions that seem to capture the underlying structure of the input domain more accurately. More importantly, it also generates significantly better-calibrated uncertainty estimates. This improvement translates into higher-quality samples when using gradients from the context-guided model for conditional sampling.



(a) Predictions and samples from a context-guided model



(b) Predictions and samples from a context-guided guidance model.

Figure 8: Comparative analysis of predictions, uncertainty estimates, and samples generated by guidance models trained with L_2 regularization and context-guidance schemes. The training and validation sets, (see Figure 7), are depicted by color-coded and black crosses, respectively. We display 50 samples generated by both unconditional and conditional models, depicted as green and blue circles, respectively. The samples are generated with the same random seed and no cherry-picking. The left panel shows the model's mean predictions across a broader input space $[-2.5, 2.5]^2 \subset \mathbb{R}^2$, with warmer colors (red) indicating higher predicted values. The right panel shows the model's log-variance estimates across the same interval, serving as a measure of uncertainty, with darker regions indicating more uncertain predictons. We observe that both L_2-regularized and context-guided models fit the training data well. However, without additional information about the input domain, the predictions of the L_2-regularized model exhibit poor generalization and miscalibrated confidence estimates when evaluated in new, unseen regions. In contrast, our context-guided approach is able to use this additional information to steer guidance model training toward functions that seem to capture the underlying structure of the input domain more accurately. More importantly, it also generates significantly better-calibrated uncertainty estimates. This improvement translates into higher-quality samples when using gradients from the context-guided model for conditional sampling.

B.2. Graph-Structured Diffusion for Small Molecules

Dataset. We evaluate all methods on the same dataset as Lee et al. (2023), consisting of 250 000 molecules sampled uniformly from the ZINC database of commercially available compounds (Irwin et al., 2012). Each molecule is annotated with ground-truth labels $\mathbf{y}$, defined in terms of

- the clipped and normalized negative QuickVina docking score $\mathbf{y}_{\text{vina}}$ (Alhossary et al., 2015),
- the normalized synthetic accessibility (SA) score $\mathbf{y}_{\text{SA}}$ (Ertl, 2003), and
- the quantitative estimate for drug-likeness (QED) score $\mathbf{y}_{\text{QED}}$ (Bickerton et al., 2012).

The resulting composite objective (for which larger values indicate better performance) is given by

$$\mathbf{y} = \frac{\text{clip}(-\mathbf{y}_{\text{vina}}, 20, 0)}{20} \cdot \frac{10 - \mathbf{y}_{\text{SA}}}{9} \cdot \mathbf{y}_{\text{QED}}\,, \text{ with } \mathbf{y} \in [0, 1]. \tag{B.10}$$

As the QuickVina docking score $\mathbf{y}_{\text{vina}}$ is a function of the specific protein pocket that a molecule is docked into, all data preparation and model training steps described in this section are repeated across five different proteins from the DUD-E dataset (Mysinger et al., 2012) selected by Lee et al. (2023), namely PARP1 (Poly [ADP-ribose] polymerase-1), FA7 (Coagulation factor VII), 5HT1B (5-hydroxytryptamine receptor 1B), BRAF (Serine/threonine-protein kinase B-raf), and JAK2 (Tyrosine-protein kinase JAK2). All compounds are featurized as topological molecular graphs $\mathbf{g}_0$, defined as tuples $\mathbf{g}_0 = (\mathbf{x}_0, \mathbf{a}_0)$ of node features $\mathbf{x}_0 \in \mathbb{R}^{M \times F}$ and adjacency matrices $\mathbf{a}_0 \in \mathbb{R}^{M \times M}$, where $M = 38$ is the maximum number of heavy atoms in the training set and $F = 10$ is the dimensionality of node features, denoting a one-hot encoding of the element types C, N, O, F, P, S, Cl, Br, I, and NONE. Different bond types are learned by discretizing the continuous adjacency matric $\mathbf{a}_0$ into bins corresponding to certain bonds, namely $a_{i,j} < 0.5 \rightarrow$ NONE, $0.5 < a_{i,j} < 1.5 \rightarrow$ single, $1.5 < a_{i,j} < 2.5 \rightarrow$ double, $a_{i,j} > 2.5 \rightarrow$ triple. Following Lee et al. (2023), we then order the molecules by their labels $\mathbf{y}$ and split them evenly into a low-value training set and a high-value test set of 125 000 data points each, repeated for each of the five protein targets. This experimental setup allows us to evaluate the ability of guided diffusion models trained on molecules with low-value properties to extrapolate to and generate data points from higher-value, out-of-distribution regions.

To generate a context set $\hat{\mathbf{x}}_{\mathcal{C}}$ for our context-guided diffusion method, an additional 500 000 unlabeled compounds were sampled uniformly from the ZINC database and processed in the same fashion as before. A comparison of the training and context set in terms of different molecular properties derived with RDKIT (Landrum et al., 2013) is presented in Figure 9.

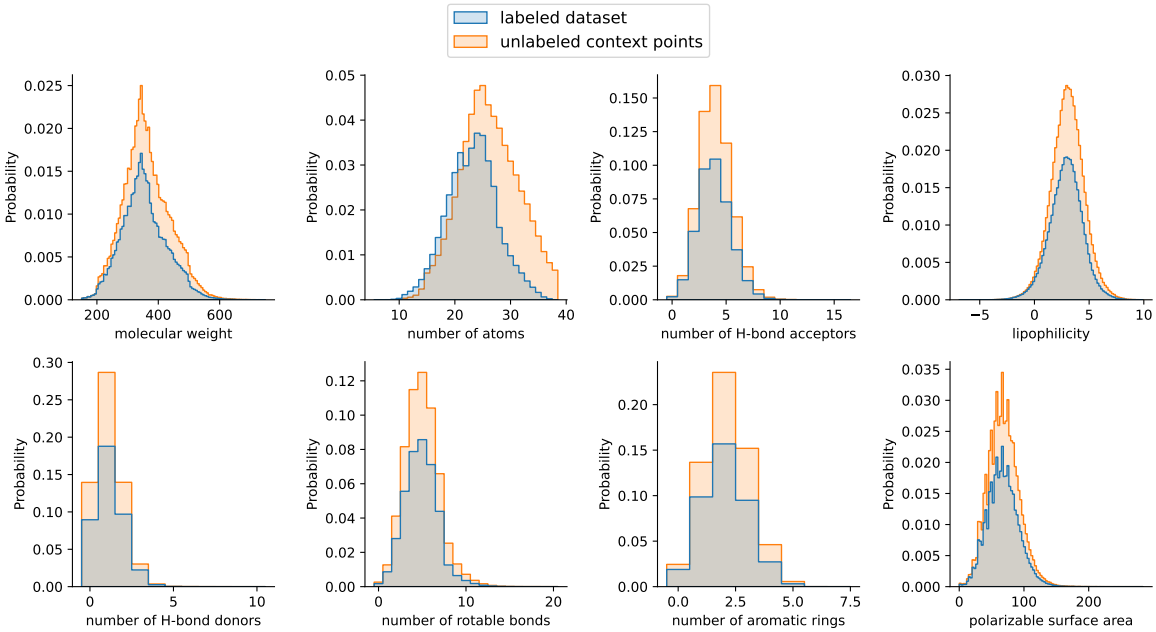


Figure 9: Comparison of the molecular properties of the labeled training and unlabeled context sets, verifying that the latter consists of meaningfully related yet distinct compounds and is suitable for use with our regularization scheme.

Diffusion Models over Graphs. The *Molecular Out-Of-distribution Diffusion* (MOOD) framework introduced in Lee et al. (2023) leverages diffusion models defined over the space of molecular graphs. Specifically, they employ the *Graph Diffusion via the System of Stochastic differential equations* (GDSS) formalism of Jo et al. (2022), which we restate in the following. In this framework, a molecular graph $\mathbf{g}_0$ is defined as a tuple $\mathbf{g}_0 = (\mathbf{x}_0, \mathbf{a}_0)$ of node features $\mathbf{x}_0 \in \mathbb{R}^{M \times F}$ and adjacency matrices $\mathbf{a}_0 \in \mathbb{R}^{M \times M}$, where M is the maximum number of heavy atoms in the training set and F is the dimensionality of the node features. Following the formalism of Song et al. (2020) outlined in Section 2, the forward diffusion process $\{\mathbf{G}_t = (\mathbf{X}_t, \mathbf{A}_t)\}_{t=0}^T$ is given by

$$\mathrm{d}\mathbf{G}_t = \mathbf{f}(\mathbf{G}_t, t)\,\mathrm{d}t + \beta_t\,\mathrm{d}\mathbf{B}_t \tag{B.11}$$

and decomposes into a system of stochastic differential equations that can be modeled jointly

$$\begin{aligned} \mathrm{d}\mathbf{X}_t &= \mathbf{f}_{\mathbf{X}}(\mathbf{X}_t, t)\,\mathrm{d}t + \beta_{\mathbf{X},t}\,\mathrm{d}\mathbf{B}_t \\ \mathrm{d}\mathbf{A}_t &= \mathbf{f}_{\mathbf{A}}(\mathbf{A}_t, t)\,\mathrm{d}t + \beta_{\mathbf{A},t}\,\mathrm{d}\mathbf{B}_t, \end{aligned} \tag{B.12}$$

where $\mathbf{f}(\mathbf{G}_t, t) = [\mathbf{f}_{\mathbf{X}}(\mathbf{X}_t, t), \mathbf{f}_{\mathbf{A}}(\mathbf{A}_t, t)]$ and $\beta_t = [\beta_{\mathbf{X},t}, \beta_{\mathbf{A},t}]$. The corresponding time reversal $\{\bar{\mathbf{G}}_t = (\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t)\}_{t \geq 0} = \{\mathbf{G}_{T-t} = (\mathbf{X}_{T-t}, \mathbf{A}_{T-t})\}_{t \in [0,T]}$ is given by

$$\mathrm{d}\bar{\mathbf{G}}_t = \left\{-\mathbf{f}(\bar{\mathbf{G}}_t, T-t) + \beta_{T-t}^2 \nabla_{\mathbf{G}} \log p_{T-t}\left(\bar{\mathbf{G}}_t\right)\right\}\,\mathrm{d}t + \beta_{T-t}\,\mathrm{d}\mathbf{B}_t, \tag{B.13}$$

flowing backward in time and decomposing into a corresponding joint system of SDEs given by

$$\begin{aligned} \mathrm{d}\bar{\mathbf{X}}_t &= \left\{-\mathbf{f}_{\mathbf{X}}(\bar{\mathbf{X}}_t, T-t) + \beta_{\mathbf{X},T-t}^2 \nabla_{\mathbf{X}} \log p_{T-t}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t\right)\right\}\,\mathrm{d}t + \beta_{\mathbf{X},T-t}\,\mathrm{d}\mathbf{B}_t \\ \mathrm{d}\bar{\mathbf{A}}_t &= \left\{-\mathbf{f}_{\mathbf{A}}(\bar{\mathbf{A}}_t, T-t) + \beta_{\mathbf{A},T-t}^2 \nabla_{\mathbf{A}} \log p_{T-t}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t\right)\right\}\,\mathrm{d}t + \beta_{\mathbf{A},T-t}\,\mathrm{d}\mathbf{B}_t, \end{aligned} \tag{B.14}$$

where $\nabla_{\mathbf{X}} \log p_t(\cdot)$ and $\nabla_{\mathbf{A}} \log p_t(\cdot)$ that are approximated by separate score networks $s_{\theta_{\mathbf{X}}}(\mathbf{X}_t, \mathbf{A}_t, t)$ and $s_{\theta_{\mathbf{A}}}(\mathbf{X}_t, \mathbf{A}_t, t)$. To encourage the exploration of novel chemical space, Lee et al. (2023) condition the denoising process on a hyperparameter $\gamma \in [0, 1]$ that shrinks the unconditional gradients to explore near out-of-distribution regions around the training distribution

$$\begin{aligned} \nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t \mid \gamma) &= \nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t) + \nabla_{\mathbf{G}} \log p_t(\gamma \mid \boldsymbol{G}_t) \\ &= \nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t) + \nabla_{\mathbf{G}} \log \left(p_t(\mathbf{G}_t)^{-\sqrt{\gamma}}\right) \\ &= (1 - \sqrt{\gamma})\,\nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t). \end{aligned} \tag{B.15}$$

In all of our experiments, we follow Lee et al. (2023) and set $\sqrt{\gamma} = 0.2$. To additionally condition the denoising process on desirable molecular properties, Lee et al. (2023) make use of standard guided diffusion as outlined in Section 2, specified as

$$\begin{aligned} \nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t \mid \gamma, \mathbf{y}) &= \nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t) + \nabla_{\mathbf{G}} \log p_t(\gamma \mid \boldsymbol{G}_t) + \nabla_{\mathbf{G}} \log p_t(\mathbf{y} \mid \mathbf{G}_t) \\ &= (1 - \sqrt{\gamma})\,\nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t) + \nabla_{\mathbf{G}} \log p_t(\mathbf{y} \mid \mathbf{G}_t). \end{aligned} \tag{B.16}$$

Here, a regression model $f_t(\mathbf{g}_t, \theta)$ is used to approximate the conditional distribution $p_t(\mathbf{y} \mid \mathbf{g}_t) \approx \exp(\alpha_t f_t(\mathbf{g}_t, \theta))/Z_t$, where α_t is an adaptive guidance scale based on a hyperparameter $r_{\mathbf{G},0}$ and given by

$$\alpha_{\mathbf{G},t} = r_{\mathbf{G},t} \frac{\|s_\theta(\mathbf{G}_t, t)\|_2}{\nabla_{\mathbf{G}} \log \|\beta_\theta(\mathbf{G}_t, t)\|_2} \quad \text{with} \quad r_{\mathbf{G},t} = 0.1^t r_{\mathbf{G},0} \tag{B.17}$$

Combining the decomposition into a system of joint SDEs from Equation (B.14), the shrinking of the unconditional gradients from Equation (B.15) and the guidance model formulation from Equation (B.17), the full conditional sample generation process used by Lee et al. (2023) is given by

$$\begin{aligned} \mathrm{d}\bar{\mathbf{X}}_t &= \beta_{\mathbf{X},T-t}^2 \left\{\frac{1}{2}\bar{\mathbf{X}}_t + (1 - \sqrt{\gamma})\, s_{\theta_{\mathbf{X}}}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t, T-t\right) + \nabla_{\mathbf{X}} \log \exp\left(\alpha_{\mathbf{X},T-t} f_{T-t}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t; \theta\right)\right)\right\} \mathrm{d}t + \beta_{\mathbf{X},T-t} \mathrm{d}\mathbf{B}_t \\ \mathrm{d}\bar{\mathbf{A}}_t &= \beta_{\mathbf{A},T-t}^2 \left\{(1 - \sqrt{\gamma})\, s_{\theta_{\mathbf{A}}}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t, T-t\right) + \nabla_{\mathbf{A}} \log \exp\left(\alpha_{\mathbf{A},T-t} f_{T-t}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t; \theta\right)\right)\right\} \mathrm{d}t + \beta_{\mathbf{A},T-t} \mathrm{d}\mathbf{B}_t \end{aligned} \tag{B.18}$$

where the drift and diffusion coefficients of $\mathrm{d}\bar{\mathbf{X}}_t$ and $\mathrm{d}\bar{\mathbf{A}}_t$ are determined by the variance-preserving (VP) and variance-exploding (VE) SDE from Song et al. (2020), respectively.

Diffusion Model Training. As in Lee et al. (2023), we adopt the pre-trained node feature and adjacency matrix score networks $s_{\theta_{\mathbf{X}}}(\mathbf{x}_t, \mathbf{a}_t, t)$ and $s_{\theta_{\mathbf{A}}}(\mathbf{x}_t, \mathbf{a}_t, t)$ from Jo et al. (2022) to define our unconditional diffusion model. These networks are built on graph multi-head attention (GMH) architectures (Baek et al., 2021) to ensure permutation-equivariance. Each network consists of two graph attention layers with four attention heads of hidden dimension 16 and $\tanh$ activation functions, followed by a multi-layer perceptron output head with three layers, each of dimension 82 and eLU activation functions (Clevert et al., 2015). The node feature SDE is based on the variance-preserving SDE from Song et al. (2020), using a linear β-schedule ($\beta_{\mathbf{X},0} = 0.1$, $\beta_{\mathbf{X},T} = 1$). In contrast, the adjacency matrix SDE uses the variance-exploding SDE also from Song et al. (2020), with an exponential β-schedule ($\beta_{\mathbf{A},0} = 0.2$, $\beta_{\mathbf{A},T} = 1$). Both models employ $T = 1000$ discretization steps. It's important to note that the unconditional score networks from (Jo et al., 2022) were trained on an i.i.d 85%-15% training-validation split of our dataset, meaning that they may have been optimized using part of the high-property validation set also used for training other guidance models. However, this is not an issue as (a) the unconditional diffusion model was trained without access to label infomration, (b) we only use the validation set for hyperparameter optimization, and (c) the vast search space ensures that most generated compounds are neither in the training nor the validation set (see Appendix B.6 for a detailed analysis).

Guidance Model Training. For training the guidance models $f_t(\cdot; \theta)$, we use the graph convolutional neural network (GNN) architecture from Lee et al. (2023), consisting of 3 GCN layers with 16 hidden units and tanh activations (Kipf & Welling, 2016). The embeddings from each layer are concatenated and fed into two fully connected heads with tanh and sigmoid activations, respectively, whose outputs of dimension 16 are then multiplied and fed into another two-layer MLP with the same dimension and ReLU activation functions (Nair & Hinton, 2010). We train separate networks for each protein target, predicting both the most likely regression label $\boldsymbol{\mu}_t(\mathbf{g}_t; \theta) = f_t^1(\mathbf{g}_t; \theta)$, as well as log-variances $\log \boldsymbol{\sigma}_t^2(\mathbf{g}_t; \theta) = f_t^2(\mathbf{g}_t; \theta)$ that serve as an estimator of their predictive uncertainty. Given the regression labels $\mathbf{y}$, the models are optimized with respect to a negative log-likelihood loss $\mathcal{L}_t = -\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2 \mathbf{I})$ Following the protocol of Lee et al. (2023), all models are trained for 10 epochs of stochastic gradient descent, using the Adam optimizer with a batch size of 1024 and a learning rate of 1×10^{-3} (Kingma & Ba, 2014). The corresponding objective $\mathcal{L}$ with an explicit L_2 regularization term is given by

$$\mathcal{L}(\theta, \mathcal{D}_{\mathcal{T}}) \doteq \mathbb{E}_{p_{\mathcal{T}}} \left[-\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathcal{T}}, \boldsymbol{\sigma}_{\mathcal{T}}^2 \mathbf{I}) \right] + \frac{1}{2\lambda} ||\theta||_2^2, \tag{B.19}$$

where the regularization strength λ is optimized as a hyperparameter. For weight decay regularization, we instead switch to the AdamW optimizer (Loshchilov & Hutter, 2017) and optimize the corresponding hyperparameter. Similarly, we train context-guided models by adding the additional regularization term $R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})$ as introduced in Section 3:

$$R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t}) = \mathbb{E}_{p_{\hat{\mathbf{X}}_t}} \left[\sum\nolimits_{j=1}^{2} \left(f_t^{(j)}(\hat{\mathbf{g}}_t; \theta) - m_t(\hat{\mathbf{g}}_t) \right)^{\top} K_t(\hat{\mathbf{g}}_t)^{-1} \left(f_t^{(j)}(\hat{\mathbf{g}}_t; \theta) - m_t(\hat{\mathbf{g}}_t) \right) \right], \tag{B.20}$$

Here, $\hat{\mathbf{g}}_t$ is a context batch sampled from an out-of-distribution context set of 500 000 unlabeled molecules that were randomly sampled from ZINC and processed identically to the training data. As described in Section 3, we use a randomly initialized set of parameters ϕ to generate 16-dimensional context set embeddings $h_t(\hat{\mathbf{g}}_t, \phi) \in \mathbb{R}^{M \times 16}$ and use them to construct the covariance matrix

$$K_t(\hat{\mathbf{g}}_t) = \sigma_t h_t(\hat{\mathbf{g}}_t, \phi) h_t(\hat{\mathbf{g}}_t, \phi)^T + \tau_t \mathbf{I}.$$

Since the model returns mean and log-variance estimates $f_t^1(\hat{\mathbf{g}}_t; \theta)$ and $f_t^2(\hat{\mathbf{g}}_t; \theta)$, the target function values $m_t(\hat{\mathbf{g}}_t)$ are given by the training set mean $m_t^1(\hat{\mathbf{g}}_t) = \text{mean}(\mathbf{y})$ and a log-variance hyperparameter $m_t^2(\hat{\mathbf{g}}_t) = \boldsymbol{\sigma}_0^2$, respectively. The latter is set to $\boldsymbol{\sigma}_0 = 0.7$, chosen to induce high predictive uncertainty estimates of $\exp(\boldsymbol{\sigma}^2) \approx 2$ in out-of-distribution regions of the input domain. The resulting training objective is

$$\mathcal{L}^*(\theta, \mathcal{D}_{\mathcal{T}}) = \mathbb{E}_{p_{\mathcal{T}}} \left[\mathcal{L}(\theta, \mathcal{D}_{\mathcal{T}}, \mathcal{T}) + R(\theta, f_{\mathcal{T}}, \mathcal{T}, p_{\hat{\mathbf{X}}_{\mathcal{T}}}) \right]. \tag{B.21}$$

The covariance scale σ_t and diagonal offset τ_t are optimized as hyperparameters. To avoid overconfident predictions as context points approach the invariant distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, we increase τ_t with the same schedule as the noise scales β_t, starting from $\beta_0 = \tau_t$ and ending at $\beta_T = 10\tau_t$. Similarly, as the distinction between in- and out-of-distribution becomes meaningless at larger noise levels, we decrease σ_t with an inverted schedule starting from $\beta_0 = \sigma_t$ and ending at $\beta_T = 0.1\sigma_t$.

The deep ensembles (Lakshminarayanan et al., 2017) comprise 5 independent models regularized with weight decay, which is optimized as a hyperparameter. The pre-trained model was optimized with a self-supervised denoising objective, which has been shown to be an effective pre-training method Zaidi et al. (2023). It was trained on all 500,000 unlabeled molecules from the context set for 50 epochs with the AdamW optimizer (Loshchilov & Hutter, 2017), using a batch size of 1024, a learning rate of 10^{-4}, and a weight decay of 10^{-4}. After pre-training, this model was fine-tuned on with weight decay.

All regularization hyperparameters were optimized with respect to the supervised loss $\mathcal{L}$ on the held-out, high-label test set by performing a grid search over the hyperparameter space presented in Table 3. The best hyperparameter combination was then selected and used to re-train five independent models using different random seeds.

Table 3: Hyperparameter search space for regularization schemes of regression models detailed in Appendix B.2.

Description	Hyperparameter	Search Space
L_2 regularization	λ	$10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}, 10^{4}$
Single model with weight decay	λ	$10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.5$
Fine-tuning with weight decay	λ	$10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.5$
Ensemble with weight decay	λ	$10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.5$
Context Guidance	covariance scale σ_t	$10^{-5}, 10^{-3}, 10^{-1}, 1, 10^{1}, 10^{3}$
	diagonal offset τ_t	$10^{-5}, 10^{-3}, 10^{-1}, 1, 10^{1}, 10^{3}$
	number of context points	$256, 512$

After retraining five independent guidance models with the best regularization hyperparameters for each of the five different proteins, we use the gradients from these models to condition the denoising process. Following Lee et al. (2023), we set the guidance strength hyperparameters to $r_{\mathbf{X},0} = 0.5$ and $r_{\mathbf{G},0} = 0$, respectively, and sample 3000 molecules each. The SA (Ertl, 2003) and QED (Bickerton et al., 2012) scores for these new molecules are derived with RDKIT (Landrum et al., 2013), while the docking score is derived with a QUICKVINA script (Alhossary et al., 2015) prepared by Lee et al. (2023).

Additional Experimental Results. We repeat the heasline results from Section 5.1 in Figure 10; following (Lee et al., 2023), we present the top 5% of normalized docking scores of compounds that are drug-like ($\mathbf{y}_{\text{SA}} < 5$ and $\mathbf{y}_{\text{QED}} > 0.5$) and novel (Tanimoto similarity of < 0.4 to the training set (Bajusz et al., 2015)). We also show the percentage of different compounds that would be classified as hits with the thresholds defined in Lee et al. (2023) in Figure 11. In addition to the main metrics from Lee et al. (2023), we also present the full distributions of:

- The objective $\mathbf{y} = \frac{\text{clip}(-\mathbf{y}_{\text{vina}}, 20, 0)}{20} \cdot \frac{10 - \mathbf{y}_{\text{SA}}}{9} \cdot \mathbf{y}_{\text{QED}}$ in Figure 12.
- The clipped and normalized negative QUICKVINA score $\frac{\text{clip}(-\mathbf{y}_{\text{vina}}, 20, 0)}{20}$ in Figure 13.
- The normalized synthetic accessibility (SA) score $\frac{10 - \mathbf{y}_{\text{SA}}}{9}$ in Figure 14.
- The quantitative estimate of drug-likeness (QED) score $\mathbf{y}_{\text{QED}}$ in Figure 15.

Furthermore, we normalize the scores of the generated compounds by their molecular weight and present the corresponding results in Figure 16, ensuring the performance differences are not caused by the bias of the docking score toward larger molecules. Finally, we show the same models trained on noisy labels $\bar{\mathbf{y}} = \mathbf{y} \cdot \epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ in Figure 17. Finally, the full distribution of models trained on an 85%-15% i.i.d. split is presented in Figure 18.

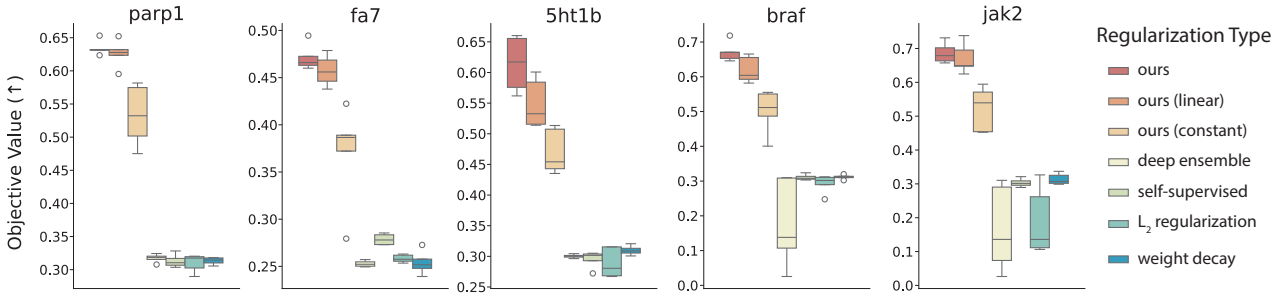


Figure 10: Comparison of the samples generated with different guided diffusion models across five distinct protein targets using the normalized top-5% docking score of novel and drug-like compounds as in Lee et al. (2023).

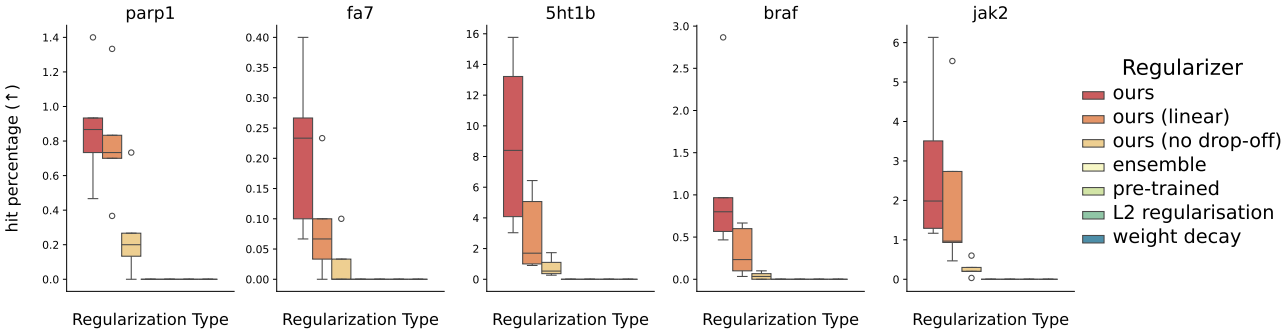


Figure 11: The percentage of small molecules denoted as hits using the thresholds from Lee et al. (2023), given by an unnormalized negative docking score of 10 for PARP1, 8.5 for FA7, 8.7845 for 5HT1B, 9.1 for JAK2, and 10.3 for BRAF.

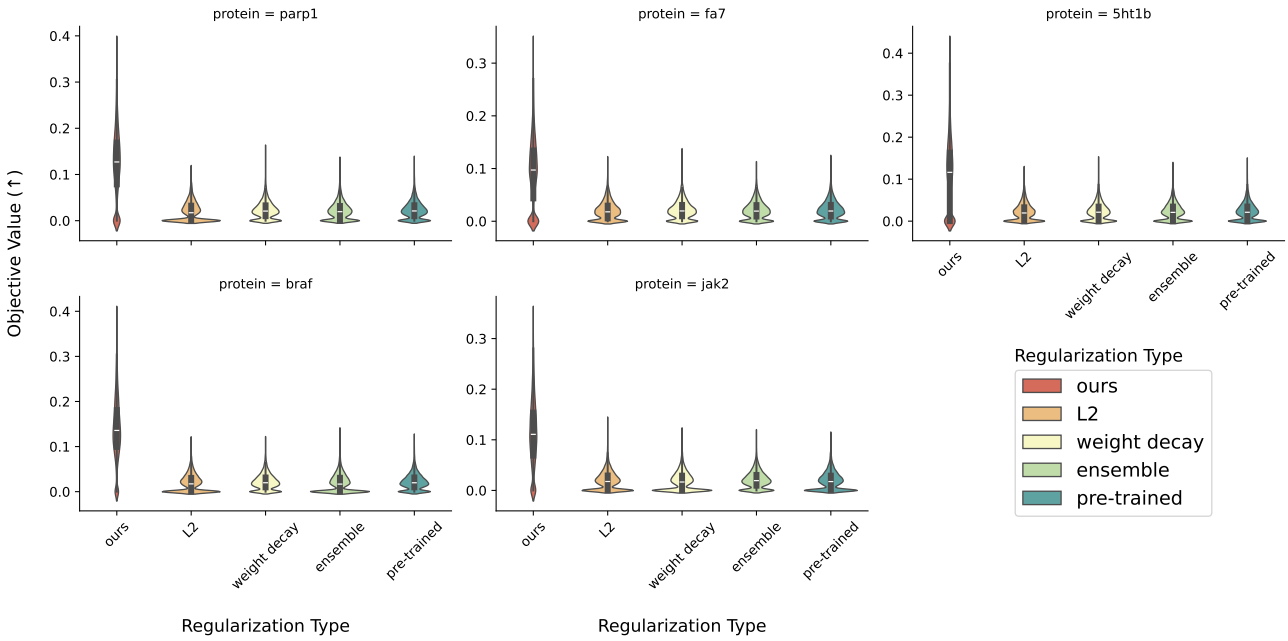


Figure 12: Full distributions of the objective values $\mathbf{y}$ of all generated compounds for each protein target across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

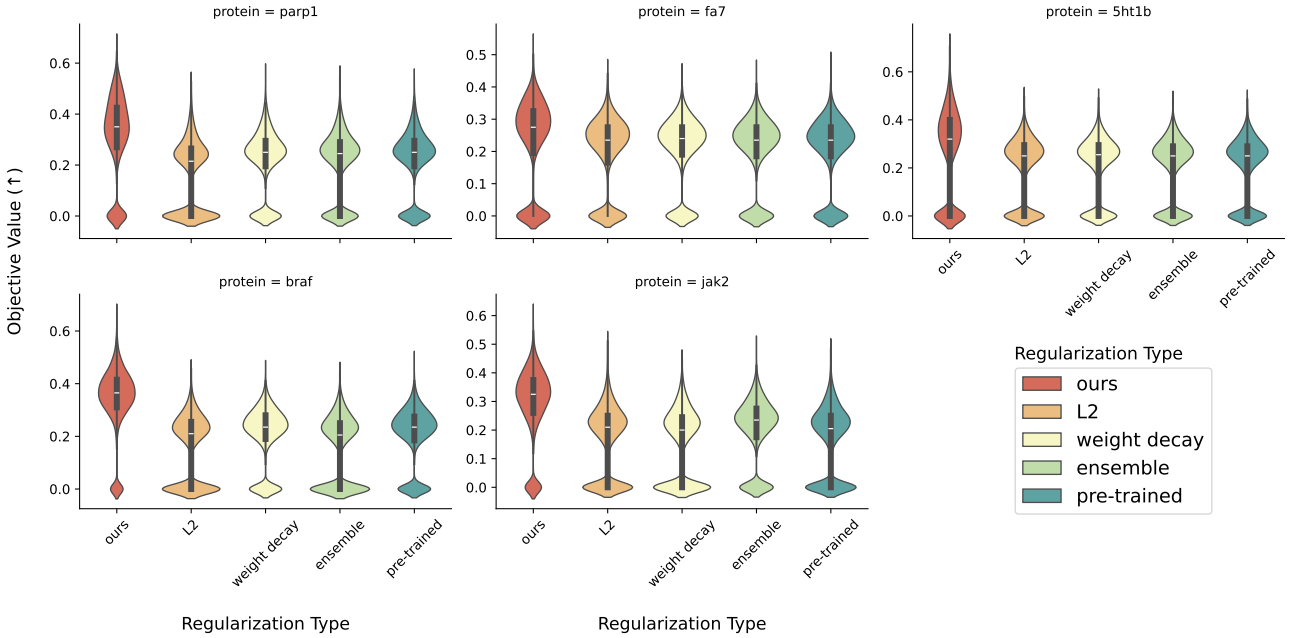


Figure 13: Full distributions of the clipped and normalized negative QUICKVINA docking score $\mathbf{y}_{\text{vina}}$ (Alhossary et al., 2015) of all generated compounds for each protein target across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

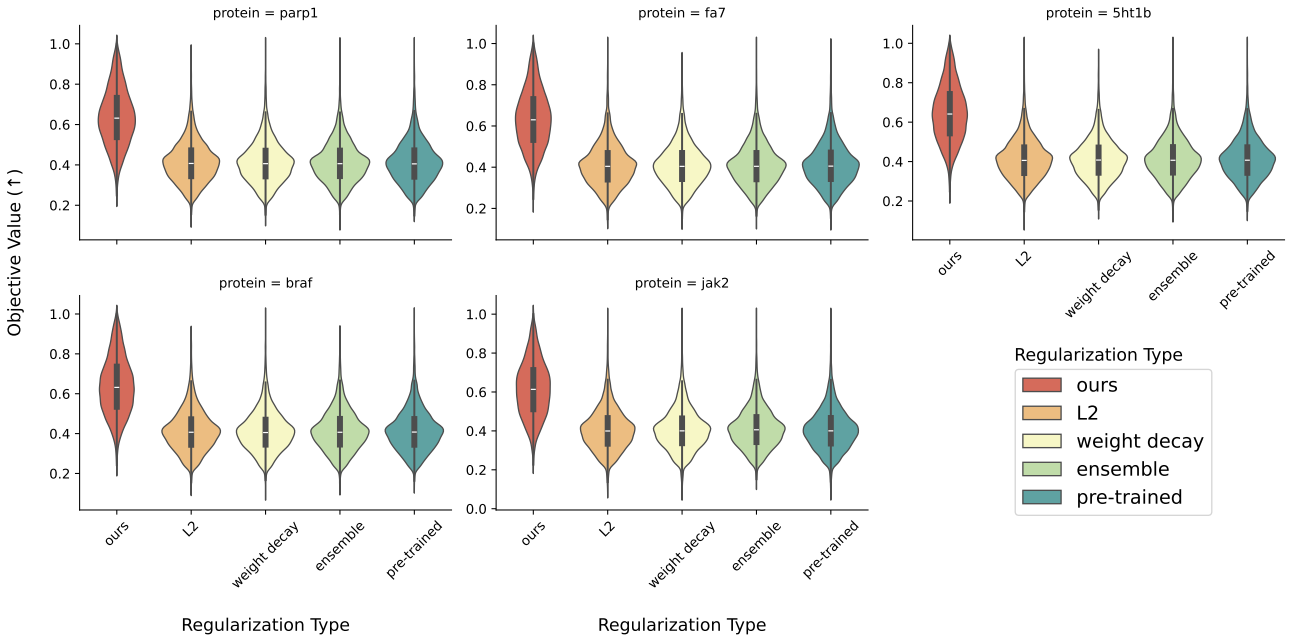


Figure 14: Full distributions of the normalized synthetic accessibility (SA) score $\mathbf{y}_{\text{SA}}$ (Ertl, 2003) of all generated compounds for each protein target across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

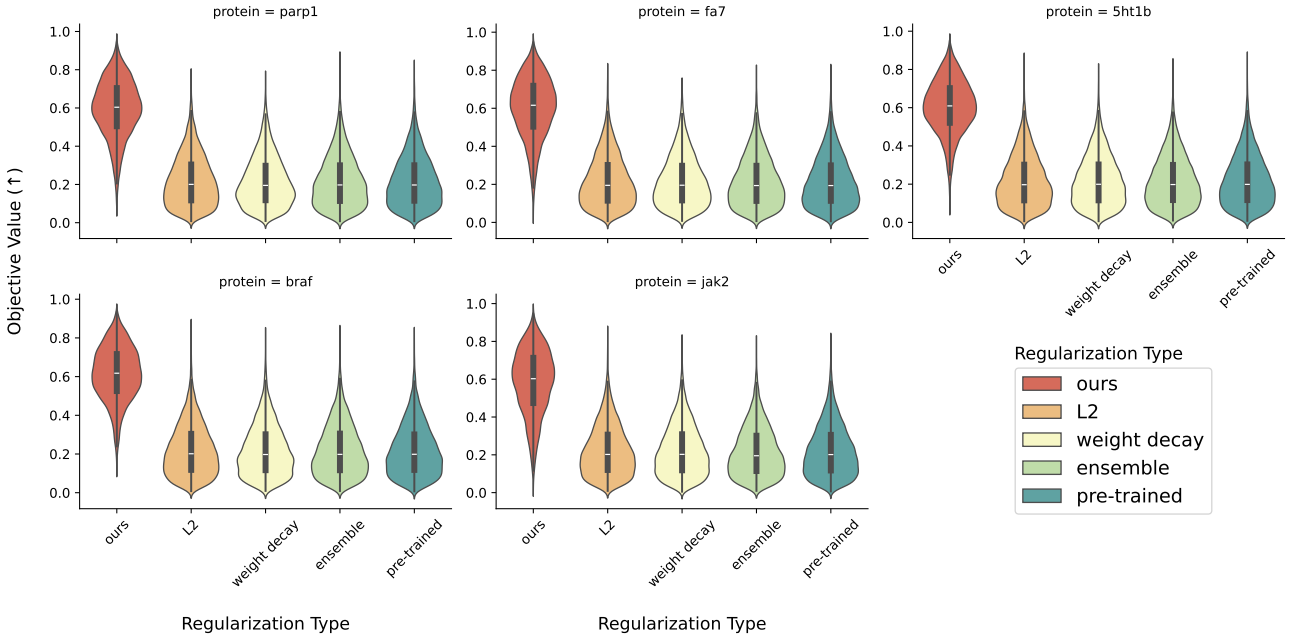


Figure 15: Full distributions of the quantitative estimate for drug-likeness (QED) score $\mathbf{y}_{\text{QED}}$ of all generated compounds for each protein target across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

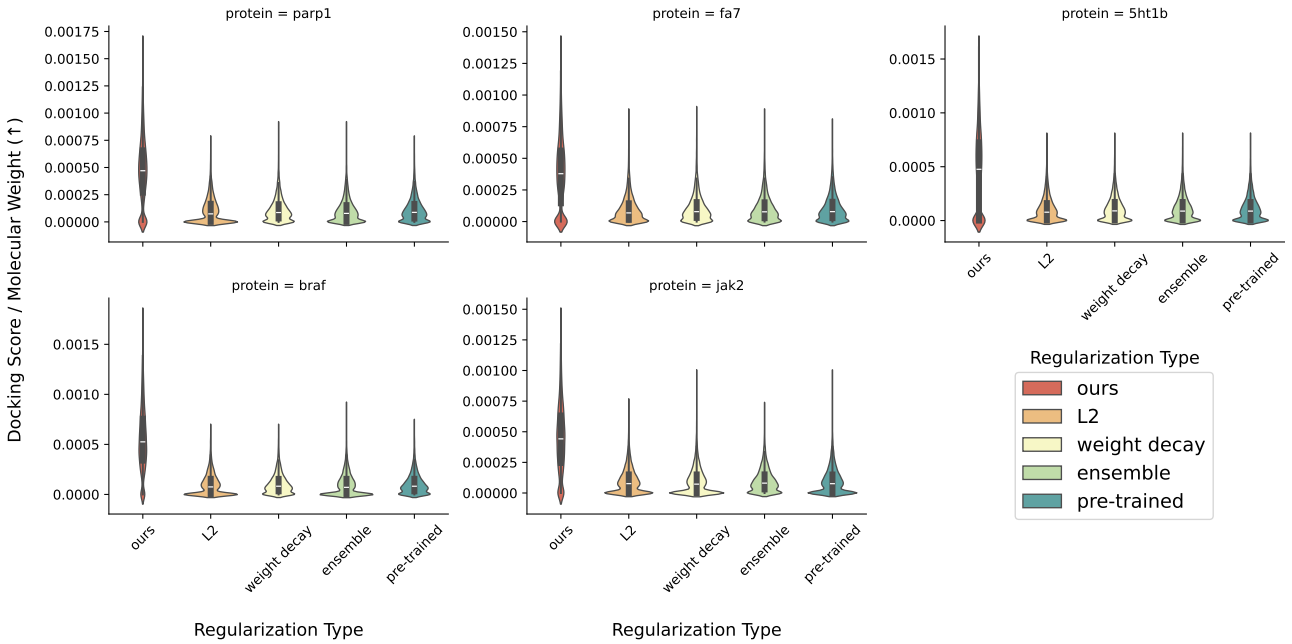


Figure 16: Full distributions of the objective values $\mathbf{y}$ of all generated compounds normalized by their molecular weight for each protein target across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

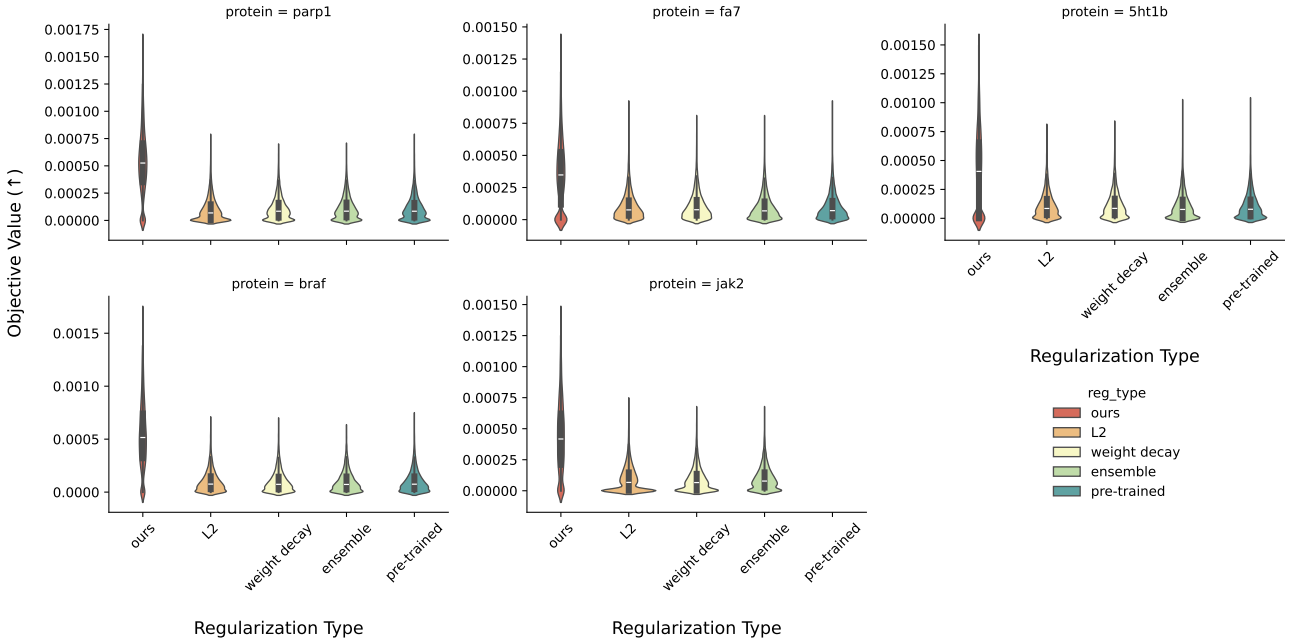


Figure 17: Full distributions of the objective values $\mathbf{y}$ of all generated compounds when models are trained on noised labels $\bar{y} = \mathbf{y} \cdot \epsilon$ with added noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

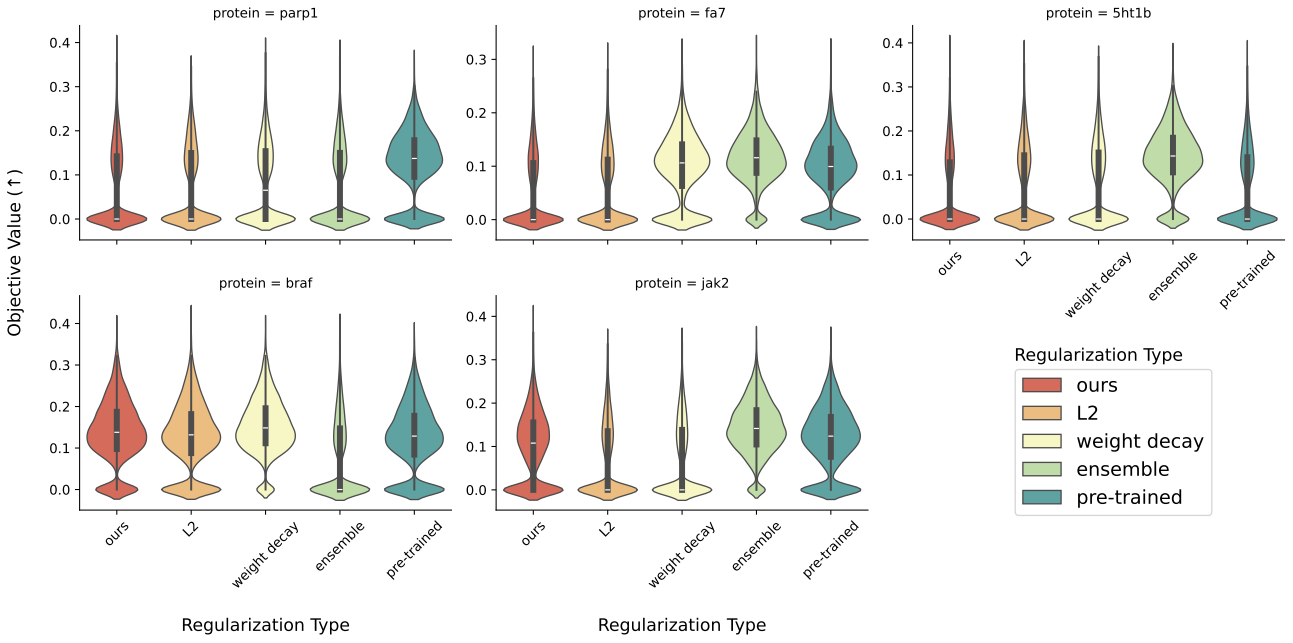


Figure 18: Full distributions of the objective values $\mathbf{y}$ of all generated compounds when guidance models that are trained on an 85%-15% i.i.d. random split, across five independent training and sampling runs with different random seeds. As expected, the deep ensembles and pre-trained methods perform particularly well in this setting. Interestingly, the performance of our domain-aware guidance method stays approximately the same between the random and label-split evaluation settings.

Comparison to Unsupervised Domain Adaptation Techniques. In addition to the models presented above, we also compare our method to guidance functions that are regularized with three robust and well-established unsupervised domain adaptation techniques, namely:

- DEEPCORAL: a method that aligns the feature distributions of the last-layer guidance model embeddings across the labeled data and context set through an auxiliary loss term (Sun & Saenko, 2016).
- Domain-Adversarial Neural Networks (DANN): a gradient-reversal approach that aligns the labeled data and context set embeddings by subtracting the gradients of a linear domain classifier from those of the guidance model (Ganin et al., 2016).
- DOMAIN CONFUSION: a method that tries to learn domain-invariant representations by maximizing the predictive entropy of an adversarially trained domain classifier (Tzeng et al., 2015).

We adapted the DEEPCORAL and DANN implementations from the DrugOOD library (Ji et al., 2023) and implemented the DOMAIN CONFUSION models from scratch, closely following the description in Tzeng et al. (2015). All models were evaluated with the same training, hyperparameter selection, and sample generation protocols as in Appendix B.2, using the domain adaptation hyperparameter search spaces presented in Table 4.

Table 4: Hyperparameter search space for the unsupervised domain adaptation algorithms.

Method	**Hyperparameter**	**Search Space**
DEEPCORAL	loss weight λ	$10^{-2}, 5\times10^{-2}, 10^{-1}, 5\times10^{-1}, 1, 5, 10^{1}, 5\times10^{1}, 10^{2}$
DANN	reversal factor α	$10^{-2}, 5\times10^{-3}, 10^{-3}, 5\times10^{-4}, 10^{-4}, 5\times10^{-5}$
DOMAIN CONFUSION	loss weight λ	$10^{-2}, 5\times10^{-2}, 10^{-1}, 5\times10^{-1}, 1, 5, 10^{1}, 5\times10^{1}, 10^{2}$

The resulting performance metrics for the first three protein targets from Lee et al. (2023) (PARP1, FA7, and 5HT1B) are presented in Table 5. While these methods consistently outperform the guidance models trained with standard approaches such as L_2 regularization and weight decay, they still perform considerably worse than our approach. One possible explanation for this performance gap is that the DEEPCORAL, DANN, and DOMAIN CONFUSION methods only act on the embedding space of a guidance model without considering the behavior of its predictions. In contrast, our context-guided approach directly regularizes the model's predictions, encouraging it to learn functions that revert to an uninformative prior in regions that are far from the training data. This allows our method to better capture the underlying structure of the data and improves its robustness in out-of-distribution settings.

Table 5: Performance comparison of our method against guidance models regularized with unsupervised domain adaptation techniques (DEEPCORAL, DANN, and DOMAIN CONFUSION) and standard regularization approaches (WEIGHT DECAY) on the first three protein targets from Lee et al. (2023) (PARP1, FA7, and 5HT1B). Our context-guided regularization scheme outperforms all domain adaptation techniques while maintaining comparable computational efficiency. Results are reported as the mean and standard deviation across five independent training and sampling runs with different random seeds. The best results for each target are shown in bold.

Regularizer	Average batch time (s)	PARP1 (↑)	FA7 (↑)	5HT1B (↑)
Ours	$0.08_{\pm 0.00}$	$\mathbf{0.64_{\pm 0.01}}$	$\mathbf{0.47_{\pm 0.00}}$	$\mathbf{0.60_{\pm 0.03}}$
DEEPCORAL	$0.08_{\pm 0.00}$	$0.36_{\pm 0.01}$	$0.32_{\pm 0.00}$	$0.35_{\pm 0.04}$
DANN	$0.07_{\pm 0.00}$	$0.37_{\pm 0.07}$	$0.27_{\pm 0.03}$	$0.34_{\pm 0.07}$
DOMAIN CONFUSION	$0.08_{\pm 0.00}$	$0.40_{\pm 0.01}$	$0.30_{\pm 0.01}$	$0.38_{\pm 0.01}$
WEIGHT DECAY	$0.02_{\pm 0.00}$	$0.31_{\pm 0.00}$	$0.26_{\pm 0.01}$	$0.31_{\pm 0.01}$

Context Set Sensitivity Studies. As the context set provides our method with structured information about a broader subset of the input domain, we found that its size and composition are key factors in determining model performance. To investigate this relationship, we carried out a series of sensitivity studies for the fist three protein targets from Lee et al. (2023) (PARP1, FA7, and 5HT1B). All experiments were performed with the same experimental setup, hyperparameter search space, and evaluation protocol as described in Appendix B.2, differing only in the choice of context set. The full results are presented in Table 6

First, we compare the performance of a model trained with the full context set (100%) to models trained with randomly sampled subsets of size 10% and 1%, respectively. Mirroring the results from Section 5.2, we observe a strong deterioration in performance across these settings, with the performance of the models trained with the smallest context set (1%) reverting to that of models trained with standard weight decay.

To explore whether an even larger and more diverse context set could lead to further performance gains, we trained an additional set of models using the 665k biologically and pharmaceutically relevant molecules from the QMUGS dataset (Isert et al., 2022), originally curated from the CHEMBL database (Gaulton et al., 2017). While we observe some performance improvements, they are relatively marginal, indicating that our original context set was likely already at or past the point of diminishing returns regarding its size.

Finally, we performed an additional sensitivity study for which we selected the 10% of molecules in the original context set that were either most or least similar to the labeled training data (as measured by their maximum ECFP4-based Tanimoto similarity, calculated with rdkit (Landrum et al., 2013)). We note that the average maximum Tanimoto similarity between these context sets and the training data is still relatively low, at 0.42 ± 0.06 and 0.31 ± 0.04, respectively. We observe that using a much smaller context set with near-OOD molecules (most similar 10%) is able to recover the performance of using the full context set (100%), while relying on less relevant molecules (least similar 10%) leads to worse performance than a random subsample (10%).

Table 6: Impact of the size and composition of the context set on the performance of our proposed regularizer for the small molecules application presented in Section 5.1. We compare the full context set (100%) to randomly sampled subsets (10% and 1%), as well as subsets of the original context set containing the 10% of molecules that are either most or least similar to the labeled training data. We additionally train models with an even larger and more diverse context set derived from QMUGS (Isert et al., 2022). Results are reported as the mean and standard deviation across five independent training and sampling runs with different random seeds.

Regularizer	parp1 (↑)	fa7 (↑)	5ht1b (↑)
ours (100%)	$0.64_{\pm 0.01}$	$0.47_{\pm 0.00}$	$0.60_{\pm 0.03}$
ours (10%)	$0.54_{\pm 0.01}$	$0.45_{\pm 0.00}$	$0.44_{\pm 0.04}$
ours (1%)	$0.37_{\pm 0.07}$	$0.27_{\pm 0.03}$	$0.30_{\pm 0.07}$
ours (QMUGS)	$0.69_{\pm 0.02}$	$0.47_{\pm 0.00}$	$0.62_{\pm 0.02}$
ours (most similar 10%)	$0.66_{\pm 0.01}$	$0.45_{\pm 0.01}$	$0.59_{\pm 0.01}$
ours (least similar 10%)	$0.51_{\pm 0.02}$	$0.32_{\pm 0.04}$	$0.42_{\pm 0.01}$
weight decay	$0.31_{\pm 0.00}$	$0.26_{\pm 0.01}$	$0.31_{\pm 0.01}$

B.3. Equivariant Diffusion for Materials.

Dataset. We build on the experimental setup from Weiss et al. (2023) and use a dataset of all 34071 cata-condensed polybenzenoid hydrocarbons (cc-PBHs) with up to 11 benzene rings (Wahab et al., 2022). In contrast to the setup in Section 5.1 and Appendix B.2, which uses a subset of the billions of compounds from the ZINC database (Tingle et al., 2023) to represent the broader chemical space of up to 1×10^{60} drug-like small molecules (Bohacek et al., 1996; Ertl, 2003), this application operates in a closed and fully enumerated input space. This allows us to construct data splits that precisely control how much generalization is required to reach held-out regions with optimal objective values.

Starting from the GFN2-xTB optimized ground-state cc-PBH structures, we adopt the graph-of-rings representation from Weiss et al. (2023), representing a compound with R rings as a point cloud of ring centroids $\mathbf{x}_t \in \mathbb{R}^{R\times 3}$. While computing electronic properties requires 3D representations, much of the combinatorial structure of this data persists in 2D topological graphs. We thus generate extended reduced graph fingerprints (Stiefl et al., 2006) in RDKIT (using a fuzziness increment of 0 and a maximum path length of 50), which correspond to a 2D equivalent of the 3D graph-of-rings framework. These fingerprints are binarized and compared via the Jaccard-Tanimoto kernel (Bajusz et al., 2015) to measure pairwise similarities. This enables us to cluster the dataset and derive maximally distinct training, validation, test, and context sets (Figure 19). Since the framework of Weiss et al. (2023) only generates compounds with a predetermined number of rings R and 75% of the dataset consists of molecules with $R = 11$, we split the data as follows: All compounds with $R < 11$ are used as part of an out-of-distribution context set. The remaining compounds are split into 3 groups using spectral clustering (scikit-learn) (Pedregosa et al., 2011). The largest cluster of 16 973 compounds is used as the training set. The second largest cluster of 4538 compounds is used as the held-out test set. The smallest cluster of 3883 compounds is used as the validation set for hyperparameter selection and early stopping. When training models with our context-guided regularization scheme, we use either all molecules with $R < 11$ as a *reduced* context set or all compounds not in the training set as a *full* context set

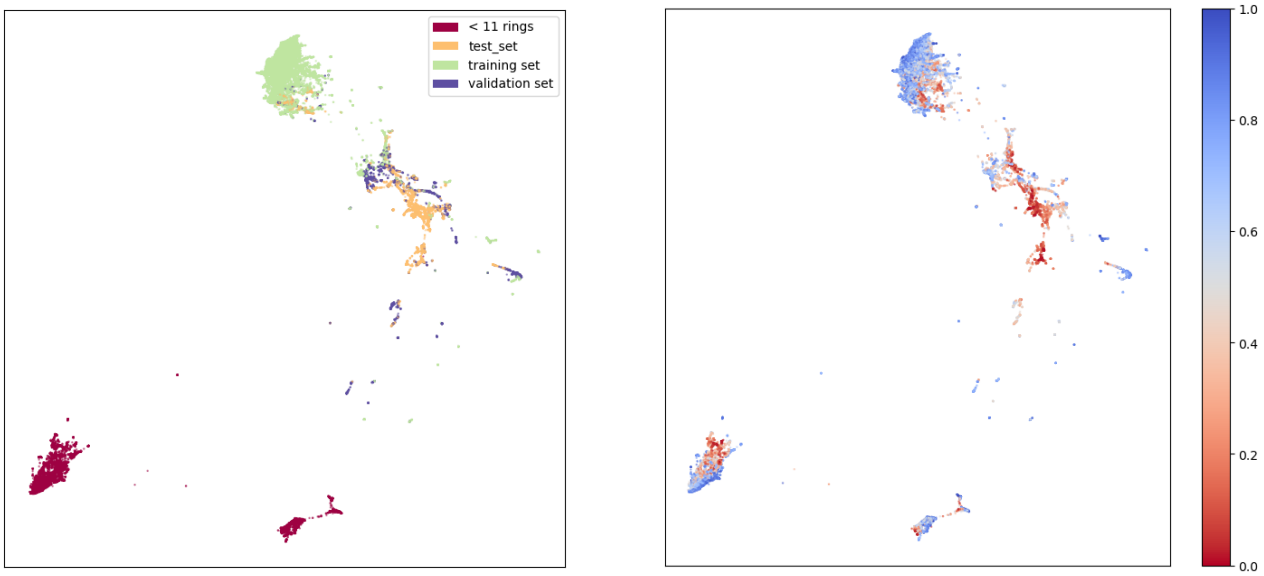


Figure 19: UMAP dimensionality reduction plots (McInnes et al., 2018) of the cc-PBH dataset using extended reduced graph fingerprints (Stiefl et al., 2006) and the Jaccard-Tanimoto distance. **Left**: Points are color-coded according to whether they belong to the out-of-distribution context set with fewer than 11 rings or the training, validation, and test set. **Right:** The same points, color-coded according to the composite objective used by Weiss et al. (2023) and defined as $\mathbf{y} = 3\,\mathbf{y}_{\text{HLG}} + \mathbf{y}_{\text{IP}} - \mathbf{y}_{\text{EA}}$. Lower values indicate more desirable properties and are, for $R = 11$, concentrated in the cluster used as a held-out test set.

Each molecule in the dataset used by Weiss et al. (2023) is annotated with five electronic properties: LUMO energy ($\mathbf{y}_{\text{LUMO}}$), HOMO-LUMO gap ($\mathbf{y}_{\text{HLG}}$), relative energy ($\mathbf{y}_{\text{REL}}$), adiabatic ionization potential ($\mathbf{y}_{\text{IP}}$), and adiabatic electron affinity ($\mathbf{y}_{\text{EA}}$) (Wahab et al., 2022). The full property distributions for each data split are shown in Figure 20. Conveniently, the held-out test set cluster contains compounds with significantly better $\mathbf{y}_{\text{HLG}}$, $\mathbf{y}_{\text{IP}}$, and $\mathbf{y}_{\text{EA}}$ values than the training set, which are the properties used to define the composite objective in Weiss et al. (2023), as is also apparent in Figure 19.

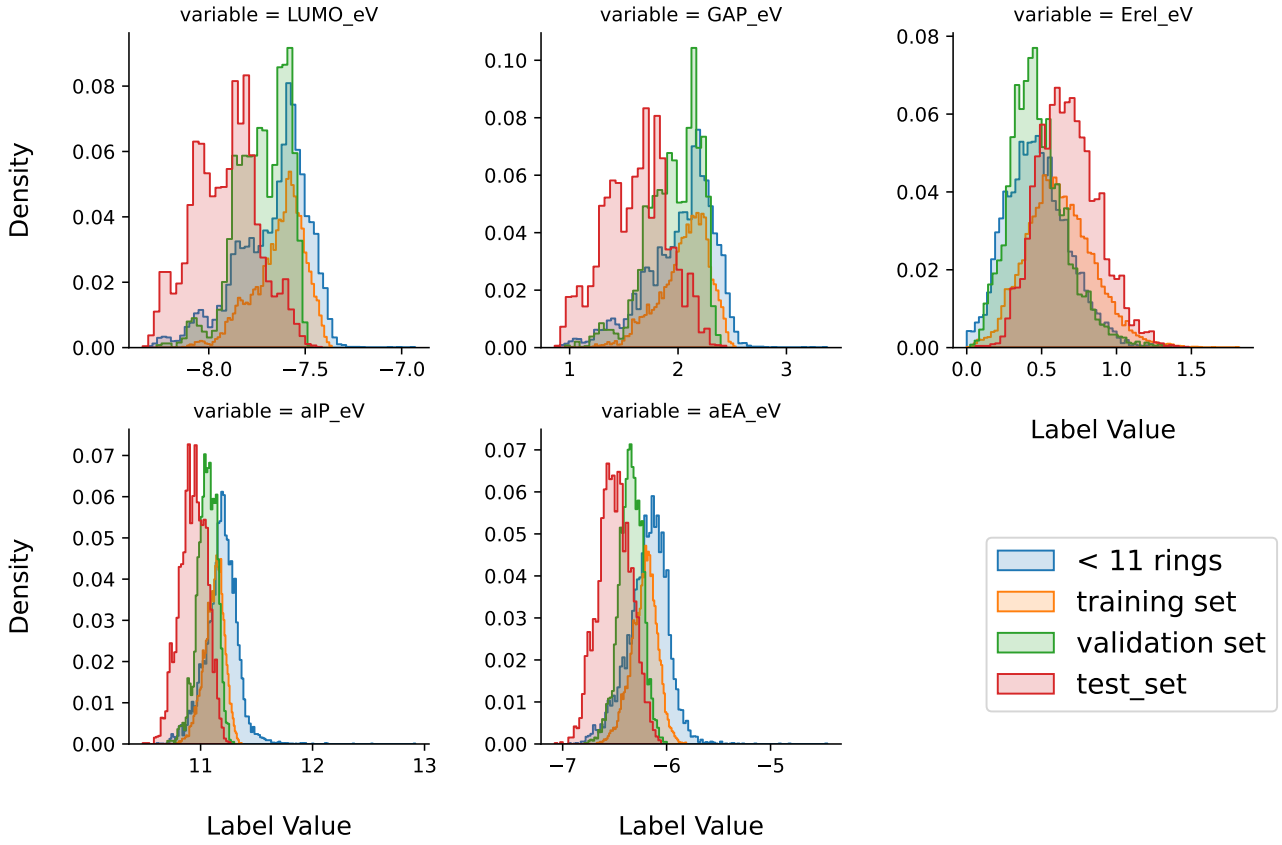


Figure 20: The full electronic property distributions for each data split.

Diffusion Model Training. Following Weiss et al. (2023), we train an E(3)-equivariant diffusion model (EDM; Hoogeboom et al. (2022)) on the graph-of-rings-derived point clouds $\mathbf{x}_t \in \mathbb{R}^{R\times 3}$. Both the forward noising and reverse denoising process outlined in Section 2 are made equivariant to permutations, translations, rotations, and reflections by using an E(3)-equivariant graph neural network (E(3)-GNN) (Satorras et al., 2021) as the score model and redefining a translation-invariant noising process in the linear subspace of $\sum_i \mathbf{x}_i = 0$ (Hoogeboom et al., 2022).

We optimized the resulting equivariant diffusion model on the 16 973 compounds in the training set using the same hyperparameter settings as in Weiss et al. (2023). Specifically, the score network $s_\theta(\mathbf{x}_t, t)$ is given by a 9-layer E(3)-GNN with 192 hidden units and $\tanh$ activations. It is trained for 1000 epochs with the ADAM optimizer (Kingma & Ba, 2014) using a learning rate of 1×10^{-3}, a batch size of 256, and a small weight decay hyperparameter of 10^{-12}. The diffusion process is given by a cosine β_t schedule (Nichol & Dhariwal, 2021) with $T = 1000$ time-steps, and $\beta_0 = 0$ and $\beta_T = 1$. Due to the reduced problem space of the graph-of-rings representation, we find that the vast majority of generated point clouds ($\geq 99\%$, as in Weiss et al. (2023)) correspond to valid molecules and can be parsed by RDKIT (Landrum et al., 2013).

Guidance Model Training. Similar to the score network, the guidance model uses a 12-layer E(3)-GNN architecture with 192 hidden units and $\tanh$ activations, matching the model hyperparameters from Weiss et al. (2023). These models are trained using the ADAM optimizer (Kingma & Ba, 2014) for up to 1000 epochs with a learning rate of 6×10^{-4} and batch size of 256, stopping if the validation set performance deteriorates for more than 10 consecutive epochs.

As in Weiss et al. (2023), we train these models as multi-property prediction networks, outputting both the regression mean $\boldsymbol{\mu}_t(\hat{\mathbf{x}}_t; \theta) = f_t^1(\hat{\mathbf{x}}_t; \theta)$ and log-variance $\log \boldsymbol{\sigma}_t^2(\hat{\mathbf{x}}_t; \theta) = f_t^2(\hat{\mathbf{x}}_t; \theta)$ for all five properties $\mathbf{y}_{\text{HLG}}$, $\mathbf{y}_{\text{IP}}$, $\mathbf{y}_{\text{LUMO}}$, $\mathbf{y}_{\text{REL}}$, and $\mathbf{y}_{\text{EA}}$. Given the true regression labels $\mathbf{y}$, the models are trained by minimizing to a negative log-likelihood loss across these five properties $\mathcal{L}_t = -\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2 \mathbf{I})$.

The corresponding objective $\mathcal{L}$ with an explicit L_2-regularization term is given by

$$\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}) \doteq \mathbb{E}_{p_\mathcal{T}}\left[-\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_\mathcal{T}, \boldsymbol{\sigma}_\mathcal{T}^2 \mathbf{I})\right] + \frac{1}{2\lambda}||\theta||_2^2, \tag{B.22}$$

where the regularization strength λ is optimized as a hyperparameter. Similarly, models are regularized with weight decay by switching to the AdamW optimizer (Loshchilov & Hutter, 2017) and optimizing the corresponding hyperparameter. Context-guided models are trained with the regularization term $R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})$ as introduced in Section 3

$$R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t}) = \mathbb{E}_{p_{\hat{\mathbf{X}}_t}}\left[\sum\nolimits_{j=1}^{2}\left(f_t^{(j)}(\hat{\mathbf{x}}_t;\theta) - m_t(\hat{\mathbf{x}}_t)\right)^\top K_t(\hat{\mathbf{x}}_t)^{-1}\left(f_t^{(j)}(\hat{\mathbf{x}}_t;\theta) - m_t(\hat{\mathbf{x}}_t)\right)\right], \tag{B.23}$$

where $\hat{\mathbf{x}}_t \sim p_{\hat{\mathbf{X}}_t}(\hat{\mathbf{x}}_t)$ is a context batch sampled from either the *full* context set covering all compounds not in the training set, or the *reduced* context set containing only compounds with fewer than 11 rings (see Figure 19). As described in Section 3, we use a randomly initialized parameter set ϕ to generate 196-dimensional context set embeddings $h_t(\hat{\mathbf{x}}_t, \phi) \in \mathbb{R}^{M\times 196}$ from the E(3)-GNN's last layer and use them to construct the covariance matrix

$$K_t(\hat{\mathbf{x}}_t) = \sigma_t h_t(\hat{\mathbf{x}}_t, \phi) h_t(\hat{\mathbf{x}}_t, \phi)^T + \tau_t \mathbf{I}.$$

Since the model predicts means and log-variances $f_t^1(\hat{\mathbf{x}}_t;\theta)$ and $f_t^2(\hat{\mathbf{x}}_t;\theta)$, the target function values $m_t(\hat{\mathbf{x}}_t)$ are given by the training set mean and a log-variance hyperparameter $m_t^2(\hat{\mathbf{x}}_t) = \boldsymbol{\sigma}_0^2$, respectively. The latter is set to $\boldsymbol{\sigma}_0 = 1$, chosen to induce high predictive uncertainty estimates of $\exp(\boldsymbol{\sigma}^2) \approx e$ in out-of-distribution regions of the input domain. The resulting training objective is

$$\mathcal{L}^*(\theta, \mathcal{D}) = \mathbb{E}_{p_\mathcal{T}}\left[\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}, \mathcal{T}) + R(\theta, f_\mathcal{T}, \mathcal{T}, p_{\hat{\mathbf{X}}_\mathcal{T}})\right]. \tag{B.24}$$

The covariance scale σ_t and diagonal offset τ_t are optimized as hyperparameters. To avoid overconfident predictions as context points approach the invariant distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, we increase τ_t with the same schedule as the noise scales β_t, starting from $\beta_0 = \tau_t$ and ending at $\beta_T = 10\tau_t$. Similarly, as the distinction between in- and out-of-distribution becomes meaningless at larger noise levels, we decrease σ_t with an inverted schedule starting from $\beta_0 = \sigma_t$ and ending at $\beta_T = 0.1\sigma_t$. The regularization hyperparameters are optimized via grid search on the held-out validation set, using the supervised loss $\mathcal{L}$ as the performance metric. The grid search was over the hyperparameter space shown in Table 7. The best hyperparameter combination was selected and used to re-train ten independent models with different random seeds.

Table 7: Hyperparameter search space for regularization schemes of regression models detailed in Appendix B.3.

Description	Hyperparameter	Search Space
L_2 regularization	λ	$10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^{-3}$
Weight decay	λ	$10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.5$
Context Guidance	covariance scale σ_t	$10^{-2}, 10^{-1}, 1, 10^1, 10^{-2}$
	diagonal offset τ_t	$10^{-2}, 10^{-1}, 1, 10^1, 10^{-2}$
	number of context points	16, 64, 256

After retraining ten independent guidance models for each regularization type, their predictions for the HOMO-LUMO gap ($\mathbf{y}_{\text{HLG}}$), adiabatic ionization potential ($\mathbf{y}_{\text{IP}}$), and adiabatic electron affinity ($\mathbf{y}_{\text{EA}}$) properties were used to define the composite objective $\mathbf{y} = 3\,\mathbf{y}_{\text{HLG}} + \mathbf{y}_{\text{IP}} - \mathbf{y}_{\text{EA}}$. The resulting gradients were then used to guide the generation of 512 samples each, using an additional guidance scale s

$$\nabla_\mathbf{X} \log p_t(\mathbf{x}_t \mid \mathbf{y}) = \nabla_\mathbf{X} \log p_t(\mathbf{x}_t) + \nabla_\mathbf{X} \log p_t(\mathbf{y} \mid \mathbf{x}_t)^s = s_\theta(\mathbf{x}_t, t) + s \cdot \nabla_\mathbf{X} \log f_t(\mathbf{x}_t;\theta)$$

Additional Experimental Results. In addition to the objective values in Figure 5, we also calculate the proportion of generated samples that are (1) valid and novel (i.e. not in the training set) in Figure 21 and (2) valid and unique (i.e. only generated once) in Figure 22. Together with Figure 5, the results suggest that as guidance strength increases, our method generates more novel molecules with better objective values, yet less diversity. This may be explained by stronger guidance signals causing the denoising process to converge to the same high-value subsets of the input domain.

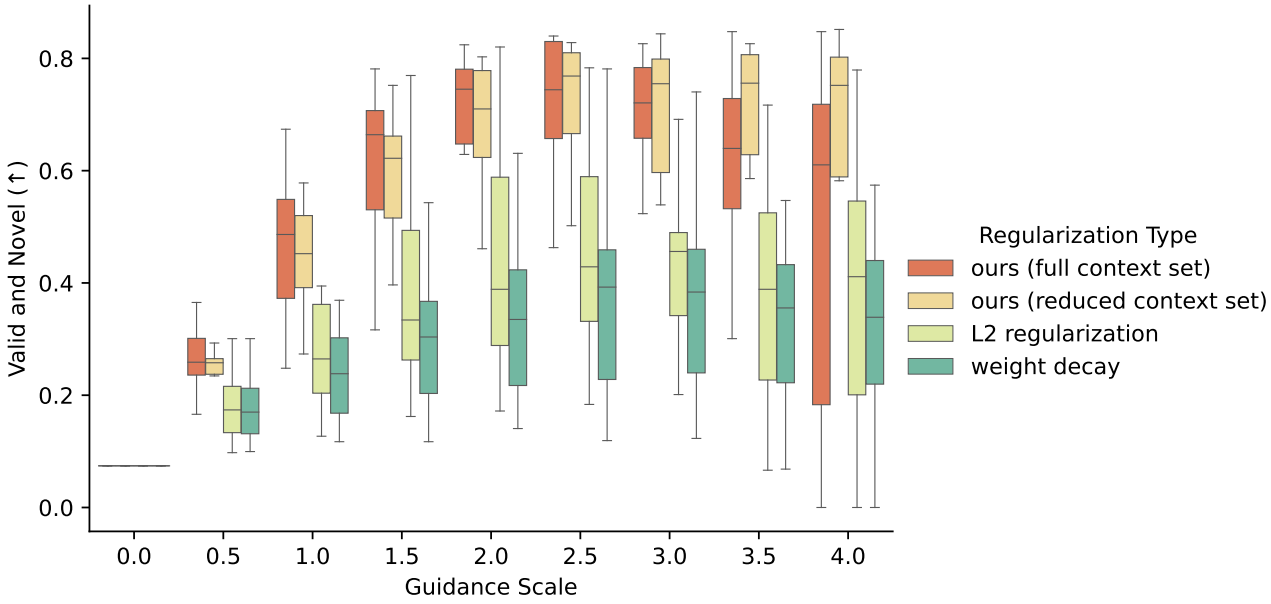


Figure 21: Performance comparison of guidance models with different regularization schemes across ten independent training and sampling runs. The proportion of compounds that are both valid and novel (i.e. not in the training set) is computed for each run and aggregated across random seeds.

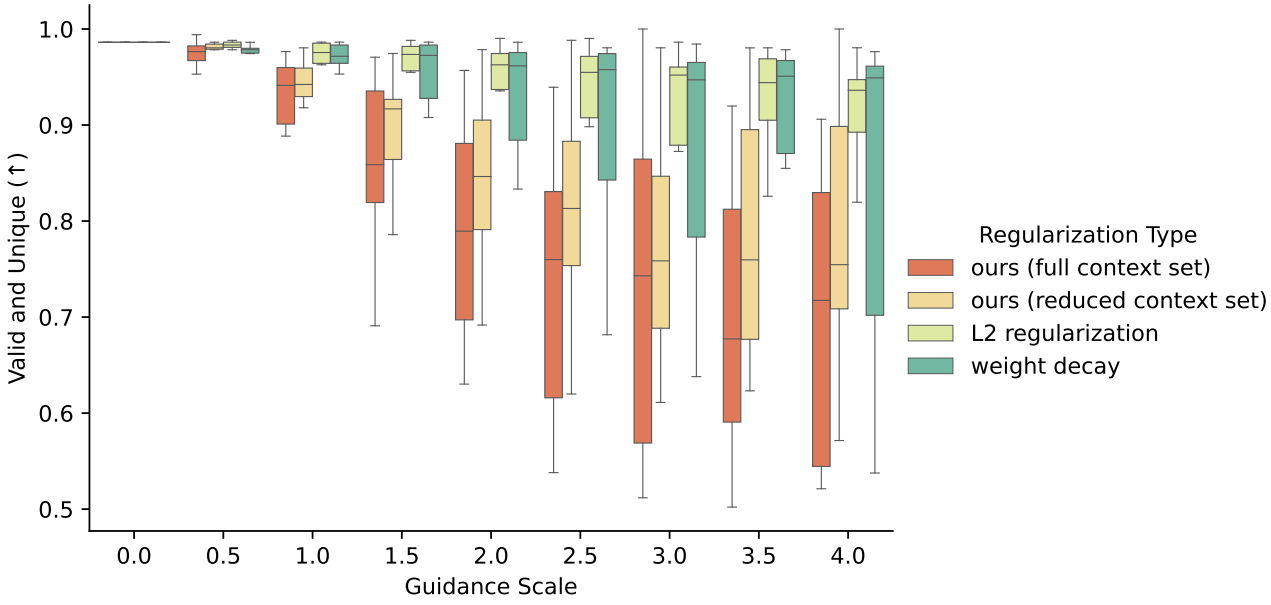


Figure 22: Performance comparison of guidance models with different regularization schemes across ten independent training and sampling runs. The proportion of compounds that are both valid and unique (i.e. only generated once) is computed for each run and aggregated across random seeds.

B.4. Discrete Diffusion for Protein Sequences.

Dataset. We use the same antibody sequence dataset as Gruver et al. (2023), consisting of approximately 100 000 paired heavy and light chain sequences from the OAS and SAbDab databases (Olsen et al., 2022; Dunbar et al., 2014) aligned with ANARCI (Dunbar & Deane, 2016) and padded to length 300. We focus on the synthetic-label applications presented by Gruver et al. (2023), namely the single-objective optimization of a protein's solvent-accessible surface area (SASA) and percentage of beta sheets. In the following experiments we optimize the latter, using code from Gruver et al. (2023) and the BioPython library (Cock et al., 2009) to derive the labels $\mathbf{y}$. Similar to Appendices B.2 and B.3, we split the data into a low-property training set and a high-property validation set to optimize for hyperparameters that perform well in label-shifted, out-of-distribution settings. Additionally, a random subsample of a third of the dataset is split off as a held-out unlabeled context set for our context-guided method, ordering the remaining two-thirds by their labels $\mathbf{y}$ and splitting them in half to generate the low-value training and high-value validation sets.

Diffusion and Guidance Model Training. In contrast to Sections 5.1 and 5.2, the guidance model of Gruver et al. (2023) is not a standalone model, but rather a regression head of the diffusion model's score network, trained jointly with the denoising objective. Specifically, a masked language model (Austin et al., 2021) learns sequence embeddings, from which two MLPs estimate the score and properties. Following Gruver et al. (2023), we use a BERT-based transformer backbone (Bhargava et al., 2021; Devlin et al., 2018) with 4 layers and 512 hidden units. The score head is a single linear layer, while the property prediction head is a feedforward network with one 512-unit hidden layer and $\tanh$ activations. This model is trained for 500 epochs with AdamW (Loshchilov & Hutter, 2017), using a batch size of 128 and a learning rate of 5×10^{-4} with a 10-step linear warmup schedule. The noising process is given by a discrete formulation from Gruver et al. (2023), using cosine β_t schedule (Nichol & Dhariwal, 2021) with $T = 1000$ time-steps, and $\beta_0 = 0$ and $\beta_T = 1$.

We train the regression head to estimate both the regression mean $\boldsymbol{\mu}_t(\hat{\mathbf{x}}_t; \theta) = f_t^1(\hat{\mathbf{x}}_t; \theta)$ and log-variance $\log \boldsymbol{\sigma}_t^2(\hat{\mathbf{x}}_t; \theta) = f_t^2(\hat{\mathbf{x}}_t; \theta)$. Given the true regression labels $\mathbf{y}$, the property prediction model is optimized with respect to the corresponding negative log-likelihood loss:

$$\mathcal{L} = -\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathcal{T}}, \boldsymbol{\sigma}_{\mathcal{T}}^2 \mathbf{I})$$

Following (Gruver et al., 2023), we take one gradient step with respect to $\mathcal{L}$ for every 5 gradient steps with respect to the score matching objective, logging the validation loss every 5 epochs and loading the best checkpoint for sample generation. We train L_2-regularized models by augmenting the supervised objective $\mathcal{L}$ with an explicit L_2 penalty

$$\mathcal{L}(\theta, \mathcal{D}_{\mathcal{T}}) \doteq \mathbb{E}_{p_{\mathcal{T}}} \left[-\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathcal{T}}, \boldsymbol{\sigma}_{\mathcal{T}}^2 \mathbf{I}) \right] + \frac{1}{2\lambda} ||\theta||_2^2, \tag{B.25}$$

where the regularization strength λ is optimized as a hyperparameter. Similarly, models are regularized with weight decay by switching to the AdamW optimizer (Loshchilov & Hutter, 2017) and optimizing the corresponding hyperparameter.

Context-guided models are trained by adding the additional regularization term $R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})$ given by

$$R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t}) = \mathbb{E}_{p_{\hat{\mathbf{X}}_t}} \left[\sum_{j=1}^{2} \left(f_t^{(j)}(\hat{\mathbf{x}}_t; \theta) - m_t(\hat{\mathbf{x}}_t) \right)^\top K_t(\hat{\mathbf{x}}_t)^{-1} \left(f_t^{(j)}(\hat{\mathbf{x}}_t; \theta) - m_t(\hat{\mathbf{x}}_t) \right) \right], \tag{B.26}$$

where $\hat{\mathbf{x}}_t \sim p_{\hat{\mathbf{X}}_t}(\hat{\mathbf{x}}_t)$ is a context batch sampled from the unlabeled context set split off as detailed before. As described in Section 3, we use a randomly initialized parameter set ϕ to generate 512-dimensional context set embeddings $h_t(\hat{\mathbf{x}}_t, \phi) \in \mathbb{R}^{M \times 512}$ from the last layer of the regression head and use them to construct the covariance matrix

$$K_t(\hat{\mathbf{x}}_t) = \sigma_t h_t(\hat{\mathbf{x}}_t, \phi) h_t(\hat{\mathbf{x}}_t, \phi)^T + \tau_t \mathbf{I}.$$

Since the model predicts mean and log-variance estimates $f_t^1(\hat{\mathbf{x}}_t; \theta)$ and $f_t^2(\hat{\mathbf{x}}_t; \theta)$, the target function values $m_t(\hat{\mathbf{x}}_t)$ are given by the training set mean and a log-variance hyperparameter $m_t^2(\hat{\mathbf{x}}_t) = \boldsymbol{\sigma}_0^2$, respectively. The latter is set to $\boldsymbol{\sigma}_0 = 1$, chosen to induce high predictive uncertainty estimates of $\exp(\boldsymbol{\sigma}^2) \approx e$ in out-of-distribution regions of the input domain. The resulting training objective is

$$\mathcal{L}^*(\theta, \mathcal{D}) = \mathbb{E}_{p_{\mathcal{T}}} \left[\mathcal{L}(\theta, \mathcal{D}_{\mathcal{T}}, \mathcal{T}) + R(\theta, f_{\mathcal{T}}, \mathcal{T}, p_{\hat{\mathbf{X}}_{\mathcal{T}}}) \right]. \tag{B.27}$$

The covariance scale σ_t and diagonal offset τ_t are optimized as hyperparameters. To avoid overconfident predictions as context points approach the invariant distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, we increase τ_t with the same schedule as the noise scales β_t,

starting from $\beta_0 = \tau_t$ and ending at $\beta_T = 10\tau_t$. Similarly, as the distinction between in- and out-of-distribution becomes meaningless at larger noise levels, we decrease σ_t with an inverted schedule starting from $\beta_0 = \sigma_t$ and ending at $\beta_T = 0.1\sigma_t$.

At sampling time, Gruver et al. (2023) generate sequences using a grid of denoising hyperparameters and examine the resulting Pareto front of objective values versus "naturalness" i.e. how likely a sequence is to be synthesizable, estimated by its likelihood under a protein language model (Ferruz et al., 2022). We use same hyperparameter grid (presented in Table 8), generating samples with five independently trained models with different random seeds.

Table 8: Hyperparameter search space for regularization schemes of regression models detailed in Appendix B.4.

Hyperparameter	Description	Sampling Range
λ	strength of naturalness regularization	$10^1, 1, 10^{-1}, 10^{-2}, 10^{-3}$
η	step size of Langevin dynamics update	1.0
K	number of steps used to update embedding	10
guidance layer	which embedding to use for update	first
return best	return samples with best guidance model predictions	true

Additional Experimental Results. In addition to the Pareto front of the generated samples shown in Figure 6, we follow Gruver et al. (2023) and also visualize the full distribution of generated samples by plotting their kernel density estimates. The results are presented in Figure 23, mirroring the conclusions drawn from the analysis of the Pareto front.

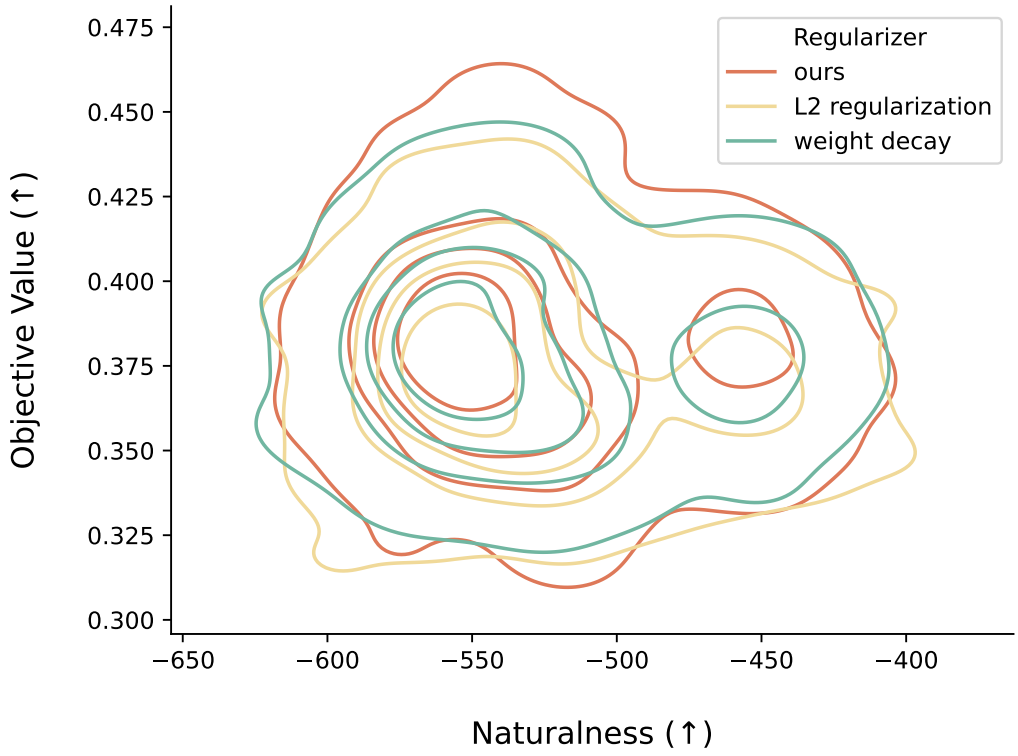


Figure 23: Kernel density estimates (KDE) of all samples generated with different regularization schemes and sampling hyperparameters, using the standard KDE parameters of the SEABORN plotting library (Waskom, 2021) and five density isocontours each. Similar to the Pareto front, this highlights the trade-off between objective value and naturalness. As samples move away from the training data and enter an out-of-distribution regime, our method consistently generates sequences with better properties at the same level of naturalness.

B.5. Runtime Comparison.

We quantify computational cost as the average execution time per mini-batch iteration, including data loading, model forward passes, backpropagation, and parameter updates. Figure 24 compares regularization schemes for the application in Section 5.1 across 5 model training runs. Figure 25 shows the same comparison for the experiments in Section 5.3. All models were trained with an identical setup and on the same NVIDIA A100 GPUs, differing only in the regularizer.

When training a standard guidance model, as in Lee et al. (2023), we observe that the added computational cost at training time is roughly equivalent to training a deep ensemble with $M = 5$ models (Figure 24). However, we note that our method does not induce any additional cost at inference, i.e., sampling time. When training the score and guidance models jointly and only updating the latter every 5 steps, as in Gruver et al. (2023), we observe that the increase in computational cost becomes negligible (Figure 25).

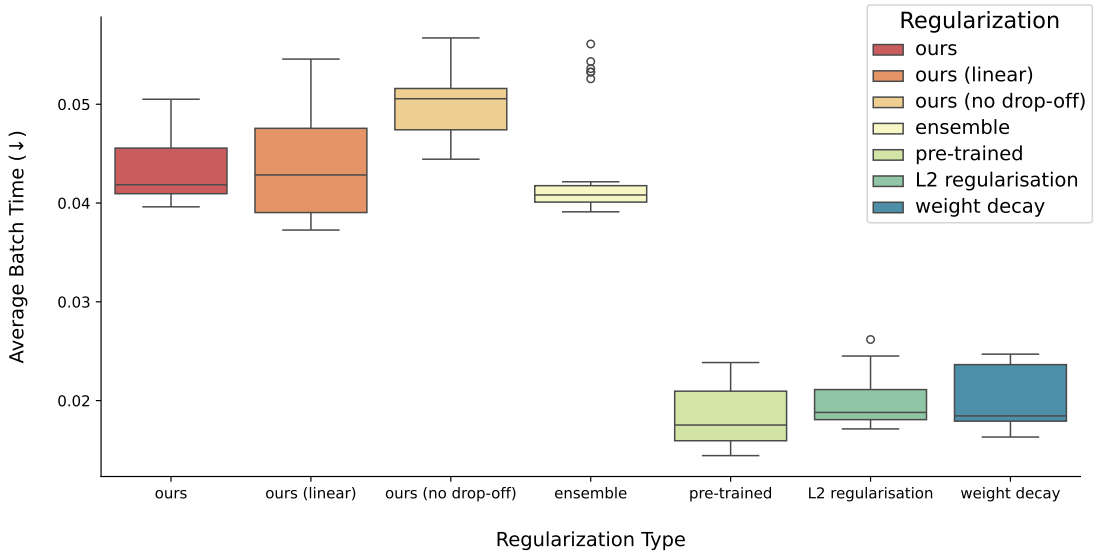


Figure 24: Average batch times of models from Section 5.1 trained with an identical setup and on the same GPUs.

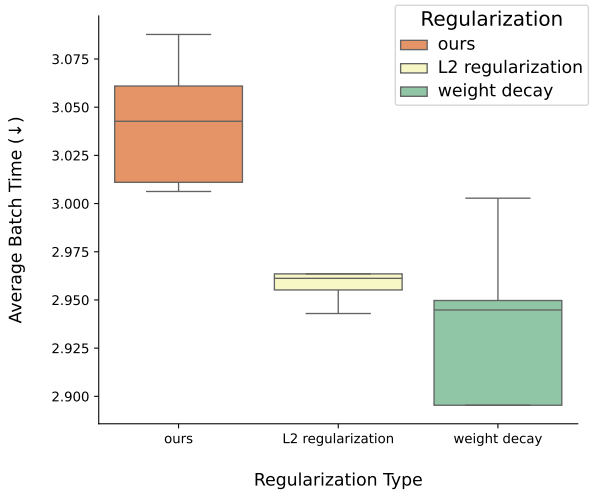


Figure 25: Average batch times of models from Section 5.3 trained with an identical setup and on the same GPUs.

B.6. Similarity of the Generated Samples to the Training and Validation Sets.

As our experimental setups select regularization hyperparameters based on an out-of-distribution validation set with more desirable properties than the training data, it is important to verify that the models do not simply generate molecules from the high-value validation set. The results are presented below and demonstrate that guidance model trained with our regularizer do not simple generate molecules from the validation set, but are instead learning to sample molecules with improved properties from novel subsets of the input domain.

B.6.1. Graph-Structured Diffusion for Small Molecules

To investigate this, we computed the maximum ECFP4-based (Rogers & Hahn, 2010) Jaccard-Tanimoto similarity of the molecules generated by our model in Section 5.1 to any compound in the training and validation sets, respectively. The results are provided in Table 9 and show that the similarities are consistently low and comparable across the training and validation sets (a threshold of 0.7 is often used to indicate molecules as closely related).

Table 9: Maximum extended-connectivity fingerprint (ECFP4; Rogers & Hahn (2010))-based Tanimoto similarity (Bajusz et al., 2015) of the small molecules generated in Section 5.1 to compounds in the training and validation sets. Means and standard deviations are reported across five independent training and sampling runs with different random seeds.

Set	PARP1	FA7	5HT1B	BRAF	JAK2
Training Set	0.33 ± 0.09	0.35 ± 0.08	0.35 ± 0.08	0.37 ± 0.07	0.36 ± 0.08
Validation Set	0.33 ± 0.09	0.36 ± 0.09	0.37 ± 0.09	0.39 ± 0.10	0.38 ± 0.09

B.6.2. Equivariant Diffusion For Materials

Similarly, we investigated the distribution of generated molecules across the training, validation, and test sets of the materials design experiments from Section 5.2. As this experimental setup operates in an exhaustively enumerated search space, we compare the proportions of generated samples across all sets with both unguided and weight decay-regularized baselines. We use a guidance scale of 4, as this is the setting in which both guided diffusion models perform best (see Figure 5). We observe that our model generates a similar proportion of molecules from the medium-value validation set as the weight decay-regularized baseline, while producing substantially more compounds from the completely held-out high-property test set.

Table 10: Distribution of generated molecules across the training, validation, and test sets for the materials science experiments in Section 5.2. Means and standard deviations are reported across ten independent training and sampling runs with different random seeds.

Model	ratio in training set	ratio in validation set	ratio in test set
Unguided	0.91 ± 0.01	0.05 ± 0.00	0.04 ± 0.00
Weight Decay	0.49 ± 0.07	0.32 ± 0.05	0.19 ± 0.03
Ours	0.19 ± 0.03	0.39 ± 0.05	0.46 ± 0.06

.

Appendix B

Supplementary to *Diffusion Models for Constrained Domains*

Supplementary to: Diffusion Models for Constrained Domains

Introduction

In this supplementary, we first recall some key concepts from Riemannian geometry in Appendix A. In Appendix B we remind the expression of the Brownian motion in local coordinates. Details about the geodesic Brownian motion are given in Appendix C. Background on the Skorokhod problem is given in Appendix D. In Appendix E we derive the implicit score matching loss. We give details about the likelihood evaluation in Appendix F. Then in Appendix G we prove the time-reversal formula of reflected Brownian motion. In Appendix I we give some background on the conformational modelling of proteins backbone for the experiment in Section 5.3. In Appendix H we detail the geometrical constraint arising from the configurational robotics arm modelling experiment from Section 5.2. Additional results, training and miscalleneous experimental details are reported in Appendix J.

A Manifold concepts

For readers unfamiliar with Riemannian geometry here we give a brief overview of some of the key concepts. This is not a technical introduction, but a conceptual one for the understanding of terms. For a technical introduction, we refer readers to Lee (2013). A Riemannian manifold is a tuple $(\mathcal{M}, \mathfrak{g})$ with $\mathcal{M}$ a smooth manifold and $\mathfrak{g}$ a metric which defines an inner product on tangent spaces.

A *smooth manifold* is a topological space which locally can be described by Euclidean space. It is characterised by a family of *charts* $\{U \subset \mathcal{M}, \phi : U \to \mathbb{R}^d\}$, homeomorphic mappings between subsets of the manifold and Euclidean space. The collection of charts must cover the whole manifold. For the manifolds to be smooth the charts must be smooth, and the transition between charts where their domains overlap must also be smooth.

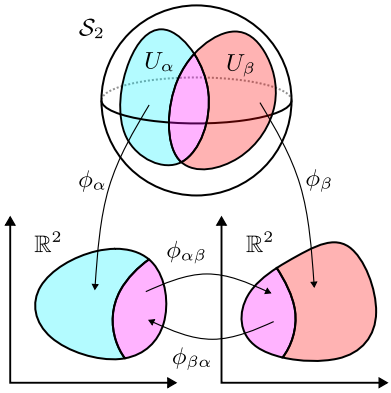


Figure 12: Example charts of the 2D manifold $\mathcal{S}^2$.

The *metric* on a Riemannian manifold gives the manifold a notion of distance and curvature. The same underlying smooth manifold with different metrics can look wildly different. The metric is defined as a smooth choice of positive definite inner product on each of the tangent spaces of the manifold. That is we have at every point a symmetric bilinear map

$$\mathfrak{g}(p) : \mathrm{T}_p\mathcal{M} \times \mathrm{T}_p\mathcal{M} \to \mathbb{R}$$

The tangent space of a point on a manifold is the generalisation of the notion of tangent planes and can be thought of as the space of derivatives of scalar functions on the manifold at that point. The collection of all tangent spaces is written as $\mathrm{T}\mathcal{M} = \bigcup_{p\in\mathcal{M}} \mathrm{T}_p\mathcal{M}$ and is called the tangent bundle. It is also manifold. Vector fields on manifolds are defined by making a choice of tangent vector at every point on the manifold in a smooth fashion. The space of vector fields is written at $\Gamma(\mathrm{T}\mathcal{M})$ and is more technically the space of *sections* of the tangent bundle.

One thing that the metric itself does not immediately define is how different tangent spaces at points on the manifold relate to one another. For this, we need additionally the concept of a *connection*. A connection is a map that takes two vector fields and produces a derivative of the first with respect to the second, that is a function $\nabla : \Gamma(\mathrm{T}\mathcal{M}) \times \Gamma(\mathrm{T}\mathcal{M}) \to \Gamma(\mathrm{T}\mathcal{M})$ and it typically written as $\nabla(X, Y) = \nabla_X Y$. Such a connection must for $X, Y, Z \in \Gamma(\mathrm{T}\mathcal{M})$ and smooth functions on the manifold $a, b : \mathcal{M} \to \mathbb{R}$ obey the following conditions:

(a) $\nabla_{aX+bY} Z = a\nabla_X Z + b\nabla_Y Z$,

(b) $\nabla_X(Y+Z) = \nabla_X Y + \nabla_X Z$,

(c) $\nabla_X(aY) = \partial_X aY + a\nabla_X Y$,

where $\partial_X aY$ is the regular *directional derivative* of aY in the direction X. These conditions ensure the connection is a well-defined derivative.

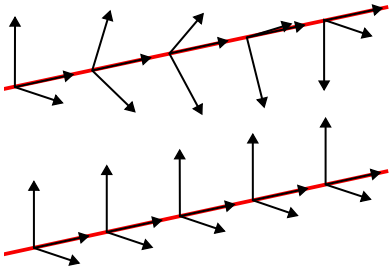


Figure 13: *Top:* Parallel transport of vectors along the red path with non-zero torsion. *Bottom:* Parallel transport of vectors along the red path with zero torsion. Both under the Euclidean metric.

On a given manifold, there are infinitely many connections. Fortunately, there is a natural choice, called the *Levi-Cevita* connection if we impose two additional conditions:

(a) $X \cdot (\mathfrak{g}(Y,Z)) = \mathfrak{g}(\nabla_X Y, Z) + \mathfrak{g}(Y, \nabla_X Z)$,

(b) $[X,Y] = \nabla_X Y - \nabla_Y X$,

where $[\cdot,\cdot]$ is the Lie bracket. The first condition ensures that the metric is preserved by the connection. That is to say, the *parallel transport* (to be defined shortly) using the connection leaves inner products unchanged on the manifold. The second ensures the connection is *torsion-free*. The change in tangent space along a geodesic (again to be defined shortly) can be described in two parts, the *curvature*, how the tangent space rotates perpendicular to the direction of travel, and the torsion, how the tangent space rotates around the axis of the direction of travel. The curvature of the connection is uniquely fixed by the other 4 conditions (the well-defined derivative and preservation of the metric). The torsion however is not fixed. By requiring it to be zero we ensure a unique connection. The requirement of zero torsion also has implications for ensuring integrability on the manifold.

With the metric and the Levi-Cevita connection in hand, we can define a number of key concepts.

We say that a vector field X is *parallel* to a curve $\gamma : (0,1) \to \mathcal{M}$ if

$$\nabla_{\gamma'} X = 0$$

where $\gamma' : (0,1) \to \mathrm{T}_{\gamma(t)}\mathcal{M}$ is the derivative of the path. For two points on the manifold $p, q \in \mathcal{M}$ and a curve between them, γ, $\gamma(0) = p$, $\gamma(1) = q$, for an initial vector $X_0 \in \mathrm{T}_p\mathcal{M}$ there is a unique vector field X that is parallel to γ such that $X(p) = X_0$. This induces a map between the tangent spaces at p and q

$$\tau_\gamma : \mathrm{T}_p\mathcal{M} \to \mathrm{T}_q\mathcal{M}$$

This map is referred as the *parallel transport* of tangent vectors between p and q, and this satisfies the condition that for $\boldsymbol{v}, \boldsymbol{u} \in \mathrm{T}_p\mathcal{M}$

$$\mathfrak{g}(p)(\boldsymbol{v}, \boldsymbol{u}) = \mathfrak{g}(q)(\tau_\gamma(\boldsymbol{v}), \tau_\gamma(\boldsymbol{u})).$$

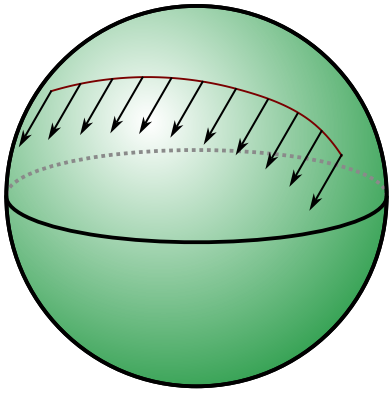


Figure 14: Parallel transport of vectors along a path on the sphere.

A *geodesic* on a manifold is the unique path on the manifold $\gamma : (0,1) \to \mathcal{M}$ such that $\nabla_{\gamma'}\gamma' = 0$. It is also the shortest path between two points on a manifold in the sense that

$$L(\gamma) = \int_0^1 \sqrt{\mathfrak{g}(\gamma(t))(\gamma'(t), \gamma(t))},$$

is minimal out of any path between the start and end of the geodesic. Geodesics give the notion of 'straight lines' on manifolds. We define the *exponential map* on a manifold as the mapping between an element of the tangent space at point p, $\boldsymbol{v} \in T_p\mathcal{M}$ and the endpoint of the unique geodesic γ with $\gamma(0) = p$ and $\gamma'(0) = \boldsymbol{v}$. In the tangent space of a manifold, we require the notion of a reflection in order to reflect geodesics off of boundary constraints. If at a point in the manifold with have $\boldsymbol{v} \in T_p\mathcal{M}$ and a unit vector normal to the constraint $\boldsymbol{n} \in T_p\mathcal{M}$, then the reflection of this vector is given by $\boldsymbol{v}' = \boldsymbol{v} - 2\mathfrak{g}(\boldsymbol{v}, \boldsymbol{n})\boldsymbol{n}$.

B Brownian motion in local coordinates

We consider a smooth function $f \in \mathrm{C}^\infty(\mathcal{M})$. The Laplace-Beltrami operator $\Delta_\mathcal{M}$ is given by $\Delta_\mathcal{M}(f) = \mathrm{div}(\mathrm{grad}(f)))$. In local coordinates we have

$$\mathrm{div}(X) = (\det(\mathfrak{g})^{-1/2}) \sum_{i=1}^d \partial_i(\det(\mathfrak{g})^{1/2} X_i),$$

as well as

$$\mathrm{grad}(f) = \mathfrak{g}^{-1}\nabla f.$$

Therefore, the Laplace-Beltrami operator is given by

$$\Delta_\mathcal{M}(f) = \sum_{i,j=1}^d \mathfrak{g}^{-1}_{i,j}\partial_{i,j} f + (\det(\mathfrak{g})^{-1/2}) \sum_{i,j=1}^d \partial_i(\det(\mathfrak{g})^{1/2}\mathfrak{g}^{-1}_{i,j})\partial_j f.$$

Therefore, in local coordinates the infinitesimal generator associated with the Laplace-Beltrami operator is given by

$$\mathcal{A}(f) = \sum_{i,j=1}^d \mathfrak{g}^{-1}_{i,j}\partial_{i,j} f + \langle b^i, \nabla f\rangle,$$

with

$$b^i = (\det(\mathfrak{g})^{-1/2}) \sum_{j=1}^d \partial_j(\det(\mathfrak{g})^{1/2}\mathfrak{g}^{-1}_{i,j}). \tag{11}$$

Therefore, the dual operator associated with $\mathcal{A}$ is given by

$$\mathcal{A}^\star(f) = \sum_{i,j=1}^d \partial_{i,j}(\mathfrak{g}^{-1}_{i,j} f) - \sum_{i=1}^d \partial_i(b^i f).$$

Note that by letting $f = \det(\mathfrak{g})^{1/2}$ we get that $\mathcal{A}^\star(f) = 0$ and therefore we recover that $p \propto \det(\mathfrak{g})$ is the invariant distribution of the Brownian motion.

Langevin dynamics on $\mathcal{M}$. We know that the Brownian motion targets $\det(\mathfrak{g})^{1/2}$. Therefore in order to correct and sample from the uniform distribution we consider the Langevin dynamics

$$\mathrm{d}\mathbf{X}_t = -\mathrm{grad}\log(\det(\mathfrak{g})^{1/2})(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}^\mathcal{M}_t.$$

Note that in the previous equation grad and $\mathbf{B}^\mathcal{M}_t$ are defined w.r.t. the metric of the manifold. In local coordinates we have

$$\mathrm{d}\mathbf{X}_t = \{b - \mathrm{grad}\log(\det(\mathfrak{g})^{1/2})\}(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathfrak{g}(\mathbf{X}_t)^{-1/2}\mathrm{d}\mathbf{B}_t. \tag{12}$$

where $b = \{b^i\}_{i=1}^d$ is given in equation 11. In addition, we have

$$\mathrm{grad}\log(\det(\mathfrak{g})^{1/2}) = \det(\mathfrak{g})^{-1/2}\mathfrak{g}^{-1}\nabla\det(\mathfrak{g}). \tag{13}$$

Using equation 11 we have

$$b^i = (\det(\mathfrak{g})^{-1/2}) \sum_{j=1}^d \partial_j(\det(\mathfrak{g})^{1/2}\mathfrak{g}^{-1}_{i,j}) = \sum_{j=1}^d \partial_j \mathfrak{g}^{-1}_{i,j} + \mathrm{grad}\log(\det(\mathfrak{g})^{1/2})_i$$

This can also be rewritten as

$$\mathrm{div}_\mathcal{M}(\mathfrak{g}^{-1}) = \mu + \mathrm{grad}\log(\det(\mathfrak{g})^{1/2}),$$

with

$$\mu_i = \sum_{j=1}^d \partial_j \mathfrak{g}^{-1}_{i,j}.$$

Combining this result and equation 13 we get that equation 12 can be rewritten as

$$\mathrm{d}\mathbf{X}_t = \mu(\mathbf{X}_t) + \sqrt{2}\mathrm{d}\mathbf{B}_t.$$

Note that (up to a factor 2) this is the same SDE as the one considered in Lee & Vempala (2017).

C Geodesic Brownian Motion

In this section, we provide some details on the geodesics Brownian motion introduced in Section 3.1. In the rest of the section, we make the following assumption.

A1. $\overline{\mathcal{M}} \subset \mathbb{R}^d$ is compact and $\mathfrak{g}^{-1} : \mathcal{M} \to \mathcal{S}_d^{++}$ can be $\mathrm{C}^\infty(\mathbb{R}^d, \mathbb{R}^{d\times d})$.

First, we start by showing that the process $(\mathbf{X}_t)_{t\geq 0}$ defined in equation 6 exists and that we have for any $t \geq 0$, $\mathbf{X}_t \in \mathcal{M}$. We recall that $\mathfrak{g} = \nabla^2\phi$ and $\lim_{x\to\partial\mathcal{M}} \Phi(x) = +\infty$.

Proposition C.1. *Assume* **A**1. *For any $x_0 \in \mathcal{M}$, there exists a unique strong solution to equation 6 denoted $(\mathbf{X}_t)_{t\geq 0}$. In addition, we have that for any $t \geq 0$, $\mathbf{X}_t \in \mathcal{M}$ almost surely. More precisely, we have $\mathbb{E}[\phi(\mathbf{X}_t)] \leq \phi(x_0) + t$.*

Proof. A unique strong solution $(\mathbf{X}_t)_{t\geq 0}$ of equation 6 with starting point $x_0 \in \mathcal{M}$ exists since the coefficients are smooth, see (Ikeda & Watanabe, 2014, Theorem 3.1, p.165). For any $A \geq 0$, we define $\tau_A = \inf\{t \geq 0 : \Phi(\mathbf{X}_t) \geq A\}$. Note that for any $t \in [0, \tau_A]$, $\Phi(\mathbf{X}_t) \in \mathcal{M}$. Using Itô formula, we have for any $t \geq 0$

$$\mathbb{E}[\Phi(\mathbf{X}_{t\wedge\tau_A})] = \Phi(x_0) + \mathbb{E}[\textstyle\int_0^{t\wedge\tau_A} \mathrm{Tr}(\mathfrak{g}^{-1}(\mathbf{X}_s)\nabla^2\Phi(\mathbf{X}_s))\mathrm{d}s] = \Phi(x_0) + \mathbb{E}[t \wedge \tau_A].$$

Using Fatou's lemma, and letting $A \to +\infty$, we conclude the proof. □

In the next result, we show that the uniform distribution is the *unique* invariant probability distribution for $(\mathbf{X}_t)_{t\geq 0}$ and that $(\mathbf{X}_t)_{t\geq 0}$ converges to this invariant distribution. We refer to (Meyn & Tweedie, 1993, Section 2, p.490) for a definition of irreducibility. We recall that the total variation of a finite (not necessarily non-negative) measure μ over $\mathbb{R}^d$ is given by $\|\mu\|_{\mathrm{TV}} = \sup\{\mu(\mathsf{A}) : \mathsf{A} \in \mathcal{B}(()\mathbb{R}^d)\}$.

Proposition C.2. *Assume* **A**1. *$(\mathbf{X}_t)_{t\geq 0}$ is π-irreducible, the uniform distribution over $\mathcal{M}$ is the only invariant probability distribution and $\lim_{t\to+\infty} \|\mathrm{P}_t - \pi\|_{\mathrm{TV}} = 0$, where P_t is the distribution of $\mathbf{X}_t$ for any $t \geq 0$ and π is the uniform distribution over $\mathcal{M}$.*

Proof. Since $x \mapsto \mathrm{div}(\mathfrak{g}^{-1})(x)$ and $x \mapsto \mathfrak{g}^{-1}$ are smooth and $\mathfrak{g}^{-1}(x)$ is positive definite for any $x \in \mathcal{M}$, we have that $(\mathbf{X}_t)_{t\geq 0}$ is π-irreducible, extending (Bhattacharya, 1978, Lemma 1.4) to $\mathcal{M}$ and using (Meyn & Tweedie, 1993, Proposition 2.1). In addition, $(\mathbf{X}_t)_{t\geq 0}$ is T-Feller using (Meyn & Tweedie, 1993, Proposition 3.3). Combining these results and the fact that $\mathcal{M}$ is bounde, we get that $(\mathbf{X}_t)_{t\geq 0}$ is positive Harris recurrent (Meyn & Tweedie, 1993, Theorem 3.2). The uniform distribution π is an invariant distribution for equation 6. Since $(\mathbf{X}_t)_{t\geq 0}$ is π-irreducible, we get that this invariant measure is unique. Hence, we conclude using (Meyn & Tweedie, 1993, Theorem 6.1). □

Note that the convergence result in total variation could be improved. In particular, quantitative geometric results could be derived. We finish this section, by applying results from the Malliavin calculus to show that for any $t > 0$, $\mathbf{X}_t$ admits a density w.r.t. the Lebesgue measure.

Proposition C.3. *Assume* **A**1. *Then, for any $t \geq 0$, $\mathbf{X}_t$ admits a smooth density p_t w.r.t. the Lebesgue measure.*

Proof. This is a direct consequence of (Nualart, 2006, Theorem 2.3.3). □

D Reflected Brownian Motion and Skorokhod problems

In this section, we provide the basic definitions and results to derive the time-reversal of the reflected Brownian motion in Appendix G. We follow closely the presentation of Lions & Sznitman (1984) and Burdzy et al. (2004). We first define the *Skorokhod problem* for deterministic problems. We consider $\mathcal{M}$ to be a smooth open bounded domain. We recall that the normal vector n is defined on $\partial\mathcal{M}$ and we set $n(x) = 0$ for any $x \notin \partial\mathcal{M}$.

Before giving the definition of the *Skorokhod problem*, we recall what is the space of functions of *bounded variations*.

Definition D.1. Let $a, b \in (-\infty, +\infty)$ and $f :\in \mathrm{C}([a, b], \mathbb{R})$. We define the *total variation* of f as

$$\mathrm{V}_{a,b}(f) = \sup\{\textstyle\sum_{i=0}^{n-1} \|f(x_{i+1}) - f(x_i)\| \ : \ (x_i)_{i=0}^{n-1}, \ a = x_0 \leq x_1 \leq \cdots \leq x_{n-1} \leq x_n = b, \ n \in \mathbb{N}\}.$$

f has bounded variations over $[a, b]$ if $\mathrm{V}_{a,b}(f) < +\infty$. Let $f \in \mathrm{C}([0, +\infty), \mathbb{R})$. f has bounded variations over $[0, +\infty)$ if for any $b > 0$, f has bounded variations over $[0, b]$.

The notion of bounded variation is a relaxation of the differentiability requirement. In particular, if $f \in \mathrm{C}^1([a, b], \mathbb{R})$, we have $\mathrm{V}_{a,b}(f) = \int_a^b \|f'(t)\| \mathrm{d}t$. In the definition of the *Skorokhod problem*, we will see that this relaxation is necessary, even in the deterministic setting.

For any function of bounded variation $f \in \mathrm{C}([a, b], \mathbb{R})$ on $[a, b]$, we define $|f| : \ [a, b] \to [0, +\infty)$ given for any $t \in [a, b]$ by $|f|_t = \mathrm{V}_{a,t}(f)$. Note that $|f|$ is non-decreasing and right-continuous. Therefore, we can define the measure $\mu_{|f|}$ on $[a, b]$, given for any $s, t \in [a, b]$ with $t \geq s$ by $\mu_{|f|}([s, t]) = |f|(t) - |f|(s)$. In particular, for any $\varphi : [a, b] \to \mathbb{R}_+$, we define

$$\textstyle\int_a^b \varphi(t) \mathrm{d}|f|_t = \int_a^b \varphi(t) \mathrm{d}\mu_{|f|}(t).$$

In addition, f can be decomposed in a difference of two non-decreasing processes right continuous processes g_1, g_2, where for any $t \in [a, b]$, $f(t) = g_1(t) - g_2(t)$, $g_1(t) = |f|_t$ and $g_2(t) = |f|_t - f(t)$. Hence, for every φ bounded on $[a, b]$, we can define

$$\textstyle\int_a^b \varphi(t) \mathrm{d}f(t) = \int_a^b \varphi(t) \mathrm{d}g_1(t) - \int_a^b \varphi(t) \mathrm{d}g_2(t).$$

Note that these definitions can be extended to the setting where $f : \ [a, b] \to \mathbb{R}^d$.

We begin with the following result, see Lions & Sznitman (1984).

Theorem D.2. *Let $(x_t)_{t\geq 0} \in \mathrm{C}([0, +\infty), \mathbb{R})$. Then, there exists a unique couple $(\bar{x}_t, k_t)_{t\geq 0}$ such that*

(a) $(k_t)_{t\geq 0}$ has bounded variation over $[0, +\infty)$.

(b) $(\bar{x}_t)_{t\geq 0} \in \mathrm{C}([0, +\infty), \overline{\mathcal{M}})$.

(c) For any $t \geq 0$, $x_t + k_t = \bar{x}_t$.

(d) For any $t \geq 0$, $|k|_t = \int_0^t \mathbf{1}_{\bar{x}_s \in \partial\mathcal{M}}(\bar{x}_s) \mathrm{d}|k|_s$ and $k_t = \int_0^t n(\bar{x}_s) \mathrm{d}|k|_s$.

Let us discuss Theorem D.2. First, Theorem D.2-(c) states the original (unconstrained) process $(x_t)_{t\geq 0}$ can be decomposed into a constrained version $(\bar{x}_t)_{t\geq 0}$ and a bounded variation process $(k_t)_{t\geq 0}$. The process $(|k|_t)_{t\geq 0}$ counts the number of times the constrained process $(\bar{x}_t)_{t\geq 0}$ hits the boundary. More formally, we have $|k|_t = \int_0^t \mathbf{1}_{x \in \partial\mathcal{M}}(\bar{x}_s) \mathrm{d}|k|_s$. When, we hit the boundary, we reflect the process. This condition is expressed in $k_t = \int_0^t n(\bar{x}_s) \mathrm{d}|k|_s$.

We now consider the extension to stochastic processes. We are given $(\mathbf{X}_t)_{t\geq 0}$ such that

$$\mathrm{d}\mathbf{X}_t = b(\mathbf{X}_t)\mathrm{d}t + \sigma(t)\mathrm{d}\mathbf{B}_t,$$

where $(\mathbf{B}_t)_{t\geq 0}$ is a d-dimensional Brownian motion. We also assume that b and σ are Lipschitz which implies the existence and strong uniqueness of $(\mathbf{X}_t)_{t\geq 0}$. We have the following result Lions & Sznitman (1984).

Theorem D.3. *There exists a unique process* $(\bar{\mathbf{X}}_t, \mathbf{k}_t)_{t\geq 0}$ *such that*

(a) $(\mathbf{k}_t)_{t\geq 0}$ *has bounded variation over* $[0, +\infty)$ *almost surely.*

(b) $(\bar{\mathbf{X}}_t)_{t\geq 0} \in \mathrm{C}([0, +\infty), \overline{\mathcal{M}})$.

(c) For any $t \geq 0$, $\bar{\mathbf{X}}_t = \bar{\mathbf{X}}_0 + \int_0^t b(\bar{\mathbf{X}}_s)\mathrm{d}s + \int_0^t \sigma(\bar{\mathbf{X}}_s)\mathrm{d}\mathbf{B}_s - \mathbf{k}_t$.

(d) For any $t \geq 0$, $|\mathbf{k}|_t = \int_0^t \mathbf{1}_{\bar{x}_s \in \partial\mathcal{M}}(\bar{x}_s)\mathrm{d}|\mathbf{k}|_s$ *and* $\mathbf{k}_t = \int_0^t n(\bar{x}_s)\mathrm{d}|\mathbf{k}|_s$.

The process $(\mathbf{X}_t)_{t\geq 0}$ is almost surely continuous, so we could apply the previous theorem almost surely for all the realizations of the process,. However, this does not tell us if the obtained solutions $(\bar{\mathbf{X}}_t, \mathbf{k}_t)_{t\geq 0}$ form themselves a process. The main difference with Theorem D.2 is in Theorem D.3-(c) which differs from Theorem D.3-(c). Note that in the case where $b = 0$ and $\sigma = \mathrm{Id}$ we recover Theorem D.3-(c). This is not true in the general case. However, it can be seen that for any realization of the process $(\bar{\mathbf{X}}_t)_{t\geq 0}$, we have that $(\bar{\mathbf{X}}_t, \mathbf{k}_t)_{t\geq 0}$ is solution of the *deterministic* Skorokhod problem by letting $x_t = \bar{\mathbf{X}}_0 + \int_0^t b(\bar{\mathbf{X}}_s)\mathrm{d}s + \int_0^t \sigma(\bar{\mathbf{X}}_s)\mathrm{d}\mathbf{B}_s$. The backward and forward Kolmogorov equations can be found in Burdzy et al. (2004). Note that the presence of the process $(\mathbf{k}_t)_{t\geq 0}$ incurs notable complications compared to unconstrained processes. In particular, there is no martingale problem associated with weak solutions of reflected SDEs but only sub-martingale problems, see Kang & Ramanan (2017) for instance.

E Implicit Score Matching Loss

E.1 Proof of ISM

Using the divergence theorem, we have

$$
\begin{aligned}
&(1/2)\int_{\mathcal{M}} \|\mathbf{s}_\theta(x) - \nabla \log p_t(x)\|^2 p_t(x) \mathrm{d}\mu(x) \\
&= (1/2)\int_{\mathcal{M}} \|\mathbf{s}_\theta(x)\|^2 p_t(x) \mathrm{d}\mu(x) - \int_{\mathcal{M}} \langle \mathbf{s}_\theta(x), \nabla \log p_t(x) \rangle p_t(x) \mathrm{d}\mu(x) + (1/2)\int_{\mathcal{M}} \|\nabla \log p_t(x)\|^2 p_t(x) \mathrm{d}\mu(x) \\
&= (1/2)\int_{\mathcal{M}} \|\mathbf{s}_\theta(x)\|^2 p_t(x) \mathrm{d}\mu(x) - \int_{\partial\mathcal{M}} \langle \mathbf{s}_\theta(x), \mathbf{n} \rangle p_t(x) \mathrm{d}\nu(x) \\
&\quad + \int_{\mathcal{M}} \mathrm{div}(\mathbf{s}_\theta)(x) p_t(x) \mathrm{d}\mu(x) + (1/2)\int_{\mathcal{M}} \|\nabla \log p_t(x)\|^2 p_t(x) \mathrm{d}\mu(x).
\end{aligned}
$$

Using that $\mathbf{s}_\theta(x) = 0$ for all $x \in \partial\mathcal{M}$, we get that

$$
\begin{aligned}
&(1/2)\int_{\mathcal{M}} \|\mathbf{s}_\theta(x) - \nabla \log p_t(x)\|^2 p_t(x) \mathrm{d}\mu(x) \\
&\qquad = (1/2)\int_{\mathcal{M}} \|\mathbf{s}_\theta(x)\|^2 p_t(x) \mathrm{d}\mu(x) + \int_{\mathcal{M}} \mathrm{div}(\mathbf{s}_\theta)(x) p_t(x) \mathrm{d}\mu(x) + (1/2)\int_{\mathcal{M}} \|\nabla \log p_t(x)\|^2 p_t(x) \mathrm{d}\mu(x),
\end{aligned}
$$

which concludes the proof.

E.2 Importance of scaling function

As discussed in Section 3.3, we include a monotone scaling function h which is zero close to the boundary to ensure the relevant conditions are met for the score matching loss and the boundary conditions. This may seem like a technical detail, but is of significant practical importance. Upon removal of the scaling function, we observe that the learned score functions behave strangely around the boundary in the reverse process, leading to samples that do not match the forward process. The problems are apparent when comparing the top three plots of Fig. 15 and Fig. 16. Interestingly, we found that these issues early on in the sampling are smoothed out by the end of the reverse process, but still lead to a failure to recover the target density.

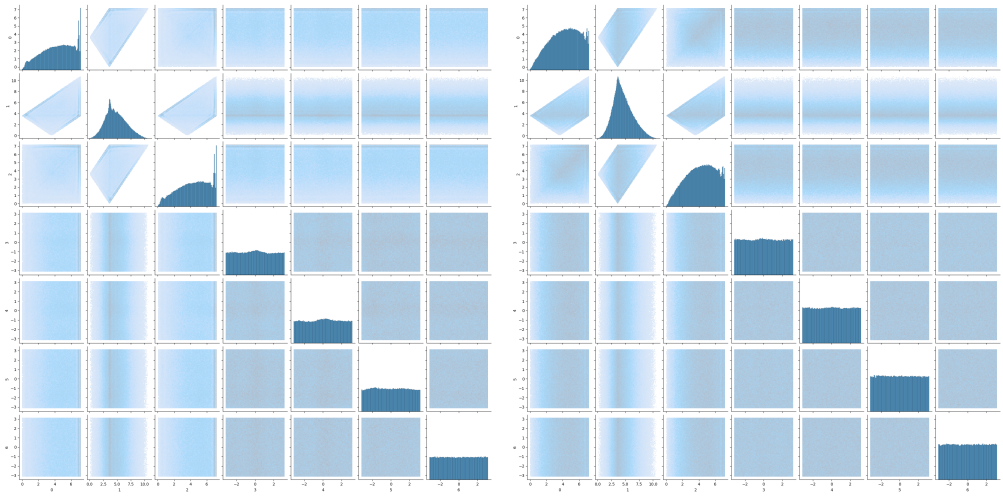


Figure 15: Reverse process samples for the cyclic peptide dataset from Section 5.3 at $t = 1.0, 0.9$ (left and right respectively) trained without the scaling function.

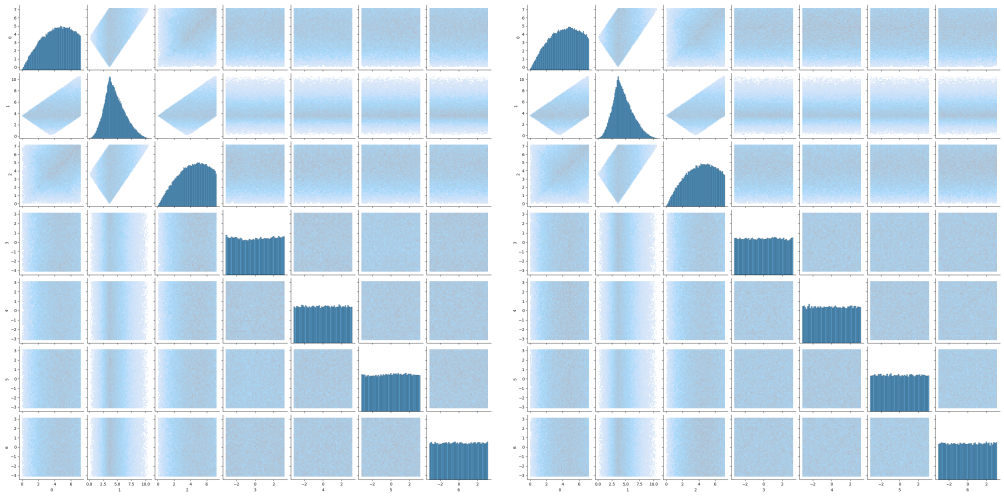


Figure 16: Reverse process samples for the cyclic peptide dataset from Section 5.3 at $t = 1.0, 0.9$ (left and right respectively) trained with the scaling function.

F Likelihood evaluation

One key advantage of constructing a continuous noising process is that, similarly to Song et al. (2021), we can evaluate the model's likelihood via the following *probability flow* Ordinary Differential Equation (ODE). In particular, for the Langevin dynamics (6) which we recall

$$\mathrm{d}\mathbf{X}_t = \tfrac{1}{2}\mathrm{div}(\mathfrak{g}^{-1})(\mathbf{X}_t)\mathrm{d}t + \mathfrak{g}(\mathbf{X}_t)^{-\frac{1}{2}}\mathrm{d}\mathbf{B}_t,$$

the following ODE has the same marginal density

$$\mathrm{d}\mathbf{Y}_t = \left[\tfrac{1}{2}\nabla \cdot \mathfrak{g}^{-1}(\mathbf{Y}_t) - \tfrac{1}{2}\mathfrak{g}^{-1}(\mathbf{Y}_t)\nabla \log p_t(\mathbf{Y}_t)\right] \mathrm{d}t.$$

We conclude this section with a derivation of the equivalent ODE. We highlight that the ODE representation for reflected diffusion models was first derived in Lou & Ermon (2023). We recall that if $\mathcal{M} \subset \mathbb{R}^d$ is a bounded open set with smooth boundary then (Burdzy et al., 2004, Theorem 2.2) ensures that the reflected Brownian motion admits a density w.r.t. the Lebesgue measure. We denote p_t this smooth density.

Proposition F.1. *Assume that $\mathcal{M} \subset \mathbb{R}^d$ is a bounded open set with smooth boundary. Assume that $(t, x) \mapsto \nabla \log p_t(x)$ is smooth on $[0, +\infty) \times \partial\mathcal{M}$. Let $(\bar{\mathbf{B}}_t)_{t\geq 0}$ be the reflected Brownian motion with $\bar{\mathbf{B}}_0 \sim p_0$ smooth and supported in $\mathcal{M}$. Let $(\mathbf{X}_t)_{t\geq 0}$ be given for any $t \geq 0$ by $\mathrm{d}\mathbf{X}_t = \frac{1}{2}\nabla \log p_t(\mathbf{X}_t)\mathrm{d}t$ and $\mathbf{X}_0 \sim p_0$, where p_t denotes the density of $\bar{\mathbf{B}}_t$ w.r.t. the Lebesgue measure for any $t > 0$. Then for any $t \in [0, T]$, $\bar{\mathbf{B}}_t$ and $\mathbf{Y}_t$ have the same distribution.*

Proof. Since the distributions of $(\bar{\mathbf{B}}_t)_{t\in[0,T]}$ and $(\mathbf{Y}_t)_{t\in[0,T]}$ satisfy the same Fokker-Planck equation whenever these processes are well-defined. Therefore, we first show that the process $(\mathbf{Y}_t)_{t\in[0,T]}$ is well-defined and stay in $\mathcal{M}$ at all times. Using (Burdzy et al., 2004, Theorem 2.2), we have that $\partial p_t(x) = \frac{1}{2}\mathrm{div}(\nabla \log p_t)(x)$, for any $t > 0$ and $x \in \mathcal{M}$. Next, we define $\mathrm{d}\mathbf{X}_t = \frac{1}{2}\nabla \log p_t(\mathbf{X}_t)\mathrm{d}t$. Note that $(\mathbf{X}_t)_{t\geq 0}$ is defined up to an explosion time T_∞ after which we fix $\mathbf{X}_t = \infty$. Denote T_0 the first time such that $\mathbf{X}_t \in \partial\mathcal{M}$. Note that since p_0 is supported on $\mathcal{M}$ we have $T_0 > 0$. We denote $(\mathbf{Y}_t)_{t\in[0,T_0]} = (\mathbf{X}_{T_0-t})_{t\in[0,T_0]}$. We have that for any $t \in [0, T_0]$, $\mathrm{d}\mathbf{Y}_t = -\frac{1}{2}\nabla \log p_{T_0-t}(\mathbf{Y}_t)\mathrm{d}t$. Since $(t, x) \mapsto \nabla \log p_t(x)$ is smooth on $[0, +\infty) \times \partial\mathcal{M}$, we get that for any $t \in [0, T_0]$, $\mathbf{Y}_t \in \partial\mathcal{M}$. In particular, we have that $\mathbf{Y}_{T_0/2} = \mathbf{X}_{T_0/2} \in \partial\mathcal{M}$ which is absurd. Therefore $T_0 = +\infty$ (which also implies that $T_\infty = +\infty$). Hence, $(\mathbf{X}_t)_{t\geq 0}$ is a flow on $\mathcal{M}$ and therefore for any $t \geq 0$, the density q_t of $\mathbf{X}_t$ is smooth and satisfies $\partial_t q_t(x) = -\frac{1}{2}\mathrm{div}(q_t \nabla \log p_t)(x)$. We conclude using the uniqueness of the solutions to the transport equation for smooth initialisation and coefficients on $\mathbb{R}^d$. □

G Time-reversal for reflected Brownian motion

We start with the following definition.

Definition G.1. Let $\mathcal{M} \subset \mathbb{R}^d$ be an open set. $\mathcal{M}$ has a smooth boundary if for any $x \in \partial\mathcal{M}$, there exists $\mathsf{U} \subset \mathbb{R}^d$ open and $f \in \mathrm{C}^\infty(\mathsf{U}, \mathbb{R})$ such that $x \in \mathsf{U}$ and (a) $\mathrm{cl}(\mathcal{M}) \cap \mathsf{U} = \{x \in \mathsf{U} : f(x) \leq 0\}$, (b) $\nabla f(x) \neq 0$ for any $x \in \mathsf{U}$ where $\mathrm{cl}(\mathcal{M})$ is the closure of $\mathcal{M}$.

We will make the use of the following lemma which is a straightforward extension of Burdzy et al. (2004, Theorem 2.6). The surface measure is defined in (Lee, 2006, Proposition 2.43). Under mild regularity assumptions, it corresponds to the Hausdorff measure of $\partial\mathcal{M}$, see Evans & Gariepy (2015).

Lemma G.2. *Let u such that $s \mapsto u(s,x) \in \mathrm{C}^1((0,T),\mathbb{R})$, for any $s \in (0,T)$, $x \mapsto u(s,x) \in \mathrm{C}^2(\mathcal{M},\mathbb{R})$ and $u \in \mathrm{C}^1(\mathrm{cl}(\mathcal{M}),\mathbb{R})$. Then for any $T \geq 0$, $s,t \in [0,T]$, we have*

$$\mathbb{E}[\int_s^t u(w, \bar{\mathbf{B}}_w)\mathrm{d}\mathbf{k}_w] = \tfrac{1}{2}\int_s^t \int_{\partial\mathcal{M}} u(x) p_w(x)\mathrm{d}\sigma(x)\mathrm{d}w.$$

Note that we recover Burdzy et al. (2004, Theorem 2.6) if we set $u = 1$. We also emphasize that the result of Burdzy et al. (2004, Theorem 2.6) is stronger than Theorem G.2 as it holds not only in expectation but also in L^2 and almost surely.

We are now ready to prove Theorem 3.2. We follow the approach of Petit (1997) which itself is based on an extension of Haussmann & Pardoux (1986). We refer to Cattiaux et al. (2021) for recent entropic approaches of time-reversal. Recall that $(\bar{\mathbf{B}}_t, \mathbf{k}_t)_{t\geq 0}$ is a solution to the *Skorokhod problem* (Skorokhod, 1961) if $(\mathbf{k}_t)_{t\geq 0}$ a bounded variation process and $(\bar{\mathbf{B}}_t)_{t\geq 0}$ a continuous adapted process such that for any $t \geq 0$, $\mathbf{B}_t = \bar{\mathbf{B}}_t + \mathbf{k}_t \in \mathcal{M}$, $(\bar{\mathbf{B}}_t)_{t\geq 0}$ and

$$|\mathbf{k}|_t = \int_0^t \mathbf{1}_{\bar{\mathbf{B}}_s \in \partial\mathcal{M}} \mathrm{d}|\mathbf{k}|_s, \quad \mathbf{k}_t = \int_0^t \mathbf{n}(\bar{\mathbf{B}}_s)\mathrm{d}|\mathbf{k}|_s, \tag{14}$$

In what follows, we define $(\mathbf{Y}_t)_{t\in[0,T]}$ such that for any $t \in [0,T]$, $\mathbf{Y}_t = \bar{\mathbf{B}}_{T-t}$. Let us consider the process $(\tilde{\mathbf{B}}_t)_{t\in[0,T]}$ defined for any $t \in [0,T]$ by

$$\tilde{\mathbf{B}}_t = -\bar{\mathbf{B}}_T + \bar{\mathbf{B}}_{T-t} + \mathbf{k}_T - \mathbf{k}_{T-t} - \int_{T-t}^T \nabla \log p_s(\bar{\mathbf{B}}_s)\mathrm{d}s.$$

First, note that $t \mapsto \tilde{\mathbf{B}}_t$ is continuous. Denote by $\mathcal{F}$, the filtration associated with $(\bar{\mathbf{B}}_{T-t})_{t\in[0,T]}$. We have that $(\tilde{\mathbf{B}}_t)_{t\in[0,T]}$ is adapted to $(\bar{\mathbf{B}}_{T-t})_{t\in[0,T]}$. Even more so, we have that $(\tilde{\mathbf{B}}_t)_{t\in[0,T]}$ satisfies the strong Markov property since $(\bar{\mathbf{B}}_t)_{t\in[0,T]}$ also satisfies the strong Markov property. Let $g \in \mathrm{C}_c^\infty(\mathrm{cl}(\mathcal{M}))$ and consider for any $0 \leq s \leq t \leq T$, $\mathbb{E}[(\tilde{\mathbf{B}}_t - \tilde{\mathbf{B}}_s)g(\bar{\mathbf{B}}_{T-t})]$. For any $0 \leq s \leq t \leq T$ we have

$$\mathbb{E}[(\tilde{\mathbf{B}}_t - \tilde{\mathbf{B}}_s)g(\bar{\mathbf{B}}_{T-t})] = \mathbb{E}[(-\bar{\mathbf{B}}_{T-s} + \bar{\mathbf{B}}_{T-t} + \mathbf{k}_{T-s} - \mathbf{k}_{T-t} - \int_{T-t}^{T-s} \nabla \log p_u(\bar{\mathbf{B}}_u)\mathrm{d}u)g(\bar{\mathbf{B}}_{T-t})]. \tag{15}$$

In what follows, we prove that for any $0 \leq s \leq t \leq T$ we have $\mathbb{E}[(\tilde{\mathbf{B}}_t - \tilde{\mathbf{B}}_s)g(\bar{\mathbf{B}}_{T-t})] = 0$. Therefore, we only need to prove that for any $0 \leq s \leq t \leq T$ we have

$$\mathbb{E}[(-\bar{\mathbf{B}}_t + \bar{\mathbf{B}}_s + \mathbf{k}_t - \mathbf{k}_s - \int_s^t \nabla \log p_u(\bar{\mathbf{B}}_u)\mathrm{d}u)g(\bar{\mathbf{B}}_t)] = 0. \tag{16}$$

Let $t \in (0,T]$. We introduce $u : [0,t] \times \mathcal{M}$ such that for any $s \in [0,t]$ and $x \in \mathcal{M}$, $u(s,x) = \mathbb{E}[g(\bar{\mathbf{B}}_t)|\bar{\mathbf{B}}_s = x]$. Using Burdzy et al. (2004, Theorem 2.8) we get that for any $x \in \mathcal{M}$, $s \mapsto u(s,x) \in \mathrm{C}^1((0,t),\mathbb{R})$ and for any $s \in (0,t)$, $x \mapsto u(s,x) \in \mathrm{C}^2(\mathcal{M},\mathbb{R})$ and $x \mapsto u(s,x) \in \mathrm{C}^1(\mathrm{cl}(\mathcal{M}),\mathbb{R})$. In addition, we have that for any $s \in (0,t)$ and for any $x \in \mathcal{M}$ and $x_0 \in \partial\mathcal{M}$

$$\partial_s u(s,x) + \tfrac{1}{2}\Delta u(s,x) = 0, \qquad \langle \nabla u(s,x_0), \mathbf{n}(x_0)\rangle = 0. \tag{17}$$

This equation is called the backward Kolmogorov equation. Using equation 17, $\bar{\mathbf{B}}_t = \mathbf{B}_t - \mathbf{k}_t$ for any $t \geq 0$ and the Itô formula for semimartingale (Revuz & Yor, 2013, Chapter IV, Theorem 3.3) we have that for any

$s \in (0, t)$

$$\begin{aligned}\mathbb{E}[u(t, \bar{\mathbf{B}}_t)\bar{\mathbf{B}}_t] = \mathbb{E}[u(s, \bar{\mathbf{B}}_s)\bar{\mathbf{B}}_s] + \mathbb{E}[\tfrac{1}{2}\int_s^t \bar{\mathbf{B}}_w \Delta u(w, \bar{\mathbf{B}}_w)\mathrm{d}w] + \mathbb{E}[\int_s^t \nabla u(w, \bar{\mathbf{B}}_w)\mathrm{d}w] \\ - \mathbb{E}[\int_s^t \bar{\mathbf{B}}_w \langle \nabla u(w, \bar{\mathbf{B}}_w), \mathbf{n}(\bar{\mathbf{B}}_w)\rangle \mathrm{d}|\mathbf{k}|_w] \\ - \mathbb{E}[\int_s^t u(w, \bar{\mathbf{B}}_w)\mathbf{n}(\bar{\mathbf{B}}_w)\mathrm{d}|\mathbf{k}|_w] \\ + \mathbb{E}[\int_s^t \bar{\mathbf{B}}_w \partial_w u(w, \bar{\mathbf{B}}_w)\mathrm{d}w] \\ = \mathbb{E}[u(s, \bar{\mathbf{B}}_s)\bar{\mathbf{B}}_s] + \mathbb{E}[\int_s^t \nabla u(w, \bar{\mathbf{B}}_w)\mathrm{d}w] - \mathbb{E}[\int_s^t u(w, \bar{\mathbf{B}}_w)\mathbf{n}(\bar{\mathbf{B}}_w)\mathrm{d}|\mathbf{k}|_w] \end{aligned} \tag{18}$$

In addition, using the Fubini theorem and the definition of $\mathbf{k}_t$ we have that for any $s \in (0, t)$

$$\mathbb{E}[\int_s^t u(w, \bar{\mathbf{B}}_w)\mathbf{n}(\bar{\mathbf{B}}_w)\mathrm{d}|\mathbf{k}|_w] = \mathbb{E}[\int_s^t \mathbb{E}[g(\bar{\mathbf{B}}_t)|\bar{\mathbf{B}}_w]\mathbf{n}(\bar{\mathbf{B}}_w)\mathrm{d}|\mathbf{k}|_w] = \mathbb{E}[g(\bar{\mathbf{B}}_t)(\mathbf{k}_t - \mathbf{k}_s)]. \tag{19}$$

Finally, using the divergence theorem and Burdzy et al. (2004, Theorem 2.6) we have that for any $s \in (0, t)$

$$\begin{aligned}\mathbb{E}[\int_s^t \nabla u(w, \bar{\mathbf{B}}_w)\mathrm{d}w] &= \int_s^t \int_{\mathcal{M}} \nabla u(w, x) p_w(x)\mathrm{d}x\mathrm{d}w \\ &= -\int_s^t \int_{\mathcal{M}} u(w, x)\nabla \log(p_w(x)) p_w(x)\mathrm{d}x\mathrm{d}w + \int_s^t \int_{\partial\mathcal{M}} u(w, x) p_w(x)\mathrm{d}x\mathrm{d}\sigma(w),\end{aligned}$$

where σ is the surface area measure on $\partial\mathcal{M}$, see Burdzy et al. (2004). Using Theorem G.2 and the Fubini theorem we get that

$$\begin{aligned}\mathbb{E}[\int_s^t \nabla u(w, \bar{\mathbf{B}}_w)\mathrm{d}w] &= -\int_s^t \int_{\mathcal{M}} u(w, x)\nabla \log(p_w(x)) p_w(x)\mathrm{d}x\mathrm{d}w + \mathbb{E}[\int_s^t u(w, \bar{\mathbf{B}}_w)\mathrm{d}\mathbf{k}_w] \\ &= -\mathbb{E}[\int_s^t g(\bar{\mathbf{B}}_t)\nabla \log(p_w(\bar{\mathbf{B}}_w))\mathrm{d}w] + 2\mathbb{E}[g(\bar{\mathbf{B}}_t)(\mathbf{k}_t - \mathbf{k}_s)]\end{aligned} \tag{20}$$

Combining equation 18, equation 19 and equation 20 we get that

$$\mathbb{E}[u(t, \bar{\mathbf{B}}_t)] = \mathbb{E}[u(s, \bar{\mathbf{B}}_s)] - \mathbb{E}[g(\bar{\mathbf{B}}_t)\int_s^t \nabla \log p_w(\bar{\mathbf{B}}_w)\mathrm{d}w] + \mathbb{E}[g(\bar{\mathbf{B}}_t)(\mathbf{k}_t - \mathbf{k}_s)].$$

Therefore, we get equation 16 and equation 15. Hence, $(\tilde{\mathbf{B}}_t)_{t\in[0,T]}$ is a continuous martingale. In addition, we have that for any $t \in [0, T]$, $\mathbb{E}[\tilde{\mathbf{B}}_t\tilde{\mathbf{B}}_t^\top] = t\,\mathrm{Id}$ and therefore, $(\tilde{\mathbf{B}}_t)_{t\in[0,T]}$ is a Brownian motion using the Lévy characterisation of Brownian motion (Revuz & Yor, 2013, Chapter IV, Theorem 3.6). Denote $(\mathbf{j}_t)_{t\in[0,T]} = (\mathbf{k}_T - \mathbf{k}_{T-t})_{t\in[0,T]}$. Using equation 15, we have that for any $t \in [0, T]$

$$\bar{\mathbf{B}}_{T-t} = \bar{\mathbf{Y}}_0 + \tilde{\mathbf{B}}_t + \int_0^t \nabla \log p_{T-s}(\mathbf{Y}_s)\mathrm{d}s - \mathbf{j}_t.$$

Using equation 14, we have for any $t \in [0, T]$

$$|\mathbf{j}|_t = \int_0^t \mathbf{1}_{\mathbf{Y}_s \in \partial\mathcal{M}}\mathrm{d}|\mathbf{j}|_s, \quad \mathbf{j}_t = \int_0^t \mathbf{n}(\bar{\mathbf{Y}}_s)\mathrm{d}|\mathbf{j}|_s,$$

which concludes the proof.

H Configurational modelling of robotic arms under manipulability constraints

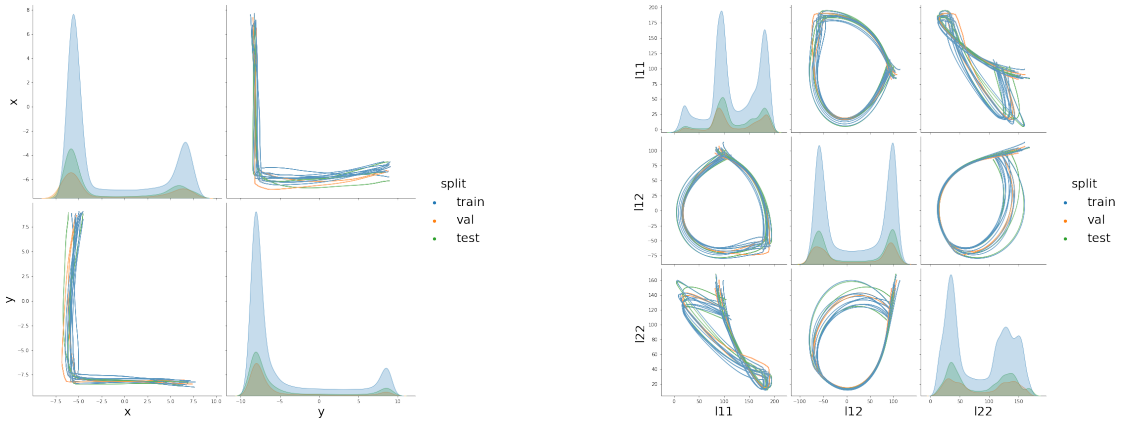
Accurately determining and specifying the movement of a robotic arm and the forces it exerts is a fundamental problem in many real-world robotics applications. A widely-used set of descriptors for modelling the flexibility of a given joint configuration are so-called manipulability ellipsoids (Yoshikawa, 1985), which are kinetostatic performance measures that quantify the ability to move or exert forces along different directions. Jaquier et al. (2021) present a geometric framework to learn trajectories of manipulability ellipsoids by making use of the fact any ellipsoid $M \in \mathbb{R}^N$ is defined by the set of points $\{\mathbf{x}|\mathbf{x}^T\mathbf{A}\mathbf{x} = 1\}$ where $\mathbf{A}$ lies on the manifold of $N \times N$ symmetric positive definite matrices S^N_{++}.

In many practical settings, it is desirable to constrain the minimal or maximal volume of a manipulability ellipsoid to retain motional flexibility or limit the magnitude of the exerted force. This necessitates lower or upper limits on the determinant of $\mathbf{A}$, translating into constraints on S^N_{++}. To model this, we make use of one of the datasets introduced by Jaquier et al. (2021), containing demonstrations of a robotic arm drawing different letters in the plane, providing the respective positional trajectories ($\mathbb{R}^2$) and velocity manipulability ellipsoids (S^2_{++}).

We use the processing routines provided by Jaquier et al. (2021) to interpolate the trajectories into 10^4 distinct points, for each of which we derive the position $\mathbf{x} \in \mathbb{R}^2$ and the PSD matrix

$$\mathbf{A} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \in S^2_{++}.$$

parametrising the velocity manipulability ellipsoid $M \in \mathbb{R}^2$. The resulting data is split into training, validation, and test sets by trajectory and visualised in Figure 17. We add a small amount of Gaussian noise to these trajectories, which is shown as the target distribution in J.2.



(a) (b)

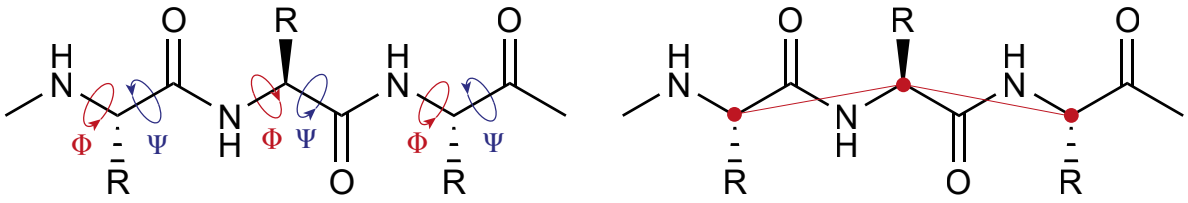
Figure 17: Positional trajectories $\mathbf{x} \in \mathbb{R}^2$ (a) and the parameters l_{11}, l_{12}, l_{22} of the the SPD matrix $\mathbf{A} \in S^2_{++}$ (b) for the letter L.

I Conformational modelling of polypeptide backbones under end point constraints

Polypeptides and proteins constitute an important class of biogenic macromolecules that underpin most aspects of organic life. Accurately modelling their conformational ensembles, i.e. the set of three-dimensional structures they assume under physiological conditions, is essential to both understanding the biological function of existing and designing the enzymatic or therapeutic properties of novel proteins (Lane, 2023). Motivated by the success of diffusion models in computer vision and natural language processing, there has been considerable interest in applying them to learn and sample from distributions over the conformational space of protein structures (Watson et al., 2022; Trippe et al., 2022; Wu et al., 2022).

I.1 Problem parameterisation

Proteins are biopolymers in which a sequence of N amino acids is joined together through $N-1$ peptide bonds, resulting in a so-called polypeptide backbone with protruding amino acid residues. As the deviation of chemical bond lengths and angles from their theoretical optimum is generally negligible, the problem of modelling the three-dimensional structure of this polypeptide chain is often reframed in the space of the internal torsion angles Φ and Ψ (see Figure 18a for an illustration), which can be modelled on a $(2N-2)$-dimensional torus $\mathbb{T}^{2N-2}$.



(a) A commonly-used approximate parameterisation of backbone geometry only considers the C_α torsion angles Φ and Ψ.

(b) As peptide bond orientations can be inferred relatively reliably, researchers often only model the C_α traces.

Figure 18: Standard approaches to modelling the conformations of polypeptide backbones.

In many data-scarce practical settings such as antibody or enzyme design, it is often unnecessary or even undesirable to model the structure of an entire protein, as researchers are primarily interested in specific functional sites with distinct biochemical properties. However, generating conformational ensembles for such substructural elements necessitates positional constraints on their endpoints to ensure that they can be accommodated by the remaining scaffold. While it is conceivable that a diffusion model could derive such constraints from limited experimental data, we argue that it is much more efficient and precise to encode them explicitly.

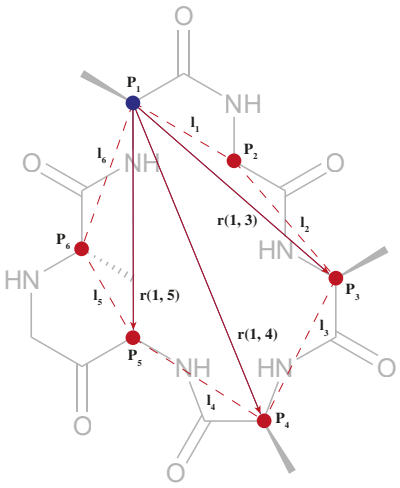
For this purpose, we adopt the distance constraint formulation from Han & Rudolph (2006) and interpret the backbone as a spatial chain with N spherical joints and fixed-length links (see Figure 19a for an illustration). After selecting a suitable anchor point, the geometry of the polypeptide chain is fully specified by (a) the set of link lengths $\ell = \{\ell_j\}_{j=1}^{N}$, (b) the set of vectors $\mathbf{r} = \{r(1,j)\}_{j=2}^{N}$ between the anchor point and each atom in the chain, and (c) the set of dihedral angles

$$\mathbf{T} = \left\{ \arccos\left(\frac{|\langle r(1,j) \times r(1,j+1), r(1,j+1) \times r(1,j+2) \rangle|}{|r(1,j) \times r(1,j+1)||r(1,j+1) r(1,j+2)|} \right) \right\}_{j=2}^{N-2} \in \mathbb{T}^{N-3}$$

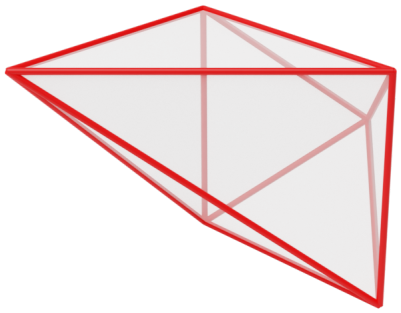
between any three consecutive vectors. After specifying the fixed bond lengths ℓ, including an arbitrary anchor point distance $d_{\text{anchor}} = \ell_N = r(1,N)$, the set of valid vectors $\mathbf{r}$ is given by the convex polytope $\mathbb{P} \subseteq \mathbb{R}^3$ defined by the following linear constraints (see Figure 19b for an illustration):

$$\begin{aligned}
r(1,3) &\leq \ell_1 + \ell_2, \\
-r(1,3) &\leq -|\ell_1 - \ell_2|, \\
\left.\begin{aligned} r(1,j) - r(1,j+1) &\leq \ell_j, \\ -r(1,j) + r(1,j+1) &\leq \ell_j, \\ -r(j) - r(j+1) &\leq -\ell_j, \end{aligned}\right\} & \; 3 \leq j \leq N-2, \\
r(1,N-1) &\leq \ell_{N-1} + d_{\text{anchor}}, \\
-r(1,N-1) &\leq -|\ell_{N-1} - d_{\text{anchor}}|.
\end{aligned}$$

This means that the set of all valid polypeptide backbone conformations is defined by the product manifold $\mathbb{P} \times \mathbb{T}^{N-3}$, enabling us to train diffusion models that exclusively generate conformations with a fixed anchor point distance d_{anchor}.



(a) An illustrative diagram of the proposed parameterisation for modelling the C_α trace geometry of the cyclic peptide c-AAGAGG.



(b) The convex polytope constraining the diagonals of the triangles for the given bond lengths in the illustrated molecule. The total design space is the product of this polytope with the 4D flat torus.

Figure 19: Parameterising the conformational space of polypeptide backbones under anchor point distance constraints.

I.2 Data generation and model training

As a proof-of-concept for the practicality of our methods, we chose to model the conformational distribution of the cyclic peptide c-AAGAGG. Cyclic peptides are an increasingly important drug modality with therapeutic uses ranging from antimicrobials to oncology, exhibiting circular polypeptide backbones (i.e. $d_{\text{anchor}} = 0$) that confer a range of desirable pharmacodynamic and pharmacokinetic properties (Dougherty et al., 2019). To reduce the dimensionality of the problem, we only consider the C_α traces (with fixed C_α-C_α link distances of 3.6 Å) instead of the full polypeptide backbone (see Figure 18b), although we note that our framework is fully applicable to both settings.

To derive a suitable dataset, the product manifold $\mathbb{P} \times \mathbb{T}^3$ describing the conformations of cyclic C_α traces of length $N = 6$ was constructed (see Figures 19a and 19b for an illustration) and used to generate 10^7 uniform samples satisfying the anchor point distance constraint $d_{\text{anchor}} = 0$. Subsequently, an estimate of the free energy E_i of each sample i was obtained by (1) reconstructing the full-atom peptide from each C_α trace using the PULCHRA algorithm (Rotkiewicz & Skolnick, 2008), (2) relaxing all non-C_α backbone and side-chain atoms (keeping the C_α positions fixed), and (3) quantifying the potential energy of each of the resulting conformations using the OpenMM suite of molecular dynamics tools (Eastman et al., 2017), and the AMBER force field (Hornak et al., 2006). These free energy estimates were then used to approximate the Boltzmann distribution over conformational states

$$p_B(i) \propto \exp\left(-\frac{E_i}{k_B T}\right),$$

where temperature was set to $T = 273.15\,\text{K}$ and $k_B = 1.380\,649 \times 10^{-23}\,\text{J}\,\text{K}^{-1}$ is the Boltzmann constant. We then apply a very minor amount of smoothing to the resulting distribution by running forward Brownian motion on both the polytope and the torus for 10 steps, using a small step size of 5×10^{-3} and the respective metrics. Finally, a subsample of 10^6 C_α traces was drawn from this distribution and used for training and evaluating our models.

J Experimental details

In what follows we describe the experimental settings used to generate results introduced in Section 5. The models and experiments have been implemented in Jax (Bradbury et al., 2018), using a modified version of the Riemannian geometry library Geomstats (Miolane et al., 2020).

Architecture. The architecture of the score network $\boldsymbol{s}_\theta$ is given by a multilayer perceptron with 6 hidden layers with 512 units each. We use sinusoidal activation functions.

Training. All models are trained by the stochastic optimizer Adam (Kingma & Ba, 2014) with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, batch-size of 256 data-points. The learning rate is annealed with a linear ramp from 0 to 1000 steps, reaching the maximum value of $2e-4$, and from then with a cosine schedule down to 0 after $100k$ iterations in total.

Diffusion. Following Song et al. (2021), the diffusion models diffusion coefficient is parametrized as $g(t) = \sqrt{\beta(t)}$ with $\beta : \ t \mapsto \beta_{\text{min}} + (\beta_{\text{max}} - \beta_{\text{min}}) \cdot t$, where we found $\beta_{\text{min}} = 0.001$ and $\beta_{\text{max}} = 6$ to work best.

Metrics. We measure the performance of trained models via the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012), which is a kernel based metric between two distributions P and Q. The MMD can be empirically approximated with the following U-statistics $\text{MMD}^2(P, Q) = \frac{1}{m(m-1)} \sum_i \sum_{j \neq i} k(x_i, x_j) + \frac{1}{m(m-1)} \sum_i \sum_{j \neq i} k(y_i, y_j) - 2\frac{1}{m^2} \sum_i \sum_j k(x_i, y_j)$ with $x_i \sim P$ and $y_i \sim Q$, where k is a kernel. For synthetic experiments we use a sum of weighted RBF kernels matching the generating distributions for the Gaussian mixtures. For the other experiments we use an RBF kernel. For all experiments we use 100,000 samples to compute the MMD.

J.1 Synthetic data on polytopes

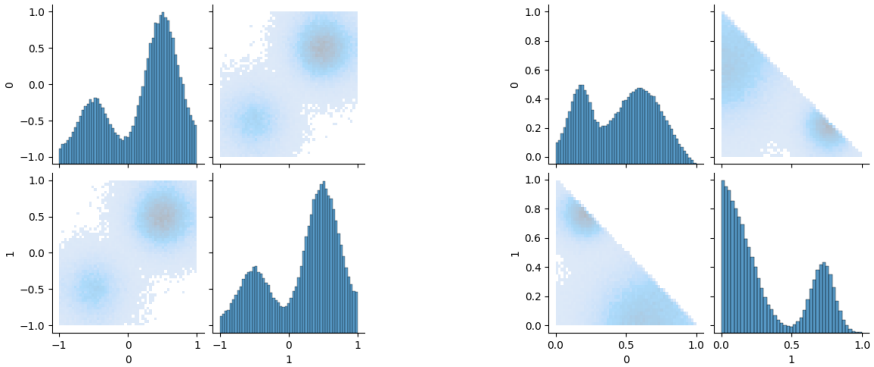
Hypercube $[-1, 1]^n$. The hypercube is a specific instance of a convex polytope where the affine constraints are given by the following coefficients:

$$A = \begin{pmatrix} 1 & \dots & 0 \\ -1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \\ 0 & \dots & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix}.$$

Where A is a $2n \times n$ matrix and b an n dimensional vector.

We construct the training and test datasets by sampling for both 100000 points from a mixture of 'wrapped normal' distributions illustrated in Figure 20a and which density is given by

$$p_0(x) = 0.7\,\text{ReflectedStep}[(0.5, 0.5), \cdot, \{f_i\}_{i\in\mathcal{I}}]\#\mathcal{N}(0, 0.25) + 0.3\,\text{ReflectedStep}[(-0.5, -0.5), \cdot, \{f_i\}_{i\in\mathcal{I}}]\#\mathcal{N}(0, 0.25).$$



(a) On the hypercube. (b) On the simplex.

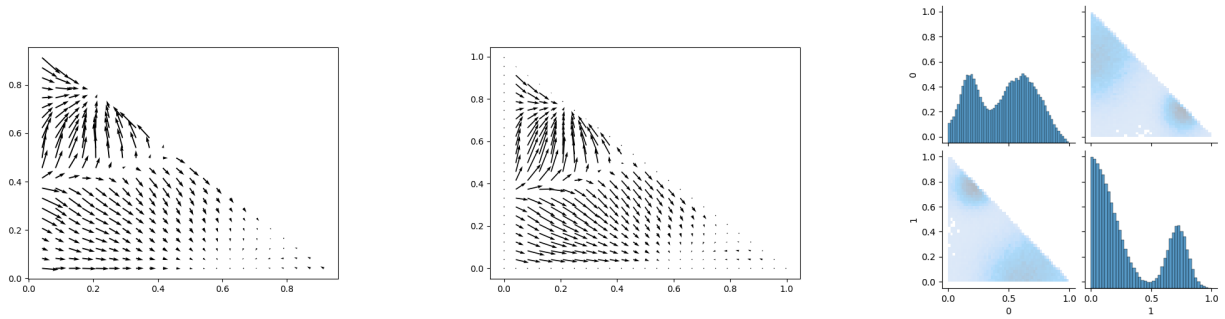
Figure 20: Pairwise and marginals samples from the synthetic data distribution.

Simplex Δ^n. Similarly, to parameterise the simplex as a convex polytope we set the matrix and constraints to be given by

$$A = \begin{pmatrix} -1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

Where A will be a $n - 1 \times n$ matrix. Essentially we perform diffusion over the first $n - 1$ components of the simplex, allowing the last component to be determined by the one minus the sum of the first $n - 1$.

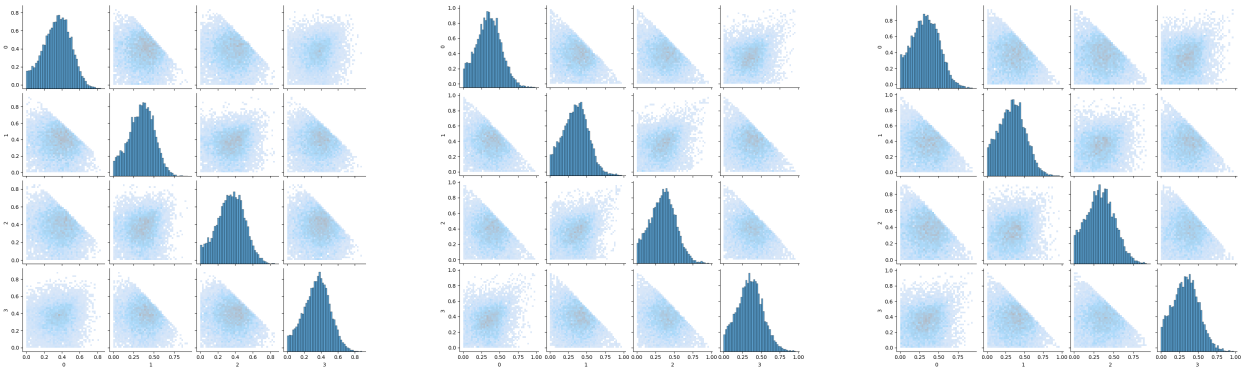
Similarly than for the hypercube, we construct the training and test datasets from generated data points which are illustrated in Figure 20b. The score network at different times is illustrated in Figure 21.



(a) $t = 0.01$. (b) $t = 0.5$. (c) Generated distribution.

Figure 21: Evolution of the score on the simplex and generated distribution.

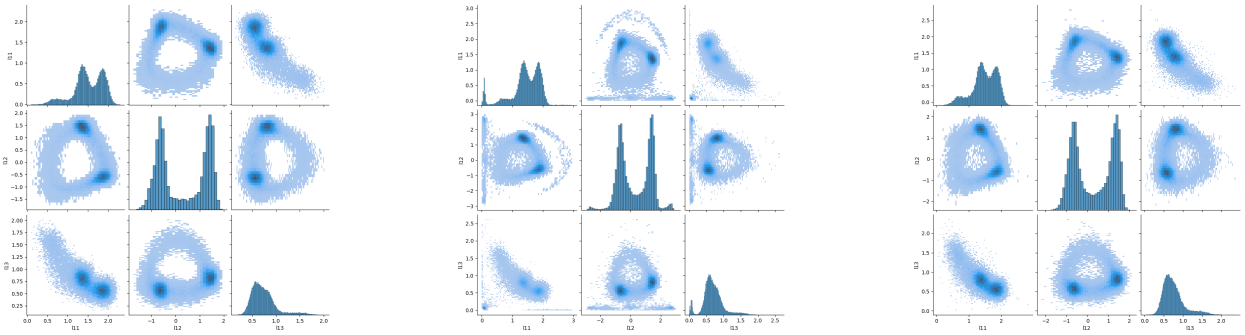
The Birkhoff polytope. The Birkhoff polytope is the space of doubly stochastic matrices, i.e. $B_n = \{\mathrm{P} \in [0,1]^{n \times n} : \sum_i^n P_{i,j} = 1, \sum_j^n P_{i,j} = 1\}$. It is a convex polytope in $\mathbb{R}^{n^2}$ and has dimension $d = (n-1)^2$.



(a) Data. (b) Log-barrier. (c) Reflected.

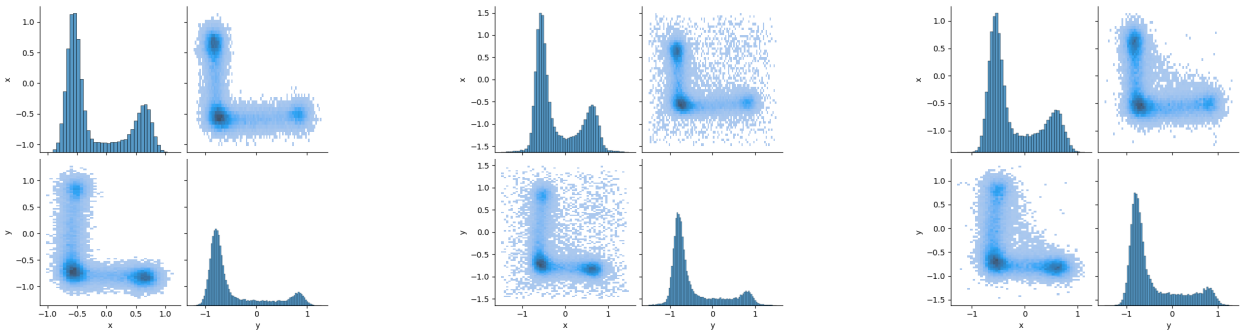
Figure 22: Pairwise and marginals samples on the Birkhoff polytope from synthetic data distribution and from trained constrained diffusion models.

J.2 Constrained SPD matrices for robotic arms modelling



(a) Data. (b) Log-barrier. (c) Reflected.

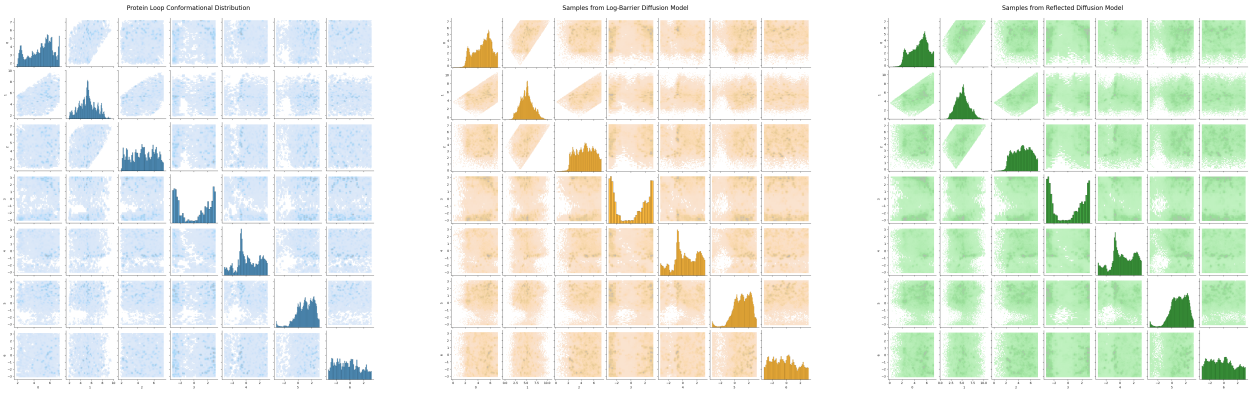
Figure 23: Pairwise and marginals distributions over the coefficients L_{11}, L_{21}, L_{22} of the lower triangle matrix parameterising SPD matrices $M = LL^\top$ (which represent the manipulability ellipsoids of the robotic arms).



(a) Data. (b) Log-barrier. (c) Reflected.

Figure 24: Pairwise and marginals distributions over the (x, y) locations of the robotic arms.

J.3 Conformational modelling of polypeptide backbones under anchor point constraints



(a) Data. (b) Log-barrier. (c) Reflected.

Figure 25: Pairwise and marginals distributions over the dimensions of the polytope and torus used to model the conformational ensembles of cyclic peptides generated by the reflected diffusion model.

.

Appendix C

Supplementary to *Metropolis Sampling for Constrained Diffusion*

Supplementary to:
Metropolis Sampling for Constrained Diffusion Models

A Overview

In Appendix B, we recall some basic concepts of Riemannian geometry which are key to defining discretisations of the reflected Brownian motion. In Appendix C, we give some details on the reflection step in reflected discretizations. In Appendix D, we prove the convergence of the rejection and Metropolis discretizations to the true reflected Brownian Motion. The geospatial dataset with non-convex constraints based on wildfire incidence rates in the continental United States is presented Appendix E. All supplementary experimental details and empirical results are gathered in Appendix F.

B Manifold concepts

In the following, we aim to introduce key concepts that underpin diffusion models on Riemannian manifolds, with a particular focus on notions relevant to the reflected Brownian motion that we build on in Appendix C. For a more thorough treatment with reference to reflected diffusion models, we refer to (Fishman et al., 2023). For a detailed presentation of smooth manifolds, see Lee (2013).

A Riemannian manifold is a tuple $(\mathcal{M}, \mathfrak{g})$ with $\mathcal{M}$ a smooth manifold and $\mathfrak{g}$ a metric that imbues the manifold with a notion of distance and curvature and is defined as a smooth positive-definite inner product on each of the tangent spaces of the manifold:

$$\mathfrak{g}(p) : \mathrm{T}_p\mathcal{M} \times \mathrm{T}_p\mathcal{M} \to \mathbb{R}.$$

The tangent space T_p of a point p on a manifold is an extension of the notion of tangent planes and can be thought of as the space of derivatives of scalar functions on the manifold at that point.

To establish how different tangent spaces relate to one another, we need to additionally introduce the concept of a *connection*. This is a map that takes two vector fields and produces a derivative of the first with respect to the second, typically written as $\nabla(X, Y) = \nabla_X Y$. While there are infinitely many connections on any given manifold, the *Levi-Cevita* emerges as a natural choice if we impose the following two conditions:

(i) $X \cdot (\mathfrak{g}(Y, Z)) = \mathfrak{g}(\nabla_X Y, Z) + \mathfrak{g}(Y, \nabla_X Z)$,
(ii) $[X, Y] = \nabla_X Y - \nabla_Y X$,

where $[\cdot, \cdot]$ is the Lie bracket. These conditions ensure that the connection is (i) metric-preserving and (ii) torsion-free, with the latter guaranteeing a unique connection and integrability on the manifold.

Using the metric and Levi-Cevita connection, we can define a number of key concepts:

Geodesic. *Geodesics* extend the Euclidean notion of 'straight lines' to manifolds. They are defined as the unique path $\gamma : (0, 1) \to \mathcal{M}$ such that $\nabla_{\gamma'}\gamma' = 0$ and are the shortest path between two points on a manifold, in the sense that $L(\gamma) = \int_0^1 \sqrt{\mathfrak{g}(\gamma(t))(\gamma'(t), \gamma(t))}\mathrm{d}t$ is minimal.

Exponential map. The *exponential map* on a manifold is given by the mapping between an element $\boldsymbol{v} \in T_p\mathcal{M}$ of the tangent space at point p and the endpoint of the unique geodesic γ with $\gamma(0) = p$ and $\gamma'(0) = \boldsymbol{v}$.

Intersection. The *intersection* along a geodesic is the first point at which the geodesic intersects the boundary. We recall that the boundary is defined by $f = 0$. We can define this via an optimisation problem: compute the minimum $t > 0$ such that we have that $\exp_{\mathfrak{g}}(x, tz)$ is a root of f: $f(\exp_{\mathfrak{g}}(x, tz)) = 0$. We will say that $\exp_{\mathfrak{g}}(x, tz) = \mathrm{intersect}_{\mathfrak{g}}(x, z; f)$ and that $t = \arg\mathrm{intersect}_{\mathfrak{g}}(x, z; f)$.

Parallel transport. We say that a vector field X is *parallel* to a curve $\gamma : (0,1) \to \mathcal{M}$ if $\nabla_{\gamma'} X = 0$, where $\gamma' : (0,1) \to \mathrm{T}_{\gamma(t)}\mathcal{M}$. For two points on the manifold $p, q \in \mathcal{M}$ that are connected by a curve γ, and an initial vector $X_0 \in \mathrm{T}_p\mathcal{M}$, there is a unique vector field X that is parallel to γ such that $X(p) = X_0$. This induces a map between the tangent spaces at p and q $\tau_\gamma : \mathrm{T}_p\mathcal{M} \to \mathrm{T}_q\mathcal{M}$, which is referred to as the *parallel transport* of tangent vectors between p and q and satisfies the condition that for $\boldsymbol{v}, \boldsymbol{u} \in \mathrm{T}_p\mathcal{M}$ $\mathfrak{g}(p)(\boldsymbol{v}, \boldsymbol{u}) = \mathfrak{g}(q)(\tau_\gamma(\boldsymbol{v}), \tau_\gamma(\boldsymbol{u}))$.

Reflection. For an element $\boldsymbol{v} \in T_p\mathcal{M}$ in the tangent space of the manifold at point p and a constraint characterised by its unit normal vector $\boldsymbol{n} \in T_p\mathcal{M}$, the *reflection* of $\boldsymbol{v}$ in the tangent space is given by $\boldsymbol{v}' = \boldsymbol{v} - 2\mathfrak{g}(\boldsymbol{v}, \boldsymbol{n})\boldsymbol{n}$.

C Full Reflected Discretisation

Here, we reproduce the central algorithm for the full discretisation of the reflected Brownian motion (Algorithm 4) derived for Euclidean models in (Lou et al., 2023) and for Reimannian models in (Fishman et al., 2023). Its central component is the *Reflected Step Algorithm* (Algorithm 3), which gives a generic computation for the reflection in any manifold. Due to the need to balance speed and numerical instability issues around the boundary, an efficient practical implementation of the reflected step is highly non-trivial, even for simple polytopes in Euclidean space. More complex geometries and boundaries make this problem significantly worse: a constraint on the trace of SPD matrices under the log-Cholesky metric of (Lin, 2019) requires solving complex non-convex optimisation problems for each sample at each discretised sampling step in both the forward and reverse process. This motivates our work in this paper.

These problems motivated the development of our Metropolis approximation, which significantly simplifies the random walk. Instead of requiring the intersection, parallel transport and reflection, we simply need to be able to evaluate the constraint functions f_i. We highlight this simplicity in Algorithm 5.

Algorithm 3 *Reflected Step Algorithm.* The algorithm operates by repeatedly taking geodesic steps until one of the constraints is violated, or until the step is fully taken. Upon hitting the boundary, we parallel-transport the tangent vector to the boundary and then reflect it against it. We then start a new geodesic from this point in the new direction. The $\mathrm{arg\,intersect}_t$ function computes the distance one must travel along a geodesic in direction $\boldsymbol{s}$ until constraint f_i is violated. For a discussion of $\mathrm{paralleltransport}$, $\exp_{\mathfrak{g}}$ and $\mathrm{reflect}$ see appendix B.

Require: $x \in \mathcal{M}$, $\boldsymbol{v} \in \mathrm{T}_x\mathcal{M}$, $\{f_i\}_{i \in \mathcal{I}}$
  $\ell \leftarrow \|\boldsymbol{v}\|_{\mathfrak{g}}$
  $\boldsymbol{s} \leftarrow \boldsymbol{v}/\|\boldsymbol{v}\|_{\mathfrak{g}}$
  while $\ell \geq 0$ **do**
    $d_i = \mathrm{arg\,intersect}_{\mathfrak{g}}(x, z; f_i)$
    $i \leftarrow \arg\min_i \; d_i$
    $\alpha \leftarrow \min(d_i, \ell)$
    $x' \leftarrow \exp_{\mathfrak{g}}(x, \alpha \boldsymbol{s})$
    $\boldsymbol{s} \leftarrow \mathrm{paralleltransport}_{\mathfrak{g}}(x, \boldsymbol{s}, x')$
    $\boldsymbol{s} \leftarrow \mathrm{reflect}(\boldsymbol{s}, f_i)$
    $\ell \leftarrow \ell - \alpha$
    $x \leftarrow x'$
  end while
  return x

Algorithm 4 *Reflected Random Walk*. Discretisation of the SDE $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t - \mathrm{d}\mathbf{k}_t$.

Require: $T, N, X_0^\gamma, \{f_i\}_{i\in\mathcal{I}}$
$\gamma = T/N$
for $k \in \{0, \dots, N-1\}$ **do**
  $Z_{k+1} \sim \mathrm{N}(0, \mathrm{Id})$
  $X_{k+1}^\gamma = \text{ReflectedStep}[X_k^\gamma, \sqrt{\gamma} Z_{k+1}, \{f_i\}_{i\in\mathcal{I}}]$
end for
return $\{X_k^\gamma\}_{k=0}^N$

Algorithm 5 *Metropolis Random Walk*. Discretisation of the SDE $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t - \mathrm{d}\mathbf{k}_t$.

Require: $T, N, X_0^\gamma, \{f_i\}_{i\in\mathcal{I}}$
$\gamma = T/N$
for $k \in \{0, \dots, N-1\}$ **do**
  $Z_{k+1} \sim \mathrm{N}(0, \mathrm{Id})$
  $X' \leftarrow \exp_{\mathfrak{g}}\big(X_k^\gamma, \sqrt{\gamma} Z_{k+1}\big)$
  if $\max_{i\in\mathcal{I}} f_i(X') \leq 0$ **then**
    $X_{k+1}^\gamma = X'$
  else
    $X_{k+1}^\gamma = X_k^\gamma$
  end if
end for
return $\{X_k^\gamma\}_{k=0}^N$

D Convergence to the reflected process

In this note, we assume that $\mathcal{M} = \{x \in \mathbb{R}^d \,:\, \Phi(x) > 0\}$ is compact, with $\Phi \in \mathrm{C}^2(\mathbb{R}^d, \mathbb{R})$. We have that $\partial\mathcal{M} = \{x \in \mathbb{R}^d \,:\, \Phi(x) = 0\}$. In addition, we assume that for any $x \in \partial\mathcal{M}$, $\|\nabla\Phi(x)\| = 1$ and that Φ is concave. The closure of $\mathcal{M}$ is denoted $\overline{\mathcal{M}}$. The assumption that Φ is concave is only used in Theorem 7-(d) and can be dropped. We consider it for simplicity.

Let $(\hat{X}_k^\gamma)_{k\in\mathbb{N}}$ given for any $\gamma > 0$ and $k \in \mathbb{N}$ by $\hat{X}_0^\gamma = x \in \overline{\mathcal{M}}$ and for $\hat{X}_{k+1}^\gamma = \hat{X}_k^\gamma + \sqrt{\gamma} Z_k^\gamma$ with Z_k^γ a Gaussian random variable conditioned on $\hat{X}_k^\gamma + \sqrt{\gamma} Z_k^\gamma \in \overline{\mathcal{M}}$. In practice, Z_k^γ is sampled using rejection sampling. We define $\hat{\mathbf{X}}^\gamma : \mathbb{R}_+ \to \overline{\mathcal{M}}$ given for any $k \in \mathbb{N}$ by $\hat{\mathbf{X}}_{k\gamma}^\gamma = \hat{X}_k^\gamma$ and for any $t \in [k\gamma, (k+1)\gamma)$, $\hat{\mathbf{X}}_t^\gamma = \hat{X}_k^\gamma$. Note that $(\mathbf{X}_t)_{t\in[0,T]}$ is a $\mathrm{D}([0,T], \overline{\mathcal{M}})$ valued random variable, where $\mathrm{D}([0,T], \overline{\mathcal{M}})$ is the space of right-continuous with left-limit processes which take values in $\overline{\mathcal{M}}$. We denote $\hat{\mathbb{P}}^\gamma$ the distribution of $(\hat{\mathbf{X}}_t^\gamma)_{t\in[0,T]}$ on $\mathrm{D}([0,T], \overline{\mathcal{M}})$.

Our goal is to show the following theorem.

Theorem 6. *For any $T \geq 0$, $(\hat{\mathbf{X}}_t^\gamma)_{t\in[0,T]}$ weakly converges to $(\mathbf{X}_t)_{t\in[0,T]}$ such that for any $t \in [0,T]$*

$$\mathbf{X}_t = x + \mathbf{B}_t - \mathbf{k}_t, \qquad |\mathbf{k}|_t = \int_0^t \mathbb{1}_{\mathbf{X}_s \in \partial\mathcal{M}} \mathrm{d}|\mathbf{k}|_s, \qquad \mathbf{k}_t = \int_0^t \mathbf{n}(\mathbf{X}_s)\mathrm{d}|\mathbf{k}|_s. \tag{6}$$

Proof. In order to prove the result, we prove that the distribution of the Markov chain seen as an element of $\mathrm{D}([0,T], \overline{\mathcal{M}})$ converges to a solution of the Skorokhod problem (6). In particular, we first show that the limiting distribution satisfies a submartingale problem following (Stroock et al., 1971, Theorem 6.3). The transition from a solution of a submartingale problem to a weak solution of the Skorokhod problem is given by (Kang et al., 2017, Theorem 1, Proposition 2.12) and (Ramanan, 2006, Corollary 2.10). In order to apply (Stroock et al., 1971, Theorem 6.3), we define an intermediate drift and diffusion matrix, see (55) and (51). To prove the theorem one needs to control the drift and diffusion matrix inside $\mathcal{M}$ and show that it converges to 0 and Id respectively. The technical part of the proof comes from the control of the drift coefficient near the boundary. In particular, we show that if the intermediate drift is large then we are close to the boundary and the intermediate drift is pointing inward. To investigate the local properties of the drift near the boundary we rely on the notion of tubular neighborhood, see (Lee et al., 2012, Theorem 6.24). □

Some key properties of the tubular neighborhood are stated in Appendix D.1. We then establish a few technical lemmas about the tail probability of some distributions in Appendix D.2. Controls on the diffusion matrix and lower bounds on the probability of belonging in $\mathcal{M}$ are given in Appendix D.3. Properties of large drift terms are given in Appendix D.4. The convergence of the drift and diffusion matrix on compact sets is given in Appendix D.5. The convergence of the boundary terms is investigated in Appendix D.6. Finally, we conclude the proof in Appendix D.7.

D.1 Properties of the tubular neighborhood

Using the results of (Lee et al., 2012), we establish the existence of an open set of $\overline{\mathcal{M}}$ (for the induced topology of $\mathbb{R}^d$) satisfying several important properties.

Theorem 7. *There exist $\mathsf{U} \subset \overline{\mathcal{M}}$ open and $C \geq 1, \bar{r} > 0$ such that for any $\gamma \in (0, \bar{\gamma})$ with $\bar{\gamma} = 1$ the following properties hold:*

(a) For any $x \in \mathsf{U}$, there exist a unique $\bar{x} \in \partial\mathcal{M}$ and $\bar{\alpha} > 0$ such that $x = \bar{x} + \bar{\alpha}\nabla\Phi(\bar{x})$.

(b) For any $\bar{\alpha} \in [0, \bar{r}]$ and $\bar{x} \in \partial\mathcal{M}$ such that $\bar{x} + \bar{\alpha}\nabla\Phi(\bar{x}) \in \overline{\mathcal{M}}$, let $x = \bar{x} + \bar{\alpha}\nabla\Phi(\bar{x})$ and $\mathsf{C}(x,\gamma)$ such that $x + \sqrt{\gamma} z \in \mathsf{C}(x,\gamma)$ if

$$-\bar{\alpha}\gamma^{-1/2} \leq \alpha < \bar{r}\gamma^{-1/2}, \qquad \|v\|^2 \leq (\alpha\gamma^{1/2} + \bar{\alpha})/(C\gamma), \tag{7}$$

with $z = \alpha\nabla\Phi(\bar{x}) + v$, with $v \perp \nabla\Phi(\bar{x})$. Then $\mathsf{C}(x,\gamma) \subset \overline{\mathcal{M}}$.

(c) $\mathsf{V} = \{\bar{x} + \alpha\nabla\Phi(\bar{x}) \,:\, \bar{x} \in \partial\mathcal{M},\ \alpha \in [0, \bar{r})\}$ is open in $\overline{\mathcal{M}}$.

(d) For any $x \in \mathsf{U}$, $x + \sqrt{\gamma} z \in \overline{\mathcal{M}} \cap \mathsf{C}(x,\gamma)^{\mathrm{c}}$ then $\alpha \geq \bar{r}\gamma^{-1/2}$ or $\|v\|^2 \geq (\alpha\gamma^{1/2} + \bar{\alpha})/(C\gamma)$ and $\alpha\gamma^{1/2} + \bar{\alpha} \geq 0$, with $z = \alpha\nabla\Phi(\bar{x}) + v$, with $\bar{x}$ given in (a) and $v \perp \nabla\Phi(\bar{x})$. .

(e) There exists $R > 0$ such that $\{x \in \overline{\mathcal{M}} \,:\, d(x, \partial\mathcal{M}) \leq 2R\} \subset \mathsf{V}$.

Proof. Let $\gamma \in (0, \bar{\gamma})$ with $\bar{\gamma} = 1$. First, note that for any $\bar{x} \in \partial\mathcal{M}$, the normal space is given by $\{\alpha\nabla\Phi(\bar{x}) \,:\, \alpha \in \mathbb{R}\}$. Using this result and (Lee et al., 2012, Theorem 6.24) there exists $\tilde{r}_0 > 0$ such that $\mathsf{U}_0 = \{\bar{x} + \alpha\nabla\Phi(\bar{x}) \,:\, \bar{x} \in \partial\mathcal{M},\ \alpha \in (-\tilde{r}_0, \tilde{r}_0)\} \subset \mathbb{R}^d$ is open[2]. We have that for any $\alpha \in [-r_0, 0)$ and $\bar{x} \in \partial\mathcal{M}$

$$\Phi(\bar{x} + \alpha\nabla\Phi(\bar{x})) = \Phi(\bar{x}) + \alpha\|\nabla\Phi(\bar{x})\|^2 + \textstyle\int_0^1 \nabla^2\Phi(\bar{x} + t\alpha\nabla\Phi(\bar{x}))(\alpha\nabla\Phi(\bar{x}))^{\otimes 2}\mathrm{d}t \tag{8}$$

$$\leq \alpha + \tilde{C}_0\alpha^2 < 0, \tag{9}$$

with $r_0 = \min(\tilde{r}_0, 1/(2\tilde{C}_0))$, where we have used that $\Phi(\bar{x}) = 0$, $\|\nabla\Phi(\bar{x})\| = 1$ and defined $\tilde{C}_0 = \sup\{\|\nabla^2\Phi(\bar{x} + \alpha\nabla\Phi(\bar{x}))\| \,:\, \bar{x} \in \partial\mathcal{M},\ \alpha \in [-\tilde{r}_0, \tilde{r}_0]\}$. Reciprocally, we have for any $\alpha \in [0, r_0)$ and $\bar{x} \in \partial\mathcal{M}$

$$\Phi(\bar{x} + \alpha\nabla\Phi(\bar{x})) = \Phi(\bar{x}) + \alpha\|\nabla\Phi(\bar{x})\|^2 + \textstyle\int_0^1 \nabla^2\Phi(\bar{x} + t\alpha\nabla\Phi(\bar{x}))(\alpha\nabla\Phi(\bar{x}))^{\otimes 2}\mathrm{d}t \geq \alpha - C_0\alpha^2, \tag{10}$$

where we have used that $\Phi(\bar{x}) = 0$, $\|\nabla\Phi(\bar{x})\| = 1$ and defined $C_0 = \sup\{\|\nabla^2\Phi(\bar{x} + \alpha\nabla\Phi(\bar{x}))\| \,:\, \bar{x} \in \partial\mathcal{M},\ \alpha \in [-r_0, r_0]\}$. Let $r_1 = \min(r_0, 1/(2C_0))$. Then, $\mathsf{U}_1 = \{\bar{x} + \alpha\nabla\Phi(\bar{x}) \,:\, \bar{x} \in \partial\mathcal{M},\ \alpha \in (-r_1, r_1)\} \subset \mathbb{R}^d$ is open and

$$\mathsf{U}_1 \cap \overline{\mathcal{M}} = \{\bar{x} + \alpha\nabla\Phi(\bar{x}) \,:\, \bar{x} \in \partial\mathcal{M},\ \alpha \in [0, r_1)\}. \tag{11}$$

In what follows, we define $\mathsf{U} = \mathsf{U}_1 \cap \overline{\mathcal{M}}$. Note that U is open for the induced topology and that $\partial\mathcal{M} \subset \mathsf{U}$. In particular, $\partial\mathcal{M}$ is compact, U^{c} is closed and $\partial\mathcal{M} \cap \mathsf{U}^{\mathrm{c}} = \varnothing$. Hence, there exists $r > 0$ such that $d(\partial\mathcal{M}, \mathsf{U}^{\mathrm{c}}) \geq 4r$. Without loss of generality we can assume that $r \leq 1/2$. We also have $\{x \in \overline{\mathcal{M}} \,:\, d(x, \partial\mathcal{M}) \leq 2r\} \subset \mathsf{U}$. The proof of (a) follows from the definition of U_0. In the rest of the proof, we define

$$C^{1/2} = 2\max(1, \sup\{\|\nabla^2\Phi(\bar{x}+u)\| \,:\, \bar{x} \in \partial\mathcal{M},\ \|u\|^2 \leq r(r+1)\}), \quad \bar{r} = \min(1/(2C^{1/2}), r/2). \tag{12}$$

Let us prove (b). Consider $x + \sqrt{\gamma}z \in \mathsf{C}(x, \gamma)$ with $\mathsf{C}(x, \gamma)$ given by (7) and $x = \bar{x} + \bar{\alpha}\nabla\Phi(\bar{x})$ and $z = \alpha\nabla\Phi(\bar{x}) + v$ with $v \perp \nabla\Phi(\bar{x})$. In particular, we recall that we have

$$-\bar{\alpha}\gamma^{-1/2} \leq \alpha < \bar{r}\gamma^{-1/2}, \qquad \|v\|^2 \leq (\alpha\gamma^{1/2} + \bar{\alpha})/(C\gamma). \tag{13}$$

This implies that

$$\bar{\alpha} + \sqrt{\gamma}\alpha \leq 2\bar{r}, \qquad \gamma\|v\|^2 \leq 2\bar{r}/C. \tag{14}$$

First, using that $C \geq 1$, $\|\nabla\Phi(\bar{x})\| = 1$, (14) and (12), we have

$$\|x + \sqrt{\gamma}z - \bar{x}\|^2 = (\bar{\alpha} + \sqrt{\gamma}\alpha)^2 + \gamma\|v\|^2 \leq r^2 + r/C \leq r(r+1). \tag{15}$$

Then, we have that

$$\Phi(x + \sqrt{\gamma}z) = \Phi(\bar{x}) + \bar{\alpha} + \sqrt{\gamma}\alpha + \textstyle\int_0^1 \nabla^2\Phi(\bar{x} + t(x + \sqrt{\gamma}z - \bar{x}))(x + \sqrt{\gamma}z - \bar{x})^{\otimes 2}\mathrm{d}t \tag{16}$$

$$\geq \bar{\alpha} + \sqrt{\gamma}\alpha - (C^{1/2}/2)((\bar{\alpha} + \sqrt{\gamma}\alpha)^2 + \gamma\|v\|^2), \tag{17}$$

where we recall that

$$C^{1/2} = 2\max(1, \sup\{\|\nabla^2\Phi(\bar{x}+u)\| \,:\, \bar{x} \in \partial\mathcal{M},\ \|u\|^2 \leq r(r+1)\}), \qquad \bar{r} = \min(1/(2C^{1/2}), r/2). \tag{18}$$

First, using that $r \leq 1/2$ and (14), we have $\bar{\alpha} + \sqrt{\gamma}\alpha \leq 2r \leq 1$. Since, $\|v\|^2 \leq (\bar{\alpha} + \sqrt{\gamma}\alpha)/(C\gamma)$ and we have that $\|v\|^2 < 1/(C\gamma)$. Let $P(X) = X - (C^{1/2}/2)X^2 - (C^{1/2}/2)\gamma\|v\|^2$. We have that $P(x) \geq 0$ if and only if $x \in [x_{\min}, x_{\max}]$ with

$$x_{\min} = (1 - (1 - C\gamma\|v\|^2)^{1/2})/C^{1/2}, \qquad x_{\max} = (1 + (1 - C\gamma\|v\|^2)^{1/2})/C^{1/2}. \tag{19}$$

Using that for any $t \in (0, 1)$, $(1-t)^{1/2} \geq 1 - t$ we have that

$$x_{\min} \leq \gamma C\|v\|^2/2, \qquad x_{\max} \geq 1/C^{1/2}. \tag{20}$$

[2]This is the tubular neighborhood theorem which is key to the rest of the proof.

Since $\|v\|^2 \leq (\sqrt{\gamma}\alpha + \bar{\alpha})/(\gamma C)$, we have that $\bar{\alpha} + \sqrt{\gamma}\alpha \geq x_{\min}$. In addition, using that $\bar{\alpha} + \sqrt{\gamma}\alpha \leq 2\bar{r} \leq 1/C^{1/2} \leq x_{\max}$, we get that $P(\bar{\alpha} + \sqrt{\gamma}\alpha) \geq 0$ and therefore $x + \sqrt{\gamma}z \in \overline{\mathcal{M}}$ since $\Phi(x + \sqrt{\gamma}z) \geq 0$. This concludes the proof of (b). Note that the condition $\alpha \geq -\gamma^{-1/2}\bar{\alpha}$ is implied by the condition $\|v\|^2 \leq (\sqrt{\gamma}\alpha + \bar{\alpha})/(\gamma C)$. Using that $\mathsf{V} \subset \{x \in \overline{\mathcal{M}} : d(x, \partial\mathcal{M}) \leq 2r\} \subset \mathsf{U}$, (c) is a direct consequence of (Lee et al., 2012, Theorem 6.24)]. Next, we prove (d). Let $x + \sqrt{\gamma}z \in \overline{\mathcal{M}} \cap \mathsf{C}(x,\gamma)^{\mathrm{c}}$. If $\alpha < -\bar{\alpha}\gamma^{-1/2}$ then since Φ is concave, we have

$$\Phi(x + \sqrt{\gamma}z) = \Phi(\bar{x}) + \bar{\alpha} + \sqrt{\gamma}\alpha + \int_0^1 \nabla^2\Phi(\bar{x} + t(x + \sqrt{\gamma}z - \bar{x}))(x + \sqrt{\gamma}z - \bar{x})^{\otimes 2}\mathrm{d}t < 0, \tag{21}$$

where we have used that $\Phi(\bar{x}) = 0$. This is absurd, hence either $\alpha \geq \bar{r}\gamma^{-1/2}$ or $\|v\|^2 \geq (\alpha\gamma^{1/2} + \bar{\alpha})/(C\gamma)$ and $\alpha\gamma^{1/2} + \bar{\alpha} \geq 0$, which concludes the proof. The proof of (e) is similar to the proof that $\{x \in \overline{\mathcal{M}} : d(x, \partial\mathcal{M}) \leq 2r\} \subset \mathsf{U}$. □

The main message of Theorem 7 is that using Theorem 7-(d), if we move in the direction of $\nabla\Phi(\bar{x})$ (the inward normal) with magnitude α then we are allowed to move in the orthonormal direction with magnitude $\alpha^{1/2}$. In the next paragraph, we discuss this fact in details and shows it is necessary for the rest of our study.

The necessity of Theorem 7-(b). At first sight one can wonder if the statement of Theorem 7-(b) could be simplify. In particular, it would be simpler to replace this statement with: for any $\bar{\alpha} \in [0, \bar{r}]$ and $\bar{x} \in \partial\mathcal{M}$ such that $\bar{x} + \bar{\alpha}\nabla\Phi(\bar{x}) \in \overline{\mathcal{M}}$, let $x = \bar{x} + \bar{\alpha}\nabla\Phi(\bar{x})$ and $\mathsf{C}(x,\gamma)$ such that $x + \sqrt{\gamma}z \in \mathsf{C}(x,\gamma)$ if

$$-\bar{\alpha}\gamma^{-1/2} \leq \alpha < \bar{r}\gamma^{-1/2}, \qquad \|v\|^2 \leq (\alpha\gamma^{1/2} + \bar{\alpha})^2/(C\gamma), \tag{22}$$

with $z = \alpha\nabla\Phi(\bar{x}) + v$, with $v \perp \nabla\Phi(\bar{x})$. Then $\mathsf{C}(x,\gamma) \subset \overline{\mathcal{M}}$. Note that $\|v\|^2 \leq (\alpha\gamma^{1/2} + \bar{\alpha})/(C\gamma)$ is replaced by $\|v\|^2 \leq (\alpha\gamma^{1/2} + \bar{\alpha})^2/(C\gamma)$, see Figure 8 for an illustration. However, in that case Theorem 7-(d) becomes: in addition, if $x + \sqrt{\gamma}z \in \overline{\mathcal{M}} \cap \mathsf{C}(x,\gamma)^{\mathrm{c}}$ then $\alpha \geq r\gamma^{-1/2}$ or $\|v\|^2 \geq (\alpha\gamma^{1/2} + \bar{\alpha})^2/(C\gamma)$ and $\alpha\gamma^{1/2} + \bar{\alpha} \geq 0$.

In what follows, when controlling the properties of large drift, see the proof of Proposition 18 and the proof of Proposition 21, we need to control quantities of the form $\mathbb{P}(x + \sqrt{\gamma}Z \in \mathsf{C}(x,\gamma)^{\mathrm{c}} \cap \overline{\mathcal{M}})/\sqrt{\gamma}$[3] Using the original Theorem 7-(d) it is possible to show that this quantity is bounded. However, if one uses the updated version of Theorem 7-(d) then one needs to show that there exists $M \geq 0$ and $\bar{\gamma} > 0$ such that for any $\gamma \in (0, \bar{\gamma})$ (here we have assumed that $\bar{\alpha} = 0$, i.e. $x \in \partial\mathcal{M}$ for simplicity)

$$\int_0^{r/\gamma^{-1/2}} \int_{\nabla\Phi(\bar{x})^\perp} \mathbb{1}_{\|v\|^2 \geq \alpha^2} \varphi(v)\varphi(\alpha)\mathrm{d}v\mathrm{d}\alpha \leq M\sqrt{\gamma}, \tag{23}$$

which is absurd.

D.2 Technical lemmas

We start with a few technical lemmas which will allow us to control some Gaussian probabilities outside of a compact set. We denote $\Psi : \mathbb{R}_+ \times \mathbb{N} \to [0,1]$ such that for any $k \in \mathbb{N}$, $\Psi(\cdot, k)$ is the tail probability of a χ-squared random variable with parameter k, i.e. for any $k \in \mathbb{N}$ and $t \geq 0$ we have

$$\Psi(t,k) = \mathbb{P}(\|Z\|^2 \geq t), \tag{24}$$

with Z a Gaussian random variable in $\mathbb{R}^k$ with zero mean and identity covariance matrix. We will make extensive use of the following lemma which is a direct consequence of (Laurent et al., 2000, Section 4, Lemma 1).

Lemma 8. *For any $k \in \mathbb{N}$ and $t \in \mathbb{R}_+$ with $t \geq 5k$, $\Psi(t,k) \leq \exp[-t/5]$.*

Proof. Let $k \in \mathbb{N}$. First, note that for any $x \geq k$, we have that $k + 2(kx)^{1/2} + 2x \leq 5x$. Combining this result and (Laurent et al., 2000, Section 4, Lemma 1, Equation (4.3)), we have that for any $x \geq k$

$$\mathbb{P}(\|X\|^2 \geq 5x) \leq \exp[-x], \tag{25}$$

with X a $\mathbb{R}^k$-valued Gaussian random variable with zero mean and identity covariance matrix. This concludes the proof upon letting $t = 5x$. □

[3] The division by $\sqrt{\gamma}$ comes from the definition of the intermediate drift (55).

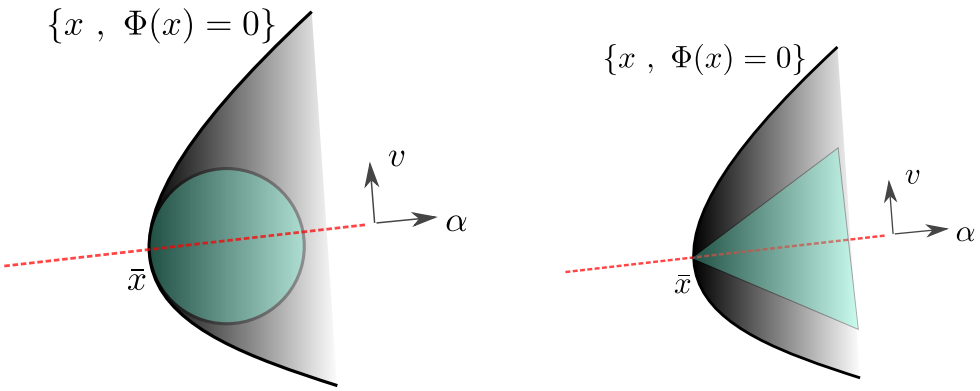


Figure 8: The grey shaded area represents $\overline{\mathcal{M}}$ while the blue shaded area represents $\mathsf{C}(x,\gamma)$ for an arbitrary value of γ and $x=\bar{x}\in\partial\mathcal{M}$.

Let $\varphi:\ \mathbb{R}^p\to\mathbb{R}_+$ given for any $u\in\mathbb{R}$ by $\varphi(u)=(2\pi)^{-p/2}\exp[-\|u\|^2/2]$[4], i.e. the density of a real Gaussian random variable with zero mean and unit variance. While Lemma 9 appears technical, it will be central to provide quantitative upper bounds on the *rejection* probability, see Lemma 12 for instance.

Lemma 9. *For any $k\in\mathbb{N}$, $\alpha_0>0$, $\beta_0\in(0,1]$ and $\delta>0$ we have*

$$\psi(\delta)=\sup\{\textstyle\int_0^{+\infty}\Psi(\alpha_0 t/\delta,k)^{\beta_0}\varphi(t-t_0/\delta)\mathrm{d}t\ :\ t_0\geq 0\}\leq C_0\delta, \tag{26}$$

with $C_0=5(2\pi)^{-1/2}(k+1)/(\alpha_0\beta_0)$.

Proof. Let $k\in\mathbb{N}$, $\alpha_0>0$, $\beta_0\in(0,1]$ and $\delta>0$. Let $t_\delta=5k\delta/\alpha_0$. Note that if $t\geq t_\delta$ then, $\alpha_0 t/\delta\geq 5k$. In addition, we have

$$\textstyle\int_0^{+\infty}\Psi(\alpha_0 t/\delta,k)^{\beta_0}\varphi(t-t_0/\delta)\mathrm{d}t\leq(2\pi)^{-1/2}\int_0^{+\infty}\Psi(\alpha_0 t/\delta,k)^{\beta_0}\mathrm{d}t \tag{27}$$

$$\leq(2\pi)^{-1/2}\textstyle\int_0^{t_\delta}\Psi(\alpha_0 t/\delta,k)^{\beta_0}\mathrm{d}t+(2\pi)^{-1/2}\int_{t_\delta}^{+\infty}\Psi(\alpha_0 t/\delta,k)^{\beta_0}\mathrm{d}t. \tag{28}$$

Using that for any $w>0$, $\int_0^{+\infty}\exp[-wt]\mathrm{d}t\leq(1/w)$, that for any $u\geq 0$, $\Psi(u,k)\leq 1$ and that if $u\geq 5k$, $\Psi(u,k)\leq\exp[-u/5]$, we get for any $t_0\geq 0$

$$\textstyle\int_0^{+\infty}\Psi(\alpha_0 t/\delta,k)\varphi(t-t_0/\delta)\leq(2\pi)^{-1/2}[5k\delta/\alpha_0+5\delta/(\alpha_0\beta_0)]\leq(5(2\pi)^{-1/2}(k+1)/(\alpha_0\beta_0))\delta, \tag{29}$$

which concludes the proof. □

Finally, we have the following lemma, which is similar to Lemma 8 but will be used to control quantities related to the norm.

Lemma 10. *For any $k\in\mathbb{N}$, $\alpha_0>0$, $\beta_0\in(0,1]$ and $\delta>0$ we have*

$$\psi(\delta)=\textstyle\int_0^{+\infty}\Psi(\alpha_0 t/\delta,k)^{\beta_0}t\varphi(t)\mathrm{d}t\leq C_0\delta^2, \tag{30}$$

with $C_0=25(2\pi)^{-1}(k^2+1)/(\alpha_0\beta_0)^2$.

Proof. Let $k\in\mathbb{N}$, $\alpha_0>0$, $\beta_0\in(0,1]$ and $\delta>0$. Let $t_\delta=5k\delta/\alpha_0$. Note that if $t\geq t_\delta$ then, $\alpha_0 t/\delta\geq 5k$. In addition, we have

$$\textstyle\int_0^{+\infty}\Psi(\alpha_0 t/\delta,k)^{\beta_0}t\varphi(t)\mathrm{d}t\leq(2\pi)^{-1}\int_0^{t_\delta}\Psi(\alpha_0 t/\delta,k)^{\beta_0}t\mathrm{d}t+(2\pi)^{-1}\int_{t_\delta}^{+\infty}\Psi(\alpha_0 t/\delta,k)^{\beta_0}t\mathrm{d}t. \tag{31}$$

[4]In the rest of the supplementary, we never precise the dimension $p\in\mathbb{N}$ which can be deduced from the variable.

In addition, using that if $u \geq 5k$ then $\Psi(u,k) \leq \exp[-u/5]$, we get

$$(2\pi)^{-1} \int_{t_\delta}^{+\infty} \Psi(\alpha_0 t/\delta, k)^{\beta_0} t \mathrm{d}t \leq (2\pi)^{-1} \int_0^{+\infty} \exp[-\alpha_0\beta_0 t/(5\delta)] t \mathrm{d}t \leq (2\pi)^{-1} 25\delta^2/(\alpha_0\beta_0)^2. \tag{32}$$

Finally, using that for any $u \geq 0$, $\Psi(u,k) \leq 1$, we have

$$(2\pi)^{-1} \int_0^{t_\delta} \Psi(\alpha_0 t/\delta, k)^{\beta_0} t \mathrm{d}t \leq (2\pi)^{-1} 25 k^2 \delta^2/\alpha_0^2, \tag{33}$$

which concludes the proof. □

D.3 Lower bound on the inside probability and control of moments of order two and higher

Lower bound on the inside probability. We begin with the following lemma which controls the expectation of $1 + \|Z\|$ *outside* of $\mathsf{C}(x,\gamma)$. We recall that V is defined in Theorem 7-(c).

Lemma 11. *Let $\bar\gamma = 1$. Let $x \in \mathsf{V}$, $Z \in\sim \mathrm{N}(0, \mathrm{Id})$ and $\gamma \in (0, \bar\gamma)$ then we have*

$$\max(\mathbb{E}[\mathbb{1}_{x+\sqrt{\gamma}Z \in \overline{\mathcal{M}} \cap \mathsf{C}(x,\gamma)^c}], \mathbb{E}[\langle Z, \nabla\Phi(\bar x)\rangle \mathbb{1}_{x+\sqrt{\gamma}Z \in \overline{\mathcal{M}} \cap \mathsf{C}(x,\gamma)^c}]) \leq \psi(\gamma), \tag{34}$$

with $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\limsup_{t\to 0} \psi(t)/t^{1/2} < +\infty$.

Proof. Let $\bar r > 0$ given by Theorem 7. First, we have that

$$\int_{\mathbb{R}} \int_{\mathbb{R}^{d-1}} (1 + |\alpha| + \|v\|) \mathbb{1}_{\alpha \geq \bar r/\gamma^{1/2}} \varphi(\alpha)\varphi(v) \mathrm{d}\alpha \mathrm{d}v \tag{35}$$

$$\leq d \int_{\mathbb{R}} (1 + |\alpha|) \mathbb{1}_{\alpha \geq \bar r/\gamma^{1/2}} \varphi(\alpha) \mathrm{d}\alpha \leq d(\Psi(\bar r^2/\gamma, 1) + \exp[-\bar r^2/(2\gamma)]). \tag{36}$$

Second, using Lemma 9, we have that

$$\int_{\mathbb{R}} \int_{\mathbb{R}^{d-1}} \mathbb{1}_{\|v\|^2 \geq (\bar\alpha + \sqrt{\gamma}\alpha)/(C\gamma)} \mathbb{1}_{\bar\alpha + \sqrt{\gamma}\alpha \geq 0} \varphi(\alpha)\varphi(v) \mathrm{d}\alpha \mathrm{d}v \tag{37}$$

$$\leq \int_{\mathbb{R}} \mathbb{1}_{\bar\alpha + \sqrt{\gamma}\alpha \geq 0} \Psi((\bar\alpha + \sqrt{\gamma}\alpha)/(C\gamma), d-1) \varphi(\alpha) \mathrm{d}\alpha \tag{38}$$

$$\leq \int_0^{+\infty} \Psi(\alpha/C\gamma^{1/2}, d-1) \varphi(\alpha - \bar\alpha/\gamma^{1/2}) \mathrm{d}\alpha \leq \Psi_1(\gamma^{1/2}). \tag{39}$$

Second, using Lemma 10, we have that

$$\int_{\mathbb{R}} \int_{\mathbb{R}^{d-1}} \alpha \mathbb{1}_{\|v\|^2 \geq (\bar\alpha + \sqrt{\gamma}\alpha)/(C\gamma)} \mathbb{1}_{\bar\alpha + \sqrt{\gamma}\alpha \geq 0} \varphi(\alpha)\varphi(v) \mathrm{d}\alpha \mathrm{d}v \tag{40}$$

$$= \int_{\mathbb{R}} \alpha \Psi((\bar\alpha + \sqrt{\gamma}\alpha)/(C\gamma), d-1) \mathbb{1}_{\bar\alpha + \sqrt{\gamma}\alpha \geq 0} \varphi(\alpha) \mathrm{d}\alpha \tag{41}$$

$$\leq \int_0^{+\infty} \Psi(\alpha/C\gamma^{1/2}, d-1) \alpha\varphi(\alpha) \mathrm{d}\alpha \leq \Psi_2(\gamma^{1/2}). \tag{42}$$

Note that we have $\limsup_{\gamma\to 0} \Psi_2(\gamma^{1/2}) + \Psi_1(\gamma^{1/2}) < +\infty$. We conclude upon combining (36), (39) and (42) with Theorem 7-(d) and the fact that $\|\Phi(\bar x)\| = 1$. □

The following lemma allow us to give a lower bound to the quantity $\mathbb{E}[\mathbb{1}_{x+\sqrt{\gamma}Z \in \overline{\mathcal{M}}}]$ uniformly w.r.t $x \in \overline{\mathcal{M}}$.

Lemma 12. *There exists $\bar\gamma > 0$ such that for any $\gamma \in (0, \bar\gamma)$ and for any $x \in \overline{\mathcal{M}}$, $\gamma \in (0, \bar\gamma)$ and $Z \sim \mathrm{N}(0, \mathrm{Id})$ we have*

$$\mathbb{E}[\mathbb{1}_{x+\sqrt{\gamma}Z \in \overline{\mathcal{M}}}] \geq 1/4\,. \tag{43}$$

Proof. Let $\gamma \in (0, \bar\gamma)$. If $x \notin \mathsf{V}$ then $\mathrm{B}(x, 2R) \subset \mathcal{M}$ using Theorem 7-(e) and therefore $\mathbb{E}[\mathbb{1}_{x+\sqrt{\gamma}Z \in \overline{\mathcal{M}}}] \geq 1/4$ for $\bar\gamma > 0$ small enough. Now, assume that $x \in \mathsf{V}$. Using Lemma 11, we have that $\mathbb{E}[\mathbb{1}_{x+\sqrt{\gamma}Z \in \overline{\mathcal{M}} \cap \mathsf{C}(x,\gamma)^c}] \leq \psi(\gamma)$. In addition, using Theorem 7-(b), we have that for any $\gamma > 0$

$$\mathbb{E}[\mathbb{1}_{x+\sqrt{\gamma}Z \in \overline{\mathcal{M}}}] \geq \mathbb{E}[\mathbb{1}_{x+\sqrt{\gamma}Z \in \mathsf{C}(x,\gamma)}] \tag{44}$$

$$\geq \int_{-\bar\alpha\gamma^{-1/2}}^{r\gamma^{-1/2}} \int_{\nabla\Phi(\bar x)^\perp} \mathbb{1}_{\|v\|^2 \leq (\bar\alpha + \gamma^{1/2}\alpha)/(C\gamma)} \varphi(\alpha)\varphi(v) \mathrm{d}\alpha \mathrm{d}v \tag{45}$$

$$\geq \int_{-\bar\alpha\gamma^{-1/2}}^{r\gamma^{-1/2}} (1 - \Psi((\bar\alpha + \gamma^{1/2}\alpha)/(C\gamma), d-1)) \varphi(\alpha) \mathrm{d}\alpha \tag{46}$$

$$\geq (1/2) - \Psi(r^2/\gamma, 1) - \int_{-\bar\alpha\gamma^{-1/2}}^{+\infty} \Psi((\bar\alpha + \gamma^{1/2}\alpha)/(C\gamma), d-1) \varphi(\alpha) \mathrm{d}\alpha. \tag{47}$$

Hence, using Lemma 8 and Lemma 9, there exists $\bar\gamma > 0$ such that for any $\gamma \in (0, \bar\gamma)$, $\Psi(r^2/\gamma, 1) + \int_0^{+\infty} \Psi(\alpha/(C\gamma^{1/2}), d)\varphi(\alpha - \gamma^{1/2}\bar\alpha) \mathrm{d}\alpha \leq 1/4$, which concludes the proof. □

Note that the result of Lemma 12 can be improved to $1/2 - \varepsilon$ for any $\varepsilon > 0$. In particular this result tells us that for $\gamma > 0$ small enough, $\overline{\mathcal{M}}$ looks like the *hyperplane* from the point of view of the Gaussian with variance γ centered on $\partial\mathcal{M}$.

Bound on moments of order two and higher. In what follows, we define for any $\gamma > 0$, $\Delta^\gamma : \overline{\mathcal{M}} \to \mathbb{R}_+$ given for any $x \in \overline{\mathcal{M}}$ by

$$\Delta^\gamma(x) = (1/\gamma) \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} \|\sqrt{\gamma}z\|^4 \varphi(z) \mathrm{d}z / \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} \varphi(z) \mathrm{d}z. \tag{48}$$

Proposition 13. *We have* $\lim_{\gamma \to 0} \sup\{\Delta^\gamma(x) \, : \, x \in \overline{\mathcal{M}}\} = 0$.

Proof. Let $\bar{\gamma} > 0$ given by Lemma 12. Let $x \in \overline{\mathcal{M}}$ and $\gamma \in (0, \bar{\gamma})$. We have using Lemma 12

$$\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} \varphi(z) \mathrm{d}z \geq 1/4. \tag{49}$$

We also have that

$$(1/\gamma) \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} \|\sqrt{\gamma}z\|^4 \varphi(z) \mathrm{d}z \leq 3\gamma d^2. \tag{50}$$

Therefore, we get that for any $\gamma \in (0, \bar{\gamma})$, $\Delta^\gamma(x) \leq 12\gamma d^2$, which concludes the proof. □

In what follows, we define for any $\gamma > 0$, $\hat{\Sigma}^\gamma : \overline{\mathcal{M}} \to \mathrm{S}_d^+(\mathbb{R})$ given for any $x \in \overline{\mathcal{M}}$ by

$$\hat{\Sigma}^\gamma(x) = \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} z \otimes z \varphi(z) \mathrm{d}z / \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} \varphi(z) \mathrm{d}z. \tag{51}$$

Proposition 14. *There exists* $\bar{\gamma} > 0$ *such that for any* $x \in \overline{\mathcal{M}}$ *and* $\gamma \in (0, \bar{\gamma})$ *we have*

$$\|\hat{\Sigma}^\gamma(x)\| \leq 4d. \tag{52}$$

Proof. Let $x \in \overline{\mathcal{M}}$ and $\bar{\gamma} > 0$ given by Lemma 12. For any $\gamma \in (0, \bar{\gamma})$, we have using Lemma 12

$$\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} \varphi(z) \mathrm{d}z \geq 1/4. \tag{53}$$

We also have that

$$\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} \|z\|^2 \varphi(z) \mathrm{d}z \leq d, \tag{54}$$

which concludes the proof. □

D.4 Properties of large drift terms

Finally, we define for any $\gamma > 0$, $\hat{b}^\gamma : \overline{\mathcal{M}} \to \mathbb{R}^d$ given for any $x \in \overline{\mathcal{M}}$ by

$$\hat{b}^\gamma(x) = \gamma^{-1/2} \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} z \varphi(z) \mathrm{d}z / \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} \varphi(z) \mathrm{d}z. \tag{55}$$

First, we show away from the boundary the drift $\hat{b}^\gamma$ converges to zero.

Proposition 15. *There exists* $\bar{\gamma} > 0$ *such that for any* $\gamma \in (0, \bar{\gamma})$, $r > 0$ *and* $x \in \overline{\mathcal{M}}$ *such that* $d(x, \partial\mathcal{M}) \geq r$ *we have* $\|\hat{b}^\gamma(x)\| \leq 2d\Psi(r/\gamma, d)^{1/2}/\gamma^{1/2}$.

Proof. Let $x \in \overline{\mathcal{M}}$ and $\bar{\gamma} > 0$ given by Lemma 12. For any $\gamma \in (0, \bar{\gamma})$ we have using Lemma 12

$$\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} \varphi(z) \mathrm{d}z \geq 1/4. \tag{56}$$

We also have that

$$\| \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} z \varphi(z) \mathrm{d}z \| \leq \| \int_{\mathbb{R}^d} \mathbb{1}_{\|z\| \leq r/\gamma^{1/2}} z \varphi(z) \mathrm{d}z \| + \int_{\mathbb{R}^d} \mathbb{1}_{\|z\| \geq r/\gamma^{1/2}} \|z\| \varphi(z) \mathrm{d}z \tag{57}$$

$$\leq 2 \int_{\mathbb{R}^d} \mathbb{1}_{\|z\| \geq r/\gamma^{1/2}} \|z\| \varphi(z) \mathrm{d}z \leq 2d\Psi(r/\gamma, d)^{1/2}/\gamma^{1/2}, \tag{58}$$

which concludes the proof. □

We have the following corollary.

Corollary 16. *There exists* $\bar{\gamma} > 0$ *such that for any* $\delta > 0$ *there exists* $M_\delta > 0$ *such that for any* $\gamma \in (0, \bar{\gamma})$ *and* $x \in \overline{\mathcal{M}}$, $\|\hat{b}^\gamma(x)\| \geq M_\delta$, *then* $\Phi(x) \leq \delta$.

Proof. Let $\bar{\gamma} > 0$ given by Lemma 12. Let $f : \mathbb{R}_+ \to \mathbb{R}_+$ given for any $r > 0$ by $f(r) = \sup\{\gamma > 0 : \Psi(r/\gamma, 1)^{1/2}/\gamma^{1/2}\}$. We have that f is non-increasing and $\lim_{r\to 0} f(r) = +\infty$. Let $\delta > 0$ and $M_\delta = 2df(\delta/C)$ with $C = \sup\{\|\nabla\Phi(x)\| : x \in \overline{\mathcal{M}}\}$. Let $\gamma \in (0, \bar{\gamma})$ and $x \in \overline{\mathcal{M}}$ such that $\|\hat{b}^\gamma(x)\| \geq M_\delta$ then using Proposition 15 we have that $d(x, \partial\mathcal{M}) \leq \delta/C$. Let $\bar{x} \in \partial\mathcal{M}$ such that $\|x - \bar{x}\| = d(x, \partial\mathcal{M})$. We have

$$\Phi(x) \leq \Phi(\bar{x}) + \int_0^1 \langle \nabla\Phi(\bar{x} + t(x - \bar{x})), x - \bar{x}\rangle \mathrm{d}t \leq \delta, \tag{59}$$

which concludes the proof. □

For ease of notation, for any $\gamma > 0$, we define $\bar{b}^\gamma = \gamma^{1/2}\hat{b}^\gamma$, the *renormalized* version of the drift. First, we have the following result which will ensure that the drift projected on the normal component does not vanish.

Lemma 17. *There exists $\bar{\gamma} > 0$ such that for any $\gamma \in (0, \bar{\gamma})$ and $x \in \mathsf{V}$ we have*

$$\langle \bar{b}^\gamma(x), \nabla\Phi(\bar{x})\rangle \geq \|\bar{b}^\gamma(x)\| - \psi(\gamma), \tag{60}$$

with $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\limsup_{\gamma\to 0} \psi(\gamma)/\sqrt{\gamma} < +\infty$.

Proof. Let $x \in \overline{\mathcal{M}}$ and $\bar{\gamma} > 0$ given by Lemma 12. For any $\gamma \in (0, \bar{\gamma})$ we have using Lemma 12

$$\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathcal{M}}\varphi(z)\mathrm{d}z \geq 1/4. \tag{61}$$

In addition, we have

$$\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathcal{M}}\langle z, \nabla\Phi(\bar{x})\rangle\varphi(z)\mathrm{d}z \geq \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathsf{C}(x,\gamma)}\langle z, \nabla\Phi(\bar{x})\rangle\varphi(z)\mathrm{d}z \tag{62}$$

$$- \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathcal{M}\cap\mathsf{C}(x,\gamma)^\mathrm{c}}\langle z, \nabla\Phi(\bar{x})\rangle\varphi(z). \tag{63}$$

Using Lemma 11, we get that

$$\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathcal{M}}\langle z, \nabla\Phi(\bar{x})\rangle\varphi(z)\mathrm{d}z \geq \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathsf{C}(x,\gamma)}\langle z, \nabla\Phi(\bar{x})\rangle\varphi(z)\mathrm{d}z - \psi(\gamma). \tag{64}$$

Let $\{e_i\}_{i=1}^{d-1}$ a basis of $\nabla\Phi(\bar{x})^\perp$. Using Theorem 7-(b), we have that for any $i \in \{1, \ldots, d-1\}$

$$\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathsf{C}(x,\gamma)}\langle z, e_i\rangle\varphi(z)\mathrm{d}z = \int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \int_{\nabla\Phi(\bar{x})^\perp} \mathbb{1}_{\|v\|^2\leq(\gamma^{1/2}\alpha+\bar{\alpha})/\gamma}\langle v, e_i\rangle\varphi(v)\varphi(\alpha)\mathrm{d}v\mathrm{d}\alpha. \tag{65}$$

Hence, combining this result and the Cauchy-Schwarz inequality we have for any $i \in \{1, \ldots, d-1\}$

$$\left(\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathsf{C}(x,\gamma)}\langle z, e_i\rangle\varphi(z)\mathrm{d}z\right)^2 = \left(\int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \int_{\nabla\Phi(\bar{x})^\perp} \mathbb{1}_{\|v\|^2\geq(\gamma^{1/2}\alpha+\bar{\alpha})/\gamma}\langle v, e_i\rangle\varphi(v)\varphi(\alpha)\mathrm{d}v\mathrm{d}\alpha\right)^2 \tag{66}$$

$$\leq \int_{\nabla\Phi(\bar{x})^\perp}\langle v, e_i\rangle^2\varphi(v)\mathrm{d}v\left(\int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \Psi((\bar{\alpha} + \alpha\gamma^{1/2})/\gamma, d-1)^{1/2}\varphi(\alpha)\mathrm{d}\alpha\right)^2 \tag{67}$$

$$\leq \left(\int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \Psi((\bar{\alpha} + \alpha\gamma^{1/2})/\gamma, d-1)^{1/2}\varphi(\alpha)\mathrm{d}\alpha\right)^2. \tag{68}$$

Hence, using Lemma 9, we get that

$$\sum_{i=1}^{d-1}\left(\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathsf{C}(x,\gamma)}\langle z, e_i\rangle\varphi(z)\mathrm{d}z\right)^2 \leq (d-1)\psi^2(\gamma), \tag{69}$$

with ψ given by Lemma 9 with $\beta_0 = 1/2$. Therefore, we get that

$$\left(\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathsf{C}(x,\gamma)}\langle z, \nabla\Phi(\bar{z})\rangle\varphi(z)\mathrm{d}z\right)^2 \tag{70}$$

$$= \left(\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathcal{M}}\varphi(z)\mathrm{d}z\right)^2\|\bar{b}^\gamma(x)\|^2 - \sum_{i=1}^{d-1}\left(\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathsf{C}(x,\gamma)}\langle z, e_i\rangle\varphi(z)\mathrm{d}z\right)^2 \tag{71}$$

$$\geq \left(\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathcal{M}}\varphi(z)\mathrm{d}z\right)^2\|\bar{b}^\gamma(x)\|^2 - \psi(\gamma)^2. \tag{72}$$

We conclude the proof upon using that for any $a, b \geq 0$, $(a+b)^{1/2} \leq a^{1/2} + b^{1/2}$ and (61). □

We are now ready to state the following lower bound on the drift.

Proposition 18. *There exist $\bar\gamma > 0$, $M \geq 0$ and $c > 0$ such that for any $x \in \overline{\mathcal{M}}$ and $\gamma \in (0, \bar\gamma)$ if $\|\hat b^\gamma(x)\| \geq M$ then $x \in \mathsf{V}$ and*

$$\min(\langle \hat b^\gamma(x), \nabla\Phi(x)\rangle, \langle \hat b^\gamma(x), \nabla\Phi(\bar x)\rangle) \geq c\|\hat b^\gamma(x)\|. \tag{73}$$

Proof. Let $\bar\gamma > 0$ given by Lemma 12 and $M_0 = 4\sup\{\psi(\gamma)/\gamma^{1/2} \,:\, \gamma \in (0, \bar\gamma]\}$. In addition, let $c = 1/4$. Using Proposition 15 and Theorem 7-(e), there exists $M_1 \geq 0$ such that for any any $x \in \overline{\mathcal{M}}$, if $\|\hat b^\gamma(x)\| \geq M_1$ then $x \in \mathsf{V}$ and $x = \bar x + \alpha\nabla\Phi(\bar x)$ with $\alpha \leq 1/(4C)$ and $C = \sup\{\|\nabla^2\Phi(x)\| \,:\, x \in \overline{\mathcal{M}}\}$. We denote $M = \max(M_0, M_1)$. Let $\gamma \in (0, \bar\gamma)$ and $x \in \overline{\mathcal{M}}$ such that $\|\hat b^\gamma(x)\| \geq M$. Using Lemma 17, we have that

$$\langle \hat b^\gamma(x), \nabla\Phi(\bar x)\rangle \geq \|\hat b^\gamma(x)\| - \psi(\gamma)/\gamma^{1/2}. \tag{74}$$

Using that $\psi(\gamma)/\gamma^{1/2} \leq M/2 \leq \|\hat b^\gamma(x)\|/2$, we have

$$\langle \hat b^\gamma(x), \nabla\Phi(\bar x)\rangle \geq (1/2)\|\hat b^\gamma(x)\|. \tag{75}$$

Since $\|x - \bar x\| \leq \alpha \leq 1/(4C)$ we have $\langle \hat b^\gamma(x), \nabla\Phi(x)\rangle \geq (1/2 - C\alpha)\|\hat b^\gamma(x)\| \geq \|\hat b^\gamma(x)\|/4$, which concludes the proof. □

D.5 Convergence on compact sets

In this section, we show the convergence of the drift and diffusion matrix on compact sets. We recall that $\mathcal{M}$ does *not* include its boundary $\partial\mathcal{M}$.

Proposition 19. *For any compact set $\mathsf{K} \subset \mathcal{M}$ and $\varepsilon > 0$, there exists $\bar\gamma > 0$ such that for any $\gamma \in (0, \bar\gamma)$ we have for any $x \in \mathsf{K}$*

$$\|\hat b^\gamma(x)\| \leq \varepsilon, \qquad \|\hat\Sigma^\gamma(x) - \mathrm{Id}\,\| \leq \varepsilon. \tag{76}$$

Proof. Let $\mathsf{K} \subset \mathcal{M}$ be a compact set and $\gamma > 0$. Since $\mathsf{K} \cap \partial\mathcal{M} = \varnothing$, there exists $r > 0$ such that for any $x \in \mathsf{K}$, $d(x, \partial\mathcal{M}) > r$. Therefore, we have that for any $x \in \mathsf{K}$

$$\|\hat b^\gamma(x)\| = \gamma^{-1/2}\|\textstyle\int_{x+\sqrt\gamma z\in\mathcal{M}} z\varphi(z)\mathrm{d}z\|/\int_{x+\sqrt\gamma z\in\mathcal{M}}\varphi(z)\mathrm{d}z. \tag{77}$$

In addition, using the Cauchy-Schwarz inequality we have

$$\begin{aligned}\|\textstyle\int_{x+\sqrt\gamma z\in\mathcal{M}} z\varphi(z)\mathrm{d}z\| &\leq \|\textstyle\int_{\mathbb{R}^d} z\varphi(z)\mathrm{d}z\| + \int_{\mathcal{M}^c}\|z\|\varphi(z)\mathrm{d}z &(78)\\ &\leq \textstyle\int_{\mathbb{R}^d}\mathbb{1}_{\|z\|\geq r/\gamma^{1/2}}\|z\|\varphi(z)\mathrm{d}z \leq \sqrt{d}\Psi(r^2/\gamma, d)^{1/2}. &(79)\end{aligned}$$

Using Lemma 8 and Lemma 12, there exists $\bar\gamma_0 > 0$ such that for any $\gamma \in (0, \bar\gamma_0)$ we have that for any $x \in \mathsf{K}$

$$\|\hat b^\gamma(x)\| \leq 4d\Psi(r^2/\gamma, 1)^{1/2}/\gamma^{1/2} \leq \varepsilon, \tag{80}$$

which concludes the first part of the proof. Similarly, we have that for any $x \in \mathsf{K}$

$$\begin{aligned}\|\textstyle\int_{x+\sqrt\gamma z\in\mathcal{M}}(z\otimes z - \mathrm{Id})\varphi(z)\mathrm{d}z\| &\leq \|\textstyle\int_{\mathbb{R}^d}(z\otimes z - \mathrm{Id})\varphi(z)\mathrm{d}z\| + \int_{\mathcal{M}^c}\|z\|\varphi(z)\mathrm{d}z &(81)\\ &\leq \textstyle\int_{\mathbb{R}^d}\mathbb{1}_{\|z\|\geq r/\gamma^{1/2}}\|z\otimes z - \mathrm{Id}\,\|\varphi(z)\mathrm{d}z &(82)\\ &\leq \sqrt{2}(1+3d^2)^{1/2}\Psi(r^2/\gamma, d)^{1/2}. &(83)\end{aligned}$$

Using Lemma 8 and Lemma 12, there exists $\bar\gamma_1 > 0$ such that for any $\gamma \in (0, \bar\gamma_1)$, we have that for any $x \in \mathsf{K}$

$$\|\hat\Sigma^\gamma(x) - \mathrm{Id}\,\| \leq 4\sqrt{2}(1+3d^2)^{1/2}\Psi(r^2/\gamma, 1)^{1/2} \leq \varepsilon, \tag{84}$$

which concludes the proof upon letting $\bar\gamma = \min(\bar\gamma_0, \bar\gamma_1)$. □

D.6 Convergence on the boundary

Finally, we investigate the behavior at the boundary of the diffusion matrix and the drift. First, we show that there is a lower bound to the diffusion matrix near the boundary. Second, we show that the renormalized drift converges to the outward normal.

Proposition 20. *There exist $c > 0$ and $\bar{\gamma} > 0$ such that for any $\gamma \in (0, \bar{\gamma})$, $u \in \mathbb{R}^d$ and $x \in \mathsf{V}$ we have*

$$\langle u, \hat{\Sigma}^\gamma(x) u \rangle \geq c \|u\|^2. \tag{85}$$

In particular, there exist $r, \varepsilon > 0$ such that for any $\gamma \in (0, \bar{\gamma})$ and $x \in \overline{\mathcal{M}}$ with $d(x, \partial\mathcal{M}) \leq r$

$$\langle \nabla\Phi(x), \hat{\Sigma}^\gamma(x) \nabla\Phi(x) \rangle \geq \varepsilon. \tag{86}$$

Proof. First, we show (85). Let $x \in \mathsf{V}$. We have for any $u \in \mathbb{R}^d$

$$\langle u, \hat{\Sigma}^\gamma(x) u \rangle = \textstyle\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} \langle z, u \rangle^2 \varphi(z) \mathrm{d}z / \int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathcal{M}} \mathrm{d}z \tag{87}$$

$$\geq \textstyle\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathsf{C}(x,\gamma)} \langle z, u \rangle^2 \varphi(z) \mathrm{d}z. \tag{88}$$

For any $u \in \mathbb{R}^d$, let $\alpha_u = \langle u, \nabla\Phi(\bar{x}) \rangle$. Using Theorem 7-(b) we have for any $u \in \mathbb{R}^d$

$$\textstyle\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathsf{C}(x,\gamma)} \langle z, u \rangle^2 \varphi(z) \mathrm{d}z \tag{89}$$

$$= \textstyle\int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \int_{\nabla\Phi(\bar{x})^\perp} (\langle u, v \rangle + \alpha\alpha_u)^2 \mathbb{1}_{\|v\|^2 \leq (\alpha\gamma^{1/2}+\bar{\alpha})/\gamma} \varphi(v)\varphi(\alpha) \mathrm{d}v \mathrm{d}\alpha \tag{90}$$

$$\geq \textstyle\int_0^{r/\gamma^{1/2}} \int_{\nabla\Phi(\bar{x})^\perp} (\langle u, v \rangle^2 + \alpha^2\alpha_u^2) \mathbb{1}_{\|v\|^2 \leq (\alpha\gamma^{1/2}+\bar{\alpha})/\gamma} \varphi(v)\varphi(\alpha) \mathrm{d}v \mathrm{d}\alpha \tag{91}$$

$$\geq \alpha_u^2 \textstyle\int_0^{r/\gamma^{1/2}} \alpha^2 \varphi(\alpha) \mathrm{d}\alpha + \int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \int_{\nabla\Phi(\bar{x})^\perp} \langle u, v \rangle^2 \mathbb{1}_{\|v\|^2 \leq (\alpha\gamma^{1/2}+\bar{\alpha})/\gamma} \varphi(v)\varphi(\alpha) \mathrm{d}v \mathrm{d}\alpha. \tag{92}$$

Using Cauchy-Schwarz inequality, we have

$$\textstyle\int_0^{r/\gamma^{1/2}} \alpha^2 \varphi(\alpha) \mathrm{d}\alpha = (1/2) - \int_{r/\gamma^{1/2}}^{+\infty} \alpha^2 \varphi(\alpha) \mathrm{d}\alpha \geq (1/2) - 3\Phi(r^2/\gamma, 1)^{1/2}. \tag{93}$$

In addition, using the Cauchy-Schwarz inequality, we have that

$$\textstyle\int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \int_{\nabla\Phi(\bar{x})^\perp} \langle u, v \rangle^2 \mathbb{1}_{\|v\|^2 \leq (\alpha\gamma^{1/2}+\bar{\alpha})/\gamma} \varphi(v)\varphi(\alpha) \mathrm{d}v \mathrm{d}\alpha \tag{94}$$

$$= \textstyle\int_{\nabla\Phi(\bar{x})^\perp} \langle u, v \rangle^2 \varphi(v) \mathrm{d}v \int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \varphi(\alpha) \mathrm{d}\alpha \tag{95}$$

$$- \textstyle\int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \int_{\nabla\Phi(\bar{x})^\perp} \langle u, v \rangle^2 \mathbb{1}_{\|v\|^2 \geq (\alpha\gamma^{1/2}+\bar{\alpha})/\gamma} \varphi(v)\varphi(\alpha) \mathrm{d}v \mathrm{d}\alpha \tag{96}$$

$$\geq (\|u\|^2 - \alpha_u^2)((1/2) - \Phi(r^2/\gamma, 1)) \tag{97}$$

$$- \sqrt{3}(d-1)\|u\|^2 \textstyle\int_0^{+\infty} \Phi(\alpha/\gamma^{1/2}, d-1)^{1/2} \varphi(\alpha - \bar{\alpha}/\gamma^{1/2}) \mathrm{d}\alpha. \tag{98}$$

Combining this result, (93), (92) and Lemma 9 there exists $\bar{\gamma} > 0$ such that for any $\gamma \in (0, \bar{\gamma}]$ and $u \in \mathbb{R}^d$

$$\textstyle\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z \in \mathsf{C}(x,\gamma)} \langle z, u \rangle^2 \varphi(z) \mathrm{d}z \geq (1/4)\|u\|^2, \tag{99}$$

which concludes the proof of (85). Finally, using Theorem 7-(e), we have that for any $x \in \overline{\mathcal{M}}$ if $d(x, \partial\mathcal{M}) \leq R$ then $x \in \mathsf{V}$. Let $r = \min(R, 1/(2C))$ with $C = \sup\{\|\nabla^2\Phi(x)\| \ : \ x \in \overline{\mathcal{M}}\}$. We have that for any $x \in \overline{\mathcal{M}}$ such that $d(x, \partial\mathcal{M}) \leq r$

$$\|\nabla\Phi(x)\| \geq \|\nabla\Phi(\bar{x}_0)\| - Cr \geq 1/2, \tag{100}$$

where $\bar{x}_0$ is such that $\|x - \bar{x}_0\| \leq r$ and $\bar{x}_0 \in \partial\mathcal{M}$. Combining this result and (99) concludes the proof upon letting $\varepsilon = 1/16$. □

Finally, we investigate the behavior of the normalized drift near the boundary.

Proposition 21. *For any $\bar{x}_0 \in \partial\mathcal{M}$ and $\varepsilon > 0$, there exist $\bar{\gamma}, r, M > 0$ such that for any $x \in \overline{\mathcal{M}}$ and $\gamma \in (0, \bar{\gamma})$ with $\|x - \bar{x}_0\| \leq r$ and $\|\hat{b}^\gamma(x)\| \geq M$*

$$\|\hat{b}^\gamma(x)/\langle \hat{b}^\gamma(x), \nabla\Phi(x) \rangle - \nabla\Phi(\bar{x}_0)\| \leq \varepsilon. \tag{101}$$

Proof. Let $\bar{\gamma}$ be given by Proposition 18. Let ψ given by Lemma 9 and $M_0 = \sup\{\psi(\gamma)/\gamma^{1/2} : \gamma \in (0, \bar{\gamma})\} < +\infty$. Let $M = 16M_0/(c\varepsilon^{1/2})$ with c given in Proposition 18. Let $R > 0$ given by Theorem 7-(e) such that for any $x \in \overline{\mathcal{M}}$ with $d(x, \partial\mathcal{M})$ there exist $\bar{x} \in \partial\mathcal{M}$ and $\alpha \in [0, c\varepsilon/(4C)]$ such that $x = \bar{x} + \alpha\nabla\Phi(\bar{x})$ with $C = \sup\{\|\nabla^2\Phi(x)\| : x \in \overline{\mathcal{M}}\}$ and c given in Proposition 18. Let $r = \min(\bar{r}, c\varepsilon/4, R)$ and $x \in \overline{M}$ with $\|x - \bar{x}_0\| \leq r$. First, since $d(x, \partial\mathcal{M}) \leq R$, there exist $\bar{x} \in \partial\mathcal{M}$ and $\alpha \in [0, \varepsilon/(4C)]$ such that $x = \bar{x} + \alpha\nabla\Phi(\bar{x})$. Therefore, we get that $\|\bar{x} - \bar{x}_0\| \leq \varepsilon/(2C)$ and therefore $\|\nabla\Phi(\bar{x}_0) - \nabla\Phi(\bar{x})\| \leq \varepsilon/2$. In addition, we have that

$$\|\hat{b}^\gamma(x)/\langle\hat{b}^\gamma(x), \nabla\Phi(x)\rangle - \hat{b}^\gamma(x)/\langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x})\rangle\| \tag{102}$$

$$\leq \|\hat{b}^\gamma(x)\|^2\|\nabla\Phi(x) - \nabla\Phi(\bar{x})\|/(\langle\hat{b}^\gamma(x), \nabla\Phi(x)\rangle\langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x})\rangle). \tag{103}$$

Using Proposition 18, we get that

$$\|\hat{b}^\gamma(x)/\langle\hat{b}^\gamma(x), \nabla\Phi(x)\rangle - \hat{b}^\gamma(x)/\langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x})\rangle\| \leq \varepsilon/4. \tag{104}$$

In what follows, we show that

$$\|\hat{b}^\gamma(x)/\langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x})\rangle - \nabla\Phi(\bar{x})\|^2 \leq \varepsilon/2. \tag{105}$$

In particular, we show that for any $u \in \nabla\Phi(\bar{x})^\perp$ with $\|u\| = 1$,

$$\langle\hat{b}^\gamma(x), u\rangle^2 \leq (\varepsilon/16)\langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x})\rangle^2. \tag{106}$$

Assuming (106), letting $u = (\hat{b}^\gamma(x) - \langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x}\rangle))/(\|\hat{b}^\gamma(x)\|^2 - \langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x}\rangle^2)^{1/2}$ and using that $\hat{b}^\gamma(x) = \langle\hat{b}^\gamma(x), u\rangle u + \langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x})\rangle\nabla\Phi(\bar{x})$ we have

$$\|\hat{b}^\gamma(x)/\langle\hat{b}^\gamma(x), \nabla\Phi(x)\rangle - \nabla\Phi(\bar{x})\| \leq \|\hat{b}^\gamma(x)/\langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x})\rangle - \nabla\Phi(\bar{x})\| \tag{107}$$

$$+ \|\hat{b}^\gamma(x)/\langle\hat{b}^\gamma(x), \nabla\Phi(x)\rangle - \hat{b}^\gamma(x)/\langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x})\rangle\| \tag{108}$$

$$\leq |\langle\hat{b}^\gamma(x), u\rangle/\langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x})\rangle| + \varepsilon/4 \leq \varepsilon/2, \tag{109}$$

which concludes the proof. Let $u \in \nabla\Phi(\bar{x})^\perp$ with $\|u\| = 1$ and $\{e_i\}_{i=1}^{d-1}$ an orthonormal basis of $\nabla\Phi(\bar{x})^\perp$. There exist $\{a_i\}_{i=1}^{d-1}$ such that $\sum_{i=1}^{d-1} a_i^2 = 1$ and $u = \sum_{i=1}^{d-1} a_i e_i$. Using Theorem 7-(b), we have that for any $i \in \{1, \ldots, d-1\}$

$$\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathsf{C}(x,\gamma)}\langle z, e_i\rangle\varphi(z)\mathrm{d}z = \int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \int_{\nabla\Phi(\bar{x})^\perp} \mathbb{1}_{\|v\|^2\leq(\gamma^{1/2}\alpha+\bar{\alpha})/\gamma}\langle v, e_i\rangle\varphi(v)\varphi(\alpha)\mathrm{d}v\mathrm{d}\alpha \tag{110}$$

$$= \int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \int_{\nabla\Phi(\bar{x})^\perp} \mathbb{1}_{\|v\|^2\geq(\gamma^{1/2}\alpha+\bar{\alpha})/\gamma}\langle v, e_i\rangle\varphi(v)\varphi(\alpha)\mathrm{d}v\mathrm{d}\alpha \tag{111}$$

Hence, combining this result and the Cauchy-Schwarz inequality we have for any $i \in \{1, \ldots, d-1\}$

$$\left(\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathsf{C}(x,\gamma)}\langle z, e_i\rangle\varphi(z)\mathrm{d}z\right)^2 = \left(\int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \int_{\nabla\Phi(\bar{x})^\perp} \mathbb{1}_{\|v\|^2\geq(\gamma^{1/2}\alpha+\bar{\alpha})/\gamma}\langle v, e_i\rangle\varphi(v)\varphi(\alpha)\mathrm{d}v\mathrm{d}\alpha\right)^2 \tag{112}$$

$$\leq \int_{\nabla\Phi(\bar{x})^\perp}\langle v, e_i\rangle^2\varphi(v)\mathrm{d}v\left(\int_{-\bar{\alpha}/\gamma^{1/2}}^{r/\gamma^{1/2}} \Psi((\bar{\alpha} + \alpha\gamma^{1/2})/\gamma, d-1)^{1/2}\varphi(\alpha)\mathrm{d}\alpha\right)^2. \tag{113}$$

Hence, we get that

$$\sum_{i=1}^{d-1} a_i^2\left(\int_{\mathbb{R}^d} \mathbb{1}_{x+\sqrt{\gamma}z\in\mathsf{C}(x,\gamma)}\langle z, e_i\rangle\varphi(z)\mathrm{d}z\right)^2 \leq \|u\|^2\psi^2(\gamma), \tag{114}$$

with ψ given by Lemma 9. Recalling that $\|\hat{b}^\gamma(x)\| \geq M$ we have

$$\langle\hat{b}^\gamma(x), u\rangle^2 \leq 16\psi(\gamma)^2/\gamma \leq c^2(\varepsilon/16)M^2 \leq (\varepsilon/16)\langle\hat{b}^\gamma(x), \nabla\Phi(\bar{x})\rangle^2, \tag{115}$$

which concludes the proof. □

D.7 Submartingale problem and weak solution

We are now ready to conclude the proof.

Theorem 22. *There exists $\mathbb{P}^\star$ a distribution on $\mathrm{D}([0,T],\overline{\mathcal{M}})$ such that $\lim_{\gamma\to 0}\hat{\mathbb{P}}^\gamma = \mathbb{P}^\star$. In addition, for any $f \in \mathrm{C}^{1,2}([0,T]\times\overline{\mathcal{M}},\mathbb{R})$ with $\langle\nabla\Phi(\bar{x}),\nabla f(x)\rangle \geq 0$ for any $t\in[0,T]$ and $x\in\partial\mathcal{M}$, we have that the process $(f(t,\omega(t)))_{t\in[0,T]}$ given for any $t\in[0,T]$*

$$f(t,\omega(t)) - \textstyle\int_0^t (\partial_s f(s,\omega(s)) + \frac{1}{2}\Delta f(s,\omega(s)))\mathbb{1}_{\mathcal{M}}(\omega(s))\mathrm{d}s, \tag{116}$$

is a $\mathbb{P}$ submartingale.

Proof. Condition (A) (Stroock et al., 1971, p.197) is a consequence of Proposition 13. Condition (B) (Stroock et al., 1971, p.197) is a consequence of Proposition 18. Condition (C) (Stroock et al., 1971, p.198) is a consequence of Corollary 16. Condition (D) (Stroock et al., 1971, p.198) is a consequence of Proposition 14. We fix $\rho = 0$ and condition (1) (Stroock et al., 1971, p.203) is a consequence of Proposition 19. Condition (2)-(iii) (Stroock et al., 1971, p.203) is a consequence of Proposition 20. Condition (2)-(iv) (Stroock et al., 1971, p.203) is a consequence of Proposition 21. We conclude upon using (Stroock et al., 1971, Theorem 6.3) and (Stroock et al., 1971, Theorem 5.8). □

We finally conclude the proof of Theorem 6 upon using the results of (Kang et al., 2017) which establish the link between a weak solution to the reflected SDE and the solution to a submartingale problem.

Theorem 23. *For any $T \geq 0$, $(\hat{\mathbf{X}}_t^\gamma)_{t\in[0,T]}$ weakly converges to $(\mathbf{X}_t)_{t\in[0,T]}$ such that for any $t\in[0,T]$*

$$\mathbf{X}_t = x + \mathbf{B}_t - \mathbf{k}_t, \qquad |\mathbf{k}|_t = \textstyle\int_0^t \mathbb{1}_{\mathbf{X}_s\in\partial\mathcal{M}}\mathrm{d}|\mathbf{k}|_s, \qquad \mathbf{k}_t = \int_0^t \mathbf{n}(\mathbf{X}_s)\mathrm{d}|\mathbf{k}|_s. \tag{117}$$

Proof. Using Theorem 22 and (Kang et al., 2017, Theorem 1, Proposition 2.12), we have that $\mathbb{P}$ in Theorem 22is associated with a solution to the extended Skorokhod problem. We conclude that a solution to the extended Skorokhod problem is a solution to the Skorokhod problem using (Ramanan, 2006, Corollary 2.10). □

D.8 Extension to the Metropolis process

We recall that the Metropolis process is defined as follows. Let $(X_k^\gamma)_{k\in\mathbb{N}}$ given for any $\gamma>0$ and $k\in\mathbb{N}$ by $X_0^\gamma = x\in\overline{\mathcal{M}}$ and for $X_{k+1}^\gamma = X_k^\gamma + \sqrt{\gamma}Z_k$ if $X_k^\gamma + \sqrt{\gamma}Z_k^\gamma \in \overline{\mathcal{M}}$ and X_k^γ otherwise, $Z_k\sim\mathrm{N}(0,\mathrm{Id})$. We recall that $\hat{b}^\gamma$, $\hat{\Sigma}^\gamma$ and $\hat{\Delta}^\gamma$ are given by (48), (51) and (55). In particular, denoting $\hat{\mathrm{K}}^\gamma$ the Markov kernel associated with $(\hat{X}_k^\gamma)_{k\in\mathbb{N}}$, i.e. $\hat{\mathrm{K}}^\gamma : \mathcal{M}\times\mathcal{B}(\mathcal{M})\to[0,1]$ such that for any $x\in\mathcal{M}$, $\hat{\mathrm{K}}^\gamma(x,\cdot)$ is a probability measure, for any $\mathsf{A}\in\mathcal{B}(\mathcal{M})$, $\hat{\mathrm{K}}^\gamma(\cdot,\mathsf{A})$ is a measurable function and $\mathbb{E}[\mathbb{1}_{\mathsf{A}}(\hat{X}_1^\gamma)\mid\hat{X}_0^\gamma = x] = \hat{\mathrm{K}}^\gamma(x,\mathsf{A})$. We have that for any $\gamma>0$ and $x\in\mathcal{M}$

$$\hat{b}^\gamma(x) = (1/\gamma)\textstyle\int_{\mathcal{M}}(y-x)\hat{\mathrm{K}}^\gamma(x,\mathrm{d}y), \tag{118}$$

$$\hat{\Sigma}^\gamma(x) = (1/\gamma)\textstyle\int_{\mathcal{M}}(y-x)^{\otimes 2}\hat{\mathrm{K}}^\gamma(x,\mathrm{d}y), \tag{119}$$

$$\hat{\Delta}^\gamma(x) = (1/\gamma)\textstyle\int_{\mathcal{M}}\|y-x\|^4\hat{\mathrm{K}}^\gamma(x,\mathrm{d}y). \tag{120}$$

In what follows, we denote $a^\gamma(x) = \mathbb{E}[\mathbb{1}_{x+\sqrt{\gamma}Z_0\in\mathcal{M}}]$. Denote K^γ the kernel associated with $(X_k^\gamma)_{k\in\mathbb{N}}$. We have that for any $\mathsf{A}\in\mathcal{B}(\mathcal{M})$, $\gamma>0$ and $x\in\mathcal{M}$

$$\mathrm{K}^\gamma(x,\mathsf{A}) = \mathbb{E}[\mathbb{1}_{X_{k+1}^\gamma\in\mathsf{A}}\mathbb{1}_{x+\sqrt{\gamma}Z_{k+1}\in\mathcal{M}}] + (1-a^\gamma(x))\mathbb{1}_{\mathsf{A}}(x) \tag{121}$$

$$= a^\gamma(x)\hat{\mathrm{K}}^\gamma(x,\mathsf{A}) + (1-a^\gamma(x))\mathbb{1}_{\mathsf{A}}(x). \tag{122}$$

We define for any $\gamma>0$ and $x\in\mathcal{M}$

$$b^\gamma(x) = (1/\gamma)\textstyle\int_{\mathcal{M}}(y-x)\mathrm{K}^\gamma(x,\mathrm{d}y), \tag{123}$$

$$\Sigma^\gamma(x) = (1/\gamma)\textstyle\int_{\mathcal{M}}(y-x)^{\otimes 2}\mathrm{K}^\gamma(x,\mathrm{d}y), \tag{124}$$

$$\Delta^\gamma(x) = (1/\gamma)\textstyle\int_{\mathcal{M}}\|y-x\|^4\mathrm{K}^\gamma(x,\mathrm{d}y). \tag{125}$$

Using (122), we get that for any $\gamma > 0$ and $x \in \mathcal{M}$

$$b^\gamma(x) = a^\gamma(x)\hat{b}^\gamma(x), \qquad \Sigma^\gamma(x) = a^\gamma(x)\hat{\Sigma}^\gamma(x), \qquad \Delta^\gamma(x) = a^\gamma(x)\hat{\Delta}^\gamma(x). \tag{126}$$

Using Lemma 12, we have that for any $\gamma \in (0, \bar{\gamma})$ and $x \in \mathcal{M}$, $a^\gamma(x) \geq 1/4$.

In order to conclude for the convergence of the Metropolis process we adapt Theorem 22 and Theorem 23. We define $\mathbf{X}^\gamma : \mathbb{R}_+ \to \overline{\mathcal{M}}$ given for any $k \in \mathbb{N}$ by $\mathbf{X}^\gamma_{k\gamma} = X^\gamma_k$ and for any $t \in [k\gamma, (k+1)\gamma)$, $\mathbf{X}^\gamma_t = X^\gamma_k$. Note that $(\mathbf{X}_t)_{t\in[0,T]}$ is a $\mathrm{D}([0,T], \overline{\mathcal{M}})$ valued random variable, where $\mathrm{D}([0,T], \overline{\mathcal{M}})$ is the space of right-continuous with left-limit processes which take values in $\overline{\mathcal{M}}$. We denote $\mathbb{P}^\gamma$ the distribution of $(\mathbf{X}_t)_{t\in[0,T]}$ on $\mathrm{D}([0,T], \overline{\mathcal{M}})$.

Theorem 24. *There exists $\mathbb{P}^\star$ a distribution on $\mathrm{D}([0,T], \overline{\mathcal{M}})$ such that $\lim_{\gamma\to 0} \mathbb{P}^\gamma = \mathbb{P}^\star$. In addition, for any $f \in \mathrm{C}^{1,2}([0,T] \times \overline{\mathcal{M}}, \mathbb{R})$ with $\langle \nabla\Phi(\bar{x}), \nabla f(x)\rangle \geq 0$ for any $t \in [0,T]$ and $x \in \partial\mathcal{M}$, we have that the process $(f(t, \omega(t)))_{t\in[0,T]}$ given for any $t \in [0,T]$*

$$f(t,\omega(t)) - \textstyle\int_0^t (\partial_s f(s,\omega(s)) + \frac{1}{2}\Delta f(s,\omega(s)))\mathbb{1}_{\mathcal{M}}(\omega(s))\mathrm{d}s, \tag{127}$$

is a $\mathbb{P}$ submartingale.

Proof. Condition (A) (Stroock et al., 1971, p.197) is a consequence of Proposition 13 and (126). Condition (B) (Stroock et al., 1971, p.197) is a consequence of Proposition 18 and (126). Condition (C) (Stroock et al., 1971, p.198) is a consequence of Corollary 16 and (126). Condition (D) (Stroock et al., 1971, p.198) is a consequence of Proposition 14 and (126). We fix $\rho = 0$ and condition (1) (Stroock et al., 1971, p.203) is a consequence of Proposition 19 and that $\lim_{\gamma\to 0} a^\gamma = 1$ uniformly on compact subsets $\mathsf{K} \subset \mathcal{M}$. Condition (2)-(iii) (Stroock et al., 1971, p.203) is a consequence of Proposition 20 and (126). Condition (2)-(iv) (Stroock et al., 1971, p.203) is a consequence of Proposition 21 and (126). We conclude upon using (Stroock et al., 1971, Theorem 6.3) and (Stroock et al., 1971, Theorem 5.8). □

Theorem 25. *For any $T \geq 0$, $(\mathbf{X}^\gamma_t)_{t\in[0,T]}$ weakly converges to $(\mathbf{X}_t)_{t\in[0,T]}$ such that for any $t \in [0,T]$*

$$\mathbf{X}_t = x + \mathbf{B}_t - \mathbf{k}_t, \qquad |\mathbf{k}|_t = \textstyle\int_0^t \mathbb{1}_{\mathbf{X}_s\in\partial\mathcal{M}}\mathrm{d}|\mathbf{k}|_s, \qquad \mathbf{k}_t = \textstyle\int_0^t \mathbf{n}(\mathbf{X}_s)\mathrm{d}|\mathbf{k}|_s. \tag{128}$$

Proof. The proof is identical to Theorem 23. □

E Modelling geospatial data within non-convex boundaries

To demonstrate the ability of the proposed method to model distributions whose support is restricted to manifolds with highly non-convex boundaries, we derived a geospatial dataset based on the historical wildfire incidence rate within the continental United States (described in in Appendix E.1) and, using the corresponding country borders, trained a constrained diffusion model by adapting the point-in-spherical-polytope conditions outlined in (Ketzner et al., 2022) (described in Appendix E.2).

E.1 Derivation of bounded geospatial dataset

Specifically, we retrieved the rasterised version of the wildfire data provided by Welty et al. (2020), converted it to a spherical geodetic coordinate system using the CARTOPY library (Met Office, 2010 - 2015), and drew a weighted subsample of size 1×10^6. We then retrieved the country borders of the continental United States from (Natural Earth, 2023) and mapped them to the same geodetic reference frame as the wildfire data. A visualization of the resulting dataset is presented in Figure 9.

E.2 Point-in-spherical-polytope algorithms

The support of the data-generating distribution we aim to approximate is thus restricted to a highly non-convex spherical polytope $\mathbb{P} \in \mathcal{S}^2$ given by the country borders of the continental United States. To determine whether a query point $q \in \mathcal{S}^2$ is within $\mathbb{P}$, we adapt an efficient reformulation of the point-in-spherical-polygon algorithm (Bevis et al., 1989) presented in (Ketzner et al., 2022). The algorithm requires the provision of a reference point $r \in \mathcal{S}^2$ known to be located in $\mathbb{P}$ and determines



Figure 9: Orthographic projection of the wildfire dataset described in Appendix E. The projection is aligned with the centroid of the continental United States and zoomed in ten-fold for visual clarity. All visualisations of geospatial data were generated using the GEOVIEWS (Rudiger et al., 2023) and DATASHADER (Bednar et al., 2023) libraries.

whether q is inside or outside the polygon by checking whether the geodesic between r and q crosses the polygon an even or odd number of times. Letting $\hat{x} \in \mathbb{R}^3$ denote the Cartesian coordinates of a point $x \in \mathcal{S}^2$, (Ketzner et al., 2022) rely on a Cartesian reference coordinate system $\hat{Q}$ (with its z-axis given by $\hat{r}$) and the corresponding spherical coordinate system Q to decompose the edge-crossing condition of Bevis et al. (1989) into two efficiently computable parts. That is, the geodesic between q and r crosses an edge $e_i = (v_i, v_j)$ of the polygon if:

(i) the longitude of q in Q is bounded by the longitudes of v_i and v_j in Q, i.e.

$$\phi_Q(q) \in [\min(\phi_Q(v_i), \phi_Q(v_j)), \max(\phi_Q(v_i), \phi_Q(v_j))],$$

(ii) the plane specified by the normal vector $\hat{p}_i = \hat{v}_i \times \hat{v}_j$ represents an equator that separates q and r into two different hemispheres, i.e.

$$\text{sign}(\langle \hat{p}_i, \hat{r} \rangle \cdot \langle \hat{p}_i, \hat{q} \rangle) = -1.$$

Especially when $\mathbb{P}$ is fixed and the corresponding coordinate transformations and normal vectors can be precomputed for each edge, this algorithm affords an efficient and parallelisable approach to determining whether any given point on $\mathcal{S}^2$ is contained by a spherical polytope.

F Supplementary Experimental Results

F.1 Evaluating log-barrier and Euclidean models

Following (Fishman et al., 2023), we approached the empirical evaluation of our Metropolis model by computing the maximum mean discrepancy (MMD) (Gretton et al., 2012) between samples from the true distribution and the trained diffusion models. The MMD is a statistic that quantifies the similarity of two samples by computing the distance of their respective mean embeddings in a reproducing kernel Hilbert space. For this, we use an RBF kernel with the same length scales as the standard deviations of the normal distributions used to generate the synthetic distribution. We sum these RBF kernels by the weights of the corresponding components of the synthetic Gaussian mixture model.

This is essential to be able to include the log-barrier in the comparison since the log-barrier methods suffer severe instabilities around the boundary, as the space is stretched to more and more. These instabilities cause the problems in fitting the log-barrier model and in computing the likelihood using the log-barrier model.

Table 5: Maximum mean discrepancy (MMD) (↓) of a held-out test set from a synthetic bimodal distribution over convex subsets of $\mathbb{R}^d$ bounded by the hypercube $[-1, 1]^d$ and unit simplex Δ^d. Means and standard deviations are computed over 3 different runs.

Manifold	Dimension	Process	MMD		% in Manifold
			mean	std	mean
Δ^d	2	Euclidean	0.027	0.011	0.969
		Log-Barrier	0.050	0.012.	1.000
		Reflected	0.041	0.008	1.000
		Rejection	0.030	0.002	1.000
	3	Euclidean	0.032	0.015	0.969
		Log-Barrier	0.238	0.009	1.000
		Reflected	0.179	0.013	1.000
		Rejection	0.111	0.002	1.000
	10	Euclidean	0.028	0.001	0.946
		Log-Barrier	0.275	0.0015	1.000
		Reflected	0.233	0.004	1.000
		Rejection	0.226	0.005	1.000
$[0, 1]^d$	2	Euclidean	0.069	0.004	0.992
		Log-Barrier	0.66	0.006	1.000
		Reflected	0.048	0.012	1.000
		Rejection	0.025	0.005	1.000
	3	Euclidean	0.074	0.004	0.991
		Log-Barrier	0.209	0.0077	1.000
		Reflected	0.085	0.006	1.000
		Rejection	0.049	0.006	1.000
	10	Euclidean	0.086	0.007	0.968
		Log-Barrier	0.330	0.004	1.000
		Reflected	0.314	0.049	1.000
		Rejection	0.138	0.007	1.000

From the results in Table 5, it is clear that the log-barrier approach performs significantly worse than the Reflected model and the Metropolis models across all settings. This, in conjunction with numerical instabilities we encountered when attempting to evaluate sample likelihoods with the log-barrier models as presented in (Fishman et al., 2023), motivated us to focus on the Reflected and Metropolis models in the main text.

Additionally, we note that the unconstrained Euclidean models outperform the constrained methods on both the simplex and the hypercube as the dimensionality of the problem space increases. Especially on the simplex, we attribute this performance primarily to the fact that the synthetic distribution is simply a standard Normal with only a small portion close to the boundary. The amount of reflection needed to model the distribution decreases in higher dimensions, as the mass of the Normal distribution gets increasingly concentrated—which Euclidean diffusion models will fit well. This same dynamic is partially responsible for the hypercube performance.

F.2 Implementational details

All source code that is needed to reproduce the results presented below is made available under https://github.com/oxcsml/score-sde/tree/metropolis, which requires a supporting package to handle the different geometries that is available under https://github.com/oxcsml/geomstats/tree/polytope.

We use the same architecture in all of our experiments: a 6-layer MLP with 512 hidden units and sine activation functions, except in the output layer, which uses a linear activation function. Following (Fishman et al., 2023), we implement a simple linear function that scales the score by the distance to the boundary, approaching zero within $\epsilon = 0.01$ of the boundary. This ensures the score obeys the Neumann boundary conditions required by the reflected Brownian Motion. For the geospatial dataset within non-convex country borders, we do not use distance rescaling. Instead, we substitute

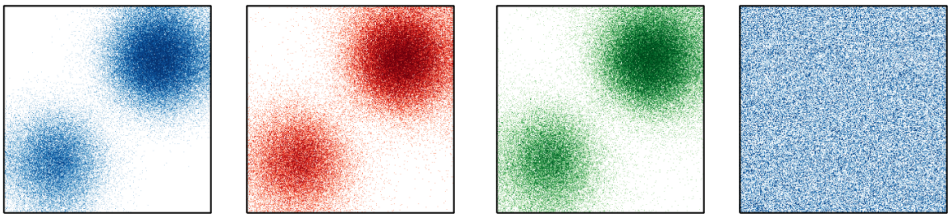
it with a series of step functions to rescale the score. This is a proof-of-concept to show that even when computing the distance is hard, simple and efficient approximations suffice. When constructing Riemannian diffusion models on the torus and sphere for the protein and geospatial datasets, we follow (De Bortoli et al., 2022) and include an additional preconditioner for the score on the manifold. We *do not* use the residual trick or the standard deviation trick, which are both common score-rescaling functions in image model architectures; in our setting, we find that they adversely affect model training.

For the forward/reverse process we always set $T = 1$, $\beta_0 = 1 \times 10^{-3}$ and then tune β_1 to ensure that the forward process just reaches the invariant distribution with a linear β-schedule. At sampling time we use $N = 100$ steps of the discretised process. We discretise the training process by selecting a random N between 0 and 100 for each example, rolling out to that time point. This lets us cheaply implement a simple variance reduction technique: we take multiple samples from this trajectory by selecting multiple random N to save for each example. This technique was originally described in (Fishman et al., 2023) and we find it is also helpful for our Metropolis models. For all experiments, we use the ism loss with a modified weighting function of $(1 + t)$, which we found to be essential to model training. All experiments use a batch size of 256 with 8 repeats per batch. For training, we use a learning rate of 2×10^{-4} with a cosine learning rate schedule. We trained for 100,000 batches on the synthetic examples and 300,000 batches on the real-world examples (robotics, proteins, wildfires).

We selected these hyperparameters from a systematic search over learning rates (6×10^{-4}, 2×10^{-4}, 6×10^{-5}, 2×10^{-5}), learning rate schedules (cosine, log-linear), and batch sizes (128, 256, 512, 1024) on synthetic examples for the reflected and log-barrier models. Similar parameters worked well for both, and we used those for our Metropolis models to allow a straightforward comparison. We tried $N = 100, 1000$ for several synthetic examples but found that very large rollout times actually hurt performance for the Metropolis model, though the log-barrier performed a bit better with longer rollouts and the reflected was the same.

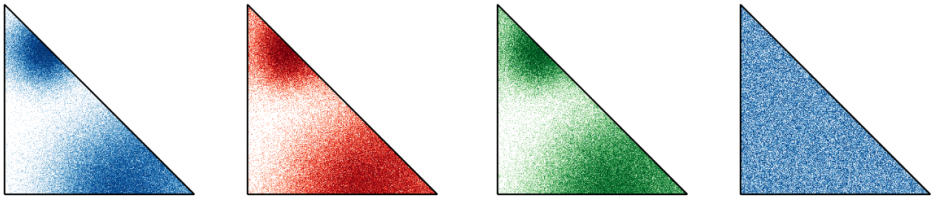
All models were trained on a single NVIDIA GeForce GTX 1080 GPU. All of the Metropolis models presented here can easily be trained on this hardware in under 4 hours. The runtime for the log-barrier and reflected models is considerably longer.

F.3 Synthetic Distributions on Constrained Manifolds of Increasing Dimensionality



(a) Data distribution (b) Metropolis model (c) Reflected model (d) Uniform distribution

Figure 10: Qualitiative comparison of samples from the data distribution, our Metropolis model, a Reflected model and the uniform distribution for a synthetic bimodal distribution on $[-1, 1]^2$.



(a) Data distribution (b) Metropolis model (c) Reflected model (d) Uniform distribution

Figure 11: Qualitiative comparison of samples from the data distribution, our Metropolis model, a Reflected model and the uniform distribution for a synthetic bimodal distribution on Δ^2.

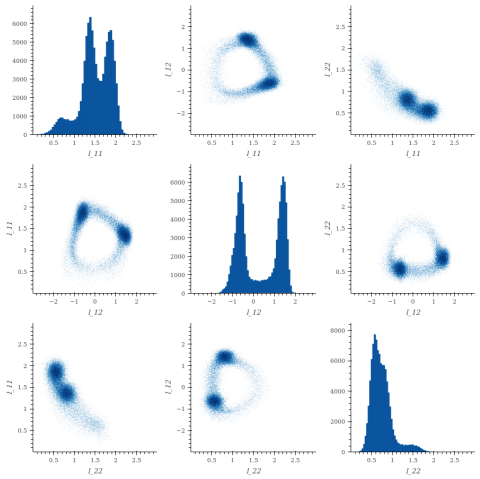
F.4 Constrained Configurational Modelling of Robotic Arms

The following univariate marginal and pairwise bivariate plots visualise the distribution of different samples in

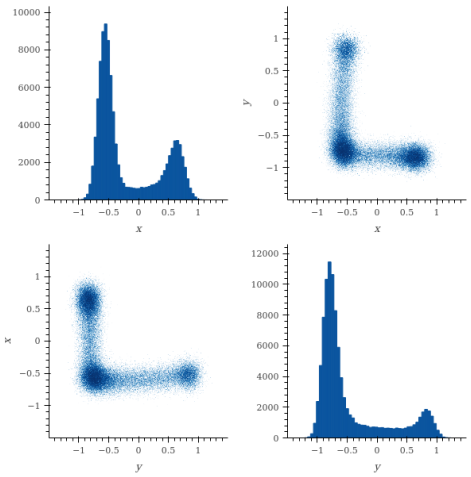
(i) the three dimensions needed to describe an ellipsoid $M = \begin{bmatrix} l_1 & l_2 \\ l_2 & l_3 \end{bmatrix} \in \mathcal{S}^2_{++}$ and

(ii) the two dimensions needed to describe a location in $\mathbb{R}^2$.

F.4.1 Visualisation of samples from the data distribution



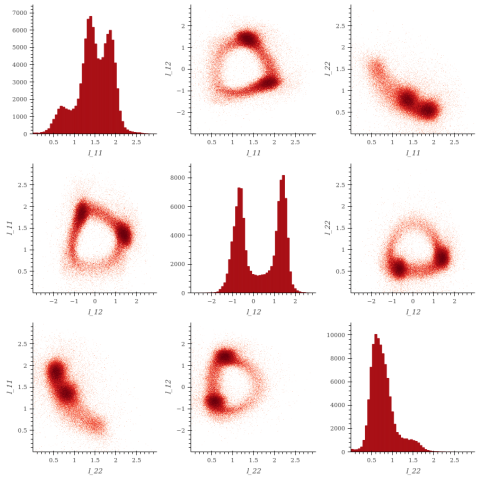
(a) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from the data distribution in $\mathcal{S}^2_{++}$.



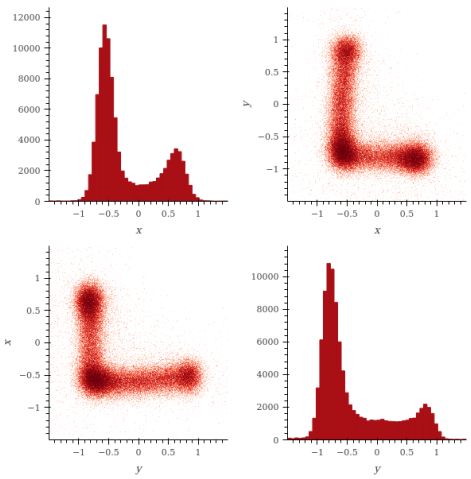
(b) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from the data distribution in $\mathbb{R}^2$.

Figure 12: Visualisation of the data distribution in $\mathcal{S}^2_{++} \times \mathbb{R}^2$ using univariate marginal and pairwise bivariate plots.

F.4.2 Visualisation of samples from our Metropolis sampling-based diffusion model



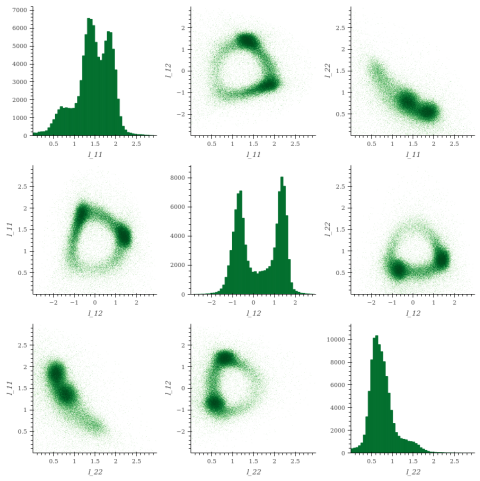
(a) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from our Metropolis sampling-based diffusion model in $\mathcal{S}^2_{++}$.



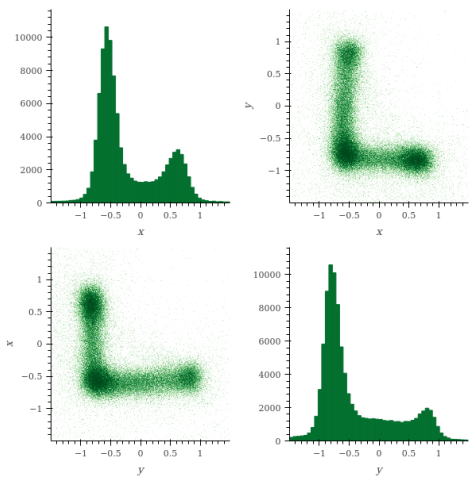
(b) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from our Metropolis sampling-based diffusion model in $\mathbb{R}^2$.

Figure 13: Visualisation of the distribution learned by our Metropolis sampling-based diffusion model in $\mathcal{S}^2_{++} \times \mathbb{R}^2$ using univariate marginal and pairwise bivariate plots.

F.4.3 Visualisation of samples from a reflected Brownian motion-based diffusion model



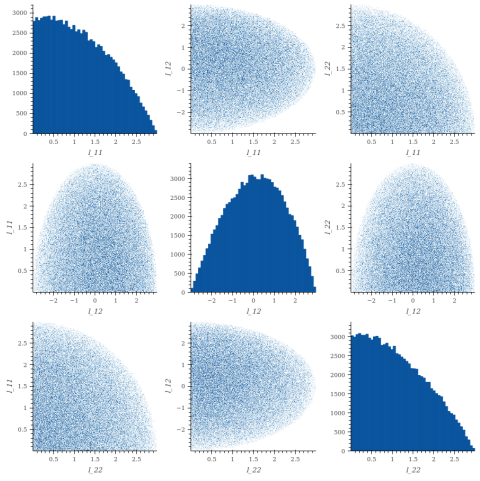
(a) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from a reflected Brownian motion-based diffusion model in $\mathcal{S}^2_{++}$.



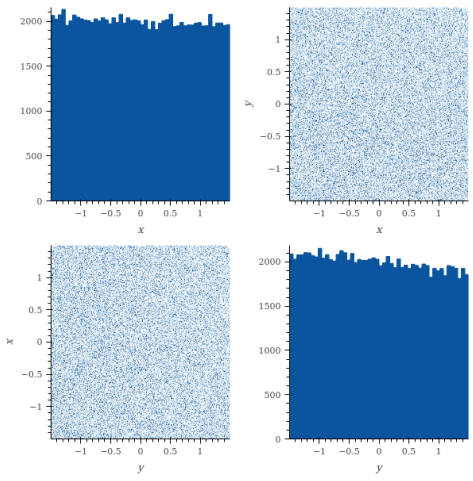
(b) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from a reflected Brownian motion-based diffusion model in $\mathbb{R}^2$.

Figure 14: Visualisation of the distribution learned by a reflected Brownian motion-based diffusion model in $\mathcal{S}^2_{++} \times \mathbb{R}^2$ using univariate marginal and pairwise bivariate plots.

F.4.4 Visualisation of samples from the uniform distribution



(a) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from the uniform distribution in $\mathcal{S}^2_{++}$.



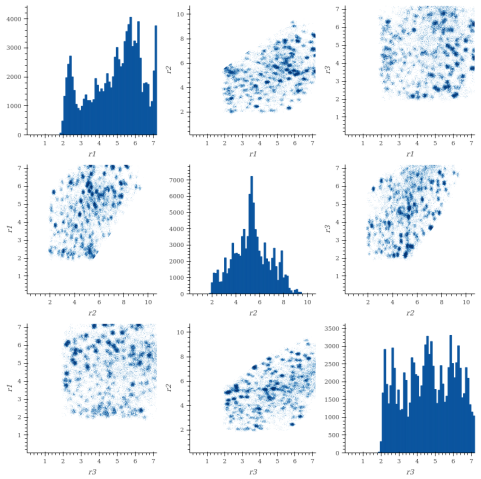
(b) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from the uniform distribution in $\mathbb{R}^2$.

Figure 15: Visualisation of the uniform distribution in $\mathcal{S}^2_{++} \times \mathbb{R}^2$ using univariate marginal and pairwise bivariate plots.

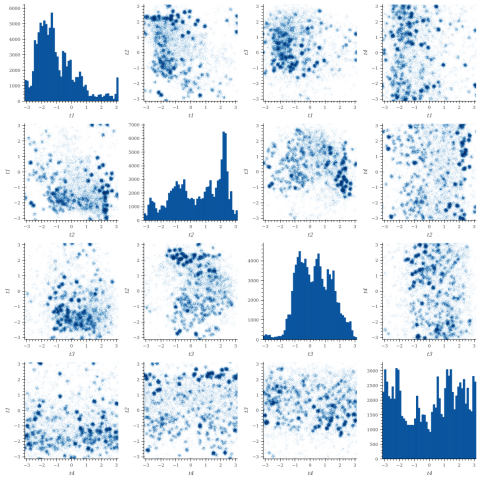
F.5 Conformational Modelling of Protein Backbones

The following univariate marginal and pairwise bivariate plots visualise the distribution of different samples in (i) the polytope $\mathbb{P} \subset \mathbb{R}^3$ and (ii) the torus $\mathbb{T}^4$ used to parametrise the conformations of a polypeptide chain of length $N = 6$ with coinciding endpoints. We refer to (Han et al., 2006) for full detail on the reparametrisation and to (Fishman et al., 2023) for a full description of the dataset.

F.5.1 Visualisation of samples from the data distribution



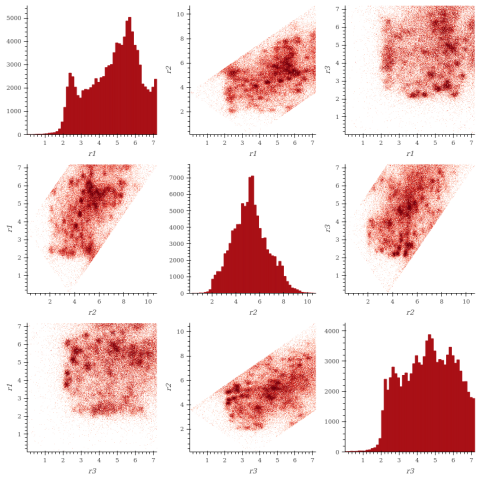
(a) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from the data distribution in $\mathbb{P} \subset \mathbb{R}^3$.



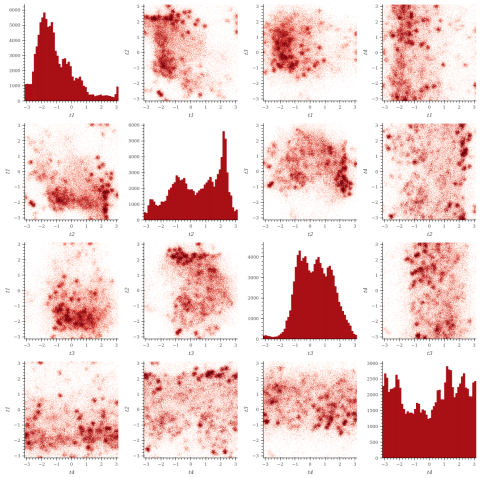
(b) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from the data distribution in $\mathbb{T}^4$.

Figure 16: Visualisation of the data distribution in $\mathbb{P} \subset \mathbb{R}^3 \times \mathbb{T}^4$ using univariate marginal and pairwise bivariate plots.

F.5.2 Visualisation of samples from our Metropolis sampling-based diffusion model



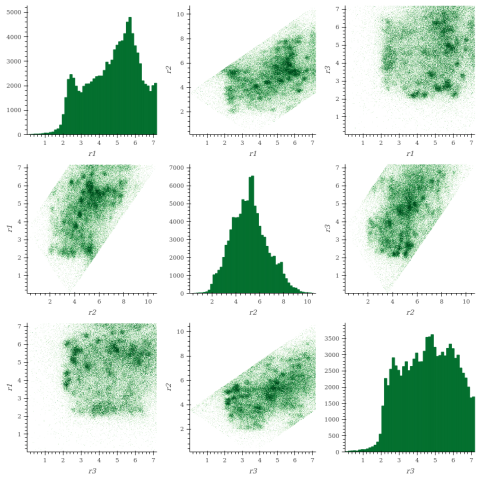
(a) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from our Metropolis model in $\mathbb{P} \subset \mathbb{R}^3$.



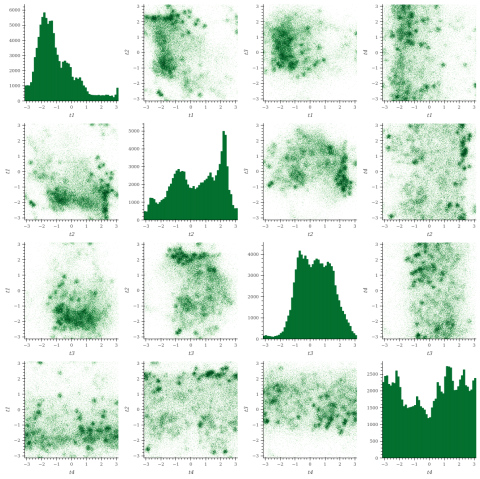
(b) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from our Metropolis model in $\mathbb{T}^4$.

Figure 17: Visualisation of the distribution learned by our Metropolis model in $\mathbb{P} \subset \mathbb{R}^3 \times \mathbb{T}^4$ using univariate marginal and pairwise bivariate plots.

F.5.3 Visualisation of samples from a reflected Brownian motion-based diffusion model



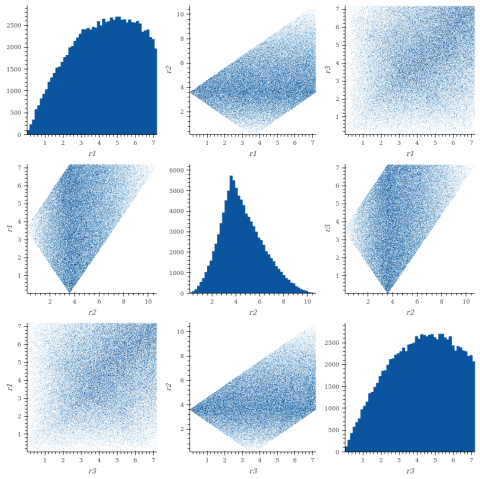
(a) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from a reflected Brownian motion-based diffusion model in $\mathbb{P} \subset \mathbb{R}^3$.



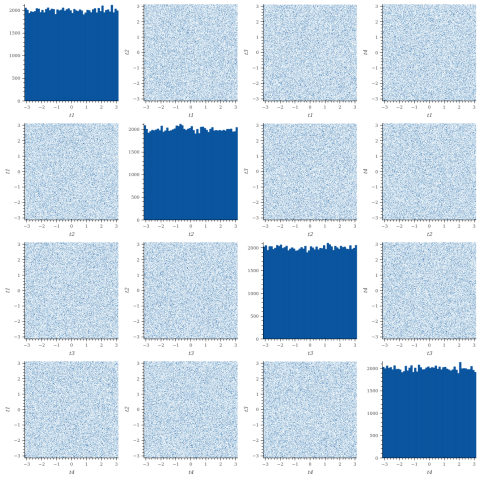
(b) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from a reflected Brownian motion-based diffusion model in $\mathbb{T}^4$.

Figure 18: Visualisation of the distribution learned by a reflected Brownian motion-based diffusion model in $\mathbb{P} \subset \mathbb{R}^3 \times \mathbb{T}^4$ using univariate marginal and pairwise bivariate plots.

F.5.4 Visualisation of samples from the uniform distribution



(a) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from the uniform distribution in $\mathbb{P} \subset \mathbb{R}^3$.



(b) Plots of the univariate marginal and pairwise bivariate distributions of 1×10^5 samples from the uniform distribution in $\mathbb{T}^4$.

Figure 19: Visualisation of the uniform distribution in $\mathbb{P} \subset \mathbb{R}^3 \times \mathbb{T}^4$ using univariate marginal and pairwise bivariate plots.

.

Appendix D

Supplementary to *Drug Discovery under Covariate Shift*

Additional Methodological Details

To provide a more detailed exposition of how the function-space prior in the work presented in Chapter 6 is constructed, the following section elaborates on the technical details of the employed approach.

It is based on the function-space variational inference (FSVI) framework first introduced by Rudner et al. (2020) that aims to define meaningful priors in Bayesian neural networks by specifying the desired behaviour over the space of learned neural network functions. This allows for an intuitive *function-space prior* $p_{f(\cdot;\Theta)}$ that is designed to enforce high predictive uncertainty on inputs dissimilar to the training data. In the present work, this takes the form of a high-entropy prior characterized by a near-zero logit-space mean and minimal off-diagonal covariance structure. Since it is not tractable to specify this function-space prior directly, it is instead *induced* by an appropriate parameter-space prior p_Θ, which we choose to be an isotropic Gaussian. The push-forward mechanism relies on a first-order Taylor expansion of the network mapping $f(\cdot;\Theta)$ around the mean m of the parameter distribution:

$$\tilde{f}(\cdot; m, \Theta) \approx f(\cdot; m) + \mathcal{J}(\cdot; m)(\Theta - m)$$

where $\mathcal{J}(\cdot; m) \coloneqq (\partial f(\cdot;\Theta)/\partial\Theta)|_{\Theta=m}$ is the network's Jacobian with respect to its parameters. Under this local linearization scheme, a Gaussian parameter-space prior $p_\Theta = \mathcal{N}(m, S)$ induces a Gaussian function-space prior whose mean and covariance, evaluated at inputs $\mathbf{X}$ and $\mathbf{X}'$, are:

$$\begin{aligned}\mathbb{E}\left[\tilde{f}(\mathbf{X};\Theta)\right] &= f(\mathbf{X}; m)\\ \operatorname{Cov}\left[\tilde{f}(\mathbf{X};\Theta), \tilde{f}(\mathbf{X}';\Theta)\right] &= \mathcal{J}(\mathbf{X}; m) S \mathcal{J}(\mathbf{X}', m)^\top.\end{aligned}$$

At each training iteration, this prior is enforced by optimizing the Kullback-Leibler divergence objective $\mathbb{D}_{KL}(q_{f(\cdot;\Theta)} \| p_{f(\cdot;\Theta)})$ with respect to this prior and the the learned function distribution, $q_{f(\cdot;\Theta)}$, evaluated on a set of unlabelled context points sampled uniformly from the ZINC database.

Appendix

A. Data Curation and Pre-Processing

To generate an appropriate dataset of reliably labeled bioactivity measurements, we retrieved and reprocessed high-throughput screening data generated by Antonova-Koch et al. (2018) as part of a campaign to discover novel chemoprotective antimalarial drug candidates.

The authors established a cell-based phenotypic screening pipeline to identify compounds that inhibit the development of luciferase-expressing liver-stage *Plasmodium falciparum* parasites. After assaying a commercially-available chemical library of 538 273 of drug-like small molecules in a single-point primary screen, they selected the 9963 most promising compounds for a series of confirmatory dose-response screens. Specifically, an 8-point dilution series was used to assess, in duplicate, the potency and efficacy of each compound in the original assay (*Pbluc*). Additionally, the tendency of the assayed compounds to produce false positives and other experimental artifacts was investigated by performing a series of counter-screens that measure hepatic cytotoxicity (*HepG2tox*) and interference with the luciferase-based luminescent readout (*Ffluc*). The fact that all bioactivity measurements are (1) generated using biological duplicates and (2) associated with quantitative measures that reflect their likelihood to produce confounding experimental artifacts substantially improves the reliability of the resulting labels.

To facilitate the integration of bioactivity and counter-screen measurements and make the data more amenable to predictive modeling, the IC_{50} values that quantify the concentration at which a molecule produces half of its maximum inhibitory effect were converted to binary labels. Specifically, all compounds with an $\mathrm{IC}_{50} \leq 1.5\mu\mathrm{M}$ were denoted as active while all compounds with an $\mathrm{IC}_{50} \geq 3\mu\mathrm{M}$ were denoted as inactive, discarding 652 compounds with $1.5\mu\mathrm{M} \leq \mathrm{IC}_{50} \leq 3\mu\mathrm{M}$ and assigning qualified IC_{50} values to the appropriate class (see Figure 6 for a diagram of the IC_{50} distribution and the applied thresholds).

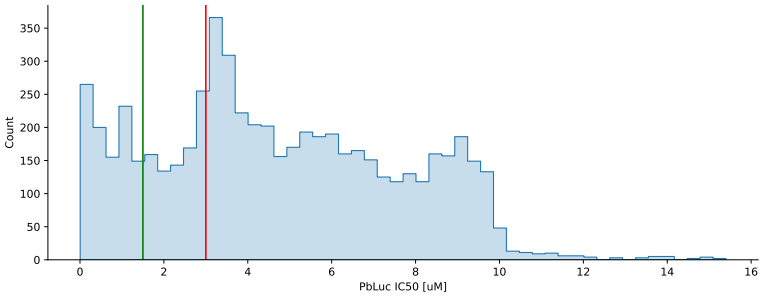


Figure 6. A histogram of the distribution of non-qualified *Pbluc* IC_{50} values. The green and red vertical lines indicate the thresholds (set to $\mathrm{IC}_{50} = 1.5\mu\mathrm{M}$ and $\mathrm{IC}_{50} = 3\mu\mathrm{M}$) that determine whether a compound is assigned to the active or inactive class.

In order to integrate information from the *HepG2tox* and *Ffluc* counter-screens and filter out problematic compounds that are likely false positives or risk confounding the evaluation in other ways, the thresholds outlined in Antonova-Koch et al. (2018) were applied. In particular, problematic compounds were discarded due to causing hepatotoxicity or assay interference if their respective IC_{50} values met at least one of the criteria outlined in Equation (A.1) and Equation (A.2)

$$\textit{HepG2tox}\ \mathrm{IC}_{50} < 2 \cdot \textit{Pbluc}\ \mathrm{IC}_{50} \wedge \textit{HepG2tox}\ \mathrm{IC}_{50} < c_{\max} \tag{A.1}$$

$$\textit{Ffluc}\ \mathrm{IC}_{50} < 2 \cdot \textit{Pbluc}\ \mathrm{IC}_{50} \wedge \textit{Ffluc}\ \mathrm{IC}_{50} < c_{\max}, \tag{A.2}$$

where $c_{\max}$ denotes the maximum concentration a compound was assayed at. These filtering criteria categorized 764 compounds as inhibiting hepatocyte viability and 446 compounds as interfering with the luminescence readout, including an overlap of 49. Removing these compounds from the dataset results in a total of 8150 compounds, of which 7301 (90%) are labeled as inactive and 849 (10%) are labeled as active.

B. Additional Experimental Details

This section provides additional implementational details of our experimental setup. Appendix B.1 presents the test set performance of each model under different data splitting and featurization techniques using the AUC-PRC and the ACE scores. Additional experimental details are provided in Appendix B.2, describing the implementation and hyperparameter screening ranges for each model in our empirical evaluation, namely logistic regression (Appendix B.2.1), random forest classifiers (Appendix B.2.2), multi-layer perceptrons (Appendix B.2.3), deep ensembles (Appendix B.2.4), pre-trained graph neural networks (Appendix B.2.5), GROVER (Appendix B.2.6), various domain adaptation and generalization techniques (Appendix B.2.7), and Q-SAVI (Appendix B.2.8). Detailed ablation plots that explore the influence of different hyperparameters on the performance of Q-SAVI are presented in Appendix B.3.

All experiments and analyses were performed in Python (Van Rossum and Drake Jr, 1995), using a range of general-purpose packages to aid model development and analysis (Harris et al., 2020; Waskom, 2021; Wes McKinney, 2010; Virtanen et al., 2020). All code that is necessary to reproduce the results presented in this work is available in the following repository: https://github.com/leojklarner/Q-SAVI.

B.1. Test Set Performances in AUC-PRC and ACE

Following model training and hyperparameter selection, the predictive accuracy and calibration of the estimated test-set label probabilities were assessed using the area under the ROC curve (AUC-ROC) and the Brier score. These metrics allow for a direct comparison of models across test sets with different label distributions (see Table 2). In addition, the performance of each algorithm was evaluated using the area under the precision-recall curve (AUC-PRC) and the adaptive calibration error (ACE; Nixon et al. (2019)). AUC-PRC and ACE are particularly well-suited for imbalanced datasets, and provide a characterization of model performance that closely aligns with AUC-ROC and BRIER SCORE (see Table 5). Note that their performance is not comparable across data splits, as it depends on the label distribution of a given test set—while the AUC-ROC of a no-skill classifier is 0.5, its AUC-PRC is given by the positive label probability $p(y = 1)$ of the test set.

Table 5. An overview of the test set performance of each model for each data splitting and featurization technique, quantified by the AUC-PRC (↑) and the ACE (↓) scores. All entries indicate the mean and standard error computed over 10 independent training runs with different random seeds. The best models within a margin of statistical significance are highlighted in bold.

	Model & Featurization	Spectral Split ECFP	Spectral Split rdkitFP	Weight Split ECFP	Weight Split rdkitFP	Scaffold Split ECFP	Scaffold Split rdkitFP	Random Split ECFP	Random Split rdkitFP
AUC-PRC (↑)	Logistic Regression	.211±.000	.140±.000	.106±.000	**.112±.000**	.211±.000	.211±.000	.248±.000	.225±.000
	Random Forest	.207±.005	.141±.002	.090±.002	.089±.003	.165±.002	.200±.002	**.292±.002**	**.294±.002**
	MLP	.208±.003	.144±.001	.090±.003	.089±.004	.180±.005	.184±.007	.270±.004	.233±.006
	Deep Ensemble	.217±.003	.144±.001	.114±.001	.102±.003	.209±.002	.209±.002	.288±.002	.266±.004
	GIN	.183±.006	.149±.005	.082±.004		.202±.005		.218±.005	
	GIN (attr masking)	.192±.003	.152±.005	.108±.003		**.245±.004**		.251±.003	
	GIN (context pred)	.188±.004	.152±.004	.098±.002		.206±.005		.254±.005	
	Grover	.199±.001	.139±.002	.106±.001		.204±.004		.227±.003	
	IRM	.154±.004	.145±.004	.086±.004		.178±.005		.176±.004	
	GroupDRO	.166±.003	.159±.002	.102±.003		.202±.003		.172±.005	
	DANN	.156±.003	.155±.005	.095±.005		.184±.004		.202±.003	
	DeepCoral	.154±.004	.151±.004	.091±.003		.194±.003		.212±.003	
	Q-SAVI	**.221±.003**	**.165±.004**	**.121±.002**	**.111±.003**	.197±.003	.216±.003	.239±.002	.208±.004
ACE (↓)	Logistic Regression	.061±.000	.055±.000	.041±.000	.034±.000	.026±.000	**.025±.000**	.018±.000	.024±.000
	Random Forest	.078±.001	**.033±.001**	.074±.001	.087±.001	.029±.001	**.025±.001**	**.016±.001**	.035±.001
	MLP	.079±.003	.052±.003	.035±.003	.055±.007	.029±.002	.044±.011	.029±.001	.026±.002
	Deep Ensemble	.078±.004	.050±.001	.025±.001	.053±.005	**.022±.001**	**.025±.001**	.023±.001	.019±.001
	GIN	.064±.004	.047±.007	.036±.003		.033±.003		.026±.003	
	GIN (attr masking)	**.053±.002**	.057±.009	.038±.002		.030±.001		.020±.001	
	GIN (context pred)	.078±.002	.051±.005	.034±.003		.028±.002		**.015±.001**	
	Grover	.074±.004	**.035±.002**	.036±.002		.038±.002		.020±.001	
	IRM	.071±.003	.067±.002	.044±.002		.035±.001		.024±.002	
	GroupDRO	.060±.003	**.035±.003**	.039±.002		.036±.002		.026±.001	
	DANN	.057±.002	.046±.003	.035±.003		.028±.001		.030±.002	
	DeepCoral	.097±.006	**.035±.002**	.041±.004		.036±.002		.026±.002	
	Q-SAVI	**.052±.001**	.043±.013	**.015±.001**	**.016±.001**	.036±0.002	**.025±.002**	.021±.001	.024±.002

B.2. Model Implementations and Hyperparameter Ranges

To ensure a fair and meaningful comparison of the evaluated machine learning models, the hyperparameters of each algorithm were independently optimized for every data split and featurization technique. The following sections provide comprehensive details about the implementation and hyperparameter ranges used for each model in our empirical evaluation.

- Logistic Regression (Section B.2.1)
- Random Forest Classifiers (Section B.2.2)
- Multi-layer Perceptrons (Section B.2.3)
- Deep Ensembles (Section B.2.4)
- Pre-trained Graph Neural Networks (Section B.2.5)
- GROVER (Section B.2.6)
- Domain Adaptation and Generalization Techniques (Section B.2.7)
- Our Probabilistic Regularization Scheme (Section B.2.8)

B.2.1. Logistic Regression

The **logistic regression models** were trained with the scikit-learn library (Pedregosa et al., 2011) using the LIBLINEAR solver (Fan et al., 2008) with a maximum of 1000 iterations and a stopping tolerance of 1×10^{-4}. They were independently fit for all hyperparameter combinations specified inTable 6, using the combination with the best unweighted validation set log-likelihood to choose the best hyperparameter setting to evaluate on the held-out test set.

Table 6. Hyperparameters for Logistic Regression

Model	Hyperparameter	Search Space
Linear Regression	regularization type	$\ell 1$, $\ell 2$
	regularization strength	1.0×10^{-4}, $2.6 \times 10^{-4} \ldots$, 3.8×10^{3}, 1.0×10^{4}
	class weight	none, balanced

B.2.2. Random Forest Classifiers

The **random forest models** were trained with the scikit-learn library (Pedregosa et al., 2011) using 100 decision trees and the GINI splitting criterion. They were independently fit for all hyperparameter combinations specified in Table 7, using the combination with the best unweighted validation set log-likelihood to choose the best hyperparameter setting to evaluate on the held-out test set.

Table 7. Hyperparameters for Random Forest Classifiers

Model	Hyperparameter	Search Space
Random Forest	maximum depth	5, 15, 26, 36, 47, 57, 68, 78, 89, 100
	min. samples per split	5, 15, 50, 100
	min. samples per leaf	1, 5, 10, 30, 100
	class weight	none, balanced

B.2.3. MULTI-LAYER PERCEPTRONS

The **multi-layer perceptrons** were implemented with the PyTorch library (Paszke et al., 2019), using rectified linear units (Nair and Hinton, 2010) as activation functions. Their weights were initialized using a Normal distribution $\mathcal{N}(0, 1)$ truncated at $\pm 2\sigma$, with biases initialized at zero. These parameters were optimized on the training set using the ADAMW stochastic gradient descent optimizer (Loshchilov and Hutter, 2017) with a batch size of 128 and the cross-entropy loss for a maximum of 500 epochs, using early stopping to terminate training if the unweighted log-likelihood on the validation set did not decrease for more than 10 epochs, reverting to the checkpoint with best validation set log-likelihood for evaluating their performance for hyperparameter optimization and the subsequent on the held-out test set. Batch normalization and dropout were applied after the ReLU non-lineary. The full hyperparameter search space is presented in Table 8.

Table 8. Hyperparameters for Multi-layer Perceptrons

Model	**Hyperparameter**	**Search Space**
Multi-layer Perceptron	learning rate	1×10^{-4}, 1×10^{-3}
	weight decay	1×10^{-3}, 1×10^{-2}, 1×10^{-1}
	number of layers	2, 4, 6
	embedding dimension	32, 64
	batch normalization (BN)	yes, no
	BN running statistics	yes, no
	dropout	0.0, 0.2, 0.5
	class weight	none, balanced

B.2.4. DEEP ENSEMBLES

The **deep ensembles** were trained using an identical setup to the multi-layer perceptrons, with the distinction that $M = 5$ independent networks were trained with different random seeds and evaluated with respect to their average log-likelihood on the validation set. Similarly, at inference time the class probabilities were averaged across ensembles. The full hyperparameter search space is presented in Table 9 and is identical to Table 8.

Table 9. Hyperparameters for Deep Ensembles

Model	**Hyperparameter**	**Search Space**
Deep Ensemble	learning rate	1×10^{-4}, 1×10^{-3}
	weight decay	1×10^{-3}, 1×10^{-2}, 1×10^{-1}
	number of layers	2, 4, 6
	embedding dimension	32, 64
	batch normalization (BN)	yes, no
	BN running statistics	yes, no
	dropout	0.0, 0.2, 0.5
	class weight	none, balanced

B.2.5. Pre-trained Graph Neural Networks

The graph featurization pipeline, architectures, and pre-trained initializations of the graph isomorphism networks presented in Hu et al. (2019) were retrieved from the paper's official GitHub repository and fine-tuned on the training set using the AdamW optimizer (Loshchilov and Hutter, 2017) with a batch size of 128 and the cross-entropy loss for a maximum of 500 epochs, using early stopping to terminate training if the unweighted log-likelihood on the validation set did not decrease for more than 10 epochs and reverting to the checkpoint with best validation set log-likelihood for evaluating their performance for hyperparameter optimization and the subsequent on the held-out test set. The full hyperparameter search space is presented in Table 10. The pre-trained initializations were provided for networks with 5 layers of 300 hidden units, set up using batch normalization with running statistics.

Table 10. Hyperparameters for Pre-trained GINs

Model	**Hyperparameter**	**Search Space**
Pre-trained GINs	learning rate	1×10^{-4}, 3×10^{-3}, 1×10^{-3}, 3×10^{-3}, 1×10^{-2}
	weight decay	1×10^{-3}, 1×10^{-2}, 1×10^{-1}
	dropout	0.0, 0.2, 0.5
	class weight	none, balanced

B.2.6. GROVER

All code, models, and initializations required to fine-tune the pre-trained graph transformers presented in Rong et al. (2020) was retrieved from the paper's official GitHub repository and fine-tuned on the training set with a batch size of 128 for a maximum of 500 epochs, using early stopping to terminate training if the unweighted log-likelihood on the validation set did not decrease for more than 10 epochs and reverting to the checkpoint with best validation set log-likelihood for evaluating their performance for hyperparameter optimization and the held-out test set. The hyperparameters specifying the number of layers and their embedding dimension indicate the size of the MLP fit on top of the pre-trained molecular representations produced by the GROVER base model and were chosen to be identical to the other MLP-based deep learning algorithms. The full hyperparameter search space is presented in Table 11.

Table 11. Hyperparameters for GROVER

Model	**Hyperparameter**	**Search Space**
GROVER	learning rate	1×10^{-4}, 1×10^{-3}
	weight decay	1×10^{-3}, 1×10^{-2}, 1×10^{-1}
	dropout	0.0, 0.2, 0.5
	number of layers	2, 4, 6
	embedding dimension	32, 64

B.2.7. Domain Adaptation Techniques

All code and featurization utilities required to run the evaluated domain adaptation and generalization techniques, namely invariant risk minimization (**IRM**), group-distributionally robust training (**GroupDRO**), domain-adversarial networks (**DANN**) and deep correlation alignment (**DeepCoral**), were adapted from Ji et al. (2022) and provided with data split-specific domain indicators. For this, the training set was additionally split into three domains, either using spectral clustering, molecular weight thresholds, a grouped scaffold split, or random partitions. All models used the default architecture choice in Ji et al. (2022)—a graph isomorphism network with 4 layers and 128 hidden units—and trained according to the respective optimization procedures. The full hyperparameter range is presented in Table 12.

Table 12. Hyperparameters for Domain Adaptation Techniques

Model	**Hyperparameter**	**Search Space**
IRM/GroupDRO/DANN/DeepCoral	learning rate	1×10^{-4}, 1×10^{-3}
	weight decay	1×10^{-3}, 1×10^{-2}, 1×10^{-1}
	dropout	0.0, 0.2, 0.5

B.2.8. Q-SAVI

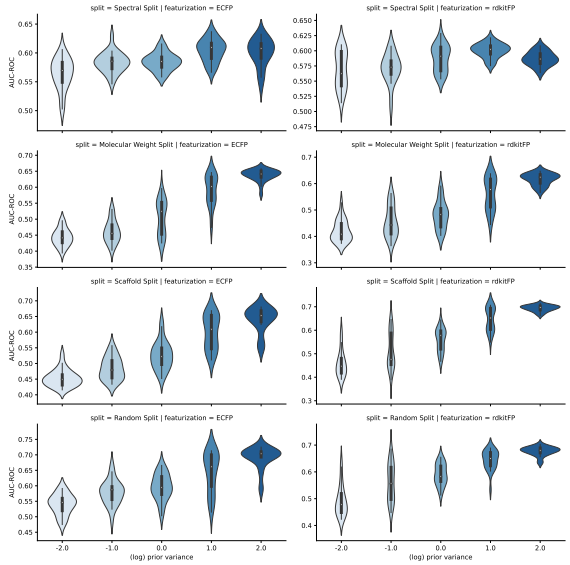
The models based on our probabilistic regularization scheme were trained using the implementation of the local linearization scheme presented in Rudner et al. (2022a;b) provided by the authors and using the exact same architecture, initialization, and optimization procedures as for the multi-layer perceptrons and deep ensembles—differing only in the objective function. Specifically, at each gradient step iteration, a sample of M molecules (where M is a hyperparameter) was drawn from a uniform distribution over the ZINC database (Irwin et al., 2020), providing a set of context points on which to evaluate the objective in Equation (9), using the Bernoulli likelihood to specify $\log p(\mathbf{y}_\mathcal{D} \mid f(\mathbf{x}_\mathcal{D}; \boldsymbol{\theta}))$. To construct a prior distribution over parametric function mappings $p_{f(\{\mathbf{X}, \mathbf{X}_\mathcal{C}\}; \boldsymbol{\Theta})}$ that maximizes predictive uncertainty away from the training data, it was defined as a distribution over functions with a logit-space mean vector of approximately zero and minimal structure in the off-diagonal entries of its covariance matrix. We refer to our code repository for further implementational details. The full hyperparameter search space is presented in Table 13.

Table 13. Hyperparameters for Our Model

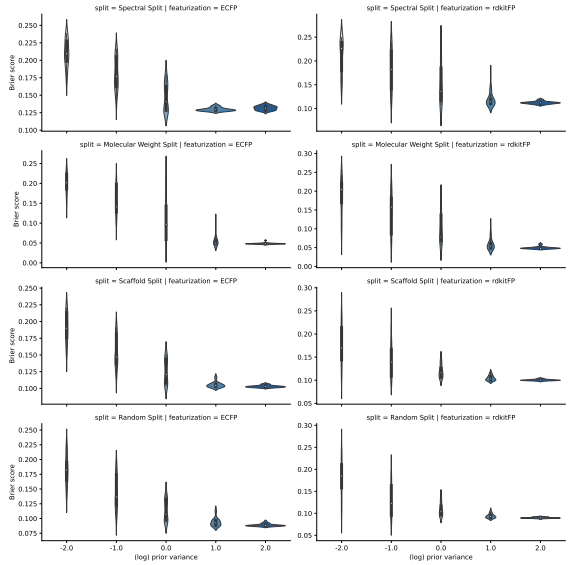
Model	**Hyperparameter**	**Search Space**
Q-SAVI	learning rate	1×10^{-4}, 1×10^{-3}
	number of layers	2, 4, 6
	embedding dimension	32, 64
	prior variance	1×10^{-2}, 1×10^{-1}, 1, 1×10^{1}, 1×10^{2}
	context points per sample	16, 128

B.3. Ablation Studies

To understand the impact of different hyperparameters on the performance of our proposed probabilistic regularization scheme, we conducted a series of ablation experiments. In these experiments, we systematically varied the hyperparameters relevant to evaluating the objective in Equation (9)—namely the prior variance and the number of sampled context points—while keeping others fixed, and measured their effects on the test set AUC-ROC and BRIER SCORE. The resulting ablation plots are presented in Figures 7 and 8 and show that larger prior covariances are strongly correlated with more robust test-set performances across splits—while the effect of larger context point samples is less pronounced.

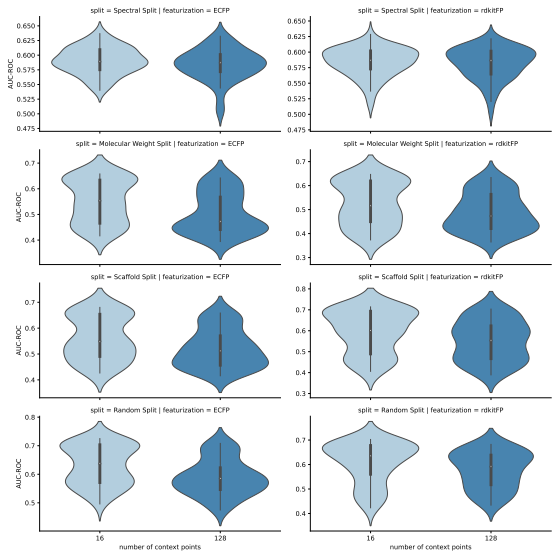


(a) Ablation plot showing the effect of the prior covariance on the test set AUC-ROC.

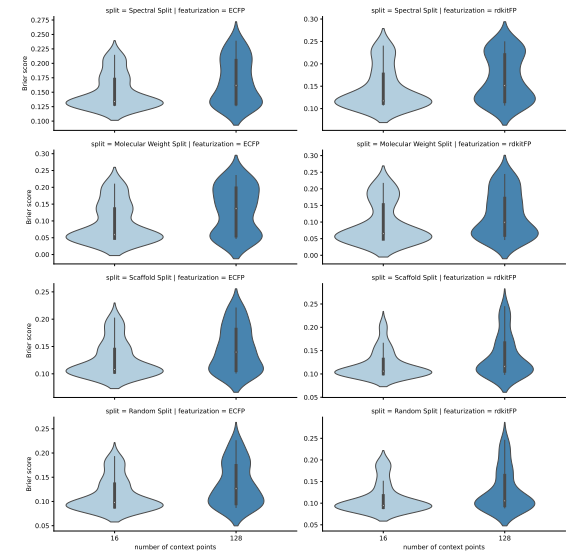


(b) Ablation plot showing the effect of the prior covariance on the test set BRIER SCORE.

Figure 7. Effect of (log) prior variance on the test set performance metrics.



(a) Ablation plot showing the effect of the number of context points on the test set AUC-ROC.



(b) Ablation plot showing the effect of the number of context points on the test set BRIER SCORE.

Figure 8. Effect of the number of context points on the test set performance metrics.

C. Additional Experimental Details for the Merck Molecular Activity Challenge Data

In order to assess the practical utility of our method in real-world production settings, an evaluation on the Merck Molecular Activity Challenge datasets (Ma et al., 2015) was conducted. This data consists of 15 datasets from real-world production environments with time-split training and test sets. As the compound structures are only provided in the form of anonymized atom-pair descriptors, it is not possible to use a uniform subsample of a large chemical database as a context point distribution. Instead, the evaluation focused on the three most covariate- and label-shifted datasets, see Figure 9, using a uniform distribution over molecules from the remaining datasets as a context point distribution. To select these datasets, the multiset version of the standard Jaccard/Tanimoto index

$$k_{\text{jac-multiset}}(\mathbf{x}, \mathbf{y}) = \frac{\sum_i \min(\mathbf{x}_i, \mathbf{y}_i)}{\sum_i \max(\mathbf{x}_i, \mathbf{y}_i)}$$

was used to evaluate the MMD statistic between two sets of count vectors and quantify covariate shift. Label shift between the regression targets of every training and test set was quantified through the two-sample Kolmogorov-Smirnov test statistic.

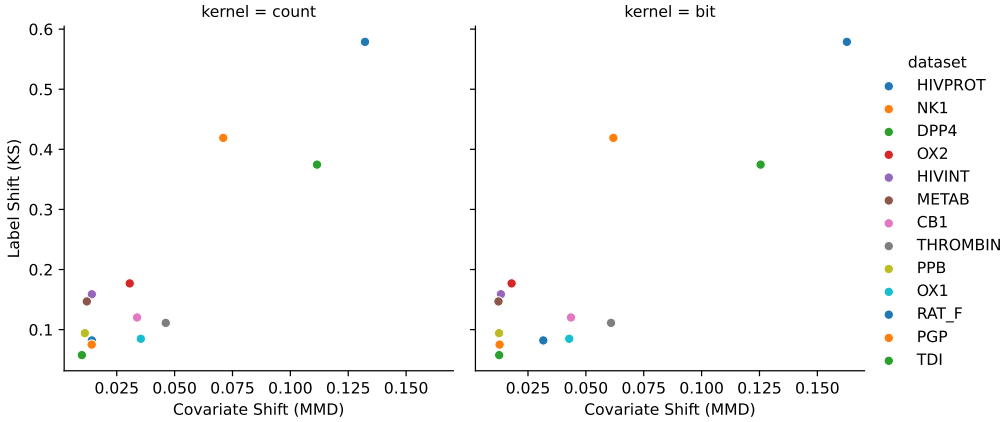


Figure 9. Scatterplot illustrating the covariate and label shifts in the Merck Molecular Activity Challenge datasets

As shown in Figure 10, the direct overlap between the HIVPROT, NK1, and DPP4 datasets with the remaining data is minimal, warranting its use as a general and diverse context point distribution. Using this evaluation setup, 10% of the training sets was randomly split off as a validation set for hyperparameter optimization and, where applicable, early stopping. Model-specific details are outlined below, including implementational details and hyperparameter ranges for regularized linear regressions (Appendix C.1), random forest regressors (Appendix C.2), and an adapted version of our probabilistic regularization scheme (Appendix C.3).

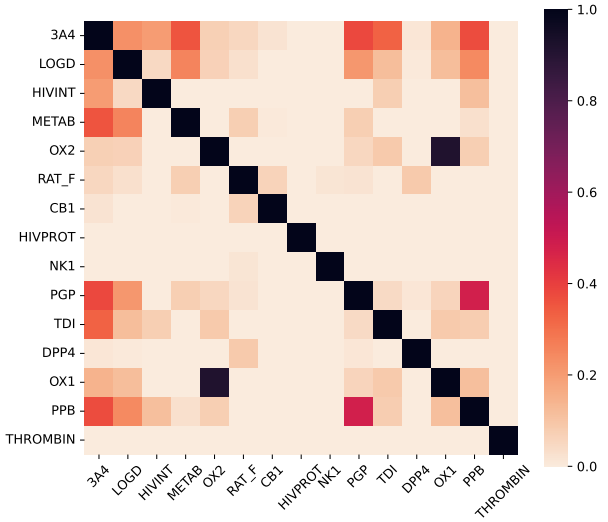


Figure 10. Heatmap illustrating the pairwise overlap between different datasets from the Merck Molecular Activity Challenge, defined as the proportion of molecules from the smaller dataset that are found in the larger dataset, i.e., $|\mathbf{X}_1 \cup \mathbf{X}_2|/\min(|\mathbf{X}_1|, |\mathbf{X}_2|)$

C.1. Regularized Linear Regression

The **regularized linear regression models** were trained using the scikit-learn library (Pedregosa et al., 2011). The LIBLINEAR solver (Fan et al., 2008) was employed with a maximum of 1,000 iterations and a stopping tolerance of 1×10^{-4}. The models were independently fitted for all specified hyperparameter combinations presented in Table 14. The combination yielding the lowest validation set mean squared error was selected to evaluate the model on the held-out test set.

Table 14. Hyperparameters for $L-1$ and L_2-Regularized Linear Regression

Model	Hyperparameter	Search Space
Linear Regression	regularization type	$\ell 1, \ell 2$
	regularization strength	100 values spaced log-linearly in $[1 \times 10^{-4}, 1 \times 10^{4}]$

C.2. Random Forest Regressors

The **random forest regression models** were trained using the scikit-learn library (Pedregosa et al., 2011). The models consisted of 100 decision trees with the GINI splitting criterion. Each model was independently fitted for all specified hyperparameter combinations shown in Table 15. The combination with the lowest validation set mean squared error was selected to evaluate the model on the held-out test set.

Table 15. Hyperparameters for Random Forest Regressor

Model	Hyperparameter	Search Space
Random Forest	maximum depth	50 values spaced linearly in [5, 500]
	min. samples per split	5, 15, 50, 100
	min. samples per leaf	1, 5, 10, 30, 100

C.3. Q-SAVI

The regression variant of our probabilistic regularization scheme was set up identically to the classification variant described in Appendix B.2.8, the only difference being the likelihood function used to evaluate Equation (9). Instead of specifying $\log p(\mathbf{y}_{\mathcal{D}} \mid f(\mathbf{x}_{\mathcal{D}}; \boldsymbol{\theta}))$ as a Bernoulli likelihood, a homoscedastic multivariate Normal likelihood with a unit diagonal covariance matrix was used. While a more expressive approach of either optimizing the covariance as a hyperparameter or letting the network predict point-wise means and variances to use in combination with a heteroscedastic likelihood function is possible, this straightforward method was found to be sufficient in this context. The full hyperparameter search space is presented in Table 16 and is identical to that in Table 13.

Table 16. Hyperparameters for Our Model

Model	Hyperparameter	Search Space
Ours	learning rate	1×10^{-4}, 1×10^{-3}
	number of layers	2, 4, 6
	embedding dimension	32, 64
	prior variance	1×10^{-2}, 1×10^{-1}, 1, 1×10^{1}, 1×10^{2}
	context points per sample	16, 128

.