Unifying Planning and Learning for Contact-Rich Manipulation

Jun Yamada

Mansfield College

University of Oxford

Supervisor: Prof. Ingmar Posner

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2025

Unifying Planning and Learning for Contact-Rich Manipulation
Candidate: Jun Yamada
Supervisors: Professor Ingmar Posner
Examiners: Professor Ioannis Havoutis, Professor Subramanian Ramamoorthy
Date of examination: 26th February, 2026

University of Oxford
Applied AI Lab (A2I)
Oxford Robotics Institute
Department of Engineering Science

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Ingmar Posner, for his invaluable guidance, continuous support, and encouragement throughout my doctoral research. His insights and mentorship have been instrumental in shaping both my academic journey and personal growth as a researcher. It has been a privilege to be part of the Applied AI Lab under his supervision.

I greatly appreciate the insightful feedback provided by my examiners, Professor Ioannis Havoutis and Professor Subramanian Ramamoorthy, which has significantly improved the final version of this thesis.

I am also deeply thankful to my collaborators and labmates – Jack Collins, Alexander L. Mitchell, Shaohong Zhong, Anson Lei, Branton Demoss, Frederik Nolte, Markus Baumgartner, Bradley Stanley-Clamp, Jan Schneider, Marc Rigter, Joe Watson, Oiwi Parker Jones, Yizhe Wu, Walter Goodwin, and Chia-Man Hung – for their stimulating discussions, generous help, and for making the lab an inspiring environment. In particular, I would like to extend special thanks to Jack Collins, whose insightful research advice and careful feedback on my writing significantly improved the quality of many of my publications. I am additionally grateful to Lara Brudermüller and Frank Fu for their insightful research discussions. I would also like to express my sincere appreciation to the engineering teams who supported my research during my DPhil, including Chris Prahacs, Tom Dobra, John Lo, Matt Towlson, Matthew Graham, and Tobit Flatscher. Without their technical support and dedication, I would not have been nearly as productive in my research.

I would also like to express my sincere gratitude to my former supervisor, Prof. Joseph Lim, as well as Younwoon Lee and Karl Pertsch, who actively supervised my research projects at the CLVR Lab. Without their mentorship and guidance, I might not have embarked on this DPhil journey in the first place.

I am further grateful for the opportunity to undertake an internship at NVIDIA, which provided valuable perspectives that enriched my research. I would especially like to thank Balakumar Sundaralingam, Adithyavairavan Murali, Ajay Mandlekar, and Yashraj Narang for their mentorship, collaboration, and support during this time.

I would also like to thank my family and friends for their unwavering encouragement, understanding, and belief in me throughout this journey. Their support has been a constant source of strength. Finally, I would like to extend my heartfelt gratitude to my partner, Yoshiko Sakuma, for her patience, encouragement, and steadfast support, which have been invaluable throughout this journey.

Abstract

Contact-rich manipulation tasks in semi-structured and unstructured environments pose significant challenges for robotic systems, such as in small-batch manufacturing and open-world scenarios where robots are required to generalise or rapidly adapt to novel objects, tasks, or scenes with minimal setup or reconfiguration. In such settings, manually engineered solutions are inherently unscalable, as they necessitate substantial task-specific design effort and are unable to accommodate the diversity and variability encountered in real-world conditions.

Model-based planning, such as motion planning and trajectory optimisation, is often effective when an accurate geometric and dynamics model is available, as it enables safe motion generation through predictive planning and is well-suited for explicit constraint handling. However, contact-rich manipulation introduces discontinuous and hybrid dynamics due to intermittent contact, frictional interactions, and mode switching. These interactions are inherently non-smooth and highly sensitive to small perturbations, making accurate modelling difficult in practice. Moreover, these approaches often struggle to find feasible solutions for complex tasks, particularly in contact-rich manipulation, where large search spaces and narrow feasible regions make optimisation difficult, while also being computationally expensive and reliant on carefully designed cost functions that are hard to specify for intricate contact interactions. As a result, many practical systems instead rely on robust feedback policies or reusable motion primitives, which can achieve reliable performance without requiring globally optimal solutions.

Recent advances in generative models offer a promising direction for addressing these challenges by learning a dynamics model from data or sampling distributions, thereby facilitating exploration and improving model-based planning. By serving as learned dynamics models or action sampling distributions, generative models can enhance motion planning and trajectory optimisation with data-driven priors, enabling more efficient planning and reactive decision-making in complex environments. Yet, learning accurate dynamics models for complex contact-rich manipulation tasks in the real world remains challenging, due to the need for a large amount of real-world data and the inherent difficulty of modelling intricate contact interactions.

In contrast, model-free learning approaches, such as model-free reinforcement learning (RL) and imitation learning (IL), offer complementary strengths: they can acquire complex manipulation skills directly from data, handle high-dimensional sensory inputs, such as images or point clouds, and potentially generalise to unseen objects without explicit modelling. However, they often suffer from sample

inefficiency and safety concerns, particularly in cluttered environments with obstacles. Moreover, IL typically requires a substantial number of high-quality expert demonstrations, which can be costly and time-consuming to collect.

This thesis develops unified frameworks that combine the complementary strengths of planning and learning for contact-rich manipulation. The initial chapters examine how model-based planning, such as motion planning and trajectory optimisation, can guide and decompose manipulation tasks, improving sample efficiency and safety while allowing learning-based methods to focus on contact-rich interaction in unstructured environments. For example, planning can navigate the robot to desired targets while avoiding obstacles or sample action candidates that satisfy physical and safety constraints. The focus of the investigation then moves towards improving model-based methods with learning-based approaches, particularly the use of generative models, while also emphasising improved data efficiency and task performance. The thesis subsequently discusses the interplay between these two themes, highlighting their complementary roles in unifying planning and learning for contact-rich manipulation.

Across extensive simulation and real-world experiments, these unified approaches demonstrate significant improvements in sample efficiency, safety, and generalisation compared to exclusively planning-based or learning-based methods. This work makes foundational contributions to scalable robot learning for real-world applications, particularly in domains such as small-batch manufacturing and open-world environments, where robots must rapidly acquire new skills or generalise to diverse objects and tasks.

Contents

1 Introduction **1**

1.1 Thesis Outline and Contributions 6

1.1.1 Efficient Skill Acquisition for Insertion Tasks in Obstructed Environments 7

1.1.2 Grasp-MPC: Closed-Loop Visual Grasping via Value-Guided Model Predictive Control 9

1.1.3 COMBO-Grasp: Learning Constraint-Based Manipulation for Bimanual Occluded Grasping 11

1.1.4 Leveraging Scene Embeddings for Gradient-Based Motion Planning in Latent Space 12

1.1.5 TWIST: Teacher-Student World Model Distillation for Efficient Sim-to-Real Transfer 14

1.1.6 D-Cubed: Latent Diffusion Trajectory Optimisation for Dexterous Deformable Manipulation 16

1.2 Publication List 17

2 Background and Related Work **20**

2.1 Model-based Planning 20

2.1.1 Motion Planning 21

2.1.2 Trajectory Optimisation for Receding-Horizon Control . . . 22

2.2 Robot Learning for Manipulation Tasks 27

2.2.1 Model-free Reinforcement Learning 28

2.2.2 Model-based Reinforcement Learning 32

2.2.3 Offline Reinforcement Learning 34

2.2.4 Imitation Learning 35

2.3 Generative Models in Robotics 37

2.3.1 Variational Autoencoders 37

2.3.2 Diffusion Models 39

2.3.3 Normalising Flows and Flow Matching 40

2.4 Unifying Planning and Learning 41

2.4.1 Learning-based Motion Planning 41

2.4.2 Sequential Integration of Motion Planning and Learning-based Approaches 42

2.4.3 Trajectory Optimisation with Learning-based Approaches . . 43

3 Efficient Skill Acquisition for Insertion Tasks in Obstructed Environments 44
3.1 Limitations and Future Work 65

4 Grasp-MPC: Closed-Loop Visual Grasping via Value-Guided Model Predictive Control 67
4.1 Limitations and Future Work 87

5 COMBO-Grasp: Learning Constraint-Based Manipulation for Bimanual Occluded Grasping 88
5.1 Limitations and Future Work 110

6 Leveraging Scene Embeddings for Gradient-Based Motion Planning in Latent Space 111
6.1 Limitations and Future Work 122

7 TWIST: Teacher-Student World Model Distillation for Efficient Sim-to-Real Transfer 123
7.1 Limitations and Future Work 134

8 D-Cubed: Latent Diffusion Trajectory Optimisation for Dexterous Deformable Manipulation 135
8.1 Limitations and Future Work 156

9 Discussion and Future Work 158
9.1 Discussion 158
9.1.1 Planning-Guided Efficient Skill Acquisition 158
9.1.2 Improving Model-Based Approaches with Generative Models 160
9.1.3 Connection Between Two Themes 162
9.2 Future Work 163

10 Conclusions 167
10.1 Key Contributions and Impact 168
10.2 Broader Implications 169
10.3 Final Remarks 169

References 171

1 Introduction

One of the central challenges in robotics is to enable robots to perform contact-rich manipulation tasks reliably in complex, uncertain, and unstructured environments. Traditional approaches have relied heavily on manual programming and carefully engineered systems [1–3], which perform well in structured industrial settings but struggle in dynamic or unstructured environments. As robotics expands beyond large-scale, tightly controlled factories into unstructured and variable environments, including home care, supermarkets, and small-batch manufacturing, the need for adaptable decision-making frameworks becomes increasingly demanding.

In these domains, robots must generalise and adapt to novel objects, tasks, and environments without extensive reprogramming. Small-batch manufacturing is one illustrative example: companies require robots that can quickly learn new tasks and maintain efficiency without the rigid infrastructure of mass production. Similar challenges arise in service robotics and logistics applications, where environments are semi-structured at best. In these domains, variability in object properties, cluttered environments, changing task goals, and partial observability make manual programming approaches prohibitively time-consuming. Each new configuration often requires extensive re-tuning or redesign of task-specific motion, which is impractical at scale. These factors make the reliable execution of complex, contact-rich manipulation tasks in such environments a fundamental challenge in robotics.

Improvements in hardware, such as soft robotic end-effectors [4, 5] and tactile sensing [6], can enhance compliance and improve contact state estimation in contact-rich manipulation. These advances improve robustness to uncertainty and provide richer feedback during interaction. However, they do not eliminate the intrinsic hybrid and frictional dynamics that make contact-rich manipulation fundamentally challenging. Furthermore, sensing modalities, including vision and force feedback, determine the extent to which the interaction state can be observed, yet in many vision-based manipulation systems the physically relevant aspects of interaction remain only partially observable. In particular, during contact, information about alignment precision, frictional regime, and contact mode is difficult to observe directly from visual input alone and may remain ambiguous even when force feedback is available. As a result, control decisions must be made under substantial uncertainty regarding the true interaction state.

These challenges have motivated extensive research into alternatives that reduce reliance on manual programming. One major line of work focuses on model-based planning, such as motion planning [7–10] and trajectory optimisation [11, 12], for their ability to simulate sequences of actions or states, anticipate future outcomes, and heuristically explore the state space. These methods offer clear advantages when an accurate geometric or dynamics model is available: they enable predictive foresight, enforce safety constraints, yield interpretable behaviours, and facilitate collision-free motion of the robot toward desired targets. However, such model-based approaches often fall short in contact-rich tasks involving uncertain dynamics, task variability, and partial observability. The high computational burden, coupled with the difficulty of finding feasible solutions in large search spaces, also limits their applicability. Moreover, in manipulation scenarios involving friction, deformation, and intermittent contacts, accurately engineering a dynamics model is often intractable, significantly limiting the capability of model-based approaches.

To address these limitations, data-driven model-based approaches, including model-based reinforcement learning (RL) [13–15], have emerged as a compelling alternative. These approaches learn a dynamics model from data and use it

to simulate future outcomes for planning or policy optimisation. To learn such dynamics models from data, generative models [16–18] have been widely employed. These models are capable of capturing high-dimensional data distributions and have proven effective at modelling the uncertainty and multimodality inherent in contact-rich manipulation. Within model-based approaches, generative models primarily serve two key roles. First, they serve as dynamics models, capable of representing complex dynamics that cannot be hand-crafted and can therefore be integrated with planning and policy optimisation. Moreover, by modelling the robot's kinematics within a structured latent space, these models support gradient-based motion planning through optimisation in a compact and differentiable representation [19]. Second, they serve as action samplers, enabling efficient exploration of complex tasks by generating high-performing candidate trajectories. Collectively, these roles position generative models as a central component in model-based decision-making for manipulation tasks. Nevertheless, learning accurate dynamics models for complex contact-rich manipulation tasks in the real world remains challenging, as it still requires a large amount of data to capture the variability and intricacies of complex contact interactions.

In contrast to model-based approaches, model-free RL [20–22] acquires control policies directly through trial-and-error interaction with the environment, without requiring explicit modelling of dynamics. This makes it appealing for contact-rich manipulation tasks, where learning accurate dynamics models from data is inherently challenging. Driven by rewards, model-free RL enables the policy to solve various contact-rich manipulation tasks. However, it is typically sample-inefficient, particularly in tasks with long horizons or sparse reward signals. Moreover, exploration in semi-structured and unstructured environments, especially those with obstacles and safety-critical constraints, is challenging, as it requires searching through narrow feasible regions and entails potential safety risks. These limitations significantly hinder the scalability and practical deployment of model-free approaches in real-world settings.

To mitigate the sample inefficiency and safety risks associated with RL, another approach is imitation learning (IL) [23–25], which learns a policy from expert demonstrations rather than relying on trial-and-error interaction. This enables policies to be trained safely and efficiently, particularly in tasks where autonomous exploration is unsafe and inefficient. However, IL typically requires a large number of high-quality demonstrations [26]. As the diversity of tasks, objects, or environments increases, collecting a sufficient amount of data becomes increasingly time-consuming and labour-intensive. Recent advances in Vision Language Action (VLA) models [27–29] have demonstrated the potential to generalise across diverse objects, tasks, and scenes by leveraging large-scale robot datasets. Although these models show promising performance in relatively simple settings, such as clean tabletop environments, their ability to execute behaviours safely and robustly remains limited in semi-structured or unstructured environments.

Given these limitations, and to leverage the complementary strengths of planning and learning, there is growing interest in unified approaches that integrate model-based planning with data-driven methods. Planning provides powerful tools for predictive foresight, constraint handling, safety assurance, and navigation to the target, while learning-based approaches enable skill acquisition for contact-rich manipulation, dynamics modelling, and the learning of action distributions that offer data-driven priors, thereby facilitating exploration. To this end, this thesis develops unified frameworks that combine planning and learning from readily available data to improve the efficiency, flexibility, safety, and robustness of skill acquisition and execution in contact-rich manipulation tasks.

In this thesis, contact-rich manipulation refers to low-level manipulation skills in which task success critically depends on complex and discontinuous contact interactions and narrow feasible regions, such that purely geometric motion planning is insufficient and successful execution requires reasoning about contact dynamics beyond kinematic feasibility. While Chapter 4 and Chapter 6 address preparatory sub-problems, including grasping and motion planning, which constitute crucial components enabling subsequent contact-rich interaction, the central emphasis lies

in the contact-rich manipulation phase itself, where hybrid interaction dynamics and narrow feasible regions necessitate adaptive control.

Within this scope, the thesis explores *Theme A: Planning-guided efficient skill acquisition* and *Theme B: Improving model-based approaches with generative models*. Collectively, these two themes are closely connected, as improvements in model-based planning developed in *Theme B* can be applied within the frameworks of *Theme A* to provide more adaptive, efficient, and effective guidance for skill acquisition.

Theme A: Planning-Guided Efficient Skill Acquisition. To address the limitations of solely planning-based or learning-based methods, this first theme focuses on unifying model-based planning, such as motion planning and trajectory optimisation, with data-driven approaches to accelerate skill acquisition for contact-rich manipulation. The core idea is to guide and decompose the exploration and execution processes by leveraging the predictive foresight, constraint-handling, motion-generation, and safe-navigation capabilities of planning, allowing the learnt policy to focus on the most challenging aspects of manipulation. Importantly, not all motion needs to be learnt: planning can generate trajectories for obstacle avoidance and target reaching, while learning-based approaches handle the contact-rich interaction phases. For example, motion planning can first ensure obstacle avoidance, after which a learning-based policy addresses the contact-rich manipulation tasks in cluttered environments. Moreover, model predictive control (MPC) generates candidate action sequences that satisfy physical and safety constraints, while the learning-based component, which captures the intricacies of successful manipulation and enables generalisation to novel objects, evaluates these candidates based on visual observations. Through this integration, planning provides structural guidance and safety, while learning extends adaptability and contact-rich control, resulting in safe, robust, and efficient skill acquisition for complex manipulation tasks in unstructured settings, even in the presence of obstacles.

Theme B: Improving Model-Based Approaches with Generative Models. While *Theme A* demonstrates how planning can guide learning for efficient skill acquisition and execution, *Theme B* explores the complementary direction:

improving model-based decision making through generative modelling. Model-based approaches are often effective when accurate geometric or dynamics models are known and the manipulation tasks involve relatively simple state spaces that are easier to explore in search of a solution. However, hand-crafting such models is often infeasible, and finding solutions in complex, contact-rich domains is further hindered by high-dimensional state spaces and discontinuous dynamics. To address these challenges, this thesis focuses on improving model-based decision-making and task performance through the use of generative models, while enhancing data efficiency by leveraging readily available datasets, such as simulated or play data. The generative models can serve as learnt dynamics models for simulating and optimising policies or planning, and as a learnt action sampler for planning, enabling effective motion generation in contact-rich scenarios. By learning forward dynamics from sensory data, these models capture the multimodal and discontinuous behaviours characteristic of contact interactions. They can also be trained to acquire informed sampling distributions for trajectory optimisation, guiding search toward promising regions of the state–action space and improving both efficiency and performance. These improvements feed back into *Theme A* by providing data-driven dynamics models in place of hand-crafted ones, along with more informative sampling strategies and faster planning capabilities, ultimately improving both skill acquisition and execution in challenging contact-rich tasks.

1.1 Thesis Outline and Contributions

The following chapter presents background material and related work that establish the context of this thesis. Since this is an integrated thesis, each subsequent chapter corresponds to an individual paper. Each of these chapters begins with an overview of the paper and its key contributions, followed by the main content. After presenting each paper, we discuss the limitations of the work and propose future research directions to address these limitations. Note that citations in the main thesis body refer to the bibliography at the end of the thesis, whereas citations within individual papers refer to the bibliography at the end of each paper.

As described in Chapter 1, this thesis explores two key themes. Chapters 3, 4, and 5 demonstrate how to effectively leverage planning to enable efficient skill acquisition and execution in complex and unstructured scenarios. Chapters 6, 7, and 8 explore model-based approaches with generative models, including gradient-based motion planning in a structured latent space, learning forward dynamics models for manipulation tasks, and leveraging a learnt generative action sampler to improve trajectory optimisation. Chapter 9 discusses how these two themes are closely connected, highlighting specific cases where methods developed in one theme enhance or extend those in the other. The following section provides a summary of the contributions of each paper that constitutes this thesis.

1.1.1 Efficient Skill Acquisition for Insertion Tasks in Obstructed Environments

In real-world applications, particularly in small-batch settings, robots must acquire contact-rich manipulation skills for previously unseen objects while maintaining data efficiency and minimising human intervention, even in obstructed environments. To operate effectively in such environments, they require robust obstacle avoidance, flexible goal conditioning, and contact-rich manipulation skills. However, existing approaches to robot control, such as motion planning and RL, struggle to meet these requirements.

Motion planning [7–10, 30] offers effective solutions for obstacle avoidance; however, it is not inherently designed to handle contact-rich manipulation tasks that require fine-grained interaction with objects. On the other hand, RL [21, 22, 31] excels at acquiring such low-level manipulation skills but often suffers from sample inefficiency and unsafe exploration, particularly in cluttered environments.

Several previous works [32–34] propose unifying motion planning and RL to enable robots to solve contact-rich manipulation tasks in obstructed environments. A key insight of our approach, similar in spirit, is that not all motions need to be learnt; for improved sample efficiency, motion planning can be used to handle long-horizon obstacle avoidance and navigation, while RL should be reserved for the most challenging aspects of the contact-rich manipulation tasks. Prior approaches

have primarily focused on relatively simple peg-insertion tasks and often rely on either object-specific classifiers [34] or goal estimators learnt using RL [32], which limits flexible goal conditioning and adaptation to new objects.

To address these challenges and enable efficient skill acquisition in small-batch settings, Chapter 3 introduces a system that combines motion planning and an RL policy, leveraging an object-centric generative model [35] trained on readily available simulation data for versatile one-shot target object identification. The target object identified by utilising the object-centric generative model serves as the goal for motion planning, guiding the robot toward the object while avoiding obstacles, after which an RL policy is executed to perform the contact-rich insertion task. The RL policy is trained locally around the target object using wrist-mounted RGB images and force torque measurements, and requires only a small number of demonstrations to structure the learning process. The overarching goal of this work is to design a robotic system that requires minimal human intervention while remaining data efficient. Although residual RL [36] can potentially achieve similar performance, it requires specifying and tuning a base policy, thereby introducing additional task-specific engineering. Instead, we leverage a small number of demonstrations to bias the policy search toward meaningful behaviours. This preserves a scalable framework while maintaining strong sample efficiency.

This approach achieves robust insertion performance at any connector position within the workspace after just one hour of training using sparse rewards. Moreover, to maximise performance, we introduce a skill transition network that is trained on a dataset collected in a self-supervised manner, thereby eliminating the need for human intervention. This model enables seamless connection between motion planning and RL by moving the robot from the terminal state of the planned trajectory, which may be out of distribution for the RL policy, to the initiation set of the policy. Combined with an object-centric generative model, Chapter 3 demonstrates that integrating motion planning with RL enables safe execution for contact-rich insertion in obstructed environments, while also supporting versatile and data-efficient skill acquisition across multiple target objects.

The contributions of Chapter 3 are the following:

- Proposing a system that combines motion planning with RL for efficient skill acquisition in obstructed environments, leveraging an object-centric generative model for versatile one-shot object identification.
- Introducing a transition network that smoothly connects the terminal states of motion planning to the initiation set of states for the learnt RL policy, significantly improving the success rate.
- Demonstrating that one-shot object identification using the object-centric generative model achieves comparable accuracy to heuristic and object-specific goal identification baselines.
- Showing that the proposed system achieves significantly better performance in real-world environments compared with baselines, including state-of-the-art RL algorithms.

1.1.2 Grasp-MPC: Closed-Loop Visual Grasping via Value-Guided Model Predictive Control

Grasping is a fundamental manipulation skill that underpins a wide range of downstream tasks, such as insertion and assembly. State-of-the-art approaches are broadly divided into open-loop and closed-loop methods. Open-loop methods [37–40] predict 6-DoF grasp poses using models trained on large synthetic datasets [41] and execute motion planning to reach the target. While effective for novel objects in cluttered scenes, they are prone to prediction errors and fail to handle moving targets due to the absence of feedback. Closed-loop methods based on RL [42–44] or IL [45, 46] enable reactive control with continuous feedback but typically operate in clean tabletop environments and generalise poorly to novel objects.

To address these issues, Chapter 4 introduces *Grasp-MPC*, a framework that integrates motion planning and model predictive control (MPC) with a data-driven value function to enable safe and generalisable closed-loop visual grasping of novel objects in cluttered and dynamic environments. Building on the approach

presented in Chapter 3, which combines motion planning with RL for manipulation tasks, *Grasp-MPC* uses motion planning to guide the robot arm to a pre-grasp pose predicted by an off-the-shelf grasp pose model [38] while avoiding obstacles. Instead of an RL policy, *Grasp-MPC* employs MPC that incorporates a learnt value function as the cost term for closed-loop visual grasping.

The value function is trained on a large-scale synthetic offline dataset of grasp trajectories over diverse objects, generated by a motion planner, which includes both feasible and infeasible grasp attempts, and estimates the likelihood of grasp success for a given observation. By integrating this cost into MPC along with constraints such as collision avoidance and motion smoothness, *Grasp-MPC* enables safe and generalisable closed-loop grasping. By unifying model-based planning, such as motion planning and MPC, with the value function learnt from data, Chapter 4 demonstrates the acquisition of generalisable, safe, robust, and adaptive manipulation skills capable of operating in cluttered environments, including multiple objects on a table or shelf, as well as moving objects.

Overall, Chapter 4 presents the following contributions:

- Introducing Grasp-MPC, a safe closed-loop 6-DoF grasping policy that generalises effectively to novel objects in cluttered environments.
- Integrating a vision-based grasp value function, trained on large-scale synthetic grasping trajectories, as a cost function within MPC for generalisable grasping.
- Generating a large-scale synthetic grasp trajectory dataset.
- Demonstrating generalisation to novel objects in cluttered scenes and achieving substantial improvements over open- and closed-loop baselines on Fetch-Bench [46] and in real-world environments.

1.1.3 COMBO-Grasp: Learning Constraint-Based Manipulation for Bimanual Occluded Grasping

Whilst Chapter 3 and Chapter 4 focus on efficient skill acquisition using a single arm by unifying planning and data-driven approaches, Chapter 5 introduces *COMBO-Grasp*, a framework that extends these principles to dual-arm coordination for occluded grasping scenarios. Occluded grasping refers to situations in which the desired grasp poses are kinematically infeasible due to environmental constraints, such as collisions with the table surface. Training an RL policy to control both arms for bimanual occluded grasping tasks is highly sample-inefficient and often struggles to find feasible solutions due to the increased dimensionality and complexity of coordination. Similarly, collecting expert demonstrations for such challenging non-prehensile manipulation tasks is time-consuming and costly.

To overcome these, *COMBO-Grasp* draws inspiration from human bimanual manipulation strategies [47–49] to coordinate two arms to stabilise and reorient the object, thereby resolving such occlusions. *COMBO-Grasp* consists of two coordinated policies: a constraint and a grasping policy. The constraint policy, trained on synthetic data collected in a self-supervised manner, generates a constraint pose to stabilise the target object, with the predicted pose subsequently used as the target for motion planning. On the other hand, the grasping policy, trained using RL in a simulated environment, reorients and grasps the target object by utilising the constraint arm. During RL training, the constraint policy generates a pose for the constraint arm at the beginning of each episode, which is then held fixed for the duration of the episode, while the grasping policy learns to control the other arm to solve the occluded grasping task by leveraging this constraint. To improve coordination, *COMBO-Grasp* introduces value-guided policy coordination, which guides the constraint policy to generate the stabilisation pose to maximise the performance of the grasping policy, inspired by classifier guidance [50] for a diffusion model [51]. This coordinated design, which integrates motion planning with an RL policy, simplifies the task, narrows the RL exploration space, accelerates training, and facilitates effective sim-to-real transfer. Both policies are trained with

privileged information for diverse objects in simulation with domain randomisation and are distilled into vision-based policies for deployment in real-world environments. By decoupling the complex bimanual occluded grasping task into two coordinated policies and leveraging motion planning to control one arm for object stabilisation and support, Chapter 5 demonstrates that this approach accelerates sample-efficient skill acquisition and achieves robust skill execution in real-world environments.

In summary, *COMBO-Grasp* makes the following key contributions.

- Presenting *COMBO-Grasp*, a novel bimanual manipulation approach in which the constraint policy predicts a target pose for motion planning that stabilises the object, while the grasping policy controls the other arm to solve occluded grasping problems by utilising the constraint.
- Leveraging force closure as a supervisory signal to collect constraint poses in a self-supervised manner, and training a constraint policy that accelerates RL training of the grasping policy.
- Introducing value-guided policy coordination that refines generated constraint poses using gradients from the value function associated with the grasping policy, thereby improving bimanual coordination.
- Conducting extensive evaluations of *COMBO-Grasp* in both simulated and real-world environments, demonstrating successful grasps on novel objects.

1.1.4 Leveraging Scene Embeddings for Gradient-Based Motion Planning in Latent Space

While Chapters 3, 4, and 5 focus on planning-guided skill acquisition by unifying planning with learning-based approaches, Chapter 6 shifts focus towards improving model-based planning, particularly motion planning, by leveraging generative models trained on readily available data. Specifically, Chapter 6 introduces *AMP-LS*, a gradient-based motion planning method that operates effectively in a learnt structured latent space to improve both planning speed and adaptability in complex environments.

Motion planners such as RRT [7, 8, 30] and PRM [9, 10] are widely used due to their reliability and well-understood properties. However, they face significant challenges in complex environments with obstacles and in dynamic settings, where planning becomes prohibitively slow and less adaptable. Recent learning-based approaches [52–54] have shown promise in improving planning speed, but they typically rely on large-scale expert demonstrations collected using geometric expert planners, which limits their flexibility and generalisability due to the costly data collection process.

Latent Space Path Planning (LSPP) [19] addresses some of these limitations by learning a structured latent space from kinematically valid static robot states, readily collected in simulation. Specifically, a variational autoencoder (VAE) [16, 17] is trained to reconstruct joint states and end-effector poses, and optimises trajectories in the latent space by minimising both the distance between the current and desired end-effector poses and the collision probability estimated by a learnt collision classifier. However, the collision classifier in LSPP relies on low-dimensional object states and supports only primitive-shaped obstacles, which restricts its applicability in complex real-world environments.

To address this limitation, AMP-LS substantially extends LSPP by introducing a collision predictor trained on point clouds from diverse, synthetically generated cluttered scenes in simulation, and by incorporating explicit collision checking during trajectory generation. Similar to LSPP, *AMP-LS* performs gradient-based motion planning by jointly minimising the predicted collision probability and the distance between the current and desired end-effector poses, backpropagating gradients through both the collision predictor and the VAE decoder. Moreover, *AMP-LS* performs explicit collision checking by interpolating between the current and the next desired states. These allow AMP-LS to generate safe and reactive collision-free motions in novel environments with diverse and complex object geometries, including dynamic obstacles. Chapter 6 demonstrates how generative models trained on readily available data can improve motion planning by learning a structured

latent space for gradient-based optimisation with a vision-based collision predictor, enhancing efficiency and adaptability in complex environments.

In summary, *AMP-LS* makes the following four contributions.

- Extending LSPP significantly by incorporating a vision-based collision predictor trained on diverse synthetic cluttered scenes, along with an explicit collision checking mechanism, into gradient-based motion planning within a learnt latent space, enabling real-time obstacle avoidance in complex environments.
- Generating collision-free trajectories with success rates comparable to traditional planning baselines, and substantially reducing planning time in simulation.
- Demonstrating zero-shot transfer to unseen real-world scenes.
- Enabling closed-loop reactive planning, allowing the system to reach moving targets while avoiding dynamic obstacles.

1.1.5 TWIST: Teacher-Student World Model Distillation for Efficient Sim-to-Real Transfer

Chapter 7 introduces *TWIST*, a sim-to-real transfer framework for model-based RL that transfers both a policy and a world model, which is trained using a generative model, from simulation to real-world environments. In contrast to Chapter 6, which learns a structured latent space from kinematically feasible static joint configurations, a form of dynamics model suitable only for kinematic planning, Chapter 7 focuses on learning a world model [55] from robot trajectories, including actions, using a generative model.

Unlike most prior work [13, 14], which primarily focuses on evaluation in simulation [13–15] or assumes access to large-scale offline datasets [56] for training, *TWIST* targets practical scenarios where such data are not accessible and instead leverages simulated data that can be readily obtained. To achieve sim-to-real transfer, domain randomisation [57] is commonly used to randomise physical and visual properties during simulation training, ensuring that the real-world domain lies

within the training distribution. However, naïvely applying domain randomisation to train a world model has been shown to be highly sample-inefficient [58].

To overcome this, *TWIST* employs a teacher–student distillation framework [59], where a world model is first trained using low-dimensional state observations in simulation under domain randomisation, in parallel with policy learning on the world model. This stage builds upon Dreamer [13, 60], which similarly trains a policy using compact latent representation in the world model. Subsequently, the state-based world model is distilled into a vision-based world model conditioned on domain-randomised visual observations, facilitating effective sim-to-real transfer. To perform the teacher-student distillation, *TWIST* exploits the generative modelling capabilities to generate imagined trajectories from both the teacher and student models. Given an initial state, the teacher world model generates a synthetic trajectory by rolling out the policy. The identical action sequence is then executed from the same starting state in the student world model, generating a corresponding trajectory in its compact latent representation space. The student model is then optimised to match its latent representations to those of the teacher, thereby aligning the predictive behaviours of the two models in latent space. The resulting policy and student world model are successfully transferred to real-world environments for contact-rich manipulation tasks, outperforming competitive baselines by a significant margin. Chapter 7 exemplifies how generative models can be leveraged not only for learning a forward dynamics model in simulation but also for enabling efficient sim-to-real transfer through teacher–student distillation, thereby enhancing their effectiveness for real-world applications. This approach significantly improves data efficiency, as it eliminates the need for large amounts of real-world interaction data.

In summary, Chapter 7 presents the following key contributions.

- Proposing *TWIST*, a framework for sim-to-real transfer in model-based RL.
- Introducing distillation in imagination that aligns the student's latent representations with those of the teacher using imagined rollouts generated by the models for effective sim-to-real transfer.

- Enabling robust and sample-efficient sim-to-real transfer of model-based RL, substantially outperforming model-based RL trained with naïve domain randomisation and a model-free approach designed for efficient sim-to-real transfer in real-world robotic manipulation tasks.

1.1.6 D-Cubed: Latent Diffusion Trajectory Optimisation for Dexterous Deformable Manipulation

Chapter 8 presents *D-Cubed*, a novel learning-based trajectory optimisation framework that addresses the challenges of trajectory optimisation for dexterous deformable object manipulation. Unlike Chapter 6 and Chapter 7, which focus on learning dynamics models using generative models, D-Cubed assumes access to a dynamics model and instead focuses on improving planning by learning an informed action distribution through diffusion-based generative modelling [18], trained on a readily available task-agnostic play dataset.

Manipulating deformable objects using a dexterous robot hand is inherently complex due to their infinite dimensionality, the numerous contacts involved, and the high degrees of freedom in dexterous robot hands. These factors make both model-based trajectory optimisation and RL challenging: trajectory optimisation struggles with hard exploration despite being effective for novel goals, such as a desired deformable object shape, while RL additionally suffers from poor generalisation and significant sample inefficiency. D-Cubed addresses the key limitation of such model-based trajectory optimisation by learning an action sampler using a latent diffusion model (LDM) [61], trained on a task-agnostic play dataset readily collected with a human hand. Since this dataset does not involve any interaction with objects, it can be collected efficiently and is far easier to obtain than collecting expert demonstrations for each individual task. Moreover, *D-Cubed* introduces gradient-free guided sampling applied to the LDM, a novel trajectory optimisation method that more effectively utilises the action sequences generated by the diffusion model.

D-Cubed first trains a skill-latent space using VAE [17] that encodes short-horizon action sequences from the play dataset and reconstructs them. Subsequently, an LDM is trained in this learnt skill-latent space to compose long-horizon skill

trajectories, capturing diverse and meaningful hand motions, and serving as an action sampler that generates long-horizon action sequences for trajectory optimisation. To generate a high-performing trajectory for a target task, *D-Cubed* introduces a novel gradient-free guided sampling approach that employs the Cross-Entropy Method (CEM) [11] within the reverse diffusion process. This allows D-Cubed to efficiently search the high-dimensional state space and find a high-performing action sequence that manipulates the deformable object to match the desired shape throughout the reverse diffusion process. The resulting framework enables generalisable goal-directed planning for deformable object manipulation without requiring task-specific demonstrations. Chapter 8 highlights the utility of generative models as an action sampler that can guide model-based planning frameworks. This illustrates a broader potential for integrating generative modelling into model-based planning to enhance exploration and optimisation in complex contact-rich manipulation tasks.

In summary, Chapter 8 presents the following contributions.

- Introducing *D-Cubed*, a trajectory optimisation framework based on latent diffusion models designed to solve dexterous deformable object manipulation tasks.
- Proposing a novel gradient-free guided sampling method that integrates CEM into the reverse diffusion process for effective trajectory optimisation.
- Demonstrating that *D-Cubed* significantly outperforms competitive baselines, including state-of-the-art RL methods, as well as gradient-based and sampling-based trajectory optimisation approaches.

1.2 Publication List

The publications [62–67] that constitute this integrated thesis are presented below.

- Chapter 3: **Jun Yamada**, Jack Collins, Ingmar Posner. Efficient Skill Acquisition for Insertion Tasks in Obstructed Environments, 6th Annual Learning for Dynamics & Control Conference (L4DC), 2024.

- Chapter 4: **Jun Yamada**, Adithyavairavan Murali, Ajay Mandlekar, Clemens Eppner, Ingmar Posner, Balakumar Sundaralingam. Grasp-MPC: Closed-Loop Visual Grasping via Value-Guided Model Predictive Control, IEEE International Conference on Robotics and Automation (ICRA), 2026.

- Chapter 5: **Jun Yamada**, Alexander L Mitchell, Jack Collins, Ingmar Posner. COMBO-Grasp: Learning Constraint-Based Manipulation for Bimanual Occluded Grasping, Conference on Robot Learning (CoRL), 2025.

- Chapter 6: **Jun Yamada***, Chia-Man Hung*, Jack Collins, Ioannis Havoutis, Ingmar Posner. Leveraging Scene Embeddings for Gradient-Based Motion Planning in Latent Space, IEEE International Conference on Robotics and Automation (ICRA), 2023.

- Chapter 7: **Jun Yamada**, Marc Rigter, Jack Collins, Ingmar Posner. TWIST: Teacher-Student World Model Distillation for Efficient Sim-to-Real Transfer, IEEE International Conference on Robotics and Automation (ICRA), 2024.

- Chapter 8: **Jun Yamada**, Shaohong Zhong, Jack Collins, Ingmar Posner: D-Cubed: Latent Diffusion Trajectory Optimisation for Dexterous Deformable Manipulation, Conference on Robot Learning (CoRL), 2025.

Additional publications to which I have contributed during the DPhil, but which fall outside the scope of this thesis, are listed below [27, 37, 68–71].

- Marc Rigter, **Jun Yamada**, Ingmar Posner. World Models via Policy-Guided Trajectory Diffusion, Transactions on Machine Learning Research (TMLR).

- Oiwi Parker Jones, Alexander L Mitchell, **Jun Yamada**, Wolfgang Merkt, Mathieu Geisert, Ioannis Havoutis, Ingmar Posner. Oscillating latent dynamics in robot systems during walking and reaching, Scientific Reports.

- Jack Collins*, Mark Robson*, **Jun Yamada***, Mohan Sridharan, Karol Janik, Ingmar Posner. RAMP: A Benchmark for Evaluating Robotic Assembly

Manipulation and Planning, IEEE Robotics and Automation Letters (RA-L), 2023.

- Adithyavairavan Murali, Balakumar Sundaralingam, Yu-Wei Chao, Wentao Yuan, **Jun Yamada**, Mark Carlson, Fabio Ramos, Stan Birchfield, Dieter Fox, Clemens Eppner, GraspGen: A Diffusion-based Framework for 6-DOF Grasping with On-Generator Training, IEEE International Conference on Robotics and Automation (ICRA), 2026.

- Kento Kawaharazuka, Jihoon Oh, **Jun Yamada**, Ingmar Posner, Yuke Zhu, Vision-Language-Action Models for Robotics: A Review Towards Real-World Applications, IEEE Access, 2025.

* denotes equal contribution.

2
Background and Related Work

This chapter presents background material and a review of related work relevant to the central themes of this thesis. Section 2.1 discusses model-based planning, such as motion planning and trajectory optimisation, which provide the basis for predictive decision-making and enable explicit constraint handling to ensure safety, thereby motivating their integration with learning-based methods. Section 2.2 reviews learning-based approaches for robotic manipulation, particularly reinforcement learning (RL), and imitation learning (IL), which provide the foundation for data-driven skill acquisition in later chapters. Section 2.3 introduces generative models, whose ability to capture complex dynamics and structured representations underpins *Theme B*, which aims to improve model-based decision making using generative models. Finally, Section 2.4 surveys prior work that unifies planning and learning, highlighting approaches most directly related to this thesis's contributions.

2.1 Model-based Planning

Model-based planning utilises a model of system dynamics or kinematics to generate trajectories or control sequences that optimise a specified objective under physical and task constraints. Such approaches include motion planning and trajectory optimisation, which enable predictive decision-making and constraint handling, critical for ensuring safety and robustness in robot manipulation. This section provides key methods in model-based planning, beginning with motion planning,

which focuses on computing feasible paths in configuration space, followed by trajectory optimisation for receding-horizon control that enables online adaptation.

2.1.1 Motion Planning

Motion planning [8, 9, 30, 72–74], which plans a collision-free trajectory from the current robot joint state to a goal joint or end-effector pose, is a fundamental technique for controlling robots. Motion planning is crucial in applications involving manipulation tasks in obstructed environments; for example, when a robot assembles a table, it must reach a table leg to grasp it while avoiding the other table legs. Motion planning is broadly categorised into sampling-based and optimisation-based methods.

Sampling-based motion planning. Sampling-based motion planners such as RRT [7, 8, 30] incrementally build trees by extending toward randomly sampled configurations. In contrast, PRM-based approaches [9, 10] first sample configurations in the free space and then construct a roadmap by connecting each configuration to nearby neighbours within a specified radius, provided the connections are collision-free. Subsequent work seeks to improve convergence efficiency and solution quality. BIT [74] integrates heuristic-guided search with batch sampling to concentrate computation on promising regions of the state space. AIT [75] further extends this idea by estimating problem-specific heuristics adaptively through an asymmetric bidirectional search, achieving both rapid initial solutions and convergence to optimal paths over time. By contrast, earlier methods such as RRT-Connect [7] prioritise speed but lack asymptotic optimality.

Despite their success in many real-world applications, these planners face challenges in environments with dense clutter, where computationally expensive collision checking can result in longer runtimes. More crucially, these approaches struggle in dynamic environments, where changes in obstacle locations, object configurations, or other aspects of the environment typically necessitate complete re-planning. Anytime planning methods [76, 77] address this limitation by incrementally repairing existing solutions, but still exhibit slow runtimes in cluttered

or frequently changing environments. To improve planning speed, CuRobo [78] introduces a GPU-accelerated motion planner that significantly improves planning speed by parallelising collision checking and trajectory optimisation, making it suitable for real-time applications and large-scale motion planning problems.

In this thesis, RRT-Connect is employed in the approaches presented in Chapter 3 and Chapter 5. Moreover, *Grasp-MPC* leverages CuRobo to efficiently control the robot arm to a pre-grasp pose.

Optimisation-based motion planning. Optimisation-based motion planning, such as Covariant Hamiltonian optimisation for motion planning (CHOMP) [79], Stochastic Trajectory Optimisation for Motion Planning (STOMP) [80], and Trajectory Optimisation for Motion Planning (TrajOpt) [81], represents another major class of motion planning methods that formulate trajectory generation as a continuous optimisation problem. CHOMP uses functional gradient descent to generate smooth, collision-free motions by minimising a cost composed of obstacle and smoothness terms. STOMP employs stochastic trajectory optimisation but does not require gradients, thereby incorporating non-differentiable cost terms. TrajOpt formulates a non-convex trajectory optimisation problem as sequential convex optimisation, achieving fast convergence while handling constraints such as joint limits and continuous-time collision checking. While these methods are capable of producing high-quality trajectories in cluttered environments, they are sensitive to initialisation and often struggle with local minima, particularly in high-dimensional or non-convex spaces.

2.1.2 Trajectory Optimisation for Receding-Horizon Control

In contrast to geometric motion planning methods, which compute complete trajectories prior to execution, receding-horizon control, commonly instantiated as Model Predictive Control (MPC), often optimises a short-horizon trajectory at each control step and applies only the first action. This enables online adaptation to dynamic environments and model uncertainty, making it well-suited for manipulation tasks in real-world settings.

Within this framework, trajectory optimisation methods, both sampling-based and gradient-based, are widely employed to generate action sequences using a known dynamics model. While these methods have demonstrated effectiveness across a range of tasks [82–85], they face significant challenges in contact-rich manipulation, where accurately modelling contact dynamics, handling the non-smooth nature of contact interactions, and designing task-specific cost functions are challenging. Moreover, cost functions in these settings are frequently sparse, poorly shaped, or misaligned with task objectives, which hinders the optimiser's ability to converge to high-quality solutions. These issues are further exacerbated in high-dimensional state-action spaces, such as those encountered in dexterous or deformable object manipulation, where the search space becomes prohibitively large. In Chapter 7, *TWIST* focuses on learning a dynamics model, whereas Chapter 8 addresses the challenge of finding high-performing solutions for complex dexterous deformable manipulation tasks given a cost function that provides limited task information. Furthermore, Grasp-MPC, presented in Chapter 4, learns a value function and employs it as the cost for grasping, rather than relying on a geometric cost such as the distance to a desired grasp pose, which is often noisy in real-world environments.

Sampling-Based Approaches. Sampling-based methods evaluate a set of control sequences through a dynamics model and select the trajectory that minimises a predefined cost function. A widely used class of such methods includes the Cross-Entropy Method (CEM) [11] and Model Predictive Path Integral (MPPI) [12], both of which optimise a sequence of control actions by iteratively refining a sampling distribution.

CEM maintains a time-indexed parametric Gaussian distribution over control inputs, $q(\mathbf{u}_{t:t+T-1}) = \prod_{\tau=t}^{t+T-1} \mathcal{N}(\mathbf{u}_\tau; \boldsymbol{\mu}_\tau, \boldsymbol{\Sigma}_\tau)$, and repeatedly samples K trajectories, evaluates their total costs, and updates the distribution parameters toward the samples with lower costs. At iteration i, CEM draws control sequences $\{\mathbf{u}^{(k)}_{t:t+T-1}\}_{k=1}^{K} \sim q_i$, rolls out dynamics $\mathbf{x}^{(k)}_{\tau+1} = f(\mathbf{x}^{(k)}_\tau, \mathbf{u}^{(k)}_\tau)$, and computes trajectory costs

$$C^{(k)} \;=\; \sum_{\tau=t}^{t+T-1} c\Big(\mathbf{x}^{(k)}_\tau,\, \mathbf{u}^{(k)}_\tau\Big) \;+\; c_T\Big(\mathbf{x}^{(k)}_{t+T}\Big)\,. \tag{2.1}$$

Let $\mathcal{E}$ denote the indices of the M elite trajectories with the lowest costs. The distribution is then updated via maximum-likelihood estimation on elites:

$$\boldsymbol{\mu}_\tau \leftarrow \frac{1}{M}\sum_{k\in\mathcal{E}} \mathbf{u}_\tau^{(k)}, \qquad \boldsymbol{\Sigma}_\tau \leftarrow \frac{1}{M}\sum_{k\in\mathcal{E}} \Big(\mathbf{u}_\tau^{(k)} - \boldsymbol{\mu}_\tau\Big)\Big(\mathbf{u}_\tau^{(k)} - \boldsymbol{\mu}_\tau\Big)^\top, \tag{2.2}$$

for $\tau = t, \ldots, t+T-1$. This iterative process concentrates sampling in promising regions of the control space, but typically requires a large number of rollouts to converge, particularly in high-dimensional tasks.

MPPI [12], on the other hand, performs a single round of weighted averaging over sampled trajectories, providing a more efficient alternative suitable for real-time control. Formally, at time t, MPPI maintains a nominal control sequence $\{\mathbf{u}_{t:t+T-1}\}$ and draws K noise sequences $\{\boldsymbol{\epsilon}_{t:t+T-1}^{(k)}\}_{k=1}^{K}$ with $\boldsymbol{\epsilon}_\tau^{(k)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Rollouts are generated via the dynamics $\mathbf{x}_{\tau+1}^{(k)} = f\Big(\mathbf{x}_\tau^{(k)},\, \mathbf{u}_\tau + \boldsymbol{\epsilon}_\tau^{(k)}\Big)$, and each trajectory incurs cost

$$C^{(k)} = \sum_{\tau=t}^{t+T-1} c\Big(\mathbf{x}_\tau^{(k)},\, \mathbf{u}_\tau + \boldsymbol{\epsilon}_\tau^{(k)}\Big) \;+\; c_T\Big(\mathbf{x}_{t+T}^{(k)}\Big)\,. \tag{2.3}$$

Then, importance weights are computed as

$$w_k \;=\; \exp\!\Big(-\frac{1}{\lambda}C^{(k)}\Big), \qquad \tilde{w}_k \;=\; \frac{w_k}{\sum_{j=1}^{K} w_j}, \tag{2.4}$$

with temperature $\lambda > 0$. The control update is the weighted average of the exploration noise:

$$\mathbf{u}_\tau \leftarrow \mathbf{u}_\tau \;+\; \sum_{k=1}^{K} \tilde{w}_k\, \boldsymbol{\epsilon}_\tau^{(k)}, \qquad \tau = t, \ldots, t+T-1, \tag{2.5}$$

after which the first control $\mathbf{u}_t$ is applied and the horizon is shifted forward.

To enable real-time deployment for robot manipulation tasks, CuRobo [78] presents a GPU-accelerated implementation of MPPI, significantly improving computational efficiency and making MPC feasible for real-world robotic applications. In CuRobo, given a robot dynamics model, MPPI samples sequences of joint accelerations $\mathbf{u} \in \mathbb{R}^d$ as actions and integrates them forward to obtain the corresponding joint velocities and positions. We employ a kinematic model in joint space. The robot state is defined as

$$\mathbf{x} = [\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}] \in \mathbb{R}^{3d}.$$

At each optimisation iteration, the state across the horizon is computed in a fully batched manner by integrating the sampled control sequences from the current robot state

$$\mathbf{x}_{\text{init}} = [\boldsymbol{\theta}_{\text{init}}, \dot{\boldsymbol{\theta}}_{\text{init}}, \ddot{\boldsymbol{\theta}}_{\text{init}}].$$

Following the formulation in CuRobo, the integration is implemented as

$$\ddot{\boldsymbol{\Theta}} = \mathbf{u}, \tag{2.6}$$

$$\dot{\boldsymbol{\Theta}} = \dot{\boldsymbol{\theta}}_{\text{init}} + S_\ell(1)\,\text{diag}(\mathbf{dt})\,\ddot{\boldsymbol{\Theta}}, \tag{2.7}$$

$$\boldsymbol{\Theta} = \boldsymbol{\theta}_{\text{init}} + S_\ell(1)\,\text{diag}(\mathbf{dt})\,\dot{\boldsymbol{\Theta}}, \tag{2.8}$$

where $S_\ell(1)$ denotes a strictly lower triangular matrix of ones and $\mathbf{dt}$ contains the timestep durations across the planning horizon. This formulation realises cumulative Euler integration in tensor form, enabling efficient parallel rollout over both the batch dimension and the planning horizon. This first-order integration assumes accelerations are constant within each interval and provides a computationally efficient approximation suitable for real-time sampling-based optimisation.

The optimisation is performed subject to constraints, including (1) joint limits on position, velocity, acceleration, and jerk; (2) self-collision avoidance; and (3) robot–world collision avoidance. For real-world deployment, particularly in Grasp MPC presented in Chapter 4, acceleration commands are not executed directly. Instead, a proportional-derivative controller tracks desired joint positions and velocities and outputs target joint velocity commands.

These sampling-based methods are well-suited for high-dimensional control tasks and exhibit robustness to non-convex objectives and discontinuities. However, these approaches still struggle with contact-rich manipulation tasks due to insufficient exploration and the difficulty of defining suitable task-specific cost functions.

In this thesis, *D-Cubed*, presented in Chapter 8, demonstrates that learning a sampling distribution over control inputs using a latent diffusion model enables effective exploration of high-dimensional state spaces and introduces a novel sampling-based trajectory optimisation method that integrates the CEM into the

reverse diffusion process for dexterous deformable manipulation tasks. Moreover, *Grasp-MPC*, presented in Chapter 4, integrates CuRobo's MPPI with a value function learnt from offline data, which serves as a cost function. This enables the robot to grasp novel objects in cluttered environments by leveraging both model-based control and generalisable vision-based cost function trained from data.

Gradient-Based Approaches. Gradient-based trajectory optimisation methods formulate control as a constrained optimisation problem and solve it using iterative updates based on gradient information. Established methods, such as the iterative Linear Quadratic Regulator (iLQR) [86] and Differential Dynamic Programming (DDP) [87], linearise system dynamics and approximate the cost function quadratically around a nominal trajectory, enabling efficient trajectory updates through backwards passes. These methods have been extensively studied in robotics and applied to tasks such as locomotion and, to a more limited extent, manipulation, where they generate dynamically feasible trajectories within control limits and contact constraints. Such frameworks are attractive for their sample efficiency and rapid convergence, particularly when accurate and differentiable dynamics models are available. However, they are sensitive to initialisation and typically assume smooth cost and dynamics functions.

More recently, gradient-based optimisation has been extended through the use of differentiable physics simulators [88–91], which allow the direct computation of trajectory gradients through full physical simulation, including contact-rich and discontinuous dynamics. While these tools broaden the applicability of gradient-based methods to manipulation scenarios, they remain computationally intensive and are often sensitive to simulator stability. In addition, gradients computed from differentiable simulators are often noisy and unreliable [92], making optimisation challenging, as observed in the baseline experiments presented in *D-Cubed* (see Chapter 8).

2.2 Robot Learning for Manipulation Tasks

Robot manipulation is often modelled within the Markov Decision Process framework when formulated as a sequential decision making problem, particularly in learning based approaches. The MDP abstraction provides a unified representation of state transitions, actions, and rewards. However, robot manipulation has traditionally been studied through motion planning, trajectory optimisation, and feedback control, where dynamics, constraints, and stability are treated explicitly. In many manipulation settings, contact dynamics and feasibility considerations are central and cannot be reduced solely to reward specification. In this thesis, we adopt the MDP formulation as a convenient and expressive framework for integrating learning and planning, while recognising that it represents one of several complementary perspectives on robotic manipulation.

An MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$ the transition dynamics, $r(\mathbf{s}_t, \mathbf{a}_t)$ the reward function, and $\gamma \in [0, 1]$ the discount factor. At each timestep t, the policy observes a state $\mathbf{s}_t \in \mathcal{S}$, selects an action $\mathbf{a}_t \in \mathcal{A}$, receives a reward $r(\mathbf{s}_t, \mathbf{a}_t)$, and transitions to the next state $\mathbf{s}_{t+1} \sim p(\cdot \mid \mathbf{s}_t, \mathbf{a}_t)$. In many real-world robotic settings, the full state $\mathbf{s}_t$ is not directly observable due to sensor noise or occlusions. These scenarios are formalised as partially observable MDPs (POMDPs), in which the policy receives observations $\mathbf{o}_t$ from an observation model $p(\mathbf{o}_t \mid \mathbf{s}_t)$ and selects actions based on these observations. The objective is to learn a policy $\pi(\mathbf{a}_t \mid \mathbf{o}_t)$ that maximises the expected cumulative return $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right]$.

Prior work has addressed the challenges of contact-rich manipulation through both analytical and algorithmic approaches. For example, compliance control [1, 93–95] compensates for position uncertainty by adjusting the robot's response to contact forces. Moreover, spiral search [2, 3] systematically probes the environment to identify task-relevant contact configurations when exact state information is unavailable. While effective in such tightly controlled settings, these methods typically rely on manually designed strategies and strong assumptions about object

geometry, contact dynamics, or environmental structure, which restricts their applicability in more variable or unstructured scenarios. In contrast, learning-based approaches offer greater generalisation, adaptability, and dexterity, making them particularly suitable for unstructured or uncertain environments. Thus, recent work has increasingly explored RL and IL to acquire adaptive and generalisable contact-rich manipulation skills. To this end, this section provides an overview of learning-based approaches for robot manipulation tasks.

2.2.1 Model-free Reinforcement Learning

Deep reinforcement learning (RL) [21, 96, 97] has shown remarkable results in controlling a robot for complex manipulation tasks. To mitigate manual engineering by users with domain-specific knowledge [98] for real-world robotic applications, deep RL enables a robot to autonomously acquire skills given a reward function. Driven by rewards, model-free RL [21, 31, 97, 99] enables robots to perform diverse manipulation tasks, including grasping [43, 100, 101], picking and placing [102], and insertion [103, 104], through interaction with the environment without explicitly modelling the dynamics.

Model-free RL methods are broadly categorised into on-policy [31, 105] and off-policy [21, 22] approaches. In on-policy RL, the policy is updated using data collected from the same policy currently being optimised. As a result, data generated by earlier versions of the policy cannot be reused, leading to poor sample efficiency. Despite this limitation, on-policy methods tend to offer more stable policy updates, as the data distribution is closely aligned with the target policy.

Proximal Policy Optimisation (PPO) [31] exemplifies such methods, where the objective is to maximise a clipped surrogate function that constrains large policy updates:

$$L^{PPO}(\theta) = \mathbb{E}[\min(\frac{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{old}(\mathbf{a}_t|\mathbf{s}_t)}\hat{A}_t, \text{clip}(\frac{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{old}(\mathbf{a}_t|\mathbf{s}_t)}, 1-\epsilon, 1+\epsilon)\hat{A}_t] \tag{2.9}$$

where $\hat{A}_t$ is the advantage estimate computed using Generalised Advantage Estimation [106]. While PPO is typically sample-inefficient, it can achieve effective learning

within shorter wall-clock time when trained in parallelised simulation environments such as IsaacSim [107], due to the rapid collection of on-policy experience. *COMBO-Grasp*, introduced in Chapter 5, employs PPO to train a policy in IsaacSim.

On the other hand, off-policy RL such as Soft Actor-Critic (SAC) [21] and Twin Delayed Deep Deterministic Policy Gradient (TD3) [22] updates the policy using data generated by a behaviour policy that is distinct from the current policy being optimised. The collected transitions are stored in a replay buffer, from which mini-batches are sampled to update both the actor and critic networks.

In particular, SAC maximises a stochastic objective that augments the expected return with an entropy regulariser to encourage exploration:

$$J(\pi) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi}[r(\mathbf{s}_t,\mathbf{a}_t) + \alpha\mathcal{H}(\pi(\cdot|\mathbf{s}_t))] \tag{2.10}$$

where $\mathcal{H}(\pi(\cdot|\mathbf{s}_t)) = -\mathbb{E}_{\mathbf{a}_t\sim\pi}[log\pi(\mathbf{a}_t|\mathbf{s}_t)]$ represents the policy entropy and α controls the trade-off between exploitation and exploration.

In contrast to on-policy methods, off-policy RL is generally more sample-efficient as it enables extensive reuse of past experiences. Therefore, off-policy RL [62, 104, 108, 109] is well-suited for training policies in real-world environments. The approach presented in Chapter 3 utilises SAC to leverage these advantages and train a policy in real-world environments. However, this comes at the cost of potential instability, due to the distributional mismatch between the behaviour policy and the target policy, and often requires careful hyperparameter tuning. Furthermore, Q-value learning with function approximation is known to suffer from overestimation bias [110], which can degrade policy performance. Both SAC and TD3 address this issue by employing two critic networks and computing a target using the minimum of the two estimated Q-values, thereby mitigating overestimation and improving training stability. In particular, the critic in SAC is trained to minimise the temporal-difference loss between the predicted Q-value and a target value that includes both the reward and the entropy term:

$$L^Q(\phi_i) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t,r_t,\mathbf{s}_{t+1})\sim\mathcal{D}}\Big[\Big(Q_{\phi_i}(\mathbf{s}_t,\mathbf{a}_t) - y_t\Big)^2\Big] \tag{2.11}$$

$$y_t = r_t + \gamma\Big(\min_{j=1,2} Q_{\bar{\phi}_j}(\mathbf{s}_{t+1},\mathbf{a}'_{t+1}) - \alpha\log\pi_\theta(\mathbf{a}'_{t+1}|\mathbf{s}_{t+1})\Big), \tag{2.12}$$

where $\bar{\phi}_j$ denotes the target network parameters, which are updated as an exponential moving average of the critic parameters ϕ, and $\mathbf{a}'_{t+1} \sim \pi_\theta(\cdot|\mathbf{s}_{t+1})$ is sampled from the current policy. This formulation combines entropy-regularised value learning with double-Q estimation to stabilise training and reduce overestimation bias. Moreover, for efficient learning, RLPD [111] performs multiple update steps per environment timestep, known as the update-to-data (UTD) ratio, and regularises the critic with layer normalisation to mitigate catastrophic overestimation. Moreover, in this thesis, Chapter 3 builds upon FERM [100], a framework for sample-efficient vision-based RL under sparse reward settings, implemented on top of SAC. Specifically, a visual encoder is first pre-trained using a small set of demonstrations via contrastive learning. The encoder is then jointly fine-tuned with the policy and critic networks during RL, while incorporating data augmentation, particularly random cropping, to improve robustness and sample efficiency.

While several prior works [112–115] have attempted to learn RL policies for bimanual manipulation, the problem remains inherently challenging due to the high-dimensional action space and the difficulty of achieving effective coordination between the two arms. To address this, prior works [113–115] have introduced inductive biases into RL methods. Predefined, parameterised skills restrict the policy to operate over temporally extended primitives rather than low-level motor commands [113]. This structured action space reduces the need for extensive exploration and alleviates long-horizon credit assignment. In addition, intrinsic motivation reshapes the reward landscape to favour informative state transitions [114], guiding exploration toward behaviours that accelerate skill acquisition. Finally, symmetry-aware actor-critic architectures embed known geometric symmetries directly into the network design [115]. By enforcing equivariance, these models eliminate redundant representations and generalise more effectively across symmetric states.

Examples include predefined parameterised skills [113], intrinsic motivation mechanisms for more efficient exploration [114], and symmetry-aware actor-critic architectures [115], all of which have demonstrated improved learning efficiency and

performance. Similarly, *COMBO-Grasp*, presented in Chapter 5, uses motion planning to decompose bimanual occluded grasping tasks. This allows two coordinated policies to specialise: one policy controls the constraint arm to stabilise the object, while the other learns to perform the grasp by exploiting the stabilisation.

Using a small number of demonstrations for guided exploration. To improve sample efficiency and facilitate effective exploration for complex manipulation tasks, policies are often trained with manually designed dense reward functions, which require substantial domain expertise. Such dense reward functions are rarely available in real-world environments. On the other hand, training RL policies from sparse rewards, which are more realistic in real-world environments, is significantly more challenging due to the hard exploration problem. This difficulty is further exacerbated in contact-rich manipulation, where precisely coordinated motions are required and small deviations can quickly lead to failure, making unguided exploration unlikely to succeed. To address this challenge, several previous works [104, 111, 116] have leveraged a small number of demonstrations to guide exploration, thereby overcoming the hard exploration problem under sparse rewards. For instance, a small number of demonstrations can be stored in a separate replay buffer or mixed with exploration data [104, 111]. This enables the policy to focus exploration on promising states observed in the expert demonstrations. Similarly, the approach presented in Chapter 3 stores a small number of expert demonstrations in a replay buffer to train a policy for insertion tasks under sparse reward conditions. Moreover, prior works [108, 109] leverage a small number of expert demonstrations and additionally incorporate human interventions to correct errors made by the behaviour policy.

More recently, Imitation Bootstrapped RL [116] trains a policy using IL, followed by an RL stage that trains another policy. During RL training, the Q-value function evaluates actions from both the imitation and RL policies, selecting the one that maximises the estimated value to facilitate efficient exploration and effective exploitation of the demonstrations. Similarly, residual RL is another common

approach, where a base policy is trained from expert demonstrations, and an RL policy is subsequently trained to adjust the output of the IL policy [117].

Sim-to-real transfer. Training policies directly in the real world is often impractical due to safety concerns and sample inefficiency. To address this limitation, sim-to-real transfer [57] is commonly employed, in which a policy is trained in simulation and subsequently deployed in real-world environments. However, domain gaps between simulated and real-world environments, such as differences in physics parameters, textures, and controllers, degrade transfer performance. To mitigate these effects, domain randomisation [57] is a widely adopted technique that randomises such parameters during training in simulation to expose the policy to a wide range of variations, thereby increasing the likelihood that real-world conditions fall within the training distribution encountered in simulation. In particular, *COMBO-Grasp* and *TWIST* presented in Chapter 5 and 7 apply domain randomisation for effective sim-to-real transfer.

Although not the focus of this thesis, system identification through real-to-sim transfer [118, 119] represents an alternative approach to domain randomisation. In particular, this method estimates the physics parameters of real-world objects to build accurate simulation models, trains a policy in simulation, and then transfers the policy back to the real world.

In addition, several prior works employ teacher–student distillation frameworks [33, 59, 65, 66], which leverage low-dimensional state observations available in simulation to efficiently train a state-based teacher policy. The trained teacher policy then supervises a vision-based student policy that operates on high-dimensional visual inputs. This approach is employed in *COMBO-Grasp*, presented in Chapter 5.

2.2.2 Model-based Reinforcement Learning

In contrast to model-free RL, model-based RL learns a dynamics model from data and optimises a policy or plans an action sequence within the learnt dynamics, effectively unifying model-based planning and learning under a single framework. Model-based RL is generally more sample-efficient and safer than

model-free methods, as it can train policies or evaluate action sequences using imagined rollouts in the learnt dynamics model, thereby reducing reliance on costly and potentially unsafe real-world interactions. Early approaches such as PILCO [120] demonstrated remarkable sample efficiency on low-dimensional state spaces by employing Gaussian Process dynamics models. To improve scalability, PETS [121] introduced probabilistic ensembles of neural networks combined with trajectory optimisation using the Cross-Entropy Method (CEM) [11], achieving strong performance on high-dimensional continuous control tasks. While these methods assumed access to structured state observations, the visual foresight framework [122] extended model-based RL to pixel observations by predicting future visual outcomes conditioned on action, thereby enabling planning directly in image space, often via sampling-based optimisation such as CEM.

Building on these foundations, world models [55] employ a variational autoencoder (VAE) [17] to learn compact latent representations jointly with predictive dynamics models for effective planning and policy optimisation. *PlaNet* [60] advances this and proposes the Recurrent State Space Model (RSSM), which models both deterministic and stochastic transitions in the latent space. This improves predictive accuracy and supports long-horizon planning in simulated environments such as Atari [123] and the DeepMind Control Suite [124]. Dreamer [13, 14, 125] builds upon PlaNet and optimises a policy on the world model to tackle more complex continuous control tasks and simple real-world manipulation tasks [126], demonstrating strong performance from raw pixel observations without access to privileged state information.

In addition to policy optimisation, several prior works [15, 127] leverage model predictive control (MPC) on the world model, integrating forward planning with policy learning to enhance sample efficiency. Recently, akin to model-free RL, MoDem [128, 129] leverages a small number of demonstrations to pre-train a world model, and subsequently collects real-world data through online interaction to fine-tune the dynamics model, thereby improving both policy and planning. In contrast to approaches that rely on offline datasets or direct real-world interaction,

which are often costly or unsafe, Chapter 7 introduces an efficient sim-to-real transfer framework for model-based RL.

Moreover, recent studies [130–132] seek to improve the accuracy of modelling robot–environment interactions by leveraging Transformer [133] architectures and diffusion models [18, 69].

2.2.3 Offline Reinforcement Learning

Offline RL [134–136] facilitates policy learning from offline datasets without requiring interaction with the environment, utilising both successful and failed trajectories to enhance policy performance. However, as discussed in prior work [137], the main challenge often lies in policy extraction rather than in value function estimation. Policy extraction refers to the procedure that derives a policy from learnt value estimates. In offline settings, naïvely maximising a learnt Q-function is problematic, as the maximisation operator tends to favour out-of-distribution actions whose values are inaccurately estimated, leading to poor generalisation at test time. Notably, advantage-weighted methods such as AWR [134] and IQL [135] address this issue by constraining policy improvement to actions observed in the dataset.

In particular, IQL, used as a baseline in Chapter 4, first learns a state value function $V_\psi(s)$ using expectile regression. Specifically, the value function is obtained by minimising

$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[L_2^\tau\Big(Q_\theta(s,a) - V_\psi(s)\Big)\right],$$

where $L_2^\tau(u) = |\tau - \mathbf{1}(u < 0)|u^2$ is the asymmetric squared loss with expectile parameter $\tau \in (0.5, 1)$. This objective estimates an upper expectile of $Q(s,a)$ without performing an explicit maximisation over actions, thereby mitigating overestimation on out-of-distribution actions.

Then, the estimated value function is used to form the regression target in the Q-function update.

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\Big(Q_\theta(s,a) - (r + \gamma V_\psi(s'))\Big)^2\right].$$

The bootstrap target depends on the learnt value function $V_\psi(s')$ rather than on $\max_{a'} Q(s', a')$, avoiding maximisation over unseen actions. Policy extraction is then performed via advantage-weighted regression [134], optimising

$$\mathcal{L}_\pi(\phi) = -\mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\exp\left(\beta(Q_\theta(s,a) - V_\psi(s))\right) \log \pi_\phi(a|s) \right],$$

where $\beta \in [0, \infty)$ is an inverse temperature. This objective corresponds to weighted behavioural cloning, assigning greater weight to actions with higher estimated advantage while remaining within the dataset support.

By contrast, *Grasp-MPC* trains only a value function and employs sampling-MPC, particularly MPPI, as a policy at test time. This design eliminates the need for a separate extraction stage by directly translating value estimates into planned action sequences, allowing the system to actively search ahead for collision-free trajectories that achieve the target task.

2.2.4 Imitation Learning

While deep RL usually requires a large number of interactions with environments to train a policy, imitation learning (IL) [23–25] enables the robot to imitate behaviour in demonstrations provided by expert users. There are mainly two approaches in IL: behaviour cloning (BC) [138, 139] which employs supervised learning to learn a mapping function from a state to action, and inverse RL [140–143], which learns to estimate a reward function from the expert demonstrations and train a policy using RL given the learnt reward function. BC methods often suffer from compounding errors [144], where small mistakes made by the learnt policy accumulate over time, leading to significant deviations from expert behaviour. A prominent approach to mitigating this issue is DAgger [144], in which the expert policy (or a human demonstrator) labels the correct action for states encountered by the IL policy. These newly labelled state–action pairs are then aggregated with the original demonstration dataset, enabling the IL policy to learn more robust behaviour under distribution shift. Similarly, DITTO [145] addresses covariate shift

by rolling out the IL policy within a learnt dynamics model and aligning its latent trajectory representations with those of the expert demonstrations.

Recently, several prior works have proposed using Transformers [146] or Diffusion models [147] as policy architectures to capture complex multi-modal action distributions. While *COMBO-Grasp*, presented in Chapter 5, does not require demonstrations collected by expert users, it employs a diffusion policy as a vision-based student grasping policy distilled from a state-based teacher policy. In parallel, the vision-based constraint policy is trained using a Gaussian mixture model [148]. Specifically, data are collected by executing the expert (teacher) policy, after which a vision-based student policy is trained to imitate the teacher's action outputs.

Expert demonstrations are commonly collected via teleoperation, using interfaces such as game controllers, space mice, or VR controllers. Recently, the ALOHA [26] teleoperation system has enabled intuitive data collection by allowing a human operator to physically manipulate a local robot, whose joint commands are mirrored in real time by an identical remote robot that performs the task. Since both robots share the same hardware, this setup eliminates the need for complex remapping or interfaces, enabling natural and accurate demonstrations. While the ALOHA system requires two identical full-sized robot arms, which can be expensive, GELLO [149] offers a low-cost alternative by using a 3D-printed, small-scale replica with the same kinematic structure. This scaled robot enables intuitive and affordable teleoperation by directly mapping its joint positions to those of the target robot (see Figure 2.1).

Figure 2.1: GELLO for a bimanual robot. Collecting high-quality demonstrations remains time-consuming, limiting large-scale data acquisition for generalisation to novel objects.

In contrast to collecting demonstrations for a parallel gripper, acquiring demonstrations for a dexterous robot hand is particularly challenging due to its high-dimensional configuration space and complex kinematic structure. Several prior

studies [150, 151] have employed VR/AR headsets to track human hand poses, which are subsequently mapped to the corresponding states of the robot hand. Similarly, AnyTeleop [152] uses only an RGB camera to estimate human hand poses and retargets them to the robot hand states.

Nevertheless, collecting a large number of demonstrations remains time-consuming and challenging, especially for complex, contact-rich manipulation tasks. This thesis primarily aims to leverage readily available data for efficient skill acquisition rather than focusing on IL. In particular, the approach presented in Chapter 3 incorporates a small number of expert demonstrations, collected via game controllers, to guide exploration in RL. Moreover, D-Cubed, presented in Chapter 8, leverages a task-agnostic structured play dataset that records only meaningful hand motions without robot–object interaction, making it far easier to collect than expert demonstrations. Using a method similar to AnyTeleop, human hand poses are estimated from RGB images and retargeted to robot hand states. The resulting dataset is then used to train an action sampler to facilitate exploration for trajectory optimisation.

2.3 Generative Models in Robotics

This section provides an overview of generative models that are particularly relevant to robotics and to this thesis. Deep generative models have gained increasing attention in robotics due to their ability to capture complex data distributions and generate diverse samples. Generative models are commonly employed for policy learning, dynamics modelling, grasp pose generation, action sampling, and trajectory generation.

2.3.1 Variational Autoencoders

The variational autoencoder (VAE) [17] is one of the most fundamental approaches, learning a latent representation by encoding inputs into a distribution, typically parameterised as a multivariate normal, from which latent variables are sampled and decoded to reconstruct the original data. Formally, given an input $\mathbf{x}$ and

latent variable $\mathbf{z}$, a VAE maximises the evidence lower bound (ELBO) on the marginal likelihood:

$$\mathcal{L}_{\text{VAE}}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta\, D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p(\mathbf{z}))\,, \tag{2.13}$$

where $q_\phi(\mathbf{z}|\mathbf{x})$ is the encoder, $p_\theta(\mathbf{x}|\mathbf{z})$ is the decoder, and $p(\mathbf{z})$ is a prior distribution, typically $\mathcal{N}(0, I)$. The first term encourages accurate reconstruction, while the second regularises the latent distribution to align with the prior. The coefficient β controls the degree of disentanglement in the latent space, as introduced in the β-VAE [16].

VAEs have been applied to various domains, such as grasp pose prediction [153], to the learning of latent representations that support downstream policy learning [154], and as a central component of world models [13, 14, 55], where they are combined with recurrent state-space models to capture system dynamics. In Chapter 6, *AMP-LS* employs a beta-VAE [16] to learn structured latent spaces for gradient-based motion planning. Moreover, *TWIST* in Chapter 7 enables effective sim-to-real transfer of world models by leveraging the capability of generative modelling. Moreover, *D-Cubed*, presented in Chapter 8, employs a VAE to learn skill latent representations by reconstructing sequences of short-horizon actions to learn a latent diffusion model as an action sampler, thereby facilitating effective trajectory optimisation.

In contrast to learning monolithic latent representations, object-centric generative models [35, 155] have recently emerged to decompose a scene into object-centric representations, where each entity is represented individually. APEX [35], for instance, is built on a VAE and learns object-centric representations from videos. In this thesis, Chapter 3 employs APEX for versatile one-shot target object identification by comparing object-centric representations in a new scene with the reference target object representation obtained from an expert demonstration.

2.3.2 Diffusion Models

Recently, diffusion models, formulated as Denoising Diffusion Probabilistic Models (DDPMs) [18], have seen widespread adoption across robotics applications. Formally, given a data sample $\mathbf{x}_0 \sim q_{\text{data}}(\mathbf{x})$, the forward diffusion process constructs a Markov chain $\mathbf{x}_0, \mathbf{x}_1, ...\mathbf{x}_T$ such that

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\Big(\sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\, \beta_t\mathbf{I}\Big)\,, \tag{2.14}$$

where β_t denotes a small positive noise variance at each timestep t. The reverse process, which removes noise to generate samples, is defined as

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad p_\theta(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2.15}$$

with

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x}_t, t),\, \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))\,. \tag{2.16}$$

The mean μ_θ and the variance Σ_θ are defined as:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t, \tag{2.17}$$

$$\Sigma_\theta(\mathbf{x}_t, t) = \sigma_t^2\mathbf{I} = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t, \tag{2.18}$$

where $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{j=1}^{t} \alpha_j$. In practice, a denoising network learns to predict either the added noise $\boldsymbol{\epsilon}_t$ or the clean data $\mathbf{x}_0$ directly. A simplified training objective for predicting $\mathbf{x}_0$ can be written as

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0),\, t\sim[1,T]}\left[\|\mathbf{x}_0 - G_\theta(\mathbf{x}_t, t)\|_2^2\right], \tag{2.19}$$

where G_θ denotes the denoising network parameterised by θ.

Guidance mechanisms further enhance the capability of diffusion models. Classifier guidance [50] directs the reverse diffusion process using gradients from a classifier.

$$\boldsymbol{\mu}_\theta^{\text{guided}}(\mathbf{x}_t, t, y) = \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) + s\,\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\nabla_{\mathbf{x}_t}\log p_\phi(y|\mathbf{x}_t), \tag{2.20}$$

where s is a guidance scale controlling the influence of the classifier. This steers the reverse diffusion process toward regions in which the classifier outputs a higher

likelihood to the desired label y. On the other hand, classifier-free guidance [156] achieves a similar effect without requiring an external classifier.

In robotics, diffusion policies [147, 157] leverage diffusion models to learn manipulation policies from expert demonstrations. Beyond policy learning, diffusion models have been employed for world modelling [69, 158, 159], motion planning [160], and grasp synthesis [37, 161]. Diffuser [159] learns an approximate dynamics model using diffusion models and employs classifier guidance, where gradients from a learnt value function steer the reverse diffusion process to generate action sequences. *COMBO-Grasp*, presented in Chapter 5, adopts diffusion models for a teacher constraint policy and student grasping policy to solve challenging occluded grasping tasks. Moreover, inspired by Diffuser, *COMBO-Grasp* introduces value-guided policy coordination, where gradients from the value function of a grasping policy guide the reverse diffusion process of the teacher constraint policy toward generating stabilisation poses that better support successful grasps. In Chapter 8, *D-Cubed* leverages a latent diffusion model [61] operating on learnt latent representations to construct an action sampler, which facilitates efficient exploration for trajectory optimisation.

2.3.3 Normalising Flows and Flow Matching

Normalising flows [162], which learn invertible mappings between simple and complex distributions, have also been widely applied, for instance, to model sampling distributions in robotics tasks [163]. These methods have subsequently been extended to flow-matching models, which are increasingly employed for training IL and RL policies [164–166]. Although not directly applied in this thesis, such approaches are particularly relevant as they are potentially capable of learning expressive action samplers for trajectory optimisation. In future work, these could be integrated with *Grasp-MPC*, presented in Chapter 4, to improve trajectory optimisation by biasing the search towards regions of the action space that are more likely to yield successful behaviours.

2.4 Unifying Planning and Learning

This thesis investigates the unification of model-based planning, such as motion planning and trajectory optimisation, with learning-based approaches to leverage their complementary strengths for efficient skill acquisition in contact-rich manipulation tasks. Model-based planning offers predictive foresight, explicit constraint handling, including safety constraints, and the ability to generate collision-free trajectories toward desired target poses; however, it is limited by model inaccuracies, computational demands, and the difficulty of finding feasible solutions in contact-rich manipulation tasks. Learning-based approaches, in contrast, can acquire flexible, dexterous, and generalisable behaviours and model complex dynamics directly from data, yet they often struggle with sample inefficiency, safety, and long-horizon exploration. Therefore, integrating these offers a principled path toward efficient, safe, and adaptable manipulation in unstructured environments.

Combining task-level planning with low-level skill learning is essential for long-horion manipulation, as real-world manipulation tasks inherently require temporally extended reasoning and sequential decision-making. However, this thesis focuses on a fundamental subproblem within long-horizon manipulation: the safe and data-efficient acquisition and execution of contact-rich manipulation skills that serve as building blocks for more complex behaviour, and we therefore do not discuss task-level integration in this section. The following sections review recent approaches in learning-based motion planning, sequential integration of motion planning and learning-based policies, and trajectory optimisation with learning-based methods.

2.4.1 Learning-based Motion Planning

Recent work has investigated learning–based approaches to enhance the speed and reactivity of motion planning. These approaches [52, 53, 167–170] typically use an offline dataset, particularly expert demonstrations generated by sampling-based motion planners. Carvalho et al. [160] extend this paradigm by employing diffusion models to learn from diverse trajectory distributions, while guiding the reverse diffusion process using constraints such as collision avoidance constraints.

While effective in novel scenes, these methods all require large datasets of expert trajectories, making them costly and harder to scale.

Moreover, several works have explored planning in learnt latent spaces [19, 171]. Latent Space Path Planning (LSPP) [19], for example, performs planning directly in the latent space trained using a generative model from readily available kinematically feasible robot states, thereby bypassing the need for costly expert trajectories. Although demonstrated in simplified settings with primitive obstacle geometries, LSPP offers key advantages such as faster planning and ease of incorporating task constraints via learnt performance predictors. LSPP is further extended to *AMP-LS*, which enables obstacle avoidance in real-world environments with complex object shapes, as presented in Chapter 6.

2.4.2 Sequential Integration of Motion Planning and Learning-based Approaches

Motion planning is effective for controlling robots in complex, cluttered environments, but it is not designed to handle contact-rich manipulation tasks and struggles due to the difficulty of accurately modelling contact dynamics. In contrast, model-free RL can acquire manipulation skills from experience, guided by a reward function. However, it typically requires large amounts of data and performs poorly in complex settings, often limited to clean tabletop environments due to exploration challenges. To combine the strengths of both approaches, several prior works [32–34, 172] integrate motion planning with learning-based policies. For example, GUAPO [34] employs an object-specific classifier to identify the hole location in a peg-in-hole task and determines when to switch between motion planning and an RL policy. MoPA-RL [32] introduces a hierarchical RL framework where the policy autonomously learns to switch between motion planning and a low-level manipulation policy given the current state to solve contact-rich manipulation tasks in obstructed environments.

However, these approaches rely on object-specific classifiers or goal identifiers trained via RL, which restricts their adaptability to previously unseen objects commonly encountered in small-batch manufacturing and open-world environments.

In contrast, Chapter 3 introduces a system that unifies motion planning and RL while leveraging an object-centric generative model for flexible goal conditioning, thereby enabling sample-efficient adaptation to novel objects and overcoming this limitation.

2.4.3 Trajectory Optimisation with Learning-based Approaches

A growing body of work has focused on integrating learning-based methods with MPC to address these fundamental challenges. This integration spans multiple directions, including the learning of dynamic models [127, 173–177], cost functions [127, 178–181], and sampling distributions [64, 163].

Neural Motion Fields [181] exemplifies attempts to learn value functions that represent distance costs to desired target poses. However, this distance-based approach demonstrates the inherent difficulty of capturing the intricate factors that determine manipulation success through simple geometric metrics, often resulting in suboptimal performance. Similarly, CV-MPC [178] learns ensemble value functions from limited demonstrations for object transport tasks, yet remains constrained by its reliance on low-dimensional state observations, limiting its applicability to manipulation tasks involving novel objects. These limitations highlight a fundamental gap in current trajectory optimisation methods: the difficulty of effectively incorporating high-dimensional sensory information and complex task semantics into optimisation frameworks suitable for real-time manipulation control. In contrast, *Grasp-MPC*, presented in Chapter 4, learns a vision-based value function from diverse synthetic grasp trajectories, enabling generalisable, safe, and reactive visual grasping in real-world environments.

3

Efficient Skill Acquisition for Insertion Tasks in Obstructed Environments

Data efficiency is a crucial factor in small-batch manufacturing applications, where robots must quickly adapt to new tasks with minimal training time and human intervention. Moreover, while many existing approaches focus on learning contact-rich manipulation skills within restricted, clean workspaces, real-world scenarios demand a robotic system capable of operating across their entire workspace, often in the presence of obstructions.

Motion planning [7, 9, 10, 30, 182] has demonstrated its ability to generate collision-free trajectories, allowing robots to navigate complex environments safely. However, motion planning is not designed to handle complex manipulation tasks that require interaction with the environment. On the other hand, RL [21, 22, 44, 66, 100, 104, 108, 109, 183] has shown promising results in mastering contact-rich manipulation tasks. Despite these advances, RL is often limited to a narrow workspace due to sample inefficiency and safety concerns, limiting its applicability in real-world environments.

To overcome the limitations and leverage the complementary strengths of both approaches, several prior works [32, 34] have attempted to combine motion planning and RL to solve manipulation tasks in obstructed environments. A key insight is that not all aspects of motion need to be learnt. Motion planning can efficiently generate collision-free trajectories to approach and position a robot around the

target object, while RL is used to handle the contact-rich interactions. However, these prior works rely on either an object-specific pose estimator [34] or a goal estimator [32] trained using RL, which requires a significant amount of data and time to retrain them when a new object is introduced. To enable real-world applications in small-batch settings, the ability to identify objects in a versatile and efficient manner is also a crucial requirement.

To this end, this chapter introduces a robotic system designed for efficient skill acquisition, addressing complex insertion tasks in obstructed environments within small-batch settings (see Chapter 3, Fig. 2, for an overview of the system). As illustrated in Chapter 3, Fig. 3, the proposed system utilises an object-centric generative model [35] for one-shot, versatile identification of target objects. The output of this model is then used for motion planning, followed by an RL policy to execute the contact-rich insertion task.

In particular, the object-centric generative model outputs a set of object-specific representations corresponding to the entities present in a scene. From a single demonstration, the system extracts the representation of the target object, which serves as the reference representation of the target. In a novel scene containing the same object, the system compares each object-specific representation to the reference and selects the one most similar to it. Once the target object is identified, its location is used to define a goal for motion planning, enabling the robot to efficiently avoid obstacles and reach the target. An RL policy, trained locally around the target object using RGB images from a wrist camera and force/torque measurements, is then executed to perform the insertion task. Rather than adopting residual RL [36], which would require specifying and tuning a base policy and thus introducing additional task-specific engineering, we instead leverage a small number of demonstrations to bias the policy search toward meaningful behaviours, preserving a scalable and data-efficient framework. This combination of object-centric perception, motion planning, and RL enables the system to efficiently solve complex contact-rich manipulation tasks, even in obstructed environments.

Furthermore, to seamlessly integrate motion planning with RL, we introduce a transition network that connects the terminal state of motion planning to the initial state distribution of the RL policy. The network is trained in a self-supervised manner using a dataset collected by moving the robot from the initiation set of the RL policy to randomly sampled nearby states, recording an RGB image observation at each sampled state, and computing the displacement between the sampled state and the initiation set, which serves as the target for the transition network to regress.

Experimental results demonstrate that the proposed one-shot object identification approach utilising an object-centric generative model achieves greater versatility and comparable accuracy to baseline methods, without requiring manual engineering such as cropping a target object image (see Chapter 3, Table 1). This makes the approach particularly suitable for small-batch scenarios. Furthermore, the proposed system achieves an average success rate of 90.0% over four different complex insertion tasks in obstructed environments, outperforming competitive baselines, including a state-of-the-art RL algorithm and ablated variants of the proposed system, by a significant margin (see Chapter 3, Table 2).

By leveraging an object-centric generative model, this chapter demonstrates that integrating motion planning with RL enables versatile, data-efficient, and safe skill acquisition and execution across multiple objects. In summary, this chapter presents the following:

1. A system for efficient skill acquisition in obstructed environments, which leverages an object-centric generative model for object-agnostic one-shot target object identification, thereby overcoming the limitations of existing methods that unify motion planning and RL.

2. A transition network that smoothly connects the terminal states of motion planning to the feasible initiation set of a learnt RL policy, significantly improving the overall success rate.

3. Performance of one-shot target object identification using an object-centric generative model, showing accuracy comparable to traditional and object-specific goal identification baselines without manual engineering.

4. Real-world experiments demonstrating that the proposed system outperforms baselines, including a state-of-the-art RL algorithm and variations of the proposed system.

Jun Yamada, Jack Collins, and Ingmar Posner (2024). Efficient Skill Acquisition for Insertion Tasks in Obstructed Environments, *Learning for Dynamics and Control Conference (L4DC)*

Proceedings of Machine Learning Research vol 242:1–16, 2024 6th Annual Conference on Learning for Dynamics and Control

Efficient Skill Acquisition for Insertion Tasks in Obstructed Environments

Jun Yamada JYAMADA@ROBOTS.OX.AC.UK
Jack Collins JCOLLINS@ROBOTS.OX.AC.UK
Ingmar Posner INGMAR@ROBOTS.OX.AC.UK
Oxford Robotics Institute, 23 Banbury Road, Oxford, United Kingdom

Editors: A. Abate, K. Margellos, A. Papachristodoulou

Abstract

Data efficiency in robotic skill acquisition is crucial for operating robots in varied small-batch assembly settings. To operate in such environments, robots must have robust obstacle avoidance and versatile goal conditioning acquired from only a few simple demonstrations. Existing approaches, however, fall short of these requirements. Deep reinforcement learning (RL) enables a robot to learn complex manipulation tasks but is often limited to small task spaces in the real world due to sample inefficiency and safety concerns. Motion planning (MP) can generate collision-free paths in obstructed environments, but cannot solve complex manipulation tasks and requires goal states often specified by a user or object-specific pose estimator. In this work, we propose a robust system for efficient skill acquisition designed to address complex insertion tasks in obstructed environments. Our system leverages an object-centric generative model (OCGM) for versatile goal identification to specify a goal for MP combined with RL to solve complex manipulation tasks in obstructed environments. Particularly, OCGM enables one-shot target object identification and re-identification in new scenes, allowing MP to guide the robot to the target object while avoiding obstacles. This is combined with a skill transition network, which bridges the gap between terminal states of MP and feasible start states of a sample-efficient RL policy. The experiments demonstrate that our OCGM-based one-shot goal identification provides competitive accuracy to other baseline approaches and that our modular framework outperforms competitive baselines, including a state-of-the-art RL algorithm, by a significant margin for complex manipulation tasks in obstructed environments.

Keywords: Robotic Manipulation, Integrated Planning and Learning, Reinforcement Learning, Motion Planning, Learning from Demonstration

1. Introduction

Teaching new skills to robots using limited supervision is essential for maximising the up-time and productivity of robots, leading to faster return on investment. Small-batch manufacturing, where there are a limited number of parts to be produced, is an exemplary environment that would greatly benefit from efficient skill acquisition. In a small-batch setting, a robot must learn to manipulate new objects while maintaining data efficiency in potentially arbitrarily obstructed environments. However, existing methods for controlling a manipulator such as motion planning and reinforcement learning individually struggle to satisfy such requirements.

Motion planning (MP) (Amato and Wu, 1996; LaValle, 1998) generates collision-free paths capable of guiding a robot safely in obstructed environments given an explicit state of the environment and goal. However, MP is not designed to plan through complex manipulation tasks requiring environmental interaction. Furthermore, MP necessitates the specification of a goal state in the robot's frame of reference, which is typically accomplished through manual engineering (Khodeir et al.,

2021), template matching (Le et al., 2019), or an object-specific pose estimator (Lee et al., 2020) trained on manually labelled supervised data.

Deep reinforcement learning (RL), on the other hand, has shown promising outcomes in learning to control a robot for complex manipulation tasks such as grasping (Kalashnikov et al., 2018; Zhan et al., 2020) and insertion (Luo et al., 2021). However, prior works often limit operation to simulated environments (Haarnoja et al., 2018) or heavily restrict and regulate operating spaces by executing with a short horizon without obstructions (Luo et al., 2021; Zhan et al., 2020) due to the sample inefficiency and potential of executing unsafe policies.

Combining MP and RL has been investigated by several prior works (Yamada et al., 2020; Lee et al., 2020) and shows the potential of leveraging the strengths of both methods to solve manipulation tasks in obstructed environments. Yet, goal specification for MP in prior work has relied on either sample-inefficient interaction with the environment or an object-specific pose estimator, which needs retraining for each new target object. Notably, MoPA-RL (Yamada et al., 2020) attempts to solve similarly complex manipulation tasks but requires more than 1M samples to train the RL policy from state-based observations with fixed obstacle positions, limiting the real-world application.

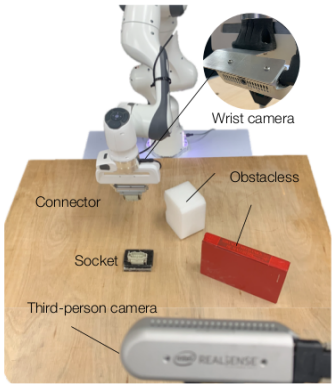


Figure 1: **Task setup.** We solve complex manipulation tasks within the entire operational space of a robot by leveraging an OCGM for versatile and efficient goal acquisition paired with MP and RL. Note that obstacles and a socket are randomly placed on the table.

Inspired by the challenges faced in small-batch manufacturing problems, we introduce a system that builds upon existing MP and RL algorithms, integrating them with an object-centric generative model (OCGM) (Wu et al., 2021) to overcome the limitations of existing methods. We posit that the integration of an OCGM leads to versatile, one-shot goal identification and re-identification, allowing for insertion tasks to be solved from a limited number of simple human demonstrations.

Specifically, we identify a target object from a *single* demonstration using an OCGM, pre-trained on diverse synthetic scenes. Matching the target object's object-centric representation to those in new scenes leads to robust object re-identification. Using the object's position as a goal, the motion planner generates a collision-free path to the target object while avoiding obstacles before a learned RL policy is executed to complete the insertion tasks. We train an RL policy for each insertion skill from a sparse reward to eliminate the need for reward engineering using specialist knowledge. We also leverage a handful of easy-to-collect demonstrations to guide exploration to achieve efficient RL policy learning. To maximise performance, we also introduce a skill transition network to reduce failures that occur when transitioning from MP to the learned RL policy.

The contributions of our work are fourfold: (1) we propose a system for efficient skill acquisition in obstructed environments that leverages an OCGM for object-agnostic to overcome the limitations of existing methods, *one-shot* goal specification, (2) we introduce a transition network that smoothly interpolates between terminal states of motion planning and feasible start states of a learned RL policy to significantly improve the successes rate of the approach, (3) we show that our OCGM-based one-shot goal specification method achieves comparable accuracy against several traditional and object-specific goal identification baselines, and (4) we demonstrate that our

system performs significantly better in real-world environments compared to baselines, including a state-of-the-art RL algorithm. In summary, our paper introduces a novel system that leverages prior MP and RL methods while distinguishing itself by eliminating the need for prior knowledge such as object geometry or object-specific detectors. We demonstrate the effectiveness of using unsupervised OCGMs to combine MP and an RL policy, making our approach particularly valuable in small-batch settings, which is our specific focus. While the individual building blocks exist, their seamless integration in a real-world robot system remains a challenging and novel achievement.

2. Related Works

Recent success in deep RL (Kalashnikov et al., 2018; Haarnoja et al., 2018) enables a robot to learn complex manipulation tasks such as grasping (Kalashnikov et al., 2018; Zhan et al., 2020) and insertion (Luo et al., 2021; Vecerik et al., 2018; Lee et al., 2018; Davchev et al., 2022; Carvalho et al., 2022) driven by a reward. To avoid the requirement of specialist knowledge for reward engineering, several prior works have proposed sample-efficient RL methods that can learn complex manipulation skills from a sparse reward by leveraging a small number of demonstrations for guided exploration (Zhan et al., 2020; Luo et al., 2021; Vecerik et al., 2017, 2018). However, due to the sample inefficiency of sparse rewards, studies have been primarily conducted in simulated environments or within limited task spaces in the real world. Learning from demonstration (LfD) (Schaal, 1999; Billard et al., 2008; Groth et al., 2021) is an alternative method for a robot to learn manipulation tasks by imitating behaviour in expert demonstrations collected by a human operator, but it often requires a large number of demonstrations to acquire manipulation skills. While InsertionNet (Spector and Di Castro, 2021) enables a robot to solve insertion tasks within the entire operational space of a robot manipulator from a small number of demonstrations, it is evaluated in a clean environment without obstruction. Successful insertion is also made possible by a small initiation set for the learnt skill. Adaptive LfD for insertion has also been proposed (Wen et al., 2022), allowing a policy to quickly adapt to new insertion objects from the same category seen in training using only a single demonstration and the object mesh. However, such mesh information is not readily available, limiting real-world applications. In our work, a manipulation skill is learnt using Framework for Efficient Robot Learning (FERM)(Zhan et al., 2020).

Motion planning (MP) (Amato and Wu, 1996; Kavraki and Latombe, 1994; LaValle, 1998) can effectively generate a collision-free path from a robot's initial configuration to a goal pose using an explicit model of the robot and environment. However, such a goal pose is often specified by a user or object-specific pose estimator. Further, complex manipulation tasks such as insertion are out of the scope of MP as MP does not model the dynamics of the surrounding environment or objects.

Several previous works (Yamada et al., 2020; Lee et al., 2020; Kuo et al., 2021) combine MP and RL to leverage the benefits of both methods to solve manipulation tasks in unstructured environments. However, these preceding works limit their real-world applicability by requiring a large number of samples to learn a goal estimator (Yamada et al., 2020) or by retraining an object-specific predictor for each new goal object (Lee et al., 2020). While MoPA-RL (Yamada et al., 2020) is most closely related to our method in spirit, it requires more than 1M samples to train an RL policy from state-based observations with fixed obstacle positions. Thus, the prior work is not directly comparable to our work due to its sample inefficiency and the need for inaccessible state observations in the real world.

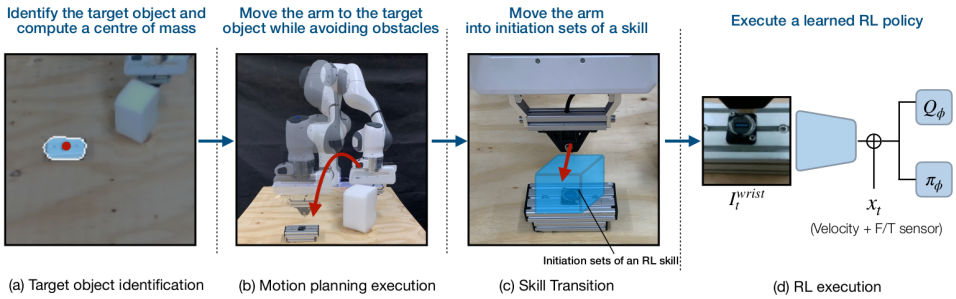


Figure 2: **Our framework architecture.** (a) We leverage an OCGM to re-identify a target object such that its object-centric representation matches one extracted from a single demonstration. The goal state is specified in the robot's reference frame using an external RGB-D camera with calibrated extrinsics. (b) Given the goal state acquired in (a), a motion planner generates a collision-free path to the goal. (c) A skill transition network guides the arm from the terminal state of the motion planning (MP) to the initiation set of the RL policy. (d) Given a wrist camera image $\boldsymbol{I}_t^{wrist}$ and robot's internal state $\boldsymbol{x}_t$, a learned RL policy executes the final interaction until task completion.

Our work leverages unsupervised OCGMs (Wu et al., 2021; Locatello et al., 2020; Lin et al., 2020) to find a target object for MP, negating the need for object-specific goal estimators. OCGMs learn structured representations of objects within complex scenes and provide a set of object-centric embeddings useful for instance matching. In contrast to goal specification methods that require human intervention or a large, object-specific datasets, i.e. template matching or object classifier, OCGMs hold the promise of versatile target object identification. While several prior works (Kirillov et al., 2023; Xie et al., 2021) introduce instance segmentation methods for unseen objects, these methods do not provide a dense description of objects suitable for instance matching. Specifically, this work leverages APEX (Wu et al., 2021) an unsupervised model trained on a wide distribution of simulated data to assist with generalisation to real-world environments.

3. Methodology

In this work, we present an efficient solution for solving insertion tasks in obstructed, real-world environments by leveraging an OCGM. We demonstrate our system on several insertion tasks as they require learning complex insertion skills and also require the identification of the target socket to complete the tasks (see Figure 1 for our task setup). We break our method down into pre-training, task-specific skill training and execution in the following subsections. The pre-training component of our method is only completed *once* and can be reused for all future insertion tasks. Task-specific skill training is required for each new insertion task and execution describes the process for autonomous task execution after training.

3.1. Pre-training

Pre-training is required for APEX (Wu et al., 2021), our choice of unsupervised OCGM, to achieve versatile one-shot target object acquisition, but it only needs to be done once as it is trained on a diverse synthetic dataset collected in simulation to encourage generalisation to a variety of real-world objects. APEX is formulated as a set of VAEs, and takes a video sequence $\boldsymbol{I}_{1:T}$ as input. Each

frame is decomposed into a set of latent representations for each discovered object j consisting of object location $\boldsymbol{z}_{t,j}^{where}$, appearance $\boldsymbol{z}_{t,j}^{what}$, and presence $z_{t,j}^{pre} \in [0, 1]$, where T is the number of frames in a video sequence. We train APEX on a synthetic dataset consisting of a set of trajectories in which a robot interacts with a diverse set of primitive shapes of differing colour and size. In order to successfully transfer APEX trained on synthetic data to the real world, we add a small amount of noise to the camera pose for each trajectory, leading to variations in the images. As a result, APEX is successfully applied to real-world scenes with similar background textures.

3.2. Task-Specific Skill Training

This section details the task-specific data and training required by our method. The data must be collected for each new task that the robot is taught, however, the supervised component only requires about 10 minutes to collect. First, a single demonstration, $\mathcal{D}^{goal} = \{(\boldsymbol{I}_t^{ext}, \boldsymbol{x}_t^{ee}), \dots\}$ consisting of a sequence of images $\boldsymbol{I}^{ext}$ from the third-person camera and robot end-effector positions $\boldsymbol{x}_t^{ee}$, of a successful task completion from anywhere within the robot's operational space is collected for goal specification using the pre-trained APEX. Additionally, 25 demonstrations, $D^{RL} = \{(\boldsymbol{I}_t^{wrist}, \boldsymbol{x}_t, \boldsymbol{a}_t), \dots\}^{|25|}$, of the insertion skill for efficient RL training, are collected from within a limited task space such that the connector is always within sight of the wrist camera, where $\boldsymbol{I}_t^{wrist}$, $\boldsymbol{x}_t$, $\boldsymbol{a}_t$ are wrist camera image, robot states including 3-dimensional Cartesian end-effector velocity and F/T sensor data, and action at time step t.

We employ FERM (Zhan et al., 2020) to train the RL policy π_θ along with a critic function Q_ϕ parameterised by π to complete the insertion task. This takes between 60 to 90 minutes to train on a desktop computer with an i7 processor and a Nvidia Titan X GPU. FERM is composed of Soft Actor Critic (Haarnoja et al., 2018), contrastive learning (Laskin et al., 2020b), and image augmentation (Laskin et al., 2020a). In addition to a gray-scale image from the wrist-mounted camera ($\boldsymbol{I}^{wrist[H\times W]}$ where H and W are 64 pixels), the policy takes as input the end-effector Cartesian velocity and F/T sensor data (see Figure 2 (d)) and outputs the desired 3-dimensional Cartesian end-effector velocity for the robot. Because the policy takes as input local information, it is able to generalise to any location in the robot's operational space. The RL policy is trained using a sparse reward $r_t = \mathbb{1}[\boldsymbol{s} \in S_g]$ where S_g is a set of goal states defined as the average termination state of the collected demonstrations within a $1cm$ tolerance. To accelerate training of the RL policy, we leverage the task demonstrations D^{RL} to initialise a replay buffer for guided exploration similar to FERM and train the policy and critic asynchronously, as inspired by prior work (Luo et al., 2021). We also limit the task space for this stage, such that the socket is always within sight of the wrist camera for training, improving sample efficiency and reducing the chance of unsafe interactions. The initial states of the policy are positioned above the socket, with a small random noise added to their positions, sampled from a uniform distribution between -0.02cm and 0.02cm.

We also introduce a skill transition network, inspired by prior work (Johns, 2021), for each new insertion task to improve the task success rate. The terminal state of the MP is not guaranteed to be within the feasible start states of the RL policy, defined as *the initiation set of the skill* (represented as a blue box in Figure 2 (c)), due to the error in estimating the MP goal state in 3D space, caused by errors in camera extrinsics and noisy depth estimation from the RGB-D camera. To mitigate this issue, a simple convolutional neural network (CNN) is trained on data collected in a self-supervised manner to predict the Cartesian offset required to move the end-effector from the terminal states of MP to the initiation set of the RL policy. The dataset is collected in less than 30 minutes by sampling

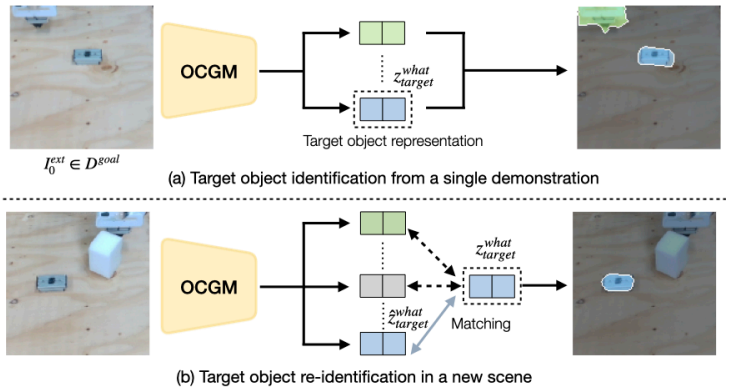


Figure 3: **Target object identification and re-identification using an OCGM.** (a) We leverage a pre-trained OCGM to extract an object-centric representation from a single task demonstration. The target object is identified in the demonstration as the object mask closest to the robot end-effector position at the end of the trajectory $\boldsymbol{x}^{ee}_T$ (see Eq 1). (b) Given a new scene, the OCGM is used to acquire object-centric representations of all objects and compare these with the already identified target object representation to re-identify the target object.

random Cartesian poses around the target object and recording the sequences of wrist camera images $\boldsymbol{I}^{wrist}_t$ and offsets between the current end-effector pose and an initial pose used for RL training. The collected data contains only local information conditioned on the wrist camera which allows the transition network to generalise to unseen target positions. Crucially, while an RL policy could be trained with a wider initial state distribution, this is well understood in literature (Yamada et al., 2020; Lee et al., 2020; Nair et al., 2018), to be sample inefficient. Instead, we train a skill transition network using a labelled dataset collected in a self-supervised fashion which is akin to Behaviour Cloning (Zhang et al., 2018).

3.3. Execution

The execution of the task can be completed from *anywhere* within the robot's operational space to any goal location. Execution follows four steps (see Figure 2) that are completed autonomously: (i) goal identification via OCGM, (ii) MP, (iii) skill transition network, and (iv) RL policy.

Goal Identification via OCGM As MP requires a goal pose to plan a collision-free path through the scene, we leverage the pretrained OCGM to identify the target object from the single demonstration D^{goal} and re-identify it in the current scene to specify the goal for MP. To identify the target object from D^{goal}, we first acquire a set of object-centric representations by encoding the first external camera image $\boldsymbol{I}^{ext}_0$ in the demonstration D^{goal} (see Figure 3 (a)). We determine the target object-centric representation $\boldsymbol{z}^{what}_{target}$ such that the object is present at the beginning of the trajectory, i.e. $p(z^{pre}_{0,j}) \geq 0.5$ and such that it is the closest to the robot end-effector position $\boldsymbol{x}^{ee}_T$ at the end of the demonstration D^{goal}. To calculate the 3D position of objects in the robot's reference frame, the centre of the object mask predicted by the OCGM is converted to Cartesian coordinates using the RGB-D camera's depth plane and the known camera extrinsics. The closest object to the robot

end-effector position at the end of the demonstration D^{goal} is calculated using L_2 distance:

$$target = \underset{j,j=1..N}{\arg\min} ||\boldsymbol{x}_T^{ee} - \boldsymbol{o}_j||_2 \tag{1}$$

where o_j is the 3D object positions in the robot reference frame and N is the number of objects discovered by the OCGM in the scene $\boldsymbol{I}_0^{ext}$. In order to re-identify the target object in the current scene (see Figure 3 (b)), we compare the target object-centric representation $\boldsymbol{z}_{target}^{what}$ with each object-centric representation $\hat{\boldsymbol{z}}_j^{what}$ discovered in the new scene (see Figure 3 (b)) using the L_2 distance and choose the object that has the most similar representation:

$$\hat{\boldsymbol{z}}_{target}^{what} = \underset{j=1..N}{\arg\min} ||\boldsymbol{z}_{target}^{what} - \hat{\boldsymbol{z}}_j^{what}||_2 \tag{2}$$

MP + Transition Policy + RL Policy Using the target object's pose $\boldsymbol{o}_{target}$ in the robot's reference frame, we use an RRT-connect motion planner to guide the robot's end-effector to the location of the target object (see Figure 2 (b)). To avoid collisions during the MP phase, an occupancy map, OctoMap (Hornung et al., 2013), is created using the point clouds captured by the calibrated external camera. After the execution of MP, we leverage the trained skill transition network to guide the arm into the initiation set of the skill (see Figure 2(c)) to maximise the outcomes of the RL policy. Finally, the learned RL policy completes the manipulation task.

4. Experiments

Our experiments are designed to answer the following guiding questions: (1) does the use of an OCGM achieve versatile and efficient target object identification for MP in the real world? (2) how well does our system perform insertion tasks in obstructed environments? (3) does a skill transition network increase task success rate?

4.1. Experimental Setup

Several insertion tasks, inspired by the NIST assembly boards (Kimble et al., 2020), are used within our experiments (see Figure 4). To verify the robustness of the target object identification using an OCGM, sockets and obstacles with different colours and sizes are used. In our experiments, we use a Franka Panda robot (7-DOF robot arm) and rigidly attach each connector to the robot's end-effector similar to prior work (Luo et al., 2021). Each phase of our framework uses different controllers: a joint position controller to follow a trajectory planned by the motion planner, a Cartesian pose controller for the skill transition network, and a Cartesian velocity impedance controller for the RL policy. For each evaluation trial, the socket, robot arm, and one or two obstacles are randomly placed in the robot's operational space.

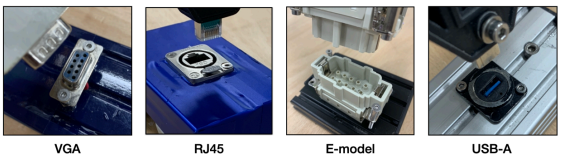


Figure 4: **Insertion tasks.** We evaluate our framework on four insertion tasks. Each socket is attached to a mount of varying size and colour to demonstrate the versatility and efficiency of our one-shot goal specification using an OCGM.

Given the pre-trained OCGM, for each insertion skill, our modular framework requires a total of 10 minutes of human-supervised demonstrations and a maximum of 130 minutes of unsupervised

	VGA		RJ45		E-model		USB-A		#Data	Intervention
Method	Accuracy	WSI	Accuracy	WSI	Accuracy	WSI	Accuracy	WSI		
Template matching	70.0%	54.7/81.9%	35.0%	22.1/50.5%	87.5%	73.9/94.5%	55.0%	39.8/69.3%	1	yes
Feature-based matching	40.0%	26.3/55.4%	87.5%	73.9/94.5%	7.5%	2.6/19.9%	77.5%	62.5/87.7%	1	yes
Object-specific classifier	80.0%	65.2/89.5%	**100.0%**	91.2/100.0%	87.5%	73.9/94.5%	75.0%	59.8/85.8%	2.5K	yes
OCGM identifier (Ours)	**82.5%**	68.0/91.3%	95.0%	83.5/98.6%	**95.0%**	83.5/98.6%	**92.5%**	80.1/97.4%	1	no

Table 1: **Accuracy of target object identification.** We evaluate our method and two baselines on 160 test scenes (40 scenes per connector), and report the accuracy, Lower Limit (LL) and Upper Limit (UL) of the Wilson score interval (WSI) with confidence interval of 95%. While template matching and object-specific classification requires human intervention, such as cropping a reference target object image and labelling training data, our OCGM identifier successfully identifies the target object from only a single demonstration without such human intervention.

training comprising of: up to 90 minutes for RL policy training and 40 minutes for data collection and training of the skill transition network.

4.2. Efficient and Versatile Target Object Identification

First, the OCGM identifier is evaluated against several baselines on 160 test scenes (40 for each of the four sockets) with the target locations hand-labelled with bounding boxes for quantitative comparison. During testing, if the intersection of union (IoU) between a ground truth and the returned bounding box from the tested algorithm is greater than 0.5, we count it as successful (Everingham et al., 2015).

We evaluate our proposed goal identification approach against three baselines. *Template matching* finds the target object in the current scene by calculating a correlation coefficient given a manually cropped target object reference image. *Feature-based matching* finds a pair of the best matched keypoints between the manually cropped target object reference image and the current scene using FLANN-based matching (Muja and Lowe, 2009) and SIFT descriptor (Lowe, 2004). *Object-specific classifier* trained on a dataset of manually cropped object images with binary labels, is queried with cropped images found using a region proposal method (Uijlings et al., 2013). Lastly, we evaluate our method by retrieving the minimum bounding box of the target object mask predicted by the OCGM.

Results. We report the accuracy of target object identification in Table 1. Our method achieves commensurate or better performance compared to other baselines, whilst not requiring human intervention or an object-specific dataset needing laborious manual data labelling. This result motivates the use of OCGMs for efficient goal acquisition for MP. The occlusion of objects that are sometimes considered as one object in APEX can lead to unsuccessful object-centric representation matching, resulting in a failure to identify the target. Also, if the object shape and colour look similar in an image, object-centric representation matching may fail. Template matching often struggles to find a target object with high confidence, potentially due to the complex scene composition and slanted third-person camera angle (see Fig. 1). Feature-based matching also shows a lower success rate for several objects due to a lack of distinguishing features, especially for small objects in a scene. The object-specific classifier, on the other hand, generally performs well because it is tailored to a single object, and could be further improved by collecting more data. However, such classifiers

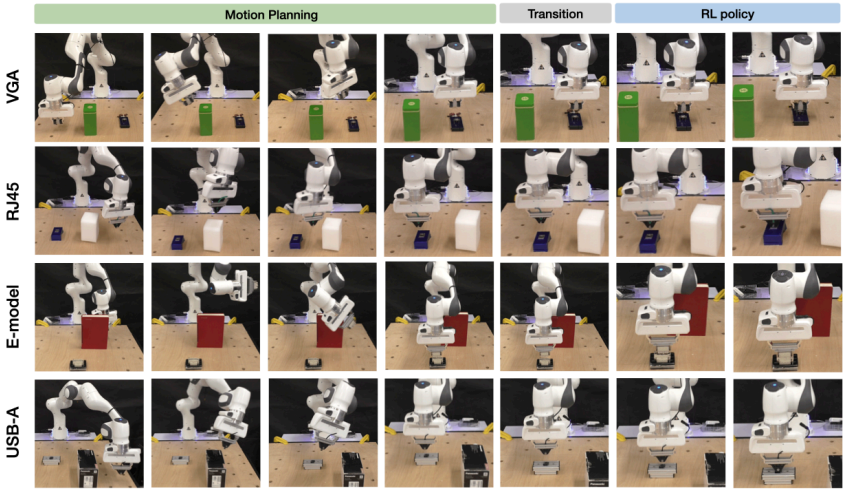


Figure 5: **Real-world industrial assembly tasks in obstructed environments.** The OCGM is used to specify a goal for MP, followed by a skill transition network and a learned RL policy. Our method successfully solves complex manipulation tasks with a high success rate.

require re-training on a new dataset manually labelled for each new object, limiting the versatility and efficiency of goal specification.

	VGA		RJ45		E-model		USB-A	
Method	Success	WSI	Success	WSI	Success	WSI	Success	WSI
SAC	0.0%	0.0/16.1%	0.0%	0.0/16.1%	0.0%	0.0/16.1%	0.0%	0.0/16.1%
MP + Demonstration Replay	3.3%	1.0/16.7%	0.0%	0.0%/16.1%	3.3%	1.0/16.7%	0.0%	0.0/16.1%
MP + BC	16.7%	7.3%/33.6%	16.7	7.3/33.6%	23.3%	11.8/40.9%	26.7%	14.2/44.5%
MP + Heuristic	10.0%	3.5/25.6%	16.7	7.3/33.6%	36.7%	21.9/54.5%	43.3%	27.4/60.8%
MP + RL w/o skill transition	73.3%	55.6/85.8%	46.7%	30.2/63.8%	80.0%	62.7/90.5%	70.0%	52.1/83.3%
MP + RL (our method)	**86.7%**	70.3/94.7%	**83.3%**	66.4/92.7%	**93.3%**	78.7/98.2%	**96.7%**	83.3/99.4%

Table 2: **Real-world assembly results.** We report the success rate, Lower Limit (LL) and Upper Limit (UL) of the Wilson score interval (WSI) with confidence interval of 95% over 30 trials. Our method outperforms, by a significant margin, all of the other methods including a state-of-the-art RL method and several comparable instantiations of our method.

4.3. Insertion Tasks in Obstructed Environments

We evaluate our proposed system on several insertion tasks in obstructed environments against a series of baselines composed of competing methods. All baselines that utilise MP make use of the OCGM for target object identification. For each task, we conduct 30 trials and report the success rate in Table 2. Figure 5 illustrates the execution of our method for each insertion task in the obstructed environments.

We compare the performance of our approach against a state-of-the-art RL algorithm and four comparable instantiations of our approach. *Soft Actor-Critic (SAC)* a state-of-the-art RL algorithm that predicts the desired Cartesian velocity from sparse rewards, trained with the same number of environmental interactions as our proposed method and similarly with 25 demonstrations preloaded

into the replay buffer, following FERM (Zhan et al., 2020). *MP+Demonstration Replay* substitutes replaying a single expert demonstration for the learned RL policy execution in our method, inspired by previous work (Johns, 2021). *MP+BC* replaces the learned RL policy in our method with Behaviour Cloning (BC) (Zhang et al., 2018), trained from 25 demonstrations. *MP+Heuristic* uses a manually designed heuristic policy (Luo et al., 2021) instead of the learned RL policy in our method to solve the task. Lastly, we evaluate our method without a skill transition network (*MP+RL w/o skill transition*) for comparison.

Results. As described in Table 2, our method (MP+RL) as outlined in Section 3 records the highest success rate for all tasks. The results for the *SAC* baseline show that it is unable to solve any of the tasks, likely because it requires a large number of samples to train the policies in the robot's operational space with obstructions. *MP+Demonstration Replay* is the most data-efficient method, however, it mostly fails to solve any of the tasks because it requires very accurate estimation of pose offsets for the demonstration replay to be successful. *MP+BC* is another efficient skill acquisition method because it does not require any additional interactions with the environments other than the given demonstrations to learn manipulation skills. However, due to the narrow state coverage, it struggles to solve the tasks. While *MP+Heuristic* is able to solve some insertion tasks, such as USB-A and E-model, almost one-third of the time, it fails to solve the tasks the majority of the time due to the need for accurate pose offset (the same reason for failure as *MP+Demonstration Replay*). While our method achieves high success rate over 4 industrial insertion tasks, the main failure case is caused by the misidentification of the target object by the OCGM. These failure modes can be readily eliminated by extended and/or augmenting the OCGM training.

Examining whether the transition network is useful for our system to solve complex manipulation tasks in obstructed environments (see Table 2), the results verify that using the skill transition network results in higher success rates than without the skill transition policy. Due to errors caused by the OCGM, camera extrinsics and estimation of the 3D goal poses, a terminal state of MP can often be outside of the initiation set of the learned RL skill. Therefore, by introducing the skill transition module to move the robot arm into the initiation set of the skill, we can mitigate these issues and achieve better performance.

5. Conclusion

In this work, we propose a modular system that leverages an OCGM for one-shot goal identification and re-identification as a vital component to combine MP and RL to solve complex manipulation tasks in obstructed environments. Specifically, the OCGM extracts a target object from only a single demonstration and re-identifies the object to determine a goal pose for MP without the need of fine-tuning on an object-specific dataset. The experimental results show that our method for goal specification using an OCGM achieves better versatility and comparable accuracy to other tested baselines. In addition, our method successfully solves real-world insertion tasks in obstructed environments from few demonstrations.

While the rotation around the z-axis of the sockets is consistent across all of the evaluation trials, we can readily extend our system to accommodate cases where the socket is rotated. We leave this extension to future work and anticipate overcoming the orientation misalignment by extending the skill transition network to additionally predict a z-axis displacement, correctly orientating the peg with respect to the socket. Any further small orientation errors could be overcome using an impedance controller and an RL policy trained with small perturbations in the z-axis orientation.

ACKNOWLEDGMENT

This work was supported by a UKRI/EPSRC Programme Grant [EP/V000748/1], we would also like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) (http://dx.doi.org/10.5281/zenodo.22558) and the SCAN facility in carrying out this work.

References

Nancy M Amato and Yan Wu. A randomized roadmap method for path and manipulation planning. In *IEEE International Conference on Robotics and Automation*, 1996.

A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Survey: Robot programming by demonstration. *Springer Handbook of Robotics*, pages 1371–1394, 2008.

Joao Carvalho, Dorothea Koert, Marek Daniv, and Jan Peters. Residual robot learning for object-centric probabilistic movement primitives, 2022.

Todor Davchev, Kevin Sebastian Luck, Michael Burke, Franziska Meier, Stefan Schaal, and Subramanian Ramamoorthy. Residual learning from demonstration: Adapting DMPs for contact-rich manipulation. *IEEE Robotics and Automation Letters*, 2022.

M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

Oliver Groth, Chia-Man Hung, Andrea Vedaldi, and Ingmar Posner. Goal-conditioned end-to-end visuomotor control for versatile skill primitives. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1319–1325, 2021.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018.

Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673, 2018.

Lydia Kavraki and Jean-Claude Latombe. Randomized preprocessing of configuration for fast path planning. In *IEEE International Conference on Robotics and Automation*, 1994.

Mohamed Khodeir, Ben Agro, and Florian Shkurti. Learning to search in task and motion planning with streams, 2021.

Kenneth Kimble, Joseph Falco, Elena Messina, Karl Van Wyk, Yu Sun, Mizuho Shibata, Wataru Uemura, and Yasuyoshi Yokokohji. Benchmarking protocols for evaluating small parts robotic assembly systems. (5), 2020.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

Cheng-Yu Kuo, Andreas Schaarschmidt, Yunduan Cui, Tamim Asfour, and Takamitsu Matsubara. Uncertainty-aware contact-safe model-based reinforcement learning. *IEEE Robotics and Automation Letters*, 6(2):3918–3925, 2021.

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020a.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *International Conference on Machine Learning, Vienna, Austria, PMLR 119*, 2020b.

Steven M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report TR 98-11, Computer Science Department, Iowa State University, 1998.

Minh-Tri Le, Chih-Hung G. Li, Shu-Mei Guo, and Jenn-Jier James Lien. Embedded-based object matching and robot arm control. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1296–1301, 2019.

Michelle A. Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks, 2018.

Michelle A Lee, Carlos Florensa, Jonathan Tremblay, Nathan Ratliff, Animesh Garg, Fabio Ramos, and Dieter Fox. Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning. *IEEE International Conference on Robotics and Automation*, 2020.

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*, 2020.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

Jianlan Luo, Oleg Sushkov, Rugile Pevceviciute, Wenzhao Lian, Chang Su, Mel Vecerik, Ning Ye, Stefan Schaal, and Jon Scholz. Robust multi-modal policies for industrial assembly via reinforcement learning and demonstrations: A large-scale study. *arXiv preprint arXiv:2103.11512*, 2021.

Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP (1)*, pages 331–340. INSTICC Press, 2009.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *IEEE international conference on robotics and automation (ICRA)*, pages 6292–6299, 2018.

Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3 (6):233–242, 1999. ISSN 1364-6613.

Oren Spector and Dotan Di Castro. Insertionnet – a scalable solution for insertion, 2021.

J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards, 2017.

Mel Vecerik, Oleg Sushkov, David Barker, Thomas Rothörl, Todd Hester, and Jon Scholz. A practical approach to insertion with variable socket position using deep reinforcement learning, 2018.

Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. 2022.

Yizhe Wu, Oiwi Parker Jones, Martin Engelcke, and Ingmar Posner. Apex: Unsupervised, object-centric scene segmentation and tracking for robot manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3375–3382. IEEE, 2021.

Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. Unseen object instance segmentation for robotic environments, 2021.

Jun Yamada, Youngwoon Lee, Gautam Salhotra, Karl Pertsch, Max Pflueger, Gaurav S. Sukhatme, Joseph J. Lim, and Peter Englert. Motion planner augmented reinforcement learning for obstructed environments. In *Conference on Robot Learning*, 2020.

Albert Zhan, Philip Zhao, Lerrel Pinto, Pieter Abbeel, and Michael Laskin. A framework for efficient robotic manipulation. *arXiv:2012.07975*, 2020.

Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *IEEE International Conference on Robotics and Automation*, pages 5628–5635, 2018.

Appendix A. Pre-training Details

A.1. APEX Training Details and Hyperparameters

We collect a synthetic dataset consisting of $30K$ trajectories each of which has 10 to 15 frames simulated using RLBench James et al. (2020). In each trajectory, a robot randomly pushes objects on a table. To apply the pre-trained APEX to the real world, we add a small amount of noise, ranging from 1cm to -1cm, to the Cartesion camera position in the simulation so that the background texture changes slightly for each trajectory and prevents APEX from overfitting to the specific background texture. Table 3 shows hyperparameters of APEX. For further details, see prior work Wu et al. (2021).

Table 3: APEX hyperparameter

Parameter	Value
Observation Rendering	(128, 128), RGB
Optimizer	Adam
Learning rate	1e-4
horizon	10
Image size	128
Foreground std	0.11
Background std	0.04
KL divergence for z_{what}	3e-4
KL divergence for z_{where}	15
KL divergence for z_{pre} (discovery)	32
KL divergence for z_{pre} (tracking)	1
Background reconstruction loss weight	10
foreground reconstruction loss weight	1

Appendix B. In-Situ Training Details

B.1. RL Training Details and Hyperparameters

We define a sparse reward function such that a policy receives 1 when the L2 distance between robot's end-effector pose and the pre-defined target pose is within a small tolerance of 0.8cm for VGA and 1cm for the other insertion tasks. The policy and critic take as input a wrist camera image, Cartesian velocity, and F/T sensor values of the end-effector. The critic also takes as input an action sampled from the policy. The architecture of the policy and the critic includes a CNN for processing the wrist camera image followed by a concatenation with state information. The concatenated features are fed to the policy and critic feedforward neural networks with the outputs being actions and values respectively. Similar to FERM Zhan et al. (2020), we apply random crops and brightness changes to the image observations to acquire a robust policy efficiently. The policy runs at 10Hz and we find that this is sufficient for completing the complex insertion tasks. Table 4 shows hyperparameters for the RL policy training.

Table 4: FERM hyperparameter

Parameter	Value
Optimizer	Adam
Task horizon	50
Learning rate	1e-3
Discount factor (γ)	0.99
Replay buffer size	10^4
Latent dimension	50
Convolution filters	$[8, 16, 32, 64]$
Convolution strides	$[2, 2, 2, 2]$
Convolution filter size	3
Hidden Units (MLP)	[128]
Nonlinearity	Leaky ReLU
Target smoothing coefficient (τ)	0.005
Target update interval	2
Actor update interval	2
Network update per environment step	10

B.2. Transition Network Details and Hyperparameters

The dataset for the transition network is collected by uniformally sampling a Cartesian offset from within ± 5cm of the initial position used for RL training. The robot takes 15 linear steps between the initial position used for RL training and the sampled offset, recording both the wrist camera image and the current offset. After collecting 100 trajectories, in less than 30 minutes, the skill transition network can be trained.

B.3. Baseline Details - SAC

To train a policy with SAC Haarnoja et al. (2018) from scratch using FERM Zhan et al. (2020), we randomly place a socket and one or two obstacles from a set of four obstacles on a table. Then, we define a reward function for collision avoidance in addition to a sparse reward for solving the insertion task within the entire operational space of the robot:

$$R = I[s \in S_g] - 0.005 \cdot I[s \in S_{coll}] \tag{3}$$

where S_g and S_{coll} are a set of goal states and states involving collisions and I is an indicator function. Furthermore, the policy also takes as input an image from the external camera in addition to a wrist camera image. We use the same hyperparameters described in Table 4 except that the task horizon of 250 is used for the baseline RL policy to solve the tasks in the entire workspace of the robot.

B.4. Baseline Details - MP + Heuristic

We design a heuristic policy for insertion tasks, inspired by previous work Luo et al. (2021) that utilises the F/T sensor data. The heuristic policy first moves the end-effector downwards until the arm contacts an object. Then, the arm follows an outwards spiral pattern on a 2D plane to find the socket hole. Finally, after the connector is aligned with the hole, the arm moves downwards again with a small circular movement in the X-Y plane to insert the connector into the socket.

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).**

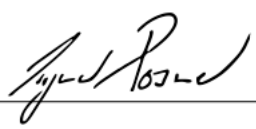
Title of Paper	Efficient Skill Acquisition for Insertion Tasks in Obstructed Environments
Publication Status	■Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Jun Yamada, Jack Collins, Ingmar Posner Published at Learning for Dynamics and Control Conference (L4DC), 2024

Student Confirmation

Student Name:	Jun Yamada		
Contribution to the Paper	- Proposed the research idea - Created and developed methodologies - Ran all experiments - Created all figures - Paper writing		
Signature		Date	19/09/2025

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: Prof. Ingmar Posner		
Supervisor comments *Jun made a substantial contribution to the publication. The description above is accurate.*		
Signature 	Date	22/09/2025

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

3.1 Limitations and Future Work

The proposed method efficiently solves contact-rich insertion tasks in obstructed environments with a high success rate by combining motion planning, an object-centric generative model, and an RL policy trained from only a few human demonstrations and autonomous interaction with the environment. Prior work either restricts the robot to a limited task space suitable for RL or relies on object-specific goal predictors to combine motion planning and an RL policy. On the other hand, this work enables operation across the entire workspace of the manipulator and achieves versatile one-shot target object identification by leveraging APEX [35], an object-centric generative model, to combine motion planning and a learnt RL policy. However, similar to several prior works [104, 184], the proposed system still requires a rigid attachment between the connector and the robot's end-effector. To address this limitation, a promising solution is to leverage more sample-efficient RL methods [108, 109] that can learn contact-rich manipulation tasks using a small number of demonstrations, with occasional human intervention to correct policy errors during the exploration phase. Alternatively, the task can be decomposed into grasping and insertion subtasks, where motion planning is used to reach the target object for each subtask, while a learning-based policy handles the contact-rich interactions with the object.

Although the successful sim-to-real transfer of APEX demonstrates effective one-shot object identification and re-identification, most object-centric generative models remain limited in their ability to represent objects with complex geometries, which are largely excluded from the current experiments. A recent approach, SlotDiffusion [155], offers a promising direction for addressing these limitations by learning richer representations of objects with complex geometries.

Another limitation of object-centric generative models is that they often produce coarse segmentation masks of the target object. To address this, a promising direction involves integrating them with semantic segmentation models such as Segment Anything Model (SAM) [185, 186], which can provide more accurate

masks. Although SAM typically relies on user-provided prompts, either textual or pixel-based, object-centric generative models can automatically identify the target object and sample relevant pixels for it to use as queries for SAM. This integration has the potential to enable more accurate and fully automated semantic segmentation within robotic manipulation systems.

Furthermore, perceptual robustness under realistic industrial conditions remains an open challenge. Highly reflective metal components under strong illumination, partial occlusions, or very small objects may degrade object identification, particularly when relying on a fixed third-person camera. In such cases, active exploration may be required to obtain more informative observations for OCGM, for example, by repositioning a wrist-mounted camera to inspect the scene from a closer range or more favourable viewpoints, rather than relying solely on a static external camera.

Another challenge arises during execution when the target object's visual observability is lost. The current formulation assumes that sufficient visual information about the target object remains within the field of view of the wrist-mounted camera at the beginning of the transition phase. If the target object moves during planning or due to perception error and falls outside the camera's view, the transition network cannot reliably predict corrective offsets. Addressing this issue may require strategies to maintain the target object's visibility, for example, by leveraging third-person camera observations in addition to the wrist-mounted camera to re-identify the target object.

Lastly, while the learnt RL policy effectively solves contact-rich insertion tasks in cluttered tabletop environments, extending it to more complex scenarios, such as manipulating objects on a shelf, remains challenging due to potential collisions with the environment. Moreover, the RL policy must be trained separately for each plug–connector pair and does not generalise to novel objects. These limitations are addressed in Chapter 4, which introduces a framework for safe and generalisable closed-loop visual grasping in cluttered environments, including both tabletop and shelf settings.

4

Grasp-MPC: Closed-Loop Visual Grasping via Value-Guided Model Predictive Control

Inspired by the unification of motion planning and learning-based policy in Chapter 3, this chapter extends the framework by replacing a learnt RL policy with model predictive control (MPC). The RL policy used in the system presented in Chapter 3 struggles in constrained environments, such as grasping an object on a shelf where collisions with the surroundings are likely. In contrast, this chapter introduces *Grasp-MPC*, which integrates MPC with a learnt value function to enable generalisable, reactive, and safe grasp execution in cluttered settings, including both tabletop and shelf environments.

Grasping is a fundamental manipulation skill that allows robots to perform a wide range of downstream tasks, including insertion and assembly. Current state-of-the-art grasping approaches are broadly categorised into open-loop and closed-loop methods, each with distinct strengths and limitations.

Open-loop approaches [37–40] typically rely on a 6-DoF grasp pose prediction model trained on large-scale synthetic datasets with annotated grasp poses [41]. These methods predict a desired grasp pose for the target object, followed by motion planning to guide the robot arm toward the desired pose. While effective for grasping novel objects in diverse and cluttered environments, they remain susceptible to prediction errors and are unable to handle moving objects due to the lack of feedback during motion. To mitigate this limitation, two complementary

strategies can be considered. One approach is to employ an object tracking model, such as FoundationPose [187], to maintain continuous pose estimation throughout execution. However, such methods typically require access to the object mesh or multiple reference views, which limit their applicability. Alternatively, the desired grasp pose can be iteratively updated using the grasp prediction models; however, successive predictions may deviate significantly from previous targets due to the stochasticity of the models or change in visual observations, introducing abrupt command changes and causing unstable robot execution.

In contrast, closed-loop approaches, commonly based on RL [42–44] or IL [45, 46], enable reactive control through continuous feedback. However, these methods often remain limited to simplified scenarios, such as grasping isolated objects on clean tabletops, and exhibit poor generalisation.

To address these limitations, this chapter introduces *Grasp-MPC*, a framework that combines the strengths of open-loop and closed-loop control for 6-DoF grasping of novel objects in cluttered environments (see Chapter 4, Fig. 1). *Grasp-MPC* leverages an off-the-shelf grasp pose prediction model to generate a (potentially noisy) pre-grasp pose. A motion planner then guides the robot toward the predicted pre-grasp pose, inspired by the system in Chapter 3. Crucially, instead of executing this open-loop linear motion from the predicted pre-grasp to grasp poses, *Grasp-MPC* employs MPC to execute closed-loop grasping as well as incorporating visual feedback.

In particular, *Grasp-MPC* incorporates a learnt value function as the task cost within the MPC optimisation to leverage the strengths of model-based and data-driven approaches. This value function is trained on a large-scale dataset of synthetic grasp trajectories, including both successful and failed attempts, enabling it to learn discriminative features that are predictive of grasp success. This approach avoids the limitation of geometric costs (e.g., distance to the predicted grasp pose), which are sensitive to the prediction error of the grasp pose prediction model, similar to the open-loop approaches.

By using MPC for action selection, *Grasp-MPC* enables real-time, feedback-driven execution that respects constraints such as collision avoidance and trajectory smoothness (e.g., minimum jerk), allowing the system to adapt dynamically to environmental changes. This capability makes it particularly well-suited for reactive grasping of moving objects, as well as for safe and reliable grasp execution in cluttered environments. To enable real-world deployment, *Grasp-MPC* utilises Model Predictive Path Integral (MPPI) [12] implemented in CuRobo [78], a GPU-accelerated MPC framework.

Through comprehensive simulated experiments on the publicly available Fetch-Bench tasks [46], *Grasp-MPC* achieves performance comparable to an open-loop approach when provided with ground-truth grasp annotations, which serve as an oracle baseline. Under noisy grasp pose inputs, *Grasp-MPC* maintains a high grasp success rate and outperforms all baselines, including state-of-the-art IL and offline RL methods, while the performance of the open-loop baseline degrades significantly. Using grasp poses predicted by M2T2 [38], *Grasp-MPC* continues to surpass all baselines in terms of grasp success. In real-world experiments with a UR10 robot, *Grasp-MPC* is evaluated against the open-loop baseline across three settings: a clean tabletop, a cluttered tabletop, and a cluttered shelf environment. *Grasp-MPC* achieves a 74.4% average success rate across these scenarios, significantly outperforming the open-loop baseline at 41.1%, thereby demonstrating its effectiveness in practical applications. Additionally, *Grasp-MPC* is successfully deployed in dynamic object grasping tasks, where it tracks and grasps moving objects, a capability inherently unsupported by the open-loop baseline.

In this chapter, in addition to leveraging motion planning to avoid obstacles and navigate the robot to the target object, the unification of MPC and learning-based approaches enables safe and generalisable closed-loop visual grasping in cluttered environments. In summary, this chapter presents the following contributions:

1. *Grasp-MPC*, a safe, generalisable, closed-loop visual grasping framework capable of handling novel objects in cluttered environments.

2. Integration of model-based control and data-driven approaches by incorporating a learnt grasp value function as a cost into MPC, enabling reactive and constraint-aware grasp execution in dynamic, cluttered settings.

3. Collection of a large-scale synthetic dataset comprising over 2 million grasp trajectories, 115 million states, and $8,515$ unique objects from the Objaverse dataset [188].

4. Comprehensive empirical validation through simulated and real-world experiments, demonstrating that *Grasp-MPC* significantly outperforms both open-loop and closed-loop baselines in terms of grasp success.

Jun Yamada, Adithyavairavan Murali, Ajay Mandlekar, Clemens Eppner, Ingmar Posner, Balakumar Sundaralingam (2026). Grasp-MPC: Closed-Loop Visual Grasping via Value-Guided Model Predictive Control, *IEEE International Conference on Robotics and Automation (ICRA)*

Grasp-MPC: Closed-Loop Visual Grasping via Value-Guided Model Predictive Control

Jun Yamada[1,2] Adithyavairavan Murali[2] Ajay Mandlekar[2]
Clemens Eppner[2] Ingmar Posner[1] Balakumar Sundaralingam[2]

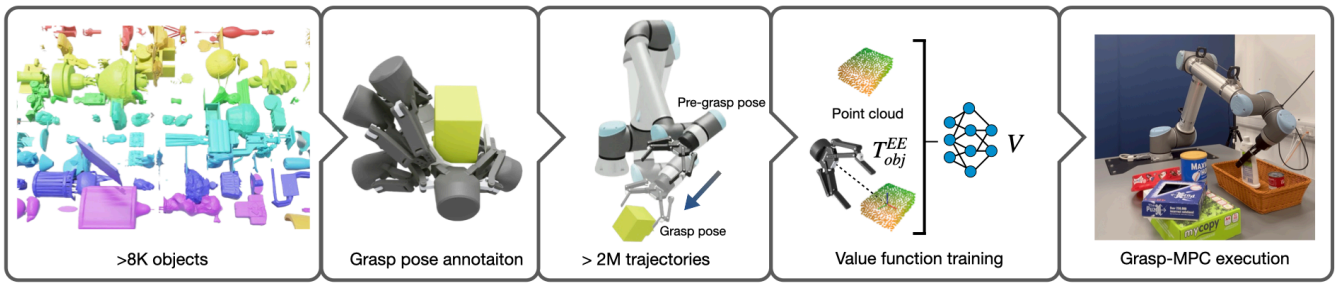


Fig. 1: *Grasp-MPC* overview. A large-scale synthetic grasp trajectory dataset is generated in simulation using a motion planner, collecting only trajectories between pre-grasp and ground-truth grasp poses across 8K Objaverse objects. A value function is trained using a sparse cost label given the target object's point cloud and the end-effector pose. The learned value function is used in an MPC framework, enabling robust and safe grasping of novel objects in cluttered environments.

***Abstract*—Grasping of diverse objects in unstructured environments remains a significant challenge. Open-loop grasping methods, effective in controlled settings, struggle in cluttered environments. Grasp prediction errors and object pose changes during grasping are the main causes of failure. In contrast, closed-loop methods address these challenges in simplified settings (e.g., single object on a table) on a limited set of objects, with no path to generalization. We propose *Grasp-MPC*, a closed-loop 6-DoF vision-based grasping policy designed for robust and reactive grasping of novel objects in cluttered environments. *Grasp-MPC* incorporates a value function, trained on visual observations from a large-scale synthetic dataset of 2 million grasp trajectories that include successful and failed attempts. We deploy this learned value function in an MPC framework in combination with other cost terms that encourage collision avoidance and smooth execution. We evaluate *Grasp-MPC* on FetchBench and real-world settings across diverse environments. *Grasp-MPC* improves grasp success rates by up to 32.6% in simulation and 33.3% in real-world noisy conditions, outperforming open-loop, diffusion policy, transformer policy, and IQL approaches. Videos and more at http://grasp-mpc.github.io.**

I. Introduction

Grasping is a foundational capability in robotics, serving as a prerequisite for physical interaction and subsequent complex manipulation tasks. However, despite decades of research, grasping remains unsolved, particularly when dealing with novel objects in cluttered scenes. State-of-the-art grasping methods can broadly be categorized into open-loop and closed-loop approaches, but both fall short of meeting the requirements of robust grasping in unstructured environments. Open-loop methods, which rely on grasp pose prediction models [1], [2], [3], [4], use motion planning to reach predicted grasp poses. Although demonstrating notable performance in grasping novel objects, these approaches are inherently unable to incorporate real-time feedback to adjust their goals, making them sensitive to grasp pose prediction errors and changes in object pose. On the other hand, closed-loop policies, including those based on reinforcement learning (RL) and imitation learning (IL), address some of these shortcomings by incorporating feedback during execution. However, prior approaches [5], [6], [7], [8] are often limited to simplified settings, such as tabletop environments, and exhibit poor generalization to novel objects, primarily due to the absence of large-scale grasp trajectory datasets for diverse objects. More importantly, safety considerations, such as collision avoidance in cluttered scenes, remain largely unaddressed during development.

To address these, we introduce *Grasp-MPC*, a framework that combines the strengths of open-loop and closed-loop methods for 6-DoF grasping in cluttered, novel-object settings. *Grasp-MPC* unifies model- and data-driven approaches by leveraging model predictive control (MPC) with a value function learned from data as a task cost function. *Grasp-MPC* uses a grasp prediction model and motion planning, similar to open-loop methods, to reach (noisy) pre-grasp poses. It addresses prediction errors and object pose changes by employing MPC for closed-loop execution, enabling real-time feedback while enforcing constraints such as collision avoidance and minimum jerk.

Furthermore, *Grasp-MPC* addresses a key challenge in applying MPC to grasping: designing a cost function that meaningfully captures grasp success. Traditional geometric cost functions based on distances to predicted grasp poses are sensitive to prediction errors and fail to exploit MPC's full closed-loop capability. To overcome this, *Grasp-MPC* leverages a vision-based value function, trained on large-scale

Work done during internship at NVIDIA.
[1]Applied AI Lab, Oxford Robotics Institute, University of Oxford
[2]NVIDIA, USA
Correspondence to: jyamada@robots.ox.ac.uk

synthetic grasping trajectories with Objaverse objects [9] using visual observations and sparse success labels (Figure 1). This value function predicts the likelihood of grasp success and serves as a task cost in MPC, guiding the robot to explore the state space toward successful grasps.

To this end, this work makes the following contributions:

1) *Grasp-MPC*, a safe closed-loop visual grasping policy for novel objects in cluttered environments.
2) *Grasp-MPC* unifies model-based control and data-driven approach by integrating a learned grasp value function into an MPC framework, enabling reactive, constraint-aware grasp execution in dynamic and cluttered environments.
3) A large-scale synthetic grasp trajectory dataset comprising over 2M trajectories, 115M states, and 8515 unique Objaverse objects, supporting scalable learning of a generalizable value function.
4) Extensive real-world and simulated evaluations show that *Grasp-MPC* significantly outperforms SOTA open-loop and closed-loop methods.

II. Related Works

Grasping. A significant body of research has focused on learning grasp prediction models [1], [3], [2], [10], [11], typically paired with motion planning for execution. These methods often rely on synthetic datasets with grasp annotations [12], enabling scalable training in simulation. While effective for novel objects, open-loop approaches suffer from prediction errors and lack feedback integration, limiting real-world robustness. In contrast, closed-loop policies learned via RL [5], [6], [7], [8] and IL [13], [14] address these issues, but often face challenges with sample inefficiency, limited generalization, and perform only in a clean tabletop.

Prior work [15], [16], [14] combines motion planning with learned policies for obstacle-aware manipulation. FetchBench [14] uses a Transformer-based IL policy [17] with motion planning for grasping in clutter, but performance is limited by dataset size and diversity. In contrast, we generate a large-scale grasping trajectory dataset, $100\times$ larger than [14], enabling better generalization.

Model Predictive Control for Robotic Manipulation. Model Predictive Control (MPC)[18], [19], [20] is a powerful framework for robotic control, enabling closed-loop optimization using real-time feedback. However, applying MPC to grasping is challenging, as it requires a cost function that captures the nuances of grasp success [21]. Even with a target grasp pose from a prediction model, MPC remains vulnerable to errors, much like open-loop methods. Recent work integrates learning with MPC to tackle challenges by learning dynamics [22], [23], [24], [25], cost functions [26], [27], [25], [28], [21], and sampling distributions [29], [30]. Chen et al. [21] propose a distance-based value function using a predicted grasp pose, but this overlooks key factors for grasp success, leading to suboptimal results. CV-MPC [26] learns value ensembles from few demonstrations but rely on low-dimensional states, limiting generalization. In contrast, *Grasp-MPC* learns a value function from large-scale synthetic trajectories using point cloud observations and sparse success/failure labels, addressing these limitations.

Offline RL Offline RL [31], [32], [33] enables policy learning using offline datasets without environment interaction, leveraging both successful and failure trajectories to improve policies. However, policy extraction often represents a bottleneck in the learning process rather than value function estimation, as discussed in prior work [34]. On the other hand, *Grasp-MPC* does not need to extract the policy from the learned value function because *Grasp-MPC* uses MPC as a policy that can explore and exploit states to minimize the learnt cost represented by the value function.

III. Preliminaries

Problem Formulation. We formulate grasping as a Partially Observable Markov Decision Process (POMDP). A trajectory is defined as $\tau = (\mathbf{x}_t, \mathbf{a}_t, c_t, \mathbf{x}_{t+1}, \mathbf{a}_{t+1}, c_{t+1}, \dots)$, where $\mathbf{x} \in \mathcal{X}$ are observations, $\mathbf{a} \in \mathcal{A}$ actions, and $c \in \mathcal{C}$ costs. The offline dataset contains N trajectories $\{\tau^i\}_{i=1}^N$, comprising both successes and failures. The objective is to minimize the discounted cumulative cost $J(\tau) = \sum t' = t^\infty \gamma^{t-t'} c(\mathbf{x}_t, \mathbf{a}_t)$, with discount factor γ.

Dynamics Model for Model Predictive Control. MPC samples action sequences and plans future states to select the next action that minimizes cost in real time, given a dynamics model. In this work, MPC optimizes for joint accelerations with Euler integration as the dynamics model. We assume that the real robot can track the generated joint position, velocity and acceleration targets accurately using a low-level controller, for example, an inverse dynamics controller, similar to prior work [35]. The environment is not explicitly modelled; thus, its dynamics, including those of objects, are unknown a priori.

IV. Approach

Grasp-MPC leverages a large-scale grasp trajectory dataset generated in a simulation (Section IV-A) to train a value function (Section IV-B). Then, the learned value function serves as a cost function within MPC (Section IV-C), enabling robust and safe closed-loop grasping that generalizes to novel objects. At deployment, *Grasp-MPC* is integrated with a grasp pose prediction model and motion planning to operate in cluttered environments (Section IV-D). Lastly, the implementation details of the value function and MPC are described in Section IV-E.

A. Data Generation

A diverse set of grasp trajectories is generated using 8515 Objaverse objects [9] (see Figure 1). *Grasp-MPC* operates from pre-grasp poses, estimated using an off-the-shelf grasp pose prediction model (Figure 2 (1)), which provides a rough estimate of viable grasp poses. Thus, each grasping trajectory is generated to move from a pre-grasp pose to the corresponding ground-truth grasp pose. The grasp pose annotations are from the GraspGen dataset [36], where candidates are generated via antipodal sampling, similar to ACRONYM [12]. For

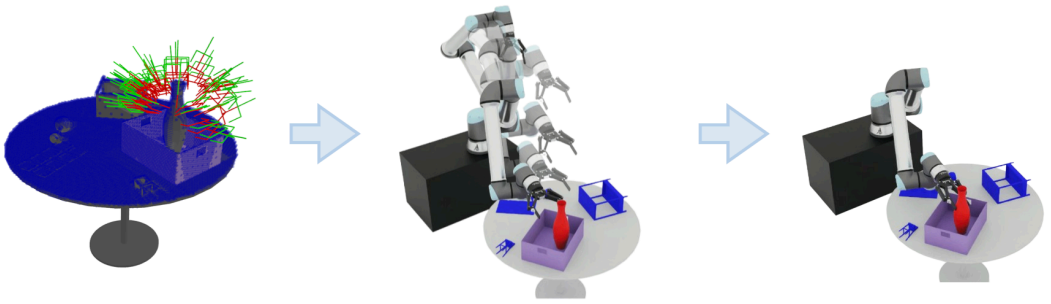


Fig. 2: *Grasp-MPC* Pipeline. *Grasp-MPC* seamlessly integrates off-the-shelf grasp prediction and motion planning with an MPC that incorporates a learned grasp value function, enabling grasping in cluttered scenes. The pipeline involves: (1) predicting grasp and pre-grasp poses using a fixed offset and filtering out in-collision poses via IK; (2) planning a trajectory to a collision-free pre-grasp pose; and (3) executing actions from the pre-grasp to grasp the object.

each object, 2K grasp transformations are uniformly sampled around the mesh. All sampled poses are evaluated in Isaac Sim to determine physical feasibility and are labeled accordingly as feasible or infeasible. Pre-grasp poses are derived by applying a fixed 15cm offset from each annotated grasp pose. To increase data coverage, we add random translation noise sampled from $\mathcal{U}(-0.04\text{cm}, 0.04\text{cm})$ and orientation noise from $\mathcal{U}(-0.04\pi, 0.04\pi)$.

We generate motions from the perturbed pre-grasp poses to the grasp poses using motion planning with CuRobo [37]. Trajectories that successfully reach physically feasible grasp poses are labelled as successful, whereas those planned toward infeasible grasp poses are labelled as failures. Incorporating both successful and failed trajectories allows the value function to learn grasp success likelihoods across the entire distribution of candidate poses (see Section IV-B). We do not validate these trajectories with simulation to accelerate data collection. Up to 256 trajectories are collected per object, with early termination if motion planning repeatedly fails. Each sample includes object poses T^{obj}_{world} and end-effector poses T^{EE}_{world}. In total, we collect $2,105,296$ trajectories ($115M$ data points), averaging 55 steps per trajectory, with lengths ranging from 31 to 233 steps. 70.2% of the collected trajectories reach a successful grasp, and the remaining trajectories are labeled as failures.

B. Value Function Training for Grasping

Grasp-MPC uses a value function as a cost for guiding MPC in grasping. The value function takes as input $\mathbf{x}$ consisting of a segmented object point cloud and the end-effector pose relative to the point cloud centroid, T^{EE}_{obj}. To standardize inputs, we center the point cloud by subtracting its mean. This setup allows *Grasp-MPC* to generalize across the workspace using only local information. The collected trajectories are labeled with sparse costs, with terminal and near-terminal states in successful grasp trajectories labeled as 0, and all others as 1. In particular, the cost c_t as timestep t is defined as:

$$c_t = \begin{cases} 0 & |q_{goal,i} - q_{t,i}| \leq 5e^{-3},\ \forall i,\ \textit{and}\ \mathbb{1}_{feasible} = 1, \\ 1 & \text{Otherwise} \end{cases} \tag{1}$$

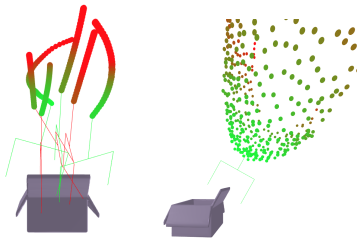


Fig. 3: Visualization of the learned value function. (a) Costs along trajectories for feasible (green) and infeasible (red) grasps, with higher final costs for infeasible poses. (b) Estimated values around a target grasp pose by varying x and y while fixing orientation; red indicates higher values and green lower values.

where $q_{t,i}$ and $q_{goal,i}$ are the i-th joint positions at time t and the goal, respectively, and $\mathbb{1}_{feasible}$ indicates whether the trajectory corresponds to a feasible grasp.

A value function $V(\mathbf{x}_t)$ is then trained to approximate the expected cost-to-go, defined as $V(\mathbf{x}_t) = \mathbb{E}_\tau[J(\tau)]$. The value function is trained using the Bellman error objective [38]:

$$y_t = c_t + \gamma V_{\phi'}(\mathbf{x}_{t+1}) \tag{2}$$

$$\ell(\phi; \mathbf{x}_t, c, \mathbf{x}_{t+1}) = \big(y_t - V_\phi(\mathbf{x}_t)\big)^2 \tag{3}$$

$$\phi^* = \arg\min_\phi \ \mathbb{E}_{(\mathbf{x}_t, c, \mathbf{x}_{t+1})} \left[\ell(\phi; \mathbf{x}_t, c, \mathbf{x}_{t+1})\right] \tag{4}$$

where y_t is the one-step target consisting of the immediate cost c_t and the discounted value of the next state $V_{\phi'}(\mathbf{x}_{t+1})$ from the target value function with exponential moving average of parameters ϕ. We set the discount factor to $\gamma = 0.99$, and the exponential moving average uses a rate of 5×10^{-3}. Figure 3 presents a visualization of the learned value function. The estimated value (cost) is lower (depicted in light green) around the target grasp pose, facilitating MPC in guiding the robot toward a successful grasp pose.

C. Integrating a Value Function as a Grasp Cost within MPC

Grasp-MPC uses learned value function as costs to guide MPC in minimizing grasping cost during online deployment.

The value function approximates the expected cost-to-go, which are integrated into the MPC objective to select control inputs. However, MPC may sample out-of-distribution actions, leading to unreliable cost estimates and reduced performance. To mitigate this, prior work [26], [39], [33] constrains MPC using pessimistic upper bounds derived from ensembles to avoid unsupported states. In our setting, ensembles do not yield performance improvements, as the large-scale synthetic data already covers a wide distribution of states (see Section V-D). Thus, in this work, the following objective is employed:

$$C_{grasp}(\mathbf{x}_{h\in H}) = \Sigma_{t'=t}^{t+H} \gamma^{t'-t} V_\theta(\mathbf{x}_{t'}) \tag{5}$$

We use Model Predictive Path Integral (MPPI) [40], a sampling-based MPC, implemented in CuRobo [37] to achieve real-time control. At each control step, MPPI samples N action sequences from a Gaussian distribution, rolls them out in parallel, evaluates the trajectories under a cost function, and computes an importance-weighted update to refine the control sequence. The first action of the optimized sequence is executed, and the horizon is shifted forward in a receding-horizon manner. *Grasp-MPC* augments standard MPC costs (e.g., minimum jerk, collision) with a value-based grasp cost. In particular, we integrate the grasp cost into CuRobo, and define the final cost as:

$$C_{\textit{Grasp-MPC}} = C_{curobo} + \omega C_{grasp} \tag{6}$$

where C_{curobo} is a set of CuRobo's default costs. ω is a weight for the pessimistic cost function, which we set to 1000. The cost term C_{curobo} in CuRobo consists of three main constraints: (1) joint limits on position, velocity, acceleration, and jerk; (2) self-collision avoidance; and (3) robot-world collision avoidance.

D. Grasp-MPC Deployment

At deployment, *Grasp-MPC* is combined with an open-loop grasp pose prediction model [1] and motion planner [37]. The grasp prediction model generates grasp poses for the target object. Pre-grasp poses are obtained by moving a negative distance along the gripper's approach vector. For real-world experiments, the standard 10cm offset distance [1], [41] is used. For simulation experiments in Fetchbench [14], 6cm offset is used, reduced from the standard 10cm due to insufficient clearance in many benchmark scenes, but increased from the original paper's 4cm, which is too restrictive. Feasible, collision-free pre-grasp poses are verified via a goalset motion plan, and the robot moves to the selected pre-grasp pose using the motion planner. Once positioned, *Grasp-MPC* grasps the object with feedback of the robot state, segmented object pointcloud, and the world signed distance field.

E. Implementation

While *Grasp-MPC* is compatible with any sampling-based MPC library, we specifically employ MPPI [40] implemented in CuRobo[37], a GPU-accelerated MPC framework, to enable fast and efficient real-time control. The value function consists of a PointNet++ [42] for point cloud input and an MLP for proprioception. Their outputs are concatenated and passed to an MLP head to output a value. A *softplus* activation ensures positive value predictions. Training uses both full and partial point clouds, with partial views rendered from randomly placed virtual cameras. Gaussian noise is added for robustness to real-world noise. The value function is trained with a mini-batch size of 1536, where 32 distinct object point clouds are sampled and 48 states are sampled for each object point cloud. The training procedure was conducted on a single RTX 4090 GPU for six days, using a learning rate of 1×10^{-4}.

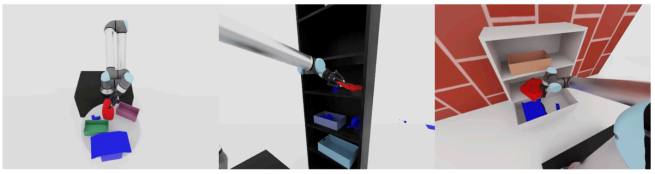


Fig. 4: Simulated environments. *Grasp-MPC* is extensively evaluated in FetchBench [14] environments.

V. Experimental Results: Simulation

In this section, we evaluate *Grasp-MPC* and competitive baselines in simulated environments designed to test grasping capabilities in cluttered settings. The experiments address: (1) How well does *Grasp-MPC* grasp unseen objects using ground-truth grasp poses? (2) How robust is it to perturbed ground-truth poses? (3) How does it perform with predicted grasp poses?

A. Experiment Setup

Grasp-MPC and baselines are evaluated in simulation using a UR10 robot with a Robotiq 2F-140 gripper. Evaluations are based on FetchBench [14] (Figure 4), adapted to use Isaac Sim for improved physics and closed-loop gripper modeling. All objects are novel and consist of both procedurally generated objects and the ACRONYM objects [12]. Three cameras capture point clouds, and the one with the greatest coverage of the target object is selected to provide observations to the policy. Experiments span 90 scenes (60 problems each), resulting in $5,400$ test cases. A $6cm$ offset is used to compute pre-grasp poses from the grasp poses.

Evaluation Metrics: The evaluation uses the metric *Grasp Success*, defined as lifting the object at least 1cm. Since the task involves motion planning to reach the pre-grasp pose, this metric accounts for motion planning failures by excluding trials where moving to the pre-grasp pose failed.

Baselines: *Grasp-MPC* is compared to the following:

- **Open-Loop Linear**: The end-effector moves linearly from the pre-grasp to the desired grasp pose. Operational Space Control [43] implemented in FetchBench is used.
- **Transformer Policy** A Transformer-based policy, inspired by the architecture used in OPTIMUS [17], and included as a baseline in FetchBench [14].
- **Diffusion Policy (DP)**: A diffusion policy [44] that takes point cloud observations as input, which is considered one of the state-of-the-art IL approaches.

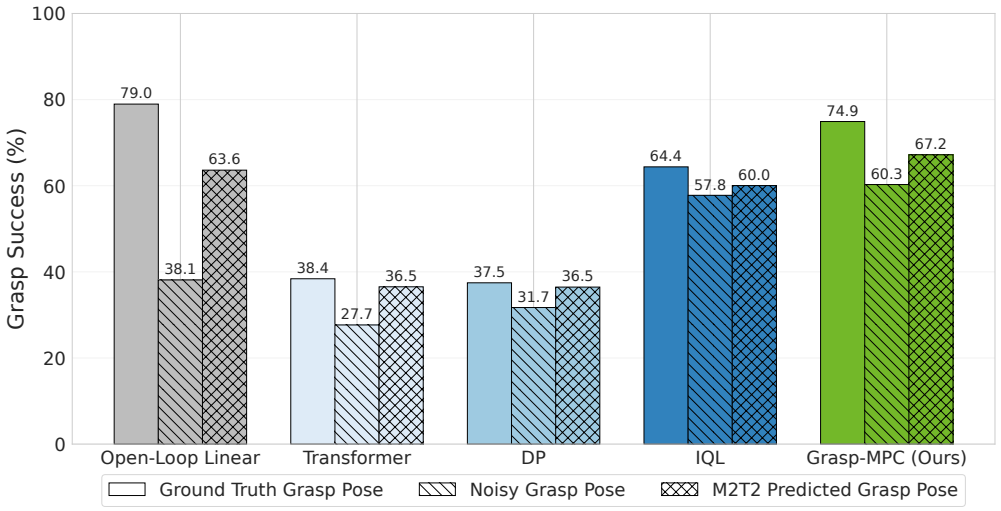


Fig. 5: Grasp performance comparison across different methods given ground truth, noisy, and M2T2-predicted grasp poses. Solid bars represent performance with ground truth grasp poses, diagonally hatched (///) bars show results with noisy perturbations ($\pm$2cm, $\pm$18deg), and cross-hatched (xx) bars indicate M2T2-predicted grasp poses. While *Grasp-MPC* maintains a high grasp success rate across all perturbations, Open-Loop Linear, an open-loop baseline, significantly drops with perturbed grasp poses.

- **IQL**: A policy trained with Implicit Q-Learning (IQL) [32], a state-of-the-art offline RL method.

All methods, including *Grasp-MPC*, use CuRobo's motion planner to generate a trajectory from the robot's initial position to a pre-grasp pose. The robot then attempts to grasp the object using one of the above methods. IL policies are trained only on successful trajectories.

B. Grasp Execution with Grasp Pose Annotations

Grasp pose annotations for objects in the Fetchbench environments were generated for the Robotiq 2f-140 following ACRONYM [12]. Up to 200 kinematically valid, collision-free grasp poses are randomly sampled during evaluation, from which one is selected for the robot to move to the corresponding pre-grasp pose using a goalset motion planner in CuRobo [37].

***Grasp-MPC* nearly matches oracle performance while surpassing closed-loop baselines.** As shown in Figure 5, *Grasp-MPC* demonstrates competitive performance compared to *Open-Loop Linear*, an open-loop approach, which serves as an oracle baseline. *Grasp-MPC* achieves a grasp success rate of 74.9%, closely matching the oracle baseline *Open-Loop Linear* at 79.0%. Moreover, *Grasp-MPC* significantly outperforms closed-loop baseline methods.

Closed-loop baselines underperform due to suboptimal data and domain mismatch. IQL achieves a grasp success of 64.4%, below *Grasp-MPC* (73.6%), likely due to limitations in its policy extraction process [34]. IL-based approaches also perform poorly, likely because motion planning, while efficient for data collection, often yields suboptimal demonstrations. Additionally, the gap between the empty scene used for data collection and the object rich evaluation environments introduce MDP mismatches that further degrade performance.

***Grasp-MPC* is robust to noisy pre-grasp poses.** *Grasp-MPC* and baseline methods are also evaluated using ground-truth grasp poses perturbed with random noise sampled from $\mathcal{U}(-2\text{cm}, 2\text{cm})$ in translation and $\mathcal{U}(-18\text{deg}, 18\text{deg})$ in orientation. As shown in Figure 5, *Open-Loop Linear* fails to recover from these perturbations due to its open-loop nature, resulting in a 40% drop in performance. In contrast, *Grasp-MPC* achieves a 60.3% grasp success rate (14% drop), outperforming baseline closed-loop control policies.

C. Grasp Execution with a Grasp Pose Prediction Model

M2T2 [1] is used as an off-the-shelf grasp prediction model to generate target grasp poses. We attempted to train M2T2 using grasp pose annotations specific to the Robotiq gripper, but could not train an optimal grasp pose prediction model. We instead use the publicly available M2T2 model trained on Franka Panda gripper, adding a 10cm offset to adapt the predicted grasp poses for the Robotiq gripper. The top-512 scored grasp poses are selected.

***Grasp-MPC* maintains strong performance despite noisy grasp poses from M2T2, outperforming all baselines.** IL-based approaches perform poorly, achieving only 36.5% grasp success rate as shown in Figure 5. The standard *Open-Loop Linear* approach achieves 63.6%, dropping by 15% from using ground truth grasp poses. *Grasp-MPC* achieves 67.2% success, highest among methods, dropping only by 8% from using ground truth grasp poses. This result highlights *Grasp-MPC*'s robustness to prediction errors from an off-the-shelf grasp prediction model and shows promise for real-world deployment, which is evaluated in the next section.

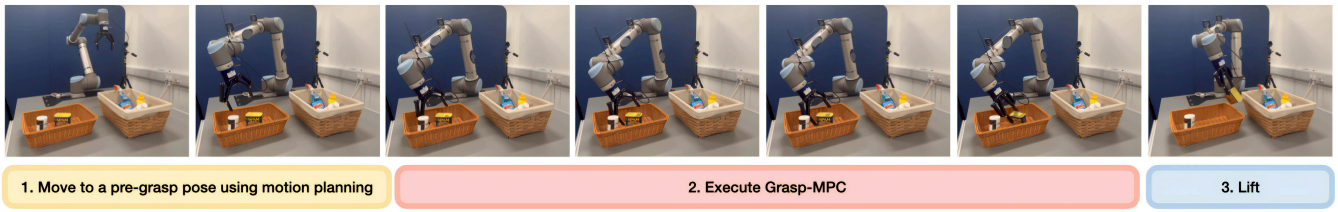


Fig. 6: *Grasp-MPC* execution in the Table Clutter scene. *Grasp-MPC* effectively grasps a novel object from the bin.

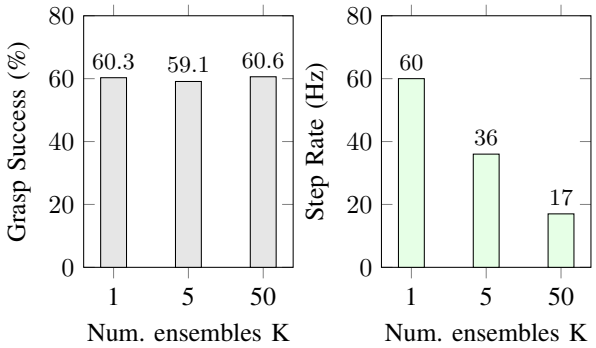


Fig. 7: Ablation on Ensemble showing Grasp Success and Step Rate for different number of ensembles K.

D. Ensemble Ablation Study

We present results using a single value function, as ensembles did not yield meaningful performance improvements. To validate this choice, we use the risk-averse objective for ensemble value functions, following the prior work [26]:

Impact of Value Function Ensembles. Figure 12 reports grasp success rates across all scenes. Using an ensemble improves performance by only 0.3%, suggesting that the dataset provides sufficient coverage for training a single value function. Although ensembles would promote safer behavior via pessimistic cost estimation, a single value function still generalizes well and enables effective grasping. The MPC step runs at $60hz$ for $K = 1$ and $17hz$ for $K = 50$, indicating the computational trade-off of using larger ensembles.

VI. Experimental Results: Real-World

Grasp-MPC is evaluated on novel objects using a UR10 robot arm with a Robotiq 2F-140 in real-world environments. The experiments are designed to address: (1) How effectively does *Grasp-MPC* grasp novel objects in challenging real-world environments compared to open-loop approaches? (2) How well does it adapt to dynamic perturbations when the target object pose changes after reaching the pre-grasp position?

A. Experiment Setup

Perception System. Two RealSense L515 cameras with known extrinsics capture 640×480 depth images to generate point cloud observations. Target object segmentation is performed using SAM-Track [45], which combines Grounding DINO [46] for detection and SAM [47] for segmentation, producing input point clouds for *Grasp-MPC*'s value function. To handle obstacles in cluttered scenes, NVBlox [48] is used to represent the environment for CuRobo's motion planning and MPC modules.

Fig. 8: Representative real-world environments: (Left) Table Empty, (Middle) Table Clutter, and (Right) Shelf Clutter.

Success Criteria. Grasp success is defined as lifting the target object and returning the arm to its home position without dropping it during execution.

Grasp Pose Prediction. As in the simulated experiments, M2T2 predicts grasp poses, with a 10cm offset applied to adapt them for the Robotiq gripper.

Baselines. *Open-Loop Linear*, implemented in CuRobo-GraspAPI, is used for comparison, as it provides a simple and reliable grasp execution pipeline suitable for real-world deployment. Other closed-loop control policies are excluded from real-world experiments due to safety concerns, as they lack collision avoidance mechanisms and are prone to collide with surrounding obstacles such as a shelf or clutter. In contrast, *Grasp-MPC* minimizes collision and learned grasp task costs, enabling safe and effective operation in cluttered real-world environments.

B. Grasping Performance Across Different Scenes

Comprehensive evaluation across three environments with increasing complexity. We compare *Grasp-MPC* against *Open-Loop Linear* using CuRobo-GraspAPI across three progressively challenging environments: *empty tabletop*, *cluttered tabletop*, and *cluttered shelf scenes* as shown in Figure 8. Each environment utilizes 5 different objects, with distinct object sets for each environment to ensure comprehensive evaluation across varying complexity levels. For each object, three different poses are considered, and two evaluation trials per method are performed to assess consistency and reliability. In total, 30 trials are conducted for each environment.

Consistent outperformance over open-loop baselines across all environments. Figure 9 presents the grasp success rates of *Grasp-MPC* and *Open-Loop Linear* across different scenes. The open-loop baseline frequently fails to grasp target objects when the predicted grasp pose deviates from the ideal configuration, as it cannot adapt during execution. In contrast,

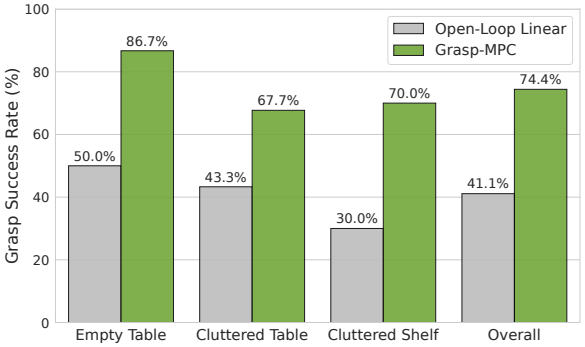


Fig. 9: Grasp performance comparison between Open-loop Linear and *Grasp-MPC*. *Grasp-MPC* consistently achieves higher grasp success rates across all scenes, highlighting its robustness in real-world environments.

Grasp-MPC continuously adjusts the gripper pose to minimize the task grasp cost function while avoiding obstacles such as a shelf, demonstrating safe and robust grasp execution across all environments. *Grasp-MPC* outperforms the baseline in all scene types, with greater performance improvements observed in complex environments.

C. Real-Time Adaptation to Object Pose Perturbations

To demonstrate the benefits of *Grasp-MPC* as a closed-loop control policy, perturbations are added to the target object pose after the robot reaches the pre-grasp pose. Since open-loop approaches cannot handle such dynamic changes by design, we evaluate only *Grasp-MPC* in this experiment to assess its real-time adaptation capabilities.

Successful grasping despite large object perturbations during execution. We evaluate 5 objects with 6 trials per object, resulting in 30 trials total. Figure 10 illustrates an example rollout demonstrating *Grasp-MPC*'s adaptability. *Grasp-MPC* achieves a 60% success rate, demonstrating its capability to adapt in real-time even when large perturbations are applied to the target object pose. *Grasp-MPC* exhibits this global behavior while having trained on grasping motions that are only 15cm away from a grasp. We are planning on training with larger grasping motions in the future to study if we can get even higher success at this task.

VII. Discussion

This work presents *Grasp-MPC*, a 6-DoF closed-loop grasping policy for novel objects in cluttered environments. The approach learns a value function from both successful and failed grasping trajectories, which is subsequently integrated into an MPC framework to generate actions during deployment. Through its modular design, *Grasp-MPC* can address deployment-time constraints such as clutter without requiring retraining. Moreover, leveraging both successful and failed trajectories enhances data efficiency.

Grasp-MPC is validated on the simulated benchmark FetchBench [14] across 5,400 grasping problems, demonstrating superior performance compared to IL methods. It also outperforms open-loop planning-based approaches in scenarios with noise in the grasp pose or when the grasp pose was provided by a learned model (e.g., M2T2). On a real robotic system, *Grasp-MPC* achieves a 30% higher success rate than a planning-based approach in cluttered table-top and shelf settings, despite being trained exclusively on empty scenes. These results demonstrate the ability of the method to operate with noisy point clouds and to handle real-world contact physics without relying on physically simulated training data.

While shown to be effective, *Grasp-MPC* presents several limitations that highlight opportunities for future improvement and extension. First, although higher success rates are achieved compared to existing methods, performance in real-world deployments does not yet reach 100%. We hypothesize that leveraging physics simulation during data generation to generate success/failure labels will improve the performance of *Grasp-MPC*. *Grasp-MPC* can also be readily finetuned with real-world data as only success/failure labels are required for trajectories. Both of these explorations are left to future work.

A further limitation is that validation is restricted to grasping tasks. Although the approach could extend to other manipulation tasks given suitable demonstration pipelines [49], evaluating value function learning across diverse tasks is left for future work.

ACKNOWLEDGMENT

This work was supported by a UKRI/EPSRC Programme Grant [EP/V000748/1]. We would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) and SCAN facilities in carrying out this work. We would like to thank Stan Birchfield for providing feedback on early versions of this paper.

References

[1] W. Yuan, A. Murali, A. Mousavian, and D. Fox, "M2t2: Multi-task masked transformer for object-centric pick and place," in *Conference on Robot Learning*, 2023.

[2] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *IEEE International Conference on Computer Vision*, 2019, pp. 2901–2910.

[3] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *IEEE International Conference on Robotics and Automation*. IEEE, 2021, pp. 13 438–13 444.

[4] J. Carvalho, A. T. Le, P. Jahr, Q. Sun, J. Urain, D. Koert, and J. Peters, "Grasp diffusion network: Learning grasp generators from partial point clouds with diffusion models in so(3)xr3," *arXiv preprint arXiv:2412.08398*, 2024.

[5] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 651–673.

[6] L. Wang, Y. Xiang, W. Yang, A. Mousavian, and D. Fox, "Goal-auxiliary actor-critic for 6d robotic grasping with point clouds," in *Conference on Robot Learning*. PMLR, 2022, pp. 70–80.

[7] R. Singh, A. Allshire, A. Handa, N. Ratliff, and K. Van Wyk, "Dextrah-rgb: Visuomotor policies to grasp anything with dexterous hands," *arXiv preprint arXiv:2412.01791*, 2024.

[8] T. G. W. Lum, M. Matak, V. Makoviychuk, A. Handa, A. Allshire, T. Hermans, N. D. Ratliff, and K. V. Wyk, "DextrAH-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics," in *Conference on Robot Learning*, 2024.

Fig. 10: *Grasp-MPC* execution for moving objects. *Grasp-MPC* adapts in real time to track and grasp moving target objects, capabilities that open-loop approaches lack.

[9] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre *et al.*, "Objaverse-xl: A universe of 10m+ 3d objects," *Neural Information Processing Systems*, vol. 36, 2024.

[10] K. R. Barad, A. Orsula, A. Richard, J. Dentler, M. Olivares-Mendez, and C. Martinez, "Graspldm: Generative 6-dof grasp synthesis using latent diffusion models," *IEEE Access*, 2024.

[11] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 441–11 450.

[12] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *IEEE International Conference on Robotics and Automation*. IEEE, 2021, pp. 6222–6227.

[13] S. Song, A. Zeng, J. Lee, and T. A. Funkhouser, "Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations," *IEEE Robotics and Automation Letters*, vol. 5, pp. 4978–4985, 2019.

[14] B. Han, M. Parakh, D. Geng, J. A. Defay, G. Luyang, and J. Deng, "Fetchbench: A simulation benchmark for robot fetching," in *Conference on Robot Learning*. PMLR, 2025, pp. 3053–3071.

[15] J. Yamada, Y. Lee, G. Salhotra, K. Pertsch, M. Pflueger, G. Sukhatme, J. Lim, and P. Englert, "Motion planner augmented reinforcement learning for robot manipulation in obstructed environments," in *Conference on Robot Learning*. PMLR, 2021, pp. 589–603.

[16] J. Yamada, J. Collins, and I. Posner, "Efficient skill acquisition for insertion tasks in obstructed environments," in *Proceedings of the 6th Annual Learning for Dynamics and Control Conference*, vol. 242. PMLR, 2024, pp. 615–627.

[17] M. Dalal, A. Mandlekar, C. Garrett, A. Handa, R. Salakhutdinov, and D. Fox, "Imitating task and motion planning with visuomotor transformers," 2023.

[18] P. Abbeel, A. Coates, and A. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, vol. 29, pp. 1608 – 1639, 2010.

[19] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1433–1440.

[20] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 1–9.

[21] Y.-C. Chen, A. Murali, B. Sundaralingam, W. Yang, A. Garg, and D. Fox, "Neural motion fields: Encoding grasp trajectories as implicit value functions," *arXiv preprint arXiv:2206.14854*, 2022.

[22] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 2786–2793.

[23] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," *Neural Information Processing Systems*, vol. 28, 2015.

[24] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 09–15 Jun 2019, pp. 2555–2565.

[25] N. A. Hansen, H. Su, and X. Wang, "Temporal difference learning for model predictive control," in *International Conference on Machine Learning*. PMLR, 2022, pp. 8387–8406.

[26] N. Jawale, B. Boots, B. Sundaralingam, and M. Bhardwaj, "Dynamic non-prehensile object transport via model-predictive reinforcement learning," 2024.

[27] M. Zhong, M. Johnson, Y. Tassa, T. Erez, and E. Todorov, "Value function approximation and model predictive control," in *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2013, pp. 100–107.

[28] K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch, "Plan online, learn offline: Efficient learning and exploration via model-based control," *arXiv preprint arXiv:1811.01848*, 2018.

[29] J. Sacks and B. Boots, "Learning sampling distributions for model predictive control," in *Conference on Robot Learning*. PMLR, 2023, pp. 1733–1742.

[30] J. Yamada, S. Zhong, J. Collins, and I. Posner, "D-cubed: Latent diffusion trajectory optimisation for dexterous deformable manipulation," 2024.

[31] X. B. Peng, A. Kumar, G. Zhang, and S. Levine, "Advantage-weighted regression: Simple and scalable off-policy reinforcement learning," *arXiv preprint arXiv:1910.00177*, 2019.

[32] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," *arXiv preprint arXiv:2110.06169*, 2021.

[33] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *Neural Information Processing Systems*, vol. 33, pp. 1179–1191, 2020.

[34] S. Park, K. Frans, S. Levine, and A. Kumar, "Is value learning really the main bottleneck in offline rl?" *arXiv preprint arXiv:2406.09329*, 2024.

[35] M. Bhardwaj, B. Sundaralingam, A. Mousavian, N. D. Ratliff, D. Fox, F. Ramos, and B. Boots, "Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 750–759.

[36] A. Murali, B. Sundaralingam, Y.-W. Chao, J. Yamada, W. Yuan, M. Carlson, F. Ramos, S. Birchfield, D. Fox, and C. Eppner, "Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training," *arXiv preprint arXiv:2507.13097*, 2025.

[37] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos *et al.*, "Curobo: Parallelized collision-free minimum-jerk robot motion generation," *arXiv preprint arXiv:2310.17274*, 2023.

[38] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[39] C.-A. Cheng, T. Xie, N. Jiang, and A. Agarwal, "Adversarially trained actor critic for offline reinforcement learning," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 17–23 Jul 2022, pp. 3852–3878.

[40] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1433–1440.

[41] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics (T-RO)*, 2023.

[42] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:1706.02413*, 2017.

[43] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[44] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, 2023.

[45] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, and Y. Yang, "Segment and track anything," *arXiv preprint arXiv:2305.06558*, 2023.

[46] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[47] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[48] A. Millane, H. Oleynikova, E. Wirbel, R. Steiner, V. Ramasamy, D. Tingdahl, and R. Siegwart, "nvblox: Gpu-accelerated incremental signed distance field mapping," *arXiv preprint arXiv:2311.00626*, 2024.

[49] C. Garrett, A. Mandlekar, B. Wen, and D. Fox, "Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment," in *Conference on Robot Learning*, 2024.

[50] V. Pravdová, L. Gajdošech, H. Ali, and V. Kocur, "On representation of 3d rotation in the context of deep learning," *arXiv preprint arXiv:2410.10350*, 2024.

[51] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[52] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022.

APPENDIX

A. Full point cloud vs partial point cloud observations.

Grasp-MPC robustly grasps novel objects even when provided with partial point cloud observations, achieving a grasp success rate of 74.9%, which is only 1.4% lower than with full point clouds (76.5%). This small performance gap highlights its practicality for real-world deployment, where sensor occlusions and incomplete observations are common, and ensures reliable grasping performance despite imperfect perception.

B. Grasp Success by Scene-Type in Simulation

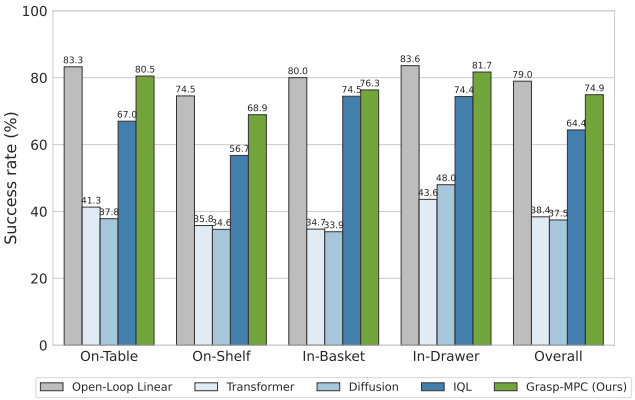
***Grasp-MPC* performs competitively with the oracle baseline (*OSC*) and consistently outperforms other baselines across all scene categories; however, *on-shelf* scenes remain particularly challenging.** Figure 11 shows the grasp success rates across various scene categories. While *OSC* serves as an oracle baseline by leveraging ground-truth grasp poses, *Grasp-MPC* achieves a comparable success rate, demonstrating strong performance. Moreover, *Grasp-MPC* consistently outperforms all non-oracle baselines across every scene category. Among all scene categories, *on-shelf* environments are the most challenging, a trend that also holds in real-world settings (see Figure 9).

***Grasp-MPC* consistently outperforms baseline methods, and the performance of *OSC* drops significantly when noisy grasp poses are given.** Figure 11-(b) presents grasp success rates across different scene categories using grasp poses perturbed with random noise sampled from $\mathcal{U}(-2\text{cm}, 2\text{cm})$ in translation and $\mathcal{U}(-18\text{deg}, 18\text{deg})$ in orientation. Unlike in the ground-truth setting, the performance of *OSC* degrades substantially under noise, as its open-loop nature prevents it from recovering from perturbations. In contrast, *Grasp-MPC* maintains strong performance even with noisy grasp inputs. Interestingly, *IQL* slightly outperforms *Grasp-MPC* in the *In-Basket* scene; however, *Grasp-MPC* achieves consistently superior results across the remaining scenes and in overall performance. We hypothesize that baseline methods such as *IQL* struggle in more constrained environments like *On-Shelf* and *In-Drawer* scenes.

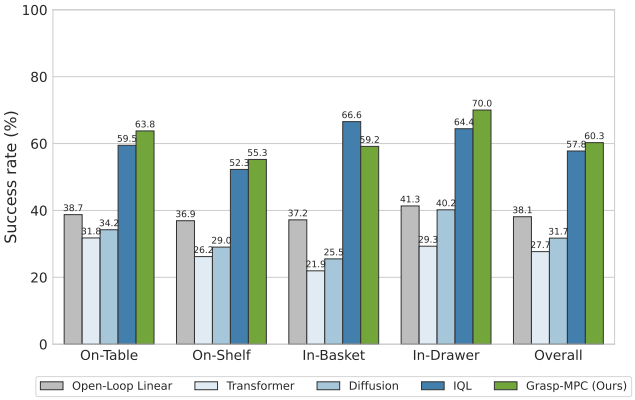
***Grasp-MPC* consistenly has a higher grasp success rate in each scene type when combined with a grasp prediction model.** Figure 11-(c) illustrates the grasp success rate of *Grasp-MPC* and the baseline approaches for each scene type. *Grasp-MPC* consistently achieves success rates exceeding 60% across all scene categories and outperforms baseline approaches by a significant margin. This highlights the superior efficacy of *Grasp-MPC* even when target grasp poses are predicted by an off-the-shelf grasp pose prediction model.

C. Ensemble Ablation Study

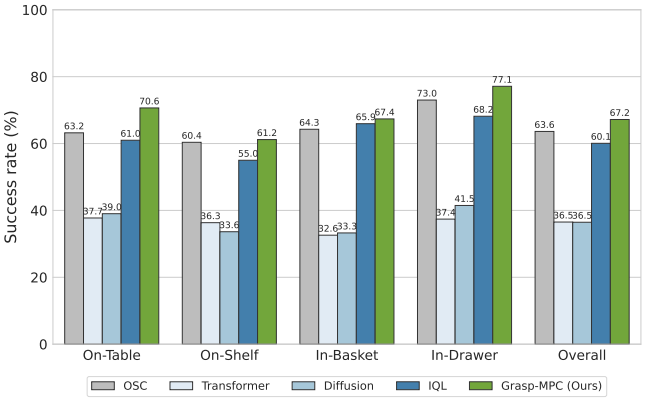
We present results using a single value function, as ensembles did not yield meaningful performance improvements. To validate this choice, we use the following risk-averse objective for ensemble value functions, following prior work [26]:



(a) Ground Truth Grasp Pose Annotations



(b) Noisy Grasp Pose Annotations



(c) M2T2 generated Grasp Poses

Fig. 11: Grasp success rate for each scene type given grasp pose annotations from ground truth (a), added noise (b), and from a trained grasp pose prediction model M2T2 (c). *Grasp-MPC* achieves a competitive or substantially higher success rate compared to the competitive baselines.

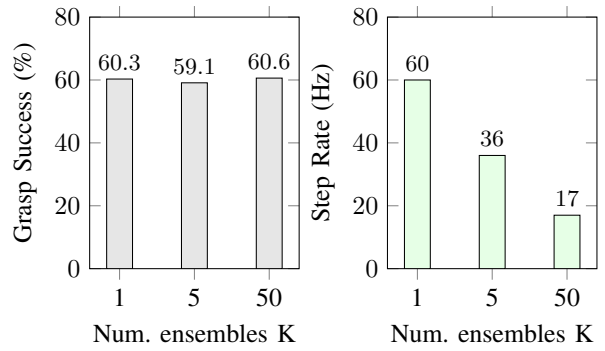


Fig. 12: Ablation on Ensemble showing Grasp Success and Step Rate for different number of ensembles K.

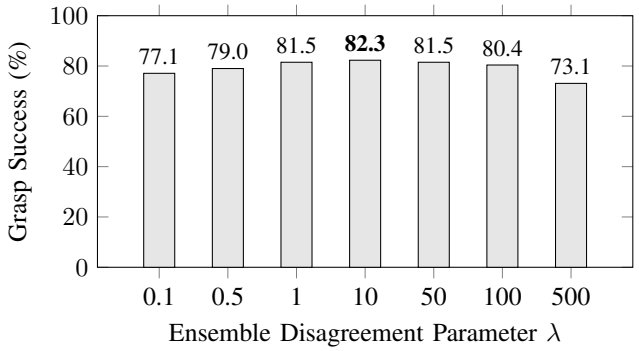


Fig. 13: Grasp Success across different λ values.

$$C_{grasp}(\mathbf{x}_{h\in H}, \ddot{\mathbf{a}}_{h\in H}) = log(\sum_{1}^{K} exp(\frac{1}{\lambda} G_i(\mathbf{x}_{h\in H}, \ddot{\mathbf{a}}_{h\in H}))) \tag{7}$$

Impact of Value Function Ensembles. Figure 12 reports grasp success rates across all scenes. Using an ensemble improves performance by only 0.3%, suggesting that the dataset provides sufficient coverage for training a single value function. Although ensembles would promote safer behavior via pessimistic cost estimation, a single value function still generalizes well and enables effective grasping. The MPC step runs at $60hz$ for $K = 1$ and $17hz$ for $K = 50$, indicating the computational trade-off of using larger ensembles.

Impact of Disagreement Hyperparameter. We analyze the effect of the disagreement hyperparameter λ in the pessimistic cost when using an ensemble value function with $K = 50$. Figure 13 shows grasp success rate of *Grasp-MPC* across 300 problems with varying λ. While performance is generally robust, overly pessimistic costs (i.e., large λ) can lead to performance degradation.

D. Analyzing Success in FetchBench

The Fetchbench benchmark contains 5400 object retrieval problems. The task success, as described by Fetchbench requires the robot to start from an initial joint configuration and grasp a specific object that's in the environment. After grasping the object, the robot then needs to move the grasped object to a retrieve pose. The evaluation done by Fetchbench focuses on the whole object retrieval pipeline while *Grasp-MPC* only solves one part of the pipeline. To make this clear, we split the task into four sequential phases:

Reach Pre-grasp Pose: The first phase requires the robot to reach a pre-grasp pose.

Pre-Grasp to Grasp: Then, from the Pre-Grasp pose, the robot moves to the grasp pose using a linear motion with OSC or with one of the learned policies described in Sec. V. After closing the gripper on the object, the robot lifts the object by 1cm with OSC.

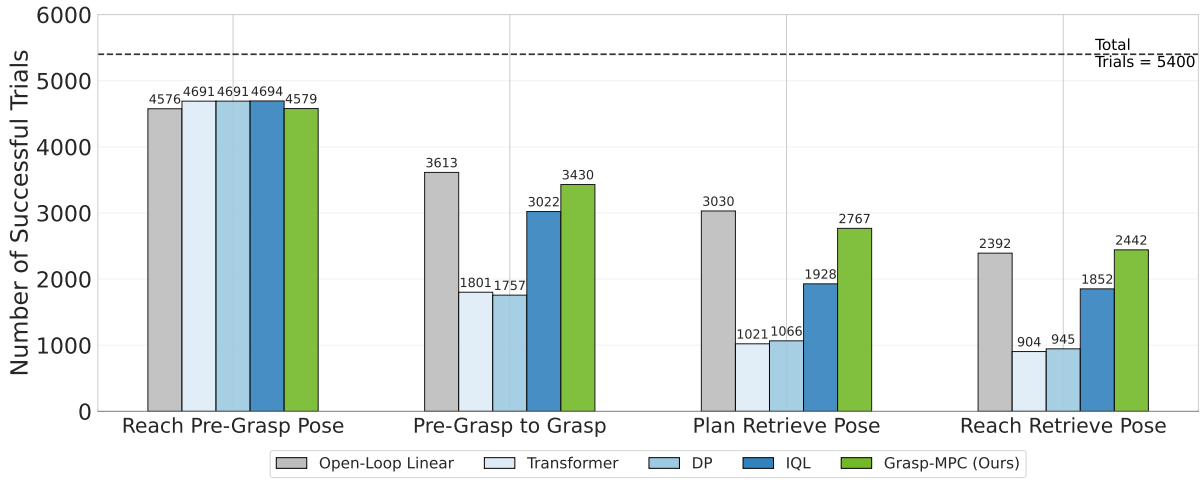
Plan Retrieve Pose: If the object is still in the gripper after the lift, the motion planner is called to plan a path to reach the retrieve pose. Failures in this phase are motion planning failures.

Reach Retrieve Pose: For those problems, where a plan is found, the robot then executes the trajectory. After reaching, if the object is still in the gripper, then task success is achieved. During this phase, failures happen because of the object slipping during motion, leading to either dropping the object or colliding with the environment.

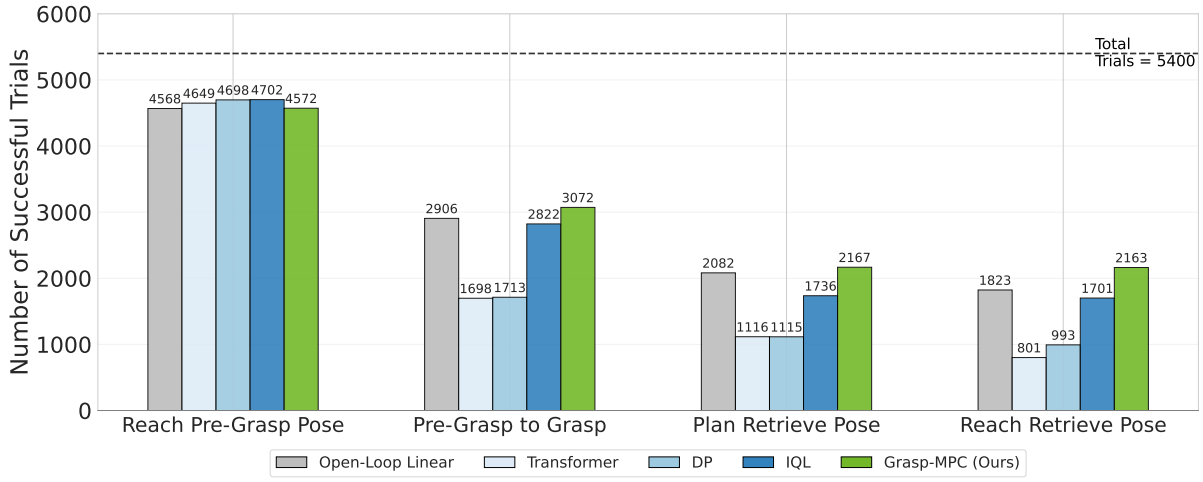
By splitting the task into four sequential phases, failures can be better understood. E.g., even with ground truth annotated grasp poses, the planner was only able to reach the pre-grasp pose 84% of the problems, as shown in Figure 14. This could mean that for the remaining problems, a more contact-rich non-prehensile action could be needed to move the object before grasping. Our experiments also showed a 3% variation in the success of Reach Pre-Grasp Pose between methods (baselines and *Grasp-MPC*) even though all methods use the same motion planner. We hypothesize that this could be because of small changes in the initial joint state due to simulation errors. To remove the effect of this slight variation, we calculate grasp success only based on trials that succeeded in Reach Pre-Grasp Pose Phase per method.

After grasping the object and lifting 1cm, we observed that not all successful trials in this phase can obtain a collision-free motion plan to move the object to the retrieve pose as seen by the drop in successful trials in *Plan Retrieve Pose* phase in Figure 14. With ground truth grasp poses, only 83% of oracle (OSC) and 80% of *Grasp-MPC* grasped objects get a successful retrieve pose plan. With grasp pose generated by M2T2, the success is worse with only 70% of OSC and 71% of *Grasp-MPC* grasped objects getting a successful retrieve motion plan. This highlights the need for grasping methods to also reason about future tasks, like retrieving paths.

Similarly, only 78% of the planned paths succeed in reaching the retreive pose with the OSC grasped object upon execution while 88% of the *Grasp-MPC* grasped object reach the retrieve pose when ground truth grasp poses are generated. When using grasps generated from M2T2, the success of reaching the retrieve pose from planned paths increases to 87.5% for OSC grasped objects and 99.8% for *Grasp-MPC* grasped objects as shown in Figure 14-(b). This is promising as even



(a) Ground Truth Grasp Poses



(b) M2T2 predicted Grasp Poses

Fig. 14: Split of fetchbench task into phases with successful trials at each phase. Each bar represents the number of successful trials for each method across the different phases of task execution in FetchBench. The dashed line indicates the total number of trials (5,400).

though our training dataset did not contain physics validated (e.g., sim) large retrieve motions to label success/failure. We hypothesize that the learned value function has attempted to go to a region with the most grasps on an object (which often is the most stable grasp). The success analysis across phases in Fetchbench evaluation provides venues for future research, including improvements to the collected dataset. Sim validation of the grasping motion and using large retrieve motions after grasping to label grasp success would improve task success. Conditioning the dataset and value function with retrieve paths will also improve success in Plan Retreive phase.

E. Synthetic Grasp Trajectory Data Generation Details

We utilize a motion planner implemented in CuRobo [37] to generate trajectories for both valid and invalid grasp poses. To build the dataset, we spawn 24 robots at a time, each planning trajectories toward different desired grasp poses for the same object. Then, we collect up to 256 trajectories for each object. During data collection, 70% of the trajectories correspond to valid grasp poses, while the remaining 30% represent invalid grasp poses. The motion planner occasionally struggles to generate viable solutions, leading to delays in the trajectory collection process. To mitigate this issue and maintain efficiency, we define a maximum failure threshold of 10 attempts per object. If the failure threshold is reached, we save the trajectories collected so far and reset the collection process to move on to a new object. This constraint prevents excessive computational overhead during data collection. This procedure is repeated across a total of $8,151$ objects sourced from the Objaverse dataset.

F. Value Function Training Details

The value function architecture consists of three main components: the PointNet++ encoder, the state encoder, and the

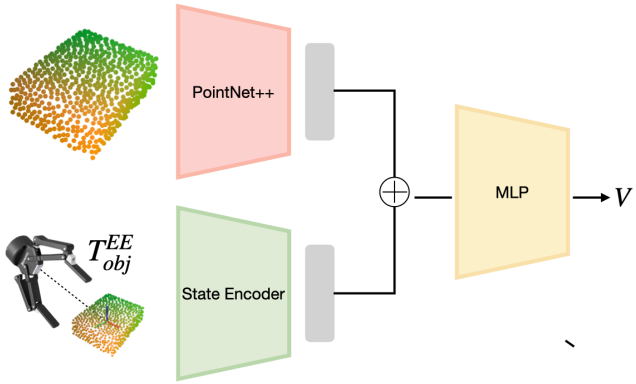


Fig. 15: *Grasp-MPC* trains a value function to serve as a cost function within MPC. The PointNet++ encoder takes a segmented point cloud as input, and the state encoder takes an end-effector pose with respect to the mean of the object point cloud T^{EE}_{obj} as input. Features from each encoder are concatenated and then used as input to the MLP head to estimate the value.

value head network (see Figure 15). The PointNet++ encoder employs three sets of abstraction layers, followed by three fully connected layers. The first set abstraction layer uses furthest point sampling (FPS) to reduce the input to 256 points. It then applies a grouping query within a radius of $3cm$, selecting up to 64 points per group. This is followed by a local PointNet consisting of three fully connected layers of size 3, 64, and 128. The second set abstraction layer further downsamples the point cloud to 64 points, using a grouping query that captures up to 128 points within a $4cm$ radius. Its local PointNet has fully connected layers of sizes 128, 128, and 256. The third abstraction layer skips FPS, instead grouping all points together, and employs a local PointNet with layer sizes of 256, 256, and 512. After the set abstraction layers, the output passes through three fully connected layers with sizes 512, 256, and 128. Between these layers, layer normalization, dropout with $p = 0.2$, and ELU activations are applied. The final output of the PointNet++ encoder is a 128 dimensional feature vector.

The state encoder processes a 9-dimensional input that encodes the pose of the robot's end-effector. The first three dimensions represent the translational components of the pose, while the remaining six dimensions correspond to the first two columns of the end-effector's rotation matrix (6D Gram-Schmidt) for its orientation [50]. The state encoder consists of two hidden layers with sizes 64 and 32, producing a 32-dimensional embedding vector. This output is concatenated with the 128-dimensional feature vector from the PointNet++ encoder, resulting in a 160-dimensional feature vector. Finally, the value head network consists of fully connected layers with three hidden layers of sizes 256, 128, and 64.

G. MPC details

We use a sampling-based MPPI optimizer with three main constraints: (1) joint limits on position, velocity, acceleration, and jerk; (2) self-collision avoidance; and (3) robot-world collision avoidance. The cost function used in *Grasp-MPC* is a grasp cost using the learned value function and the default CuRobo cost terms:

$$C_{\textit{Grasp-MPC}} = C_{\text{curobo}} + \omega C_{\text{grasp}} \tag{8}$$

where $\omega = 1000$ and C_{curobo} consist of a world collision cost, a self-collision cost, and a bounds cost for trajectory smoothness and keeping the joint states within limits.

To encourage smooth control sequences, we sample actions via a Halton sequence and fit degree-5 B-splines. Hyperparameters of MPC used in our experiments are described in Table I.

TABLE I: CuRobo MPC Hyperparameters

Parameter	**Value**
horizon	30
control_space	ACCELERATION
init_cov	0.5
gamma	0.98
n_iters	2
cold_start_n_iters	5
step_size_mean	0.9
step_size_cov	0.5
beta	1.0
alpha	1.0
num_particles	400
update_cov	True
kappa	0.01
null_act_frac	0.05
sample_mode	BEST
best_action	REPEAT
squash_fn	CLAMP
n_problems	1
random_mean	True
use_coo_sparse	True
sample_ratio: halton	0.3
sample_ratio: halton-knot	0.7
sample_ratio: random	0
sample_ratio: random-knot	0

H. Simulated Environment Setup

In this work, we evaluate *Grasp-MPC* and the competitive baselines in FetchBench [14]. However, the original FetchBench environment is incompatible with the Robotiq 2F-140 gripper due to its closed-loop kinematic chain, and its simulation platform, Isaac Gym [51], has been deprecated. To address these limitations, we replace Isaac Gym with Isaac Sim, enhancing simulation accuracy, ensuring compatibility with the Robotiq 2F-140 gripper, and benefiting from improved support and ongoing development (see Figure 16).

In the experiments, we assess *Grasp-MPC* and the baseline methods over 90 unique scenes, each containing 60 problems,

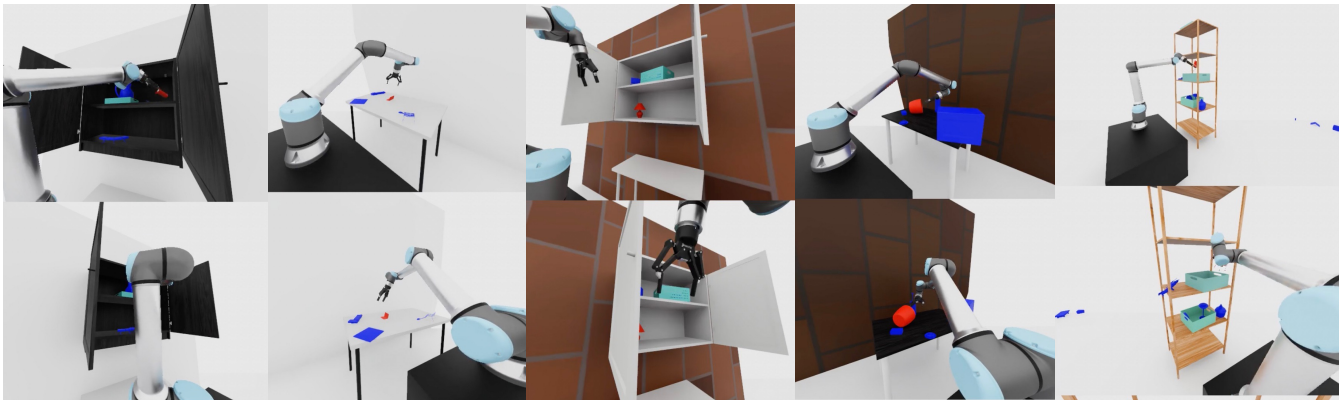


Fig. 16: Representative scenes in FetchBench [14] for grasping in clutter. We replace Isaac Gym [51] used as the underlying simulator in the original FetchBench with Isaac Sim to simulate a Robotiq 2F-140 gripper.

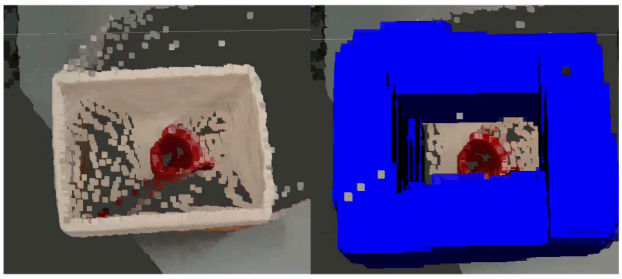


Fig. 17: Example collision voxel visualization using NVBlox. By leveraging NVBlox, a GPU-accelerated SDF library, *Grasp-MPC* successfully avoids collision and grasp the target object.

resulting in a total of $5,400$ test problems. These include $1,415$ on-table, $2,424$ on-shelf, 744 in-basket, and 817 in-drawer cases.

I. Real-world Environment Setup

The real-world experiments use SAM-Track [47], which combines Grounding-DINO [52] for object detection and SAM [47] for segmentation, to track the target object. The segmented depth image is then projected to a point cloud, and latent observation features are obtained from our observation encoder. This latent feature is then sent to our MPC, which is implemented as a ROS Python node.

We use ROS control to switch between a trajectory controller and a velocity controller. We use cuRobo's motion planner to move to the approach pose and then call our MPC to take over and grasp the object. To avoid collision, we use Nvblox [48] to represent the scene for both motion planning and MPC with the resolution of 1cm voxels. (see Figure 17) We run MPC for 120 steps and then close the gripper. Once the gripper is closed, we then lift the gripper 10cm along the opposite direction of the gravity vector using the same motion planner.

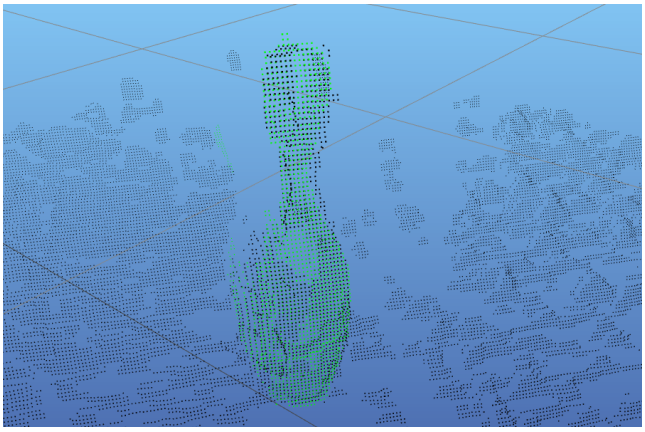


Fig. 18: Overlaying object point cloud across different methods. To maintain identical experimental conditions, the current point clouds (green) are overlaid on the reference object point clouds (black).

Identical conditions ensure fair comparison between methods. To ensure fair comparison, we maintain identical experimental conditions across both approaches: the target grasp pose, pre-grasp pose, and robot configuration at the pre-grasp position remain the same between methods. Object pose consistency is verified by overlaying the point clouds of the target object across different method evaluations, ensuring that any performance differences stem from the grasping approach rather than variations in object positioning (see Figure 18).

J. Implementation Details of Baseline Methods

GraspAPI. We utilize the grasp planning API provided by CuRobo [37] to generate a trajectory for grasping. Initially, the API plans trajectories from the robot's starting position to a set of candidate grasp poses, selecting the shortest path among these options. Once the desired grasp pose is identified, a pre-grasp pose is computed by applying a fixed linear offset to the selected grasp pose. Then, the API generates a trajectory from the robot's initial position to the pre-grasp pose. Following

this, the API employs constrained motion planning to produce a linear trajectory between the pre-grasp pose and the grasp pose. Lastly, the API returns two trajectories: one from the robot's initial position to the pre-grasp pose, and another from the pre-grasp pose to the desired grasp pose.

Operational Space Control. Similar to the baseline approach used in FetchBench [14], we use operational space control to move the end effector linearly from the pre-grasp to the grasp pose.

Diffusion Policy. We train a diffusion policy [44] using behavior cloning (BC) The diffusion policy uses a conditional Denoising Diffusion Probabilistic Models (DDPM), and its conditional noise prediction model employs the CNN-based architecture introduced in [44]. This architecture is a 3-level UNet architecture consisting of conditional residual blocks with channel dimensions $[128, 256, 512]$. The diffusion time step is encoded by a multi-layer perceptron (MLP) to produce a 128-dimensional embedding, which is concatenated with the visual embedding from the PointNet++ encoder and the state embedding from the state encoder. This concatenation yields a 288-dimensional conditional vector. The architecture of the PointNet++ encoder and the state encoder shares the same architecture as that used in the value function network in *Grasp-MPC*. In the experiments, we use an action horizon of 4 and an observation history length of 2. During training, we use 100 diffusion steps. For faster inference during evaluation, we employ Denoising Diffusion Implicit Models (DDIM) with 5 diffusion steps.

The synthetic dataset contains both successful and failed trajectories, corresponding to feasible and infeasible grasp poses. To train the diffusion policy, we utilize only the successful trajectories from the dataset. The policy produces 6-dimensional action outputs, where the first three dimensions represent the change in position, and the last three correspond to the change in the end-effector's axis-angle representation. Expert actions in the dataset are derived by computing the difference between the current and next end-effector poses. Since the motion planner generates a fine-grained trajectory during data collection, we aggregate actions by skipping 4 frames and computing the combined pose difference over this interval.

Transformer Policy. We train a transformer-based policy using BC from only successful trajectories similar to the diffusion policy described in Appendix J. To train the transformer policy, we utilize only the successful trajectories from the dataset. Our transformer policy, similar to the one utilized in FetchBench, is built upon the OPTIMUS architecture. Consistent with the value function in *Grasp-MPC*, the transformer policy processes a segmented target object point cloud alongside the robot end-effector pose relative to the center of the target object point cloud. To encode visual input, we utilize a PointNet++ encoder with the same architecture as the one used in our value function. The transformer policy uses the same action space and expert actions for training as the Diffusion Policy.

Implicit Q-Learning. We train a policy using Implicit Q-Learning, the state-of-the-art offline RL approach. In this training, the objective is to maximise the cumulative discounted rewards. Thus, instead of using a cost function (Eq. 1) defined for *Grasp-MPC*, we use the following reward function:

$$r_t = \begin{cases} 1 & |q_{goal,i} - q_{t,i}| \leq 5e^{-3},\ \forall i,\ and\ \mathbb{1}_{\text{feasible}} = 1, \\ 0 & \text{Otherwise} \end{cases} \tag{9}$$

In IQL, both the state-based value function and the Q-value function are trained. We employ the same network architecture for these value functions as that used in the value function of *Grasp-MPC*, with the key difference that the Q-value function incorporates an action as part of its input, which is included in the input to the state encoder. The policy architecture is similar to the value function architecture; however, it is parameterized as a Gaussian distribution with a fixed, state-independent standard deviation. The policy outputs the mean of the Gaussian distribution for an action space with 6 dimensions.

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).**

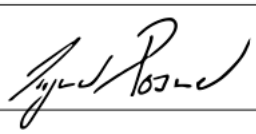
Title of Paper	Grasp-MPC: Closed-Loop Visual Grasping via Value-Guided Model Predictive Control
Publication Status	□Published □ Accepted for Publication ■Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Jun Yamada, Adithyavairavan Murali, Ajay Mandlekar, Clemens Eppner, Ingmar Posner, Balakumar Sundaralingam Under review. Available at arXiv:2509.06201

Student Confirmation

Student Name:	Jun Yamada		
Contribution to the Paper	- Developed an initial research idea - Created and developed methodologies - Ran all experiments - Created all figures, except for Figure 7 - Paper writing		
Signature		Date	19/09/2025

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: Prof. Ingmar Posner			
Supervisor comments *Jun made a substantial contribution to the publication. The description above is accurate.*			
Signature		Date	22/09/2025

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

4.1 Limitations and Future Work

While *Grasp-MPC* improves grasping performance over competitive baselines, its overall success in FetchBench remains limited because the benchmark evaluates both grasping and fetching in cluttered environments, whereas *Grasp-MPC* focuses only on the grasping phase. This limitation could be mitigated by training a value function with cost labels that account not only for grasp success but also for the quality of the grasp in facilitating subsequent fetching actions.

Currently, *Grasp-MPC* operates for a fixed number of timesteps, delaying gripper closure until the final step of the episode. This execution scheme may result in missed opportunities for earlier successful grasps, thereby requiring longer execution time. One potential solution is to train a separate Q-value function for discrete gripper actions, allowing the robot to dynamically decide when to close the gripper and terminate the episode.

Although *Grasp-MPC* demonstrates that learning a value function from data is effective for grasping, extending this approach to more contact-rich manipulation tasks is a promising research direction. *Grasp-MPC* is readily applicable to similar prehensile manipulation tasks; however, addressing non-prehensile manipulation tasks remains challenging, as it requires simulating both the target object and the robot's dynamics. One solution to this is to leverage a world model, for example, learnt using the approach presented in Chapter 7. An alternative is to train a value function that directly evaluates full action sequences [189], potentially reducing reliance on explicit rollouts during inference. Together, these directions highlight the potential of using MPC with a learnt value function as the cost, enabling the acquisition of robust, safe, and reactive closed-loop policies for more complex manipulation tasks.

5

COMBO-Grasp: Learning Constraint-Based Manipulation for Bimanual Occluded Grasping

This chapter addresses the challenge of occluded robot grasping, where desired grasp poses are kinematically infeasible due to environmental constraints, such as collisions with the table surface. While Chapter 3 and Chapter 4 demonstrate a unification of motion planning and learning-based closed-loop control to enable safe and efficient skill acquisition and execution for a single-arm system, this chapter extends this principle to a dual-arm setting. In particular, one arm is controlled by a motion planner to provide constraint-based support, stabilising the target object and enabling the other arm to tackle occluded grasping tasks through coordinated manipulation, thereby achieving sample-efficient skill acquisition and robust task execution (see Chapter 5, Fig. 1 for a rollout of *COMBO-Grasp*).

Prior works [101, 190] address occluded grasping tasks with a single arm by leveraging external constraints, such as walls. However, such constraints are often unavailable in real-world environments, which limits the applicability of these approaches. To overcome such limitations, this chapter introduces *COMBO-Grasp*, a novel bimanual manipulation framework that employs two coordinated policies to address bimanual occluded grasping tasks, drawing inspiration from nonprehensile manipulation strategies that humans perform intuitively [47–49].

The proposed approach comprises three stages, as illustrated in Chapter 5, Fig. 3. First, a synthetic dataset is generated in simulation in a self-supervised manner over a diverse set of objects to train a state-based teacher constraint policy. For each object, a random force is applied along the approach vector of the desired grasp poses that have been pre-annotated for the object. End-effector poses are randomly sampled and evaluated for force closure, where the arm must prevent the object from moving under the applied force. Poses satisfying this condition are labelled as stabilising poses for the constraint arm, used to provide support for the other arm to solve occluded grasping tasks. Effective stabilising poses are stored in the dataset to train the state-based teacher constraint policy, implemented using a diffusion model [191].

Second, a separate state-based teacher grasping policy is trained in simulation for diverse objects using Proximal Policy Optimisation (PPO) [31]. In particular, during RL training for the grasping policy, the teacher constraint policy first predicts a stabilisation pose at the beginning of each episode. The constraint arm is then positioned accordingly and remains fixed in this pose throughout the episode, while the grasping arm executes its policy to learn occluded grasping skills by utilising the constraint arm. A key contribution of this work is the value-guided policy coordination to improve bimanual coordination, inspired by classifier guidance in diffusion models [50, 159]. During RL training, we obtain gradients from the value function, which is jointly trained with the grasping policy by maximising its value, and use these gradients to guide the reverse diffusion process of the constraint policy. This enables the constraint policy to generate a stabilisation pose better aligned with the grasping policy's objectives, thereby improving sample efficiency and task performance during RL.

In the final stage, we distil the teacher constraint and grasping policies into vision-based student policies that take point cloud observations as input, ensuring robust sim-to-real transfer. To reduce the sim-to-real gap, domain randomisation [57] is applied during the distillation process. In real-world experiments, COMBO-Grasp operates through a coordinated sequence: the constraint policy generates a

stabilising pose for the right arm, which a motion planner positions to stabilise the target object. Subsequently, the grasping policy controls the left arm to perform nonprehensile manipulation such as pushing and reorienting to expose the occluded grasp pose and successfully grasp the target object.

COMBO-Grasp is extensively evaluated on novel objects, achieving 65.6% success in simulation and 68.3% in the real world, outperforming state-of-the-art RL algorithms and baseline variants. The system successfully generalises to novel objects due to training on diverse object geometries and shows robust performance across objects with varying shapes, sizes, and weights.

By decomposing the complex, occluded bimanual grasping task into two coordinated policies and employing motion planning to control one arm for object stabilisation and support, *COMBO-Grasp* demonstrates that the unification of planning and learning achieves sample-efficient skill acquisition and enables robust execution in real-world environments. Overall, this chapter presents the following contributions:

1. *COMBO-Grasp*, a novel framework for bimanual manipulation that employs two coordinated policies to address occluded grasping.

2. A self-supervised approach leveraging force-closure as a supervisory signal to collect a dataset of stabilising constraints for one arm, enabling training of a constraint policy that accelerates the RL training of the grasping policy.

3. A value function-guided policy coordination mechanism that refines predicted constraint poses using gradients from the value function of the grasping RL policy, thereby improving bimanual coordination during training.

4. Extensive evaluations in both simulated and real-world environments, demonstrating successful grasps of novel objects.

Jun Yamada, Alexander L Mitchell, Jack Collins, Ingmar Posner (2025). COMBO-Grasp: Learning Constraint-Based Manipulation for Bimanual Occluded Grasping, *Conference on Robot Learning (CoRL)*

COMBO-Grasp: Learning Constraint-Based Manipulation for Bimanual Occluded Grasping

Jun Yamada, Alexander L Mitchell, Jack Collins, Ingmar Posner
Applied AI Lab
Oxford Robotics Institute
University of Oxford

Abstract: This paper addresses the challenge of *occluded* robot grasping, i.e. grasping in situations where the desired grasp poses are kinematically infeasible due to environmental constraints such as surface collisions. Existing RL methods struggle with task complexity, and collecting expert demonstrations is often impractical. Instead, inspired by human bimanual manipulation strategies, where two hands coordinate to stabilise and reorient objects, we focus on a bimanual robotic setup to tackle this challenge. In particular, we introduce Constraint-based Manipulation for Bimanual Occluded Grasping (*COMBO-Grasp*), an approach which leverages two coordinated policies: a constraint policy trained using self-supervised datasets to generate stabilising poses and a grasping policy trained using RL that reorients and grasps the target object. A key contribution lies in value function-guided policy coordination, where gradients from a jointly trained value function refine the constraint policy during RL training to improve bimanual coordination and task performance. Lastly, *COMBO-Grasp* employs teacher-student policy distillation to effectively deploy vision-based policies in real-world environments. Experiments show that *COMBO-Grasp* significantly outperforms baselines and generalises to unseen objects in both simulation and real environments.

Keywords: Occluded Grasping, Bimanual Manipulation, Reinforcement Learning

1 Introduction

Grasping objects with kinematically infeasible grasp poses due to environmental collisions, known as occluded grasping [1], presents a significant challenge in robotics. Such kinematic infeasibility arises from supporting surfaces, such as the table that the object is resting on. For example, grasping a keyboard that rests on a desk requires reorienting the keyboard with regard to the desk surface (nonprehensile manipulation) to reveal the grasp pose (see Figure 1). Humans exhibit exceptional dexterity in solving such occluded grasping problems through coordinated bimanual manipulation, seamlessly using both hands to reposition objects for grasping. However, learning to acquire such coordinated skills for a bimanual robotic system poses significant challenges, particularly when using reinforcement learning (RL) [2, 3].

Specifically, compared to single-handed applications, bimanual manipulation exhibits a significantly increased action space with coordination requirements adding to task complexity. These challenges are exacerbated when using domain randomisation [4] to enable sim-to-real transfer and make RL approaches infeasible due to sample inefficiency. For the occluded grasping task, these challenges are particularly pronounced as the policies must enable one arm to stabilise the object while the other reorients and grasps it. More importantly, designing a reward function that facilitates the emergence of such coordinated behaviour is nontrivial. Compared to RL, learning from demonstration (LfD) necessitates a large number of expert demonstrations [5] encompassing a diverse range of objects to achieve generalisation to unseen objects.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

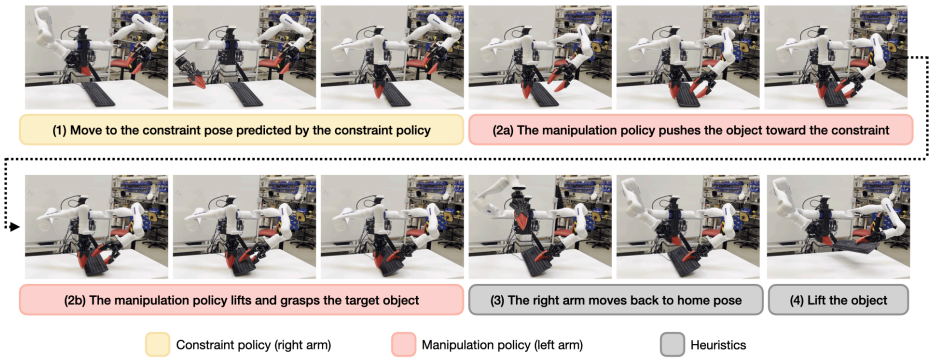


Figure 1: *COMBO-Grasp* uses two coordinated policies to tackle occluded grasping tasks. A constraint policy predicts a support pose for the right arm to assist the left arm controlled by a grasping policy. The task execution sequence is: (1) the right arm moves to the support pose, (2) the left arm grasps the object, (3) the right arm returns home, and (4) the left arm lifts the object.

We present **Co**nstraint-based **M**anipulation for **B**imanual **O**ccluded Grasping (*COMBO-Grasp*), a system designed to address occluded grasping using bimanual robot systems. Inspired by human bimanual strategies, where one hand stabilises an object while the other performs the manipulation [6, 7, 8], *COMBO-Grasp* uses two coordinated policies: a *constraint policy*, trained from dataset collected in a self-supervised manner, that generates stabilising poses, and a *grasping policy* trained using RL that reorients and grasps the target. By stabilising with one arm before grasping with the other, this coordination improves data efficiency and accelerates training for occluded grasping tasks. *COMBO-Grasp* also introduces value-guided policy coordination to refine the constraint pose, improving bimanual coordination. In particular, during RL training, gradients from the value function, trained alongside the grasping policy, optimise the constraint pose to increase grasp success. This alignment enhances object stability during bimanual grasping.

COMBO-Grasp achieves effective sim-to-real transfer via teacher-student policy distillation. A teacher trained with privileged information in simulation is distilled into a student policy that operates on point clouds. Unlike single-policy RL or LfD, *COMBO-Grasp* enables efficient bimanual coordination and generalises to unseen objects without expert demonstrations.

In summary, our contributions are four-fold:

- *COMBO-Grasp*, a novel approach to bimanual manipulation comprising two coordinated policies to solve occluded grasping problems.
- The use of force-closure as a signal to train a self-supervised constraint policy, which accelerates the subsequent RL grasping policy training.
- Value function-guided policy coordination that refines generated constraint poses using gradients from the value function to improve coordination during RL training for the grasping policy.
- Empirically demonstrating that *COMBO-Grasp* successfully grasps seen and unseen objects in both simulated and real-world environments.

2 Related Works

Learning to Grasp Objects. Grasping is a fundamental robotic skill crucial for downstream manipulation tasks [9, 10, 11]. Many prior works focus on learning grasp pose predictors with open-loop planning [11, 12, 13], typically assuming that collision-free poses are reachable via motion planning. However, these methods are often inadequate for occluded grasping scenarios, where environmental constraints may obstruct the target grasp poses. Closed-loop policies using reinforcement learning (RL)[14, 15] and imitation learning (IL)[16, 17] provide an alternative. *COMBO-Grasp* builds on this direction, addressing more challenging occluded grasping tasks that require non-prehensile manipulation before grasping. Some prior works [18, 19] address occluded grasping via extrin-

sic dexterity using a single arm. Sun et al. [18] employ dual arms for object reorientation, though still rely on external constraints such as a wall. In contrast, *COMBO-Grasp* operates without such constraints, using one arm to stabilise the object while the other performs reorientation.

Bimanual Robotic Systems. Bimanual robotic manipulation [20, 21, 8, 22] has gained increasing attention due to its flexibility and capability to handle complex tasks. RL approaches [3, 2] often require extensive exploration, particularly for high-DoF bimanual tasks. Alternatively, IL often demands a large number of expert demonstrations [5], which is often costly and impractical for complex bimanual systems, especially in non-prehensile manipulation scenarios. Several works [23, 24, 25] address these challenges by incorporating inductive biases into RL. Similarly, *COMBO-Grasp* introduces a constraint policy as an inductive bias, specifically tailored for occluded grasping tasks. Inspired by studies in biopsychology [26, 6, 7], *COMBO-Grasp* uses one arm to stabilise the object, while the other performs non-prehensile manipulation for occluded grasping.

Stabilising an object with one arm to assist the other in manipulation is a well-established strategy [27, 22]. However, these prior works require expert demonstrations [27] or nested optimisation loops [22], limiting scalability due to high supervision cost or sample inefficiency. In contrast, *COMBO-Grasp* eliminates the need for expert demonstrations or nested optimisation by using self-supervised simulation data to train a constraint policy, which stabilises objects and accelerates RL training for occluded grasping. Crucially, *COMBO-Grasp* uses value function-guided policy coordination to refine constraint poses by leveraging gradients from the grasping policy's value function during RL training. This allows the constraint policy to adapt poses that better align with the grasping policy, enhancing coordination for bimanual occluded grasping tasks.

3 Task and System Setup

Task description. To grasp a target object given a desired grasp pose that is occluded, one arm is needed to prevent the object from moving, while the dominant arm attempts to reorient and grasp the object. In this work, the left robot arm (dominant arm) always attempts to grasp a target object while the right arm (non-dominant arm) stabilises the object to assist the left arm. We leave dynamic role assignment of left and right arms to future work, similar to [27]. It is worth noting that the gripper of the left arm autonomously closes at the end of each episode to grasp the target object, and the left end-effector moves upward to lift the object.

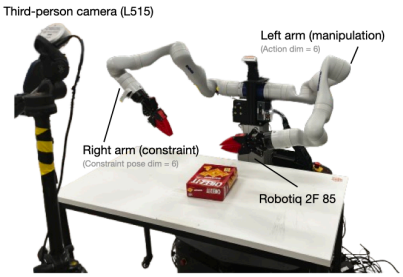


Figure 2: **Real-world system setup.** The system uses two Kinova Gen3 arms mounted perpendicularly, each with a Robotiq 2F-85 gripper and soft fingertips [28] for improved grip. A third-person RealSense L515 camera provides visual observations.

Action Space The teacher and student policies share the same action space. The grasping policy controls the left arm and outputs a six-dimensional delta pose, including translation and rotation in axis-angle representation. The constraint policy controls the right arm and outputs a six-dimensional absolute pose. Following prior work [22], our experiments assume the end-effector remains at a fixed z-coordinate, as it is typically placed on the table, with variations only in its x-y position and orientation. Thus, the first two dimensions correspond to the x and y positions, and the remaining four specify orientation as a quaternion.

Real-World Setup. We design a system for bimanual occluded grasping (Fig.2) comprising two Kinova Gen3 arms with Robotiq 2F-85 grippers, mounted perpendicularly on a central body. The grippers use deformable fingertips [29] for improved grip, replacing the original rigid ones. A calibrated Realsense L515 camera provides third-person point clouds for the vision-based student policies. To control the arms, we use a hybrid task and joint space impedance controller [30].

Simulation Setup. Isaac Sim [31] is used to train teacher policies for the occluded grasping task. To train policies, 48 objects selected from the Google Scanned Objects dataset [32] are spawned into

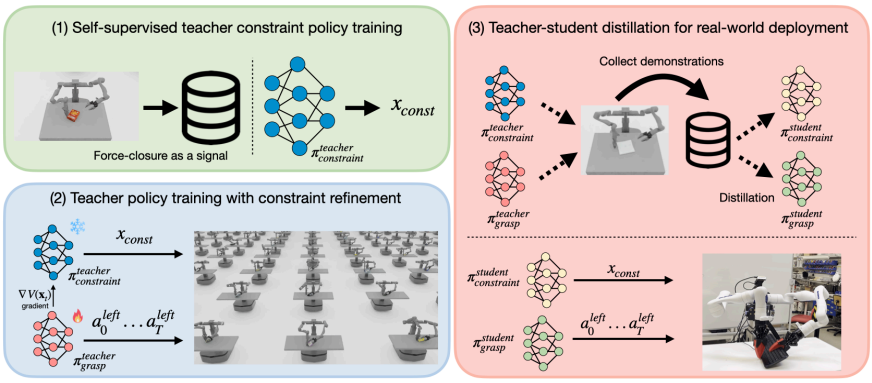


Figure 3: **Method Overview.** (1) *COMBO-Grasp* first collects a synthetic dataset in a self-supervised manner in simulation to train a state-based teacher constraint policy that outputs a right arm end-effector pose. (2) This constraint policy is frozen, and a state-based teacher grasping policy $\pi_{teacher}$ is trained with RL. To improve performance, we propose value-guided policy coordination, refining the constraint output via gradients from a jointly trained value function. (3) Both teacher policies are then distilled into vision-based student policies using point clouds, proprioception, and optionally a desired grasp pose to tackle real-world bimanual occluded grasping.

the environment (see Figure 9). We use an operational space controller [33] to control robot arms. Further information regarding the simulation setup can be found in Appendix B.

4 Approach

In this section we present *COMBO-Grasp*, a system designed to solve challenging bimanual occluded grasping tasks. COMBO-Grasp utilises two coordinated policies: a constraint policy trained on a dataset collected without human supervision within a simulation to stabilise the target object using one arm, and a grasping policy trained using RL to control the other arm and reorient the object for successful grasping.

We first present a self-supervised data collection method in simulation (Section 4.1) to train the teacher constraint policy (Section 4.2). Section 4.3 details the training of the teacher grasping policy, including value function-guided coordination for refining constraint poses. Finally, teacher-student distillation for real-world deployment is described in Section 4.5.

4.1 Self-Supervised Data Collection for Constraint Policy

Instead of relying on costly expert demonstrations, this work introduces a self-supervised data collection method using force-closure signals in simulation to train the constraint policy across diverse objects (see Figure 3 (1)). Target occluded grasp poses are generated via antipodal sampling [34] for 48 objects from the Google Scanned Objects dataset [32] (see Figure 9 in Appendix B). These poses are also used during RL training for the grasping policy (Section 4.3).

End-effector poses for the right arm are randomly sampled near the target object placed on a table, while the left arm remains fixed in its initial position. A force of $25N \times$ mass along the approach vector of a desired grasp pose is applied to the object. To assess force closure, the object's velocity is used as an approximation, as accurately evaluating force closure is often challenging, particularly in scenarios involving multiple contacts. Instead, force closure is considered successful if, after applying force, the object's velocity remains below a predefined threshold, given the specific grasp and constraint poses. The sampled end-effector pose, the corresponding desired grasp pose, and the object pose are then added to the dataset. By iterating this process in simulation, $3K$ constraint poses per object are collected. With 48 objects, this results in a total of $144K$ samples. By leveraging the object's motion as a proxy measure for the success of a constraint pose, we can generate a rich set of training data to train the constraint policy.

4.2 Teacher Constraint Policy Training

One of the central contributions of this work lies in value function-guided policy coordination, which builds upon classifier guidance used in diffusion models to refine the generated constraint pose during the training of the state-based teacher grasping policy. This is achieved by employing a diffusion model for the state-based teacher constraint policy, denoted as $\pi_{const}^{teacher}$, trained from the privileged information in the dataset (see Section 4.1). This approach leverages gradients from the value function to steer the teacher constraint policy's output, optimising the stabilising poses to align with the grasping policy's objectives to improve task performance and sample efficiency.

The teacher constraint policy uses a diffusion model formulated as a Denoising Diffusion Probabilistic Model (DDPM) [35]. Starting from x^K sampled from Gaussian noise, the DDPM performs K denoising iterations to generate a series of intermediate samples with decreasing levels of noise, $x^k, x^{k-1}, ...x^0$. To train the constraint policy, a forward diffusion process is applied to add noise to an unmodified sample, x^0, from the dataset by randomly sampling a denoising iteration k and random noise ϵ^k. The noise prediction model ϵ_θ is then trained to estimate the noise added to a sample during the forward diffusion process. Thus, the training loss is formulated as

$$\mathcal{L}_{constraint} = \mathit{MSE}(\epsilon^k, \epsilon_\theta(\mathbf{x}^0_{const} + \epsilon^k, k)) \tag{1}$$

where $\mathbf{x}_{const}$ is the constraint pose for the right arm. An MLP-based denoising model is used as the backbone for the diffusion policy (see Appendix B.1 for further details of the architecture).

The constraint policy takes as input the object pose, desired grasp pose, and object IDs. To represent Object IDs, an autoencoder [36] is trained to reconstruct object point clouds using the Chamfer distance. The resulting compact latent code replaces one-hot vectors, reducing observation dimensionality for large object sets. The state-based teacher constraint policy is used only during teacher grasping policy training (Section 4.3) and is distilled into a vision-based student policy for sim-to-real transfer.

4.3 Teacher Grasping Policy

After the constraint policy is trained, a teacher grasping policy $\pi_{grasp}^{teacher}$ is trained using Proximal Policy Optimisation (PPO) [2] on diverse objects from privileged information in simulation. To train a robust teacher grasping policy capable of performing in real-world environments, we employ domain randomisation, incorporating additive Gaussian noise into low-dimensional observations, as well as randomising the physics parameters of the target object and the controller parameters during policy training. For further information about the domain randomisation, see Appendix D.1. The teacher grasping policy receives as input the robot's proprioceptive states, object pose, object velocity, desired grasp poses, object IDs (see Section 4.2), object's mass and friction parameters, and the PID gains for the operational space control(OSC).

At the beginning of each training episode, the teacher constraint policy $\pi_{const}^{teacher}$ generates a constraint end-effector pose $\mathbf{x}_{const}$ for the right arm. Given the constraint end-effector pose, the joint positions of the right arm are computed using the CuRobo IK solver [37]. Then, the right arm moves to the computed desired constraint joint positions. Once the right arm is positioned, the grasping policy controls the left arm to attempt the occluded grasping task.

We design a reward function with six components: (1) position and (2) orientation distance to the target grasp pose for a left end-effector, (3) action penalty to penalise the large actions, (4) collision penalty (including self- and table collisions), (5) lift reward to expose occluded grasps, and (6) sparse grasp success reward. The collision penalty term is computed using the signed distance field provided by CuRobo. The final reward r is

$$r = \alpha_1 r_{dist\_pos} + \alpha_2 r_{dist\_ori} - \alpha_3 r_{collision} - \alpha_4 r_{action} + \alpha_5 r_{lift} + \alpha_6 r_{success} \tag{2}$$

where α_i is a coefficient for each reward term. For more details on teacher policy training, domain randomisation, and each reward term with the coefficient value, see Appendix B.

4.4 Value Function-guided Policy Coordination

A key aspect of *COMBO-Grasp* is to induce effective bimanual coordination using the trained constraint policy, thereby improving task performance and enhancing the sample efficiency of the RL policy training. Since the teacher constraint policy is initially trained on datasets collected using force closure as a signal, it does not inherently guarantee the generation of an optimal constraint for the grasping policy. To address this limitation, *COMBO-Grasp* draws inspiration from classifier guidance in diffusion models and we propose value function-guided policy coordination that refines the generated constraint pose using gradients from a value function $V(\mathbf{x}_t)$, which is trained alongside the grasping policy using RL. The value function of the grasping policy acts as a classifier in the classifier guidance framework, and the gradients for guidance are obtained by maximising the estimated value. This approach effectively refines the generated constraint poses to align more closely with the grasping policy's requirements, leading to improved overall performance and sample efficiency. By incorporating gradients from the value function by maximisation, the denoising process for the constraint policy is formulated as

$$\mathbf{x}_{const}^{k-1} = \alpha(\mathbf{x}_{const}^{k} - \gamma\epsilon_\theta(\mathbf{x}_{const}^{k}, k) - w\nabla V(\mathbf{x}) + \mathcal{N}(0, \sigma^2 I)) \tag{3}$$

where w is a scaling parameter, $\mathbf{x}$ is low-dimensional observation used as input to the value function $V(\cdot)$, and the constraint pose $\mathbf{x}_{const}$ is a subset of the input state $\mathbf{x}$ for the value function (i.e., $\mathbf{x}_{const} \in \mathbf{x}$). For further details on value function-guided policy coordination, see Appendix B.

4.5 Policy Distillation for Sim-to-Real Transfer

To deploy policies in real-world environments, leveraging visual observations as input is essential. Teacher-student policy distillation [38, 39] is used to transfer knowledge from trained teacher constraints and grasping policies to student policies. These student policies process point cloud observations along with state information, such as proprioceptive data and, optionally, a desired grasp pose. In *COMBO-Grasp*, we adopt a diffusion policy as the student grasping policy, similar to prior work [40]. Specifically, DP3 [40] and MLP encoders process point cloud and state observations, respectively, as illustrated in Figure 11 (Appendix C). The encoder outputs are concatenated to condition the diffusion policy. For simplicity, the student constraint policy employs a Gaussian Mixture Model (GMM). Unlike the teacher constraint policy, it does not require output steering, making the GMM approach effective and straightforward.

To distil the teacher to the student policy, we rollout the teacher in simulation and collect $10K$ expert demonstrations with visual observations. During distillation, we apply small perturbations to point cloud observations to simulate real-world noise. For further details, see Appendix C.

5 Experimental Results: Simulation

Our experiments address the following questions: (1) How successful is *COMBO-Grasp* in learning a teacher policy compared to competitive baselines? (2) How well does *COMBO-Grasp* generalise to unseen objects? (3) How does the value function-guided policy coordination affect *COMBO-Grasp*'s overall performance? For further analysis of the experiments, see Appendix A.

5.1 Evaluation Metric and Baselines

For evaluation, we assess the success rate of grasping. In particular, a trial is considered successful if the robot's left arm securely grasps and lifts the target object at least $8\ cm$ at the end of the episode.

We compare *COMBO-Grasp* with the following baselines:

- **PPO**: A PPO [41] policy that controls both arms. The policy outputs 12-dimensional actions. Compared to *COMBO-Grasp* which employs two coordinated policies, this baseline requires more extensive exploration to solve the task.

- **PPO + Constraint Reward**: A PPO policy trained with a modified reward function that adds a distance-based term between the right end-effector and the target object's center. This encourages the right arm to act as a constraint, assisting the left arm in grasping. The policy thus avoids undesirable behaviors seen with the original reward, such as high-velocity grasps by the left end-effector without support.
- ***COMBO-Grasp* with a fixed constraint**: A PPO policy is trained to control the left arm, while the right arm remains fixed in a predefined pose in contrast to *COMBO-Grasp*. This showcases the importance of a constraint policy.
- ***COMBO-Grasp* without refinmenet**: *COMBO-Grasp* without value function-guided policy coordination. This demonstrates the necessity of refining the constraint pose generated by the constraint policy to further improve performance.

5.2 Sample Efficiency in Teacher Policy Training

***COMBO-Grasp* achieves higher performance and sample efficiency.** We first evaluate teacher policy training in simulation. As shown in Fig. 4, *COMBO-Grasp* solves the occluded grasping task more efficiently and achieves better overall performance. In contrast, *PPO* struggles due to task and system complexity. More critically, it often exhibits unrealistic behaviours—e.g., the left arm grasping aggressively without right-arm support—by exploiting simulator inaccuracies, which fail to transfer to the real world.

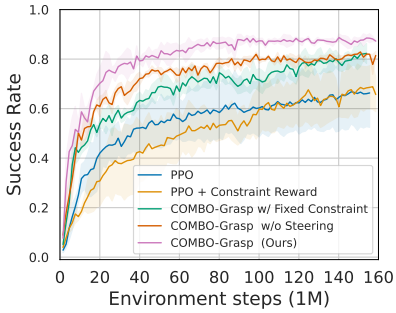


Figure 4: **Teacher policy training.** We run 3 seeds per method, with shaded regions showing standard deviation. *COMBO-Grasp* significantly outperforms baselines in both performance and sample efficiency.

Reward shaping alone is insufficient for coordination. *PPO + Constraint Reward* partially addresses these issues using a distance-based reward, but effective constraint poses are not known a priori and depend on coordinated behaviour between arms. This highlights the difficulty of inducing such coordination through reward engineering alone.

Constraint learning and coordination drive performance. *COMBO-Grasp* with fixed constraints performs poorly, as static poses may not generalise across tasks. Similarly, removing refinement degrades performance. These findings emphasise the importance of both pre-training and refining the constraint policy. In general, our coordinated approach, learning separate constraints and grasping policies, yields faster training and higher success rates than the RL baselines of a single policy.

5.3 Student Policy Performance in Simulation

***COMBO-Grasp* generalises well to both seen and unseen objects.** We evaluate the distilled student policies in simulation (Fig.5). *COMBO-Grasp* handles occluded grasping effectively across object sets. Without the target grasp pose as input, performance drops but remains competitive.

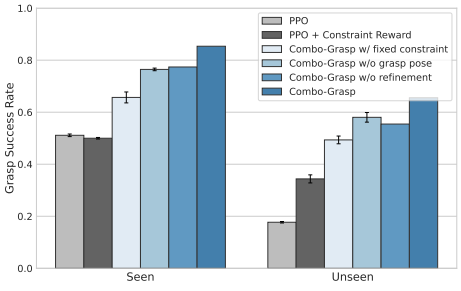


Figure 5: Student Policy Performance averaged over 3 seeds in Simulated environments. We evaluate each approach for 50 times using both seen and unseen objects.

Coordinated strategies improve generalisation to unseen objects. While *PPO* and *PPO + Constraint Reward* perform similarly on seen objects, the latter significantly outperforms on unseen ones by leveraging the right arm as a constraint instead of exploiting simulator flaws. However, it still lags behind *COMBO-Grasp* due to the difficulty of reward shaping for effective constraint learning, which limits the teacher policy and thus the student's performance.

6 Experimental Results: Real-World

We evaluate a student policy trained on simulated data in real-world settings to address (1) How does *COMBO-Grasp* perform on seen and unseen real-world objects? (2) Does conditioning on a desired grasp pose improve its performance?

6.1 Experiment Setup

Student policies are evaluated on both seen and unseen objects with diverse shapes, sizes, and weights (Figure 6). To facilitate grasping, we scan objects to reconstruct 3D meshes and generate grasp poses via antipodal sampling, avoiding the need for grasp pose prediction models [12], which are outside our evaluation scope. However, *COMBO-Grasp* is compatible with any grasp pose prediction models. When student policies are conditioned on desired grasp poses, object pose is estimated in real-time using FoundationPose[42], enabling grasp pose inference during manipulation [43].

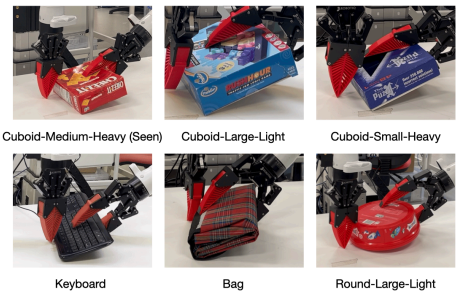


Figure 6: Selected objects of varying sizes and weights requiring occluded grasping are used to evaluate *COMBO-Grasp* in the real world.

Constraint poses from the student constraint policy are converted to joint positions using CuRobo's IK solver [37], and MoveIt [44] controls the right arm accordingly. Once positioned, the left arm executes the student grasping policy.

6.2 Results

***COMBO-Grasp* is effective in real-world occluded grasping, with trade-offs depending on input.** As shown in Table 1, *COMBO-Grasp* handles occluded grasping well for both seen and unseen objects. It struggles with the round box due to stability challenges, and performance slightly declines without the target grasp pose. In this setting, the policy cannot recover from failed nonprehensile manipulation; for instance, pushing a keyboard often fails due to its thin shape, leading to a 40% success rate.

	COMBO-Grasp	w/o grasp pose
Cuboid-Medium-Heavy (Seen)	80% (8/10)	80% (8/10)
Cuboid-Large-Light	90% (9/10)	80% (8/10)
Cuboid-Small-Heavy	50% (5/10)	60%(6/10)
Keyboard	80% (8/10)	40%(4/10)
Bag	80% (8/10)	80% (8/10)
Round-Large-Light	30% (3/10)	10% (1/10)
Average	**68.3% (41/60)**	58.3% (35/60)

Table 1: Performance of *COMBO-Grasp* in real-world environments for seen and unseen objects with varying shapes, sizes, and weights.

The target grasp pose improves robustness, but removing it increases practicality. Providing the desired grasp pose enables retries and improves success by guiding the left arm more effectively. However, omitting it increases deployment flexibility, removing the need for real-time pose estimation, which is useful in environments where tracking is infeasible. The complete baseline results, including *PPO* and various ablations of *COMBO-Grasp*, are presented in Table 2 in Appendix A.3.

7 Conclusion

We present *COMBO-Grasp*, a bimanual robotic system for occluded grasping tasks. By introducing a constraint policy and value function-guided policy coordination, which refines the constraint pose using value gradients, we show that coordinated policies efficiently solve challenging occluded grasping tasks. Furthermore, the trained teacher policies are then distilled into vision-based student policies for real-world deployment. Through empirical evaluation, we show that *COMBO-Grasp* achieves significantly better performance compared to a state-of-the-art baseline and instantiations of *COMBO-Grasp* in both simulated and real-world environments.

8 Limitations

COMBO-Grasp offers notable improvements in learning efficiency and generalisation compared to baselines and prior occluded grasping methods. However, there are some limitations to consider. Firstly, *COMBO-Grasp* struggles with unseen objects of significantly different shapes, which could be addressed by training the teacher and student policy with a more diverse set of geometries. Additionally, *COMBO-Grasp* faces challenges with round objects in the real world, where stabilisation during occluded grasping is difficult. This issue could be mitigated through a closed-loop control approach, such as learning a residual policy for real-time constraint pose adjustments.

Finally, while COMBO-Grasp is tailored for bimanual occluded grasping, we view this as a foundational step toward solving a broader range of bimanual tasks, such as threading a needle, painting, or cutting—where one arm must constrain the object while the other performs precise ma- nipulation. We see COMBO-Grasp with value-guided implicit coordination as a step towards efficiently solving this class of problem.

References

[1] A. Zhan, R. Zhao, L. Pinto, P. Abbeel, and M. Laskin. Learning visual robotic control efficiently with contrastive pre-training and data augmentation, 2022.

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[3] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[4] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[5] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid. Aloha unleashed: A simple recipe for robot dexterity, 2024. URL https://arxiv.org/abs/2410.13126.

[6] L. B. Bagesteiro and R. L. Sainburg. Handedness: dominant arm advantages in control of limb dynamics. *Journal of neurophysiology*, 88(5):2408–2421, 2002.

[7] L. Bagesteiro and R. Sainburg. Nondominant arm advantages in load compensation during rapid elbow joint movements. *Journal of neurophysiology*, 90:1503–13, 10 2003. doi:10.1152/jn.00189.2003.

[8] M. Drolet, S. Stepputtis, S. Kailas, A. Jain, J. Peters, S. Schaal, and H. B. Amor. A comparison of imitation learning algorithms for bimanual manipulation. *IEEE Robotics and Automation Letters*, 2024.

[9] J. Yamada, J. Collins, and I. Posner. Efficient skill acquisition for complex manipulation tasks in obstructed environments, 2023.

[10] J. Collins, M. Robson, J. Yamada, M. Sridharan, K. Janik, and I. Posner. Ramp: A benchmark for evaluating robotic assembly manipulation and planning. *IEEE Robotics and Automation Letters*, 2023.

[11] W. Yuan, A. Murali, A. Mousavian, and D. Fox. M2t2: Multi-task masked transformer for object-centric pick and place. *arXiv preprint arXiv:2311.00926*, 2023.

[12] A. Mousavian, C. Eppner, and D. Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2901–2910, 2019.

[13] K. R. Barad, A. Orsula, A. Richard, J. Dentler, M. Olivares-Mendez, and C. Martinez. Graspldm: Generative 6-dof grasp synthesis using latent diffusion models. *IEEE Access*, 2024.

[14] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pages 651–673. PMLR, 2018.

[15] L. Wang, Y. Xiang, W. Yang, A. Mousavian, and D. Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *Conference on Robot Learning*, pages 70–80. PMLR, 2022.

[16] A. Zhou, M. J. Kim, L. Wang, P. Florence, and C. Finn. Nerf in the palm of your hand: Corrective augmentation for robotics via novel-view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17907–17917, 2023.

[17] S. Song, A. Zeng, J. Lee, and T. A. Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *IEEE Robotics and Automation Letters*, 5:4978–4985, 2019. URL https://api.semanticscholar.org/CorpusID:209140715.

[18] Z. Sun, K. Yuan, W. Hu, C. Yang, and Z. Li. Learning pregrasp manipulation of objects from ungraspable poses, 2020.

[19] W. Zhou and D. Held. Learning to grasp the ungraspable with emergent extrinsic dexterity. In *Conference on Robot Learning*, pages 150–160. PMLR, 2023.

[20] T. Lin, Z.-H. Yin, H. Qi, P. Abbeel, and J. Malik. Twisting lids off with two hands. *arXiv preprint arXiv:2403.02338*, 2024.

[21] B. Huang, Y. Chen, T. Wang, Y. Qin, Y. Yang, N. Atanasov, and X. Wang. Dynamic handover: Throw and catch with bimanual hands, 2023.

[22] L. Shao, T. Migimatsu, and J. Bohg. Learning to scaffold the development of robotic manipulation skills. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5671–5677, 2020. doi:10.1109/ICRA40945.2020.9197134.

[23] R. Chitnis, S. Tulsiani, S. Gupta, and A. Gupta. Efficient bimanual manipulation using learned task schemas. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1149–1155. IEEE, 2020.

[24] R. Chitnis, S. Tulsiani, S. Gupta, and A. Gupta. Intrinsic motivation for encouraging synergistic behavior. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJleNCNtDH.

[25] Y. Li, C. Pan, H. Xu, X. Wang, and Y. Wu. Efficient bimanual handover and rearrangement via symmetry-aware actor-critic learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3867–3874, 2023. doi:10.1109/ICRA48891.2023.10160739.

[26] R. L. Sainburg. Evidence for a dynamic-dominance hypothesis of handedness. *Experimental Brain Research*, 142:241–258, 2001. URL https://api.semanticscholar.org/CorpusID:206924666.

[27] J. Grannen, Y. Wu, B. Vu, and D. Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=86aMPJn6hX9F.

[28] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots, 2024. URL https://arxiv.org/abs/2402.10329.

[29] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[30] M. J. Kim, F. Beck, C. Ott, and A. Albu-Schäffer. Model-free friction observers for flexible joint robots with torque measurements. *IEEE Transactions on Robotics*, 35(6):1508–1515, 2019. doi:10.1109/TRO.2019.2926496.

[31] NVIDIA. Nvidia isaac sim. URL https://developer.nvidia.com/isaac-sim.

[32] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items, 2022. URL https://arxiv.org/abs/2204.11918.

[33] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[34] C. Eppner, A. Mousavian, and D. Fox. A billion ways to grasp: An evaluation of grasp sampling schemes on a dense, physics-based grasp data set. In *The International Symposium of Robotics Research*, pages 890–905. Springer, 2019.

[35] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[36] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X.

[37] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, et al. Curobo: Parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023.

[38] J. Yamada, M. Rigter, J. Collins, and I. Posner. Twist: Teacher-student world model distillation for efficient sim-to-real transfer. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9190–9196. IEEE, 2024.

[39] J. Brosseit, B. Hahner, F. Muratore, M. Gienger, and J. Peters. Distilled domain randomization. *arXiv preprint arXiv:2112.03149*, 2021.

[40] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.

[41] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[42] B. Wen, W. Yang, J. Kautz, and S. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024.

[43] W. Wan, H. Geng, Y. Liu, Z. Shan, Y. Yang, L. Yi, and H. Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning, 2023. URL https://arxiv.org/abs/2304.00464.

[44] D. Coleman, I. A. Sucan, S. Chitta, and N. Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *ArXiv*, abs/1404.3785, 2014. URL https://api.semanticscholar.org/CorpusID:13939653.

[45] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[46] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

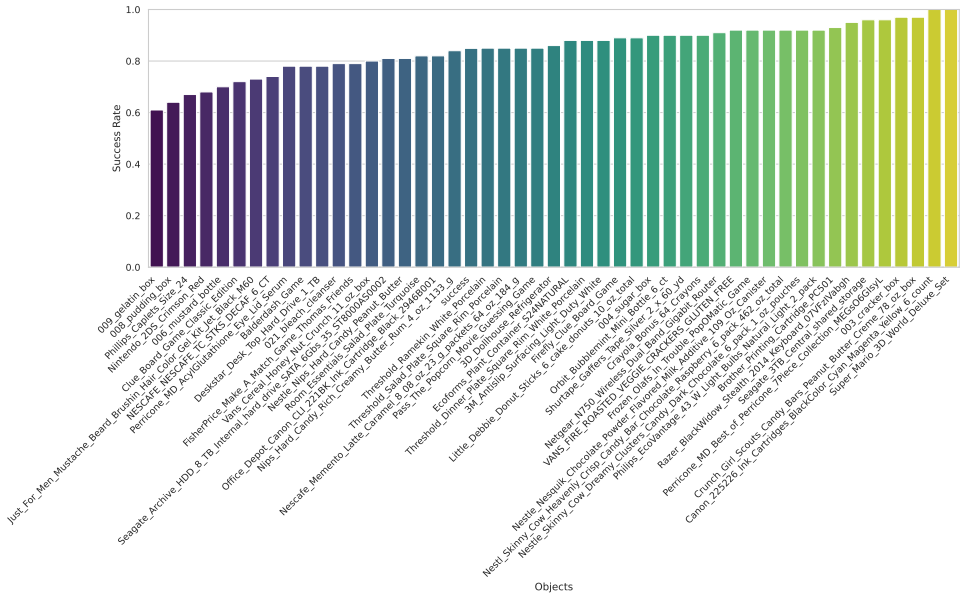


Figure 8: **Grasp success rates of *COMBO-Grasp*'s student policies.** The success rate is reported for each object in simulation, averaged over 50 trials per object.

A Additional Analysis for Experiments

A.1 Ablation of the Value Function-guided Policy Coordination

The degree to which the value function-guided policy coordination improves the task success rate is investigated here. Concretely, the impact of the scaling parameter w on the constraint diffusion policy (see Eq. 3) during teacher policy training is investigated. As illustrated in Figure 7, the teacher policy's performance decreases when value function policy coordination is not applied (i.e., $\lambda = 0$). On the other hand, incorporating value function policy coordination consistently enhances the teacher policy's overall performance. This finding suggests that the constraint policy occasionally generates constraint poses that are suboptimal for the grasping policy. Consequently, value function policy coordination promotes on-the-fly adjustments and this is cooperation between the two arms achieves higher success rates.

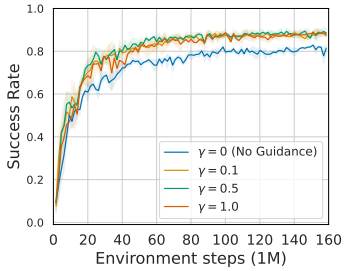


Figure 7: **Guidance scaling ablation.** We compare the guidance scaling parameter to steer the output of the constraint policy. This result indicates that *COMBO-Grasp* without guidance shows worse performance and *COMBO-Grasp*is robust to a wide range of guidance scaling parameters to achieve better performance.

A.2 Student Policy Performance per Object

Figure 8 illustrates the success rate of *COMBO-Grasp* for each object used during training. While *COMBO-Grasp* demonstrate performant success rate across diverse objects, the occluded grasp performance for small objects or objects with complex geometries is reduced when compared to that of large objects with simple geometries. In order to overcome this limitation, it is suggested that both teacher and student policies be trained using more diverse objects, such as those available in the Objaverse datasets [45].

	COMBO-Grasp	w/o grasp pose	w/ fixed constraint	w/o refinement	PPO
Cuboid-Medium-Heavy (Seen)	80% (8/10)	80% (8/10)	60% (6/10)	60% (6/10)	60% (6/10)
Cuboid-Large-Light	90% (9/10)	80% (8/10)	40%(4/10)	30% (3/10)	30% (3/10)
Cuboid-Small-Heavy	50% (5/10)	60% (6/10)	50% (5/10)	50% (5/10)	40% (4/10)
Keyboard	80% (8/10)	40%(4/10)	40% (4/10)	30% (3/10)	10% (1/10)
Bag	80% (8/10)	80% (8/10)	60% (4/10)	80% (8/10)	40% (4/10)
Round-Large-Light	30% (3/10)	10% (1/10)	0% (0/10)	10% (1/10)	0% (0/10)
Average	**68.3**% (**41**/**60**)	58.3% (35/60)	38.3%(23/60)	43.3%(26/60)	30.0% (18/60)

Table 2: Performance of *COMBO-Grasp* in real-world environments for seen and unseen objects with varying shapes, sizes, and weights.

A.3 Real-world Experiments

Table 2 presents the full results of the real-world experiments, including comparisons with all baseline methods. We observe that baseline approaches often fail to solve the tasks due to poor coordination between the left and right arms, likely because such coordination is not adequately learned during training in simulation, and consequently does not transfer well to real-world environments.

B Teacher Policy Details

B.1 Teacher Constraint Policy

We employ a diffusion policy [46] as the basis for the teacher constraint policy. The diffusion policy is implemented using a Denoising Diffusion Probabilistic Model (DDPM), with a multi-layer perceptron (MLP)-based backbone. The denoising model is built on a three-level UNet architecture, comprising residual blocks with a hidden layer size of 512. The diffusion time step is encoded as an 80-dimensional feature vector. Additionally, the desired grasp pose, $\mathbf{x} \in \mathbb{R}^9$, and the object's ID, $\mathbf{x}_{obj\_id} \in \mathbb{R}^{16}$, are encoded into an 80-dimensional vector respectively to provide task-specific context. Similarly, the noisy input representing the constraint pose is encoded into another 80-dimensional vector. These encoded vectors are summed and passed through the residual blocks. The denoising model outputs the noise added to the original input during the forward diffusion process. In this work, we use 100 diffusion time steps for both training and inference. We train the diffusion policy using an Adam optimiser with a learning rate of 1×10^{-4}.

B.2 Teacher Grasping Policy

We train a teacher grasping policy using Proximal Policy Optimisation (PPO). An actor network consists of an MLP with 2 hidden layers of sizes $[256, 256]$. The actor network is parameterized as a Gaussian distribution with a fixed, state-independent standard deviation. The critic network consists of an MLP with 3 hidden layers of sizes $[256, 256, 256]$.

We define the privileged information used to train the policy as $[\mathbf{x}_{\text{robot}}, \mathbf{x}_{\text{goal}}, \mathbf{x}_{\text{obj}}] \in \mathbb{R}^{64}$. The robot proprioceptive states, $\mathbf{x}_{\text{robot}}$, include the left end-effector pose, $\mathbf{x}_{\text{left}} \in \mathbb{R}^9$, the right end-effector pose, $\mathbf{x}_{\text{right}} \in \mathbb{R}^8$, and the translational and rotational action scale parameters for the operational space controller, $\mathbf{x}_{\text{control}} \in \mathbb{R}^2$. The right end-effector states, $\mathbf{x}_{right}$, exclude the z-coordinate position, as the table height remains constant, and the constraint pose is fixed at a predetermined z-coordinate. The goal-related states, $\mathbf{x}_{\text{goal}}$, consist of the desired grasp pose, $\mathbf{x}_{\text{grasp}} \in \mathbb{R}^7$, the distance between the left end-effector and the desired grasp position, $\mathbf{x}_{\text{dist}} \in \mathbb{R}^3$, and the orientation distance between the left end-effector and the desired grasp orientation in the axis-angle representation, $\mathbf{x}_{\text{dist\_ori}} \in \mathbb{R}^3$. The object states, $\mathbf{x}_{\text{obj}}$, comprise the object pose, $\mathbf{x}_{\text{obj\_pose}} \in \mathbb{R}^7$, the object velocity, $\mathbf{x}_{\text{obj\_vel}} \in \mathbb{R}^6$, the friction parameters, $\mathbf{x}_{\text{friction}} \in \mathbb{R}^2$, the object's mass, $x_{\text{mass}} \in \mathbb{R}^1$, and the object's ID, $\mathbf{x}_{\text{obj\_id}} \in \mathbb{R}^{16}$.

We train the policy using an Adam optimiser with an adaptive learning rate scheduler[1] based on the KL divergence between the current policy and the previous policy, whose maximum learning rate is 1×10^{-2} and the minimum is 1×10^{-6}. We use a discount factor of 0.99, a GAE lambda value of

[1] https://skrl.readthedocs.io/en/latest/api/resources/schedulers/kl_adaptive.html

Figure 9: **Training objects.** We choose 48 training objects from the Google Scanned Object Dataset [32].

Figure 10: **Test objects.** We evaluate 10 held-out objects from the Google Scanned Object Dataset.

0.95, and an entropy coefficient of $6e-3$. After each policy rollout, the policy is updated using a batch size of 2048 for 8 epochs.

B.3 Reward function

The reward function used in our experiments comprises six terms and is defined as follows:

$$\begin{aligned} r = \alpha_1 r_{dist\_pos} + \alpha_2 r_{dist\_ori} - \alpha_3 r_{collision} \\ - \alpha_4 r_{action} + \alpha_5 r_{lift} + \alpha_6 r_{success} \end{aligned} \tag{4}$$

where the weighting coefficients are set to $\alpha_1 = 0.2$, $\alpha_2 = 0.2$, $\alpha_3 = 1.0$, $\alpha_4 = 0.025$, $\alpha_5 = 0.1$, and $\alpha_6 = 40$. Each term in the reward function serves a distinct purpose in guiding the robot's behaviour:

- **Position Distance Reward (r_{dist_pos}):** This term incentivizes the left end-effector to move towards the desired grasp position. It is computed as:

$$r_{\text{dist\_pos}} = 1 - \tanh(4 \cdot ||\mathbf{p}_{\text{left}} - \mathbf{p}_{\text{grasp}}||_2), \tag{5}$$

  where $\mathbf{p}_{\text{left}} \in \mathbb{R}^3$ and $\mathbf{p}_{\text{grasp}} \in \mathbb{R}^3$ represent the current and desired positions of the left end-effector, respectively.

- **Orientation Distance Reward (r_{dist_ori}):** This term encourages the left end-effector to align its orientation with the desired grasp orientation. The orientation difference is measured in the axis-angle space‘. The reward is computed as:

$$r_{\text{dist\_ori}} = 1 - \tanh(0.2 \cdot ||\boldsymbol{\theta}_{\text{left}} - \boldsymbol{\theta}_{\text{grasp}}||_2), \tag{6}$$

  where $\boldsymbol{\theta}_{\text{left}} \in \mathbb{R}^3$ and $\boldsymbol{\theta}_{\text{grasp}} \in \mathbb{R}^3$ represent the axis-angle representations of the current and desired orientations of the left end-effector, respectively.

- **Action Penalty (r_{action}):** This term discourages large control commands by penalizing the magnitude of the action vector:

$$r_{\text{action}} = ||\mathbf{a}||_2. \tag{7}$$

- **Collision Penalty ($r_{\text{collision}}$):** To prevent self-collisions and contact with the table, we compute the signed distance (SD) using CuRobo [37]. The collision penalty is given by:

$$r_{\text{collision}} = SD_{\text{self\_col}} + SD_{\text{table}}. \tag{8}$$

The signed distance is computed for the robot arms, excluding the grippers, since the grippers must make contact with the table for occluded grasping problems. In CuRobo, a positive signed distance indicates a collision.

- **Lift Reward (r_{lift}):** This term encourages lifting the object to expose an initially occluded grasp pose. It is defined as an indicator function:

$$r_{\text{lift}} = \mathbb{1}(z_{\text{grasp}} > z_{\text{grasp,init}} + 2\text{ cm}), \tag{9}$$

where z_{grasp} and $z_{\text{grasp,init}}$ denote the current and initial heights of the desired grasp position, respectively.

- **Grasp Success Reward ($r_{success}$):** At the end of an episode, a reward of 1 is assigned if the left arm successfully grasps and lifts the object; otherwise, the reward is 0:

$$r_{\text{success}} = \begin{cases} 1, & \text{if grasp and lift are successful,} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

C Student Policy Details

We describe the architecture of the student constraint and grasping policy, as shown in Fig. 11.

C.1 Studnet Constraint Policy

The student constraint policy integrates the DP3 encoder [40] and a state encoder to process point cloud and state observations, respectively.

The DP3 encoder comprises three fully connected layers with dimensions of $[128, 256, 384]$, followed by a max pooling operation and a final fully connected layer of size 64. Layer normalization and ReLU activations are applied after each of the initial three layers preceding the max pooling operation. The state encoder consists of two hidden layers with dimensions of $[128, 256]$. The state encoder outputs a feature vector of size 32 given the desired grasp pose $\mathbf{x}_{grasp}$.

The feature vectors produced by the DP3 and state encoders are concatenated and subsequently processed through a MLP to generate a constraint pose. For this work, the student policy utilizes a Gaussian Mixture Model (GMM)-based approach due to its simplicity and effectiveness. Specifically, the GMM-based policy employs 5 modes, with a minimum standard deviation of 1×10^{-4}. We employ an AdamW optimiser with a learning rate of 5×10^{-5} and a weight decay of 5×10^{-5}.

C.2 Student Grasping Policy

We adopt the 3D Diffusion Policy (DP3) [40] as the foundation for the student grasping policy. The architecture of the DP3 encoder and the state encoder is consistent with that employed in the student constraint policy. However, the weights of these encoders are independently initialized from those of the constraint policy. Furthermore, the input dimension for the state encoder in the manipulation policy differs from that of the constraint policy. The state encoder for the manipulation policy processes $\mathbf{x}_{robot}$ and optionally $\mathbf{x}_{grasp}$ as input. During training, we employ 100 diffusion timesteps, whereas during inference a Denoising Diffusion Implicit Model (DDIMs) is used with 10 diffusion timesteps to accelerate action generation. We use an AdamW optimiser with a learning rate of 5×10^{-5} and a weight decay of 5×10^{-5}.

D Simulation Setup

D.1 Training

In order to train a teacher policy from a diverse set of objects, we select 48 objects from the Google Scanned Object dataset, as illustrated in Figure 9. To train teacher policies efficiently, we spawn 1024 robots and objects in the simulated environment.

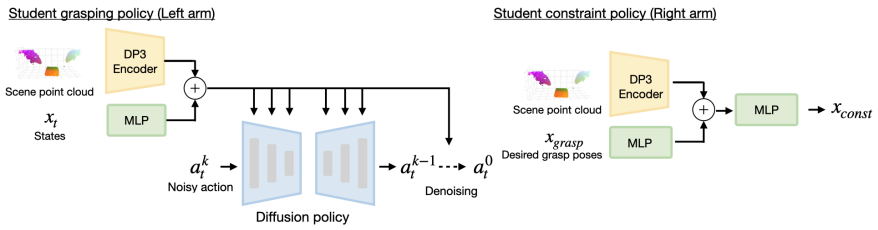


Figure 11: **Student policy architecture.** We utilize DP3 [40] as the backbone for the grasping policy. The DP3 encoder processes the scene point cloud, and its output is concatenated with a state feature vector obtained by a multi-layer perceptron (MLP). The resulting concatenated vector serves as the conditioning input for the diffusion-based policy. Similarly, the constraint student policy employs the DP3 encoder and an MLP, but it takes a desired grasp pose as input. Unlike the grasping policy, the constraint student policy employs a Gaussian Mixture Model (GMM)-based policy.

In order to train a policy robust to noises and effectively transfer it to real-world environments, we apply domain randomisation during teacher policy training. Table 3 describes the details of the randomisations used in our experiments. We also apply domain randomisation during the self-supervised data collection for the constraint policy.

Table 3: Domain Randomisation Hyperparameters

Parameter	Description
Initial robot joint positions	Add noise sampled from $\mathcal{N}(0, 0.05)$
Robot base position	Add random noise sampled from $\mathcal{U}(-0.015, 0.015)$ to the z-coordinate of the robot base
PID position action scale	Sampled from $\mathcal{U}(0.03, 0.04)$
PID rotation action scale	Sampled from $\mathcal{U}(0.1, 0.2)$
Action	Add random noise sampled from $\mathcal{N}(0, 0.01)$
Object mass	Add mass sampled from $\mathcal{U}(-0.1, 0.1)$
Static and dynamic friction	Sampled from $\mathcal{U}(0.8, 1.2)$
Grasp position	Add random noise sampled from $\mathcal{N}(0, 0.005)$
Grasp translational distance	Add random noise sampled from $\mathcal{N}(0, 0.005)$
Grasp rotational distance	Add random noise sampled from $\mathcal{N}(0, 0.005)$
End-effector position	Add random noise sampled from $\mathcal{N}(0, 0.01)$
Object position	Add random noise sampled from $\mathcal{N}(0, 0.01)$
Object orientation	Add random noise sampled from $U(-0.2\pi \text{ rad}, 0.2\pi \text{ rad})$ to the yaw axis

D.2 Evaluation

To evaluate policies for both seen and novel objects, we also select 10 held-out objects from the Google Scanned Object dataset (see Figure 10).

E Real-World Experiment Setup

E.1 Input Observation for Student Policies

The distilled student policies take point clouds as input in real-world environments. We render depth images with the size of 640×480 from a Realsense L515 camera to reconstruct point cloud observations. Similar to [40], we crop the point cloud within a pre-defined bounding box such that it includes the robot arms and the target object. Then, we remove statistical outliers from the point clouds reconstructed from depth images and apply farthest point sampling to sub-sample 1024 points.

E.1.1 Desired Occluded Grasp Pose Generation

In order to scan an object to reconstruct a mesh, we use Polycam, an application that captures pictures of objects and reconstructs an object mesh using Neural Radiance Fields (NeRF). Using the reconstructed mesh, we generate desired occluded grasp poses using antipodal sampling.

F Baseline Method Details

F.1 PPO

We train a policy using Proximal Policy Optimization (PPO) [41], where the policy outputs 12-dimensional delta end-effector poses corresponding to both the left and right arms. We use the same hyperparameters employed for training *COMBO-Grasp*, except for the entropy coefficient, which is set to 0.003. This modification was made because using the original entropy coefficient caused a continuous increase in the policy's standard deviation, resulting in the policy's inability to exploit a stable and effective strategy during training.

F.2 PPO + Constraint Reward

Similar to the *PPO* baseline, but we introduce an additional reward term that encourages the right arm to be used as a constraint. In particular, we add a reward $r_{right\_dist} = ||T^{obj} - T^{RightEE}||_2$.

F.3 *COMBO-Grasp* w/ Fixed Constraint

Instead of employing a trained constraint policy, we place the right arm as a constraint at a fixed pose. To accommodate objects of varying sizes and orientations, the constraint is positioned at the right hand side of the workspace rather than at the centre. This policy is trained using the same hyperparameters as those employed by *COMBO-Grasp*.

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).**

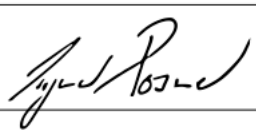
Title of Paper	COMBO-Grasp: Learning Constraint-Based Manipulation for Bimanual Occluded Grasping
Publication Status	□Published ■ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Jun Yamada, Alexander L Mitchell, Jack Collins, Ingmar Posner Accepted to publication at Conference on Robot Learning (CoRL) 2025

Student Confirmation

Student Name:	Jun Yamada		
Contribution to the Paper	- Proposed the research idea - Created and developed methodologies - Ran all experiments - Created all figures - Paper writing		
Signature		Date	19/09/2025

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: Prof. Ingmar Posner		
Supervisor comments *Jun made a substantial contribution to the publication. The description above is accurate.*		
Signature 	Date	22/09/2025

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

5.1 Limitations and Future Work

COMBO-Grasp demonstrates significant improvements in both learning efficiency and generalisation compared to competitive baselines and its own ablated variants. Despite these advances, several limitations remain. First, *COMBO-Grasp* struggles with novel objects that differ substantially in shape from those seen during training. This limitation could be addressed by exposing both the teacher and student policies to a broader distribution of object geometries during training. Moreover, *COMBO-Grasp* struggles with round objects in real-world settings, where stabilising the object during occluded grasping proves challenging. A promising direction for mitigating this issue is the incorporation of closed-loop control, for example, by learning an additional policy that enables real-time adjustment of the constraint pose during execution. While such an extension does not eliminate the fundamental challenges of reward specification and sample efficiency, it nevertheless remains promising, as it can provide online corrective behaviour that improves robustness under execution-time disturbances.

Finally, although *COMBO-Grasp* is designed specifically for bimanual occluded grasping, we view it as a foundational step toward solving a broader class of bimanual manipulation tasks, such as threading a needle, painting, or cutting, where one arm must constrain the object while the other performs precise actions. *COMBO-Grasp*, through value-guided policy coordination, lays the groundwork for solving such tasks more efficiently.

6

Leveraging Scene Embeddings for Gradient-Based Motion Planning in Latent Space

While Chapters 3, 4, and 5 focus on planning-guided skill acquisition by unifying planning with learning-based approaches, this chapter shifts focus toward improving model-based approaches through the use of generative models. Specifically, this chapter introduces *AMP-LS*, a gradient-based motion planning framework that optimises a trajectory within a learnt structured latent space, enabling fast and reactive motion planning in cluttered environments with complex object geometries. Although *AMP-LS* is not specifically designed for contact-rich manipulation, its reactivity and rapid planning speed reduce execution time and improve overall task performance, thereby strengthening the approaches presented in Chapters 3, 4, and 5.

Motion planning is a core capability for robotic manipulation tasks, which aims to plan a collision-free path from the current state of a robot to a predefined goal joint or end-effector pose configuration. While traditional motion planning algorithms, such as Rapidly-exploring Random Trees (RRT) [7, 30, 182] and Probabilistic Roadmaps (PRM) [9, 10], are widely used within the robotics community, these approaches become intractable as the problem size increases, such as with higher DoF of the robot or more complex environments. Furthermore, such approaches are typically incapable of handling dynamic environments due to their slow planning time.

Recently, several learning-based motion planning methods [52, 170] have been proposed to improve planning speeds. However, these approaches often require large-scale demonstrations obtained through sampling-based motion planning for supervised learning, which can be time-consuming and costly to generate.

On the other hand, Latent Space Path Planning (LSPP) [19] presents an alternative approach by learning a structured latent space from static robot kinematics data. This method employs optimisation-based motion planning within the latent space learnt using a variational autoencoder (VAE) [16, 17] and incorporates a collision predictor for obstacle avoidance. However, the collision predictor requires low-dimensional state information about obstacles and is only compatible with objects of primitive shapes, which restricts its application in complex real-world scenarios.

To address these limitations, *AMP-LS* significantly extends LSPP by incorporating a collision predictor that leverages scene embeddings (see Chapter 6, Fig. 2) and explicit collision checking for safer collision avoidance (see Chapter 6, Alg. 1). Similar to LSPP, *AMP-LS* learns a structured latent space from readily available robot kinematics data by employing a β-VAE [16]. In contrast to LSPP, which trains a collision predictor from low-dimensional state observations of obstacles, we train a vision-based collision predictor using diverse synthetic scenes generated in simulation. These scenes include pairs of randomly sampled diverse robot joint configurations and collision labels, in which the objects are drawn from ShapeNet [192]. *AMP-LS* generates a trajectory by minimising a combination of collision probability and end-effector distance losses. Starting from an initial latent representation encoded by a VAE from the robot's initial joint states, the gradients of these losses are backpropagated through the collision predictor and VAE decoder, both dependent on the latent representation, thereby updating it to generate a collision-free trajectory toward the desired end-effector pose. Moreover, we explicitly check for collisions in the interpolated trajectory between the current and next joint state. This allows *AMP-LS* to achieve safer collision avoidance in complex environments (see Chapter 6, Alg. 1).

The experiments demonstrate that *AMP-LS* can generate collision-free trajectories in novel, complex scenes, achieving success rates comparable to traditional planning baselines while substantially reducing planning time in simulation (Chapter 6, Table 1), by leveraging only a readily collected dataset. Furthermore, we qualitatively show that *AMP-LS* transfers effectively to real-world environments, including both complex static scenes and dynamic scenarios involving reaching a moving target while avoiding a moving obstacle (see Chapter 6, Fig. 3).

This chapter shows that a structured latent space trained on readily available data using generative models, and a vision-based collision predictor trained on synthetic data, enable gradient-based motion planning with improved planning speed and reactivity in complex, novel environments. In summary, this chapter presents the following key contributions:

1. *AMP-LS*, a significant extension of LSPP that integrates a vision-based collision predictor trained on diverse synthetic cluttered scenes, together with explicit collision checking, into gradient-based motion planning within a learned latent space, enabling closed-loop, reactive obstacle avoidance in complex, novel environments.

2. Demonstration of generating collision-free trajectories in previously unseen, complex scenes, achieving success rates comparable to traditional baselines while substantially reducing planning time in simulation.

3. Successful zero-shot transfer of AMP-LS to real-world environments.

4. Closed-loop reactive control capabilities, allowing the robot to reach moving targets while actively avoiding dynamic obstacles.

Jun Yamada*, Chia-Man Hung*, Jack Collins, Ioannis Havoutis, Ingmar Posner, (2023). Leveraging Scene Embeddings for Gradient-Based Motion Planning in Latent Space, *IEEE International Conference on Robotics and Automation (ICRA)*

Leveraging Scene Embeddings for Gradient-Based Motion Planning in Latent Space

Jun Yamada*[1], Chia-Man Hung*[1,2], Jack Collins[1], Ioannis Havoutis[2], Ingmar Posner[1]

***Abstract*— Motion planning framed as optimisation in structured latent spaces has recently emerged as competitive with traditional methods in terms of planning success while significantly outperforming them in terms of computational speed. However, the real-world applicability of recent work in this domain remains limited by the need to express obstacle information directly in *state-space*, involving simple geometric primitives. In this work we address this challenge by leveraging learned scene embeddings together with a generative model of the robot manipulator to drive the optimisation process. In addition, we introduce an approach for efficient collision checking which directly regularises the optimisation undertaken for planning. Using simulated as well as real-world experiments, we demonstrate that our approach, AMP-LS, is able to successfully plan in novel, complex scenes while outperforming traditional planning baselines in terms of computation speed by an order of magnitude. We show that the resulting system is fast enough to enable closed-loop planning in real-world dynamic scenes.**

I. Introduction

Motion planning is a core capability for robotic manipulation tasks [1], [2] with the fundamental aim of planning a collision-free path from the current state of an articulated configuration of joints to a predefined goal joint or end-effector pose configuration. Sampling-based motion planning algorithms, such as Rapidly-Exploring Random Trees (RRT) [3] and Probabilistic Roadmap (PRM) [4], are widely used within the robotics community as they have well understood properties in regards to planning time and collision avoidance. However, sampling-based methods become increasingly intractable as the problem size increases (i.e., Degrees-of-Freedom (DoF) of the robot, environment complexity, and length of the path) and are also typically too slow to be used for closed-loop planning, as any change to the environment requires re-planning [5].

Recently, learning-based motion planning [6], [7] has gained the attention of the robotics community with the promise of increased computational efficiency and faster planning speed. Notably, Latent Space Path Planning (LSPP) [8] introduces motion planning via gradient-based optimisation in the latent space of a VAE. The success rate of LSPP is commensurate with that of commonly used sampling and gradient-based motion planners, but with significantly reduced planning time. By learning a structured latent space using kinematically feasible and easily generated robot states, a learned latent space that is optimised via activation maximisation (AM) [9] can produce diverse and adaptive behaviours [10]. However, LSPP relies on state-based obstacle representation with predefined object shapes, which do not easily transfer to real-world environments.

*Equal contribution.
[1]Applied AI Lab (A2I), [2]Dynamic Robot Systems (DRS)
Oxford Robotics Institute (ORI), University of Oxford
Correspondence to: jyamada@robots.ox.ac.uk
Project page: https://amp-ls.github.io/

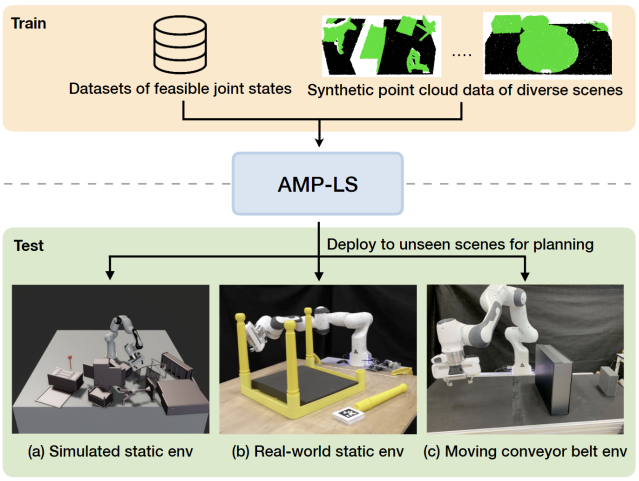


Fig. 1: **Problem setup.** AMP-LS generates a collision-free trajectory via gradient-based optimisation by leveraging scene embeddings. Our model is trained on kinematically feasible robot joint states and synthetic point clouds of diverse scenes. For evaluation, our method is deployed to unseen scenes including: (a) *Simulated static env*: Novel scenes generated by randomly placing obstacles on a table. (b) *Real-world static env*: A robot avoids the table legs to reach the pre-grasp location of the unassembled table leg. (c) *Moving Conveyor Belt env*: A robot reaches a moving target object while avoiding an obstacle on the conveyor belt by using closed-loop planning.

To address the limitation of LSPP, we introduce a method significantly extending the prior work by incorporating a collision predictor that leverages scene embeddings and efficient collision checking, which regularises the optimisation during planning for safe collision avoidance. We name this new method Activation Maximisation Planning in Latent Space (AMP-LS). Specifically, we adapt SceneCollisionNet [11], trained on diverse synthetic point clouds of scenes generated with objects from ShapeNet datasets [12], for our purpose to facilitate zero-shot transfer to unseen environments, including real-world scenes (see Fig. 1). Due to the speed of our approach, we also show that our method can be applied to closed-loop settings where both the obstacles and goal pose are moving.

The contributions of our work are threefold: (1) we present Activation Maximisation Planning in Latent Space (AMP-LS), which significantly extends LSPP by incorporating a collision predictor that leverages scene embeddings and explicit collision checking in order to regularise optimisation when planning for obstacle avoidance; (2) we empirically demonstrate that our approach can be zero-shot transferred to unseen scenes, including real-world environments, through the

use of a collision predictor that is trained on diverse synthetic scenes; (3) we show that our method can be applied to closed-loop settings with reactive behaviour, capable of reaching a *moving* target while also avoiding a *moving* obstacle.

II. Related Works

Sampling-based motion planning approaches such as RRT [3], [13] and PRM [4] are widely used to generate collision-free trajectories in robotics. PRM requires a pre-computed roadmap; RRT often struggles to find the solution with the shortest path. While several extensions such as RRT* [13] and BIT* [14] have been proposed to achieve asymptotic optimality and reduce computational cost, these approaches typically demand many samples—a runtime problem that compounds with increases in robot DoF, environmental complexity, or path length [15]. Another limitation of sampling-based motion planners is that they do not support real-time planning, as re-planning is required to navigate dynamic environments.

Optimisation-based planning approaches such as covariant Hamiltonian optimisation for motion planning (CHOMP) [16] and Stochastic Trajectory Optimisation for Motion Planning (STOMP) [17] require a large number of trajectories when given multiple constraints. These approaches typically start from an initial guess, a trajectory linking the start and desired end states, which is refined through minimisation of a cost function. Computation terminates when a stop condition is met or the algorithm times out. The artificial potential algorithm [18], [19] is perhaps the closest optimisation-based planning approach to our work. It achieves real-time obstacle avoidance by creating attractive and repulsive fields around goals and obstacles. End-effector movement is then guided by the gradient of these fields. Although appealing in its simplicity, it struggles to handle additional constraints on properties that cannot be fully determined by robot joint configuration.

Several recent works attempt to leverage neural networks for motion planning. Neural motion planning methods [20], [21], [6], [22] employ imitation learning (IL) on expert demonstrations generated by a sampling-based motion planner or reinforcement learning (RL) [23] to learn motion policies. Notably, Motion Policy Network [24] achieves commensurate success rates when compared against traditional planning approaches and even generalises well to unseen environments. However, these methods require a large number of trajectories, often generated by an expert planner, to train a motion planning policy.

Another set of works performs planning in a learned latent space [25], [8]. L2RRT [21] plans a path in a learned latent space using RRT. Our work builds upon *Latent Space Path Planning* (LSPP) [8]. LSPP plans a trajectory for a robot via iterative optimisation using activation maximisation (AM) [9] in a latent space of the robot kinematics learned by a generative model. Leveraging a collision predictor as a constraint, LSPP successfully plans a collision-free path with improved efficiency in planning time. However, LSPP approximates a scene as a set of cylindrical obstacles and requires state-based knowledge of the scene, such as position and shape of obstacles. Such narrow scene definitions and lack of complete information limits the application of this method to real-world problems.

To successfully generate a collision-free path in a scene with obstacles, learning a collision predictor to identify the collision between a robot and the scene is essential. Prior neural motion planning methods [20], [21], [6], [26] learn obstacle representations either from 2D images, occupancy grids, or point clouds, instead of explicitly predicting a probability of collision. SceneCollisionNet [11] learns the scene embeddings for a collision predictor from a large number of synthetic scenes generated with diverse objects from ShapeNet [12]. To leverage a collision predictor as a constraint for motion planning, we utilise SceneCollisionNet and adapt it to work within our latent planning framework.

III. Approach

In this work, we introduce a method significantly extending the prior work [8] and name it Activation Maximisation Planning in Latent Space (AMP-LS). Similar to the prior work [8], AMP-LS leverages a variational autoencoder (VAE) [27], [28] to learn a structured latent space to generate kinematically feasible joint trajectories. While a collision predictor in the prior work relies on state-based obstacle representations, our collision predictor leverages scene embeddings obtained from SceneCollisionNet [11] to readily achieve zero-shot transfer to unseen environments. Further, we present an approach for explicit collision checking to directly regularise the optimisation to plan collision-free trajectories. In the following section, we describe an overview of our model (see Fig. 2) and optimisation objective for planning.

A. Problem Formulation

Similar to LSPP [8], we consider the problem of generating a collision-free trajectory consisting of robot joint configurations $\{\mathbf{q}_0, \ldots, \mathbf{q}_T\}$ for a robot in an environment with obstacles. A state $\mathbf{x}_t$ at time t consists of a kinematically feasible robot joint configuration $\mathbf{q}_t$, and its end-effector position $\mathbf{e}_t^{pos}$ and orientation $\mathbf{e}_t^{ori}$. While LSPP considers only the joint states and end-effector position, adding orientation is essential to tackle motion planning problems. The end-effector orientation $\mathbf{e}^{ori}$ employs a 6D representation of SO(3), which consists of the first two column vectors in a rotation matrix $\mathbf{R}$. This representation is suitable for learning rotations using neural networks due to its property of continuity [29]. Note that no prior information of obstacles (e.g., mesh) is given. In contrast to LSPP, which leverages low-dimensional state information as an observation, AMP-LS utilises point cloud observations $\mathbf{o}_t \in \mathcal{R}^{n\times 3}$ with n points from a third-person camera, which includes only scene information. Thus, the robot point cloud is filtered out from the raw point cloud.

B. Learning Latent Representations of Robot State

To plan cohesive paths for the manipulator using a learned latent space, the latent space must be structured such that representations of similar joint states are close to each other. Leveraging a VAE [27], [28], prior work [8] successfully learns such a latent space and captures a notion of local distance in joint space. In their representation, poses that are close to each other in joint space are also close in latent space. Similarly, we also learn a VAE consisting of an encoder $q_\phi(\mathbf{z}|\mathbf{x})$ and decoder $p_\theta(\mathbf{x}|\mathbf{z})$, where $\mathbf{z}$ is the latent representation. To train the VAE, rather than directly maximise the evidence, $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x} \mid \mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z}$, which

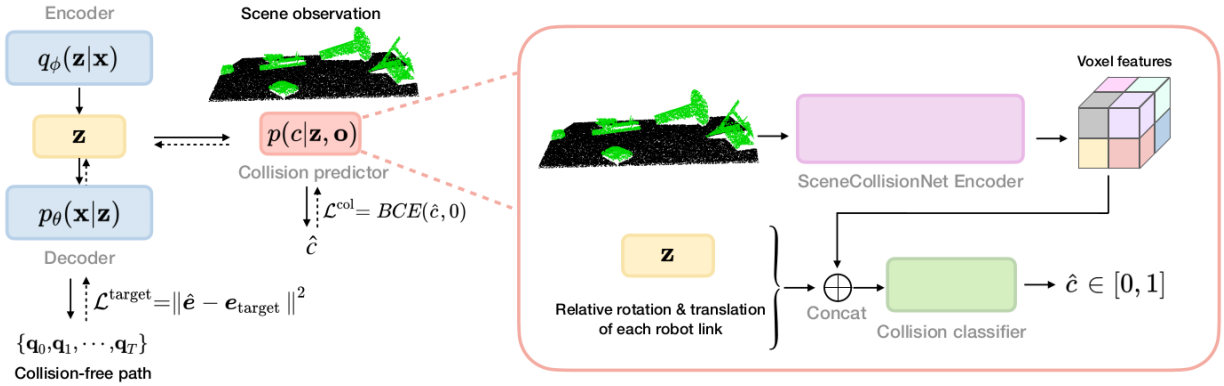


Fig. 2: **Our method overview.** A VAE (**blue**) is trained using feasible robot states $\mathbf{x}$ consisting of joint states, end-effector position, and end-effector orientation to learn structured latent representations $\mathbf{z}$ (**yellow**). Then, freezing the weights of the pre-trained encoder in the VAE, the collision predictor (**red**) takes as input the learned latent representation $\mathbf{z}$ and a scene point cloud observation $\mathbf{o}$. The collision predictor built upon SceneCollisionNet learns to output a probability $\hat{c}$ of collision between the robot arm and obstacles. To plan a collision-free trajectory, gradient-based optimisation is applied to produce a sequence of latent representations $\{\mathbf{z}_t\}_{t=1}^T$ each of which has a low probability of collision with the scene using the learned collision predictor. A sequence of joint states $\{\mathbf{q}_t\}_{t=1}^T$ is generated by decoding the sequence of latent representations $\{\mathbf{z}_t\}_{t=1}^T$ using the trained decoder in the VAE.

is generally intractable, we instead optimise the evidence lower bound (ELBO) $\mathcal{L}^{\text{ELBO}} \leq p(\mathbf{x})$:

$$\mathcal{L}^{\text{ELBO}} = \underbrace{\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z})}_{\text{Reconstruction Accuracy}} - \underbrace{D_{\text{KL}}\left[q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})\right]}_{\text{KL Term}} \tag{1}$$

There is a trade-off in the ELBO loss between the reconstruction accuracy and the KL term: accurate reconstruction at the cost of poorly structured latent space, on one hand, or well-structured latent space but noisy reconstruction, on the other. These terms are often manually weighted in the ELBO formulation [30]. An alternative to manually tuning the weight is to use GECO [31]. GECO adaptively tunes the trade-off between reconstruction and regularisation by formulating the ELBO loss as a constrained optimisation problem with a Lagrange multiplier λ:

$$\mathcal{L}^{\text{GECO}} = \underbrace{-D_{\text{KL}}\left[q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})\right]}_{\text{KL Term}} + \lambda \underbrace{\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\mathcal{C}(\mathbf{x}, \hat{\mathbf{x}})]}_{\text{Reconstruction Error Constraint}} \tag{2}$$

This encourages the model to optimise the reconstruction accuracy first, until it reaches a predefined target. The KL term is then optimised. The generative model is trained on a dataset of kinematically feasible joint states of the robot.

C. Activation Maximisation for Motion Planning

Our goal is to plan a trajectory consisting of robot joint configurations towards a target pose. That is, given a target end-effector position $\mathbf{e}_{\text{target}}^{pos}$ and orientation $\mathbf{e}_{\text{target}}^{ori}$, we expect our method to generate a sequence of joint configurations $\{\mathbf{q}_0, \ldots, \mathbf{q}_T\}$ that leads a robot to the target pose. Leveraging the trained VAE inspired by prior work [8], we can compute such a sequence of robot joints by decoding the latent representation of the VAE model $\{\mathbf{z}_0, \ldots, \mathbf{z}_T\}$. This sequence of the latent representation is computed in a probabilistic model through activation maximisation (AM) [9]:

$$\mathbf{z}_{t+1} = \mathbf{z}_t - \alpha_{\text{AM}} \nabla \mathcal{L}^{\text{AM}} \tag{3}$$

where

$$\mathcal{L}^{\text{AM}} = \lambda_{\text{target}} \Big(\underbrace{\left\| \hat{\boldsymbol{e}}^{pos} - \boldsymbol{e}_{\text{target}}^{pos} \right\|_2}_{\text{Target Position Loss}} + \underbrace{\left\| \hat{\boldsymbol{e}}^{ori} - \boldsymbol{e}_{\text{target}}^{ori} \right\|_2}_{\text{Target Orientation Loss}} \Big) + \underbrace{(-\log p(\mathbf{z}))}_{\text{Prior Loss}} \tag{4}$$

In contrast to the prior work [8], we also introduce an end-effector orientation constraint, which is generally useful for reaching a pre-grasp pose. The first latent representation $\mathbf{z_0}$ is acquired by encoding the current/starting robot state $\mathbf{z}_0 \sim q_\phi(\mathbf{z}|\mathbf{x} = \mathbf{x}_0)$. Note that model parameters are not updated, but only the parameterised latent variable $\mathbf{z}$ is iteratively updated. The first two terms in $\mathcal{L}^{\text{AM}}$ (Eq. 4) guide the latent representation to decode robot joint states that approach the target pose. The third term is the likelihood of the current representation under its prior, which is introduced in [8] to encourage the latent representation to stay close to the training distribution, thus decoding to kinematically feasible pair of joint position and end-effector pose.

D. Collision Constraints

To generate a collision-free trajectory, similar to that used in prior work [8], we add collision constraints to the objective function in Eq. 4 by introducing a collision predictor. While the prior work uses narrowly defined state-based obstacle representations as input to the collision predictor, in our approach, we adapt SceneCollisionNet [11] to embed scene observations for zero-shot transfer to unseen environments. The voxel features from SceneCollisionNet are concatenated with the latent representation of the VAE $\mathbf{z}$ and the rotation and relative translation from each robot link to the centre of the closest voxel to form the input to the collision classifier. The classifier predicts the probability of collision $\hat{c}$ between the robot and obstacles (see Fig. 2). Note that we train the

Algorithm 1 Planning a collision-free path in latent space via activation maximisation

Initialise a buffer $D = \{\mathbf{q}_0\}$, λ_{pos}, λ_{ori}, λ_{col}, $\mathbf{q}_{\text{prev}} = \mathbf{q}_0$
$\mathbf{z}_0 \sim q_\phi(\mathbf{z}|\mathbf{x} = \mathbf{x}_0)$
for $t = 0, 1, 2, \ldots, H$ **do**
  $\{\hat{\mathbf{q}}_t, \hat{\mathbf{e}}_t^{pos}, \hat{\mathbf{e}}_t^{ori}\} \sim p_\theta(\mathbf{x}|\mathbf{z} = \mathbf{z}_t)$
  if $t > 0$ and $p_\vartheta(\mathbf{z}_t, \mathbf{o}_t) < \gamma_{\text{col}}$ **then**
    $\{\mathbf{q}_{\text{prev}}, \ldots, \hat{\mathbf{q}}_t\} = f_{\text{interpolate}}(\mathbf{q}_{\text{prev}}, \hat{\mathbf{q}}_t)$
      ▷ Linear interpolation between $\mathbf{q}_{\text{prev}}$ and $\hat{\mathbf{q}}_t$
    if collision in $\{\mathbf{q}_{\text{prev}}, \ldots, \hat{\mathbf{q}}_t\}$ **then**
      $i \leftarrow$ index of the first joint state with collision in the interpolated trajectory
      $m \leftarrow |\{\mathbf{q}_{\text{prev}}, \ldots, \hat{\mathbf{q}}_t\}|$
      Reduce λ_{pos} and λ_{ori} by a factor of $\frac{i}{m}$
      $\hat{\mathbf{q}}_t \leftarrow \mathbf{q}_{\text{prev}}$, $\mathbf{z}_t \leftarrow \mathbf{z}_{\text{prev}}$
      ▷ Back trace to the previous joint and latent representations $\mathbf{q}_{\text{prev}}$ and $\mathbf{z}_{\text{prev}}$ for replanning
    else
      $D \leftarrow D \cup \{\mathbf{q}_{\text{prev}}, \ldots \hat{\mathbf{q}}_t\}$
      if $d(\hat{\boldsymbol{e}}_t, \boldsymbol{e}_{\text{target}}) < \gamma$ **then**
        break
      end if
      $\mathbf{q}_{\text{prev}} \leftarrow \hat{\mathbf{q}}_t$, $\mathbf{z}_{\text{prev}} \leftarrow \mathbf{z}_t$
    end if
  end if
  Compute losses (Eq. 5)
  Update λ_{pos}, λ_{ori}, and λ_{col} using GECO
  $\mathbf{z}_{t+1} \leftarrow \mathbf{z}_t - \alpha_{\text{AM}} \nabla \mathcal{L}_t^{\text{AM}}$
end for

collision predictor only on features of voxels closest to each robot link to ignore unnecessary voxel information. While training the collision predictor, the weights of the pre-trained VAE are frozen so that the pre-trained latent space does not change. The collision predictor is trained using the binary cross-entropy (BCE) loss with ground truth collision labels. To drive the latent representation away from obstacles, we incorporate the collision predictor loss into Eq. 4:

$$\begin{aligned}\mathcal{L}^{\text{AM}} = \lambda_{\text{target}}(&\underbrace{\left\|\hat{\boldsymbol{e}}^{pos} - \boldsymbol{e}^{pos}_{\text{target}}\right\|_2}_{\text{Target Position Loss}} + \underbrace{\left\|\hat{\boldsymbol{e}}^{ori} - \boldsymbol{e}^{ori}_{\text{target}}\right\|_2}_{\text{Target Orientation Loss}}) \\ &+\lambda_{\text{col}} \underbrace{(-\log\left(1 - p_\vartheta\left(\mathbf{z}, \mathbf{o}\right)\right))}_{\text{Collision Loss}} + \underbrace{(-\log p(\mathbf{z}))}_{\text{Prior Loss}}\end{aligned} \tag{5}$$

During planning, three coefficients λ_{target} and λ_{col} are automatically and dynamically adjusted by GECO [31]. Minimising the collision loss during AM optimisation drives the latent representation $\mathbf{z}$ towards the representation whose decoded joint configuration is collision-free.

E. Collision Checking

While prior work [8] simply optimises the objective function until it reaches a target, we observe that it is hard to perfectly balance multiple loss terms and that such simple optimisation often results in collision between the robot and obstacles. In contrast to the target losses, the collision loss is inherently a hard constraint that should not be violated at any point in the trajectory. To address this issue, our high-level idea is that collision can be predicted and avoided before execution and the coefficients of the objective function determine the direction in which the latent representation is heading towards. Specifically, we introduce explicit collision checking using the learned collision predictor and automatic rescaling for coefficients during the planning to avoid obstacles more safely. That is, if a collision probability of the decoded joint configuration is higher than a predefined threshold γ_{col}, we reject such robot configuration that is highly likely to be in collision and keep optimising the latent space until the decoded joint state is collision-free. Then, we interpolate a trajectory between the current and decoded collision-free joint state in m steps and pass them to the collision predictor to check for collision. If there is any collision in the interpolated trajectory, we obtain its index i of the joint state with collision closest to the current joint state and reduce the coefficient of the target position and orientation loss by multiplying by $\frac{i}{m}$, to encourage the optimisation to minimise the collision loss. Intuitively, this scaling induces the robot to deviate from the original route drastically depending on how close it is to an obstacle. This process continues until the collision-free next joint state is found and there is no collision in the interpolated trajectory between the current joint state and the next joint state. In our experiments, we use the threshold of $\gamma_{\text{col}} = 0.4$. For further details, see Algorithm 1.

IV. Implementation Details

A. Architecture Details

Our VAE encoder and decoder consist of three fully connected hidden layers with 512 units and ELU activation functions [32]. The input dimension to the VAE is 16, consisting of robot joint states $\mathbf{q} \in \mathcal{R}^7$, end-effector position $\mathbf{e}^{pos} \in \mathcal{R}^3$ and 6D representation of end-effector rotation matrix $\mathbf{e}^{ori} \in \mathcal{R}^6$. The dimension of the latent space $\mathbf{z}$ is 7. The collision classifier consists of fully connected layers with units of $[1024, 256]$.

B. Training Details

The VAE is trained using kinematically feasible robot joint configurations. To generate such joint states, we leverage the Flexible Collision Library (FCL) [33] for self-collision checking. The VAE model is trained with a batch size of 256 for about $2M$ training iterations using the Adam optimiser [34] with a learning rate of $3\text{e}{-4}$ on a GeForce RTX 3090. Throughout the training, valid robot configurations are generated on the fly as it is cheap to do so. In total, the model is exposed to around $500M$ configurations.

The collision predictor is trained on diverse synthetic point cloud data to assist zero-shot transfer to scenes with unseen obstacles. Such scenes are generated by placing objects randomly sampled from the ShapeNet dataset [12], consisting of 8828 3D meshes. Each object is placed on a planar surface with a random position and rotation. We sample the number of objects placed on the surface from a uniform distribution between 4 and 8. To train the collision predictor, a new scene is procedurally generated for each training iteration similar to the prior work [11], and we randomly sample 2048 instances of kinematically feasible robot joint configurations and check for collisions between each robot configuration and the generated scene using FCL. A third-person RGB-D camera is directed towards the centre of the scene to sample

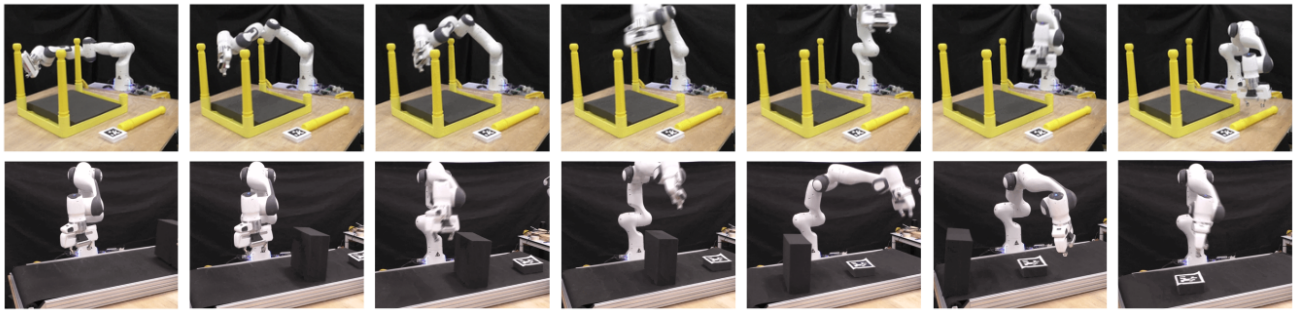


Fig. 3: Visualisation of real-world experiments. **Top**: Our method successfully plans a collision-free trajectory in a complex real-world scene from an impeded start configuration to a pre-grasp goal configuration. By training a collision predictor on diverse synthetic scenes, our method can readily transfer to such unseen scene. **Bottom**: AMP-LS can be applied to closed-loop planning to avoid moving obstacles and reach a moving target object on a conveyor. This reader is referred to our supplementary video for better visualisation.

point clouds. The camera extrinsics are randomly sampled for each query from a predefined range of roll, yaw, and pitch parameters. Thus, 2048 unique valid robot configurations and point clouds are procedurally generated for each iteration to train the collision predictor. We train the collision predictor for $1M$ training iterations using SGD with a learning rate of $1\mathrm{e}{-3}$ and with momentum 0.9 for approximately 7 days, which is similar to the training time requirement of SceneCollisionNet.

C. Deploymenet details

In open-loop planning, the current state $\mathbf{x}_0$ is encoded to a latent representation $\mathbf{z}_0$. Then, the encoded latent representation is iteratively optimised through AM optimisation (see Eq. 5) until the end-effector reaches the target pose with a tolerance of γ. In closed-loop planning, while the latent representation is similarly optimised, a point cloud input for the collision predictor and the target pose in the objective function (see Eq. 5) are updated at each time step for reactive motion.

V. Experiments

We design our experiments to answer the following guiding questions: (1) how does AMP-LS perform compared to traditional motion planning methods such as sampling and optimisation-based approaches in open-loop settings? (2) does AMP-LS transfer zero-shot to real-world static environments? (3) does AMP-LS cope with dynamic environments using closed-loop planning?

A. Experimental Setup

We evaluate our approach in both simulated and real-world environments. In simulated experiments, we use the Gazebo simulator [35] with ROS. In all of the simulated and real-world experiments, we use a 7-DoF Franka Panda robot.

B. Open-Loop Planning for Reaching Static Targets

We evaluate AMP-LS in an open-loop planning setup in a simulated environment. In this experiment, obstacles in the environment are static. We select a range of sampling and optimisation-based motion planners typically used by the robotics community and available within the unified MoveIt! library. We compare our method against several sampling-based motion planners and an optimisation-based motion planner: RRT-Connect [36], RRT* [13], Lazy PRM* [37], LBKPIECE [38], BIT* [14], and CHOMP [16]. CHOMP uses a linear initialisation from start to goal joint positions. Since we assume that complete knowledge of the environment is not available, occupancy maps [39] generated from point clouds are used for collision checking in motion planning baseline methods. We evaluate the methods on 100 novel scenes where objects are randomly placed on a table (see Fig. 1 (a)). The hyperparameters used for GECO to determine coefficients of our objective function (see Eq. 5) are found via a grid search similar to that of prior work [8]. For the baselines, we use the default parameters provided by MoveIt OMPL. For RRT*, Lazy PRM*, and BIT*, the same 1 second planning budget is given. Across all methods, a motion plan is considered to be successful if a robot reaches a target within a distance tolerance of 1cm and orientation tolerance of 15 degrees.

As illustrated in Table I, our method achieves a reasonable success rate with improved planning time compared to most of the motion planning baselines. Specifically, AMP-LS outperforms CHOMP, which is also an optimisation-based motion planner, by a significant margin because CHOMP requires a large number of trajectories to find a feasible path in complex scenes, in contrast to AMP-LS. AMP-LS still has a commensurate success rate against RRT-Connect, but the planning time of AMP-LS is an order of magnitude faster than the baseline. Traditional motion planning baselines often fail to find a collision-free path within a short time and sometimes plan a path with collision due to occlusions in the scenes. In contrast, our collision predictor is trained on diverse synthetic scenes with occlusion and can therefore reason about occluded regions, similar to SceneCollisionNet [11]. While our method demonstrates reasonable accuracy and improved planning efficiency, the path length is longer than most of the other baselines. The longer path length is due to the design of the planning strategy [8] that tunes the coefficients of losses automatically to avoid obstacles, thus not directly minimising the path length. To address this issue, additional optimisation constraints could be explored in the future that focus on reducing the path length.

As illustrated in Table I, we also ablate constraints, such as prior loss, collision loss, and explicit collision checking. The success rate of AMP-LS without a collision loss significantly drops, indicating that our collision predictor successfully

	Success rate	Planning time (s)	Path length
AMP-LS (ours)	**0.88 ± 0.06**	**0.16 ± 0.13**	3.61 ± 1.05
AMP-LS w/o col. loss	0.46 ± 0.10	0.12 ± 0.04	3.68 ± 1.29
AMP-LS w/o prior loss	0.35 ± 0.09	0.24 ± 0.21	3.23 ± 1.12
AMP-LS w/o explicit collision	0.75 ± 0.08	0.15 ± 0.21	3.50 ± 1.12
RRT-Connect	**0.86 ± 0.07**	1.60 ± 0.89	**2.17 ± 0.84**
RRT*	0.36 ± 0.09	N/A	2.25 ± 0.78
Lazy PRM*	**0.82 ± 0.08**	N/A	2.26 ± 0.82
LBKPIECE	0.23 ± 0.08	2.54 ± 1.12	2.34 ± 0.92
BIT*	0.63 ± 0.09	N/A	2.42 ± 1.02
CHOMP	0.39 ± 0.10	2.24 ± 0.79	2.41 ± 0.90

TABLE I: Comparison of performance of our method AMP-LS against baseline motion planning algorithms with ablations. We also report 95% confidence interval of Wilson score [40] for success rate and standard deviation for planning time and path length. The path length is normalised by dividing the actual path length by the distance between the initial and target end-effector positions for fairer comparison.

constrains the latent space even in novel scenes. Similar to the prior work [8], AMP-LS without the prior loss results in significantly poorer performance as the latent representation is optimised to drive into unseen latent representations, which decode to kinematically inconsistent configurations. Furthermore, Table I shows that the success rate of AMP-LS without explicit collision checking drastically decreases because perfectly optimising multiple loss terms is often challenging, resulting in the collision with obstacles.

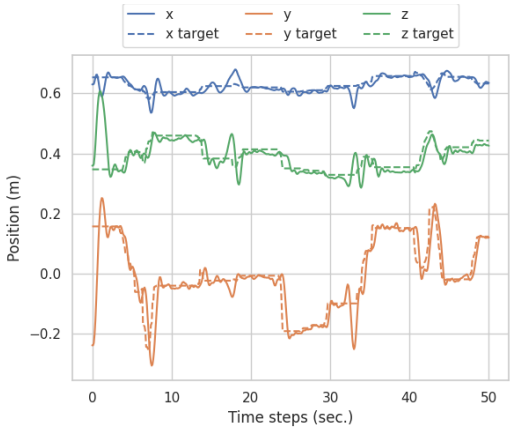


Fig. 4: **Coordinates of end-effector and moving targets in closed-loop settings.** To verify the ability of closed-loop planning in our method, we deploy our method to the real-world robot arm to reach a moving target.

C. Real-World Open-Loop Planning in a Complex Scene

Our method readily transfers to complex real-world scenes. To verify this, we qualitatively evaluate our method in a complex real-world static scene using open-loop planning as illustrated in Fig. 1 (c). In this task, the robot needs to reach the unassembled table leg while avoiding the other table legs to achieve a pre-grasp pose in a furniture assembly task. We control the robot arm using an impedance controller. As shown in Fig. 3 Top, our method can successfully plan a collision-free trajectory for a robot starting next to the table legs to avoid obstacles and reach the unassembled table leg on the table. This demonstrates that our collision predictor, trained on diverse synthetic scenes, is transferable to real-world environments.

D. Closed-Loop Planning for Moving Obstacles and Targets

As our method is, by design, an efficient local planner, AMP-LS is able to act reactively when operated as a closed-loop system. To verify the closed-loop potential of AMP-LS, we deploy our method on a robot with the goal of reaching a moving target without obstacles. To control the real-world robot, a desired next joint position is sent to an impedance controller at 10Hz. Fig. 4 illustrates coordinates of the moving target and the end-effector position over 50 seconds. Since our method can predict the next desired joint state quickly, the robot can reactively follow the moving target.

To further demonstrate the ability of reactive motion using AMP-LS, we evaluate our method on a setup where the robot needs to avoid moving obstacles and reach a target object on a conveyor in both simulated and real-world environments (see Fig. 3 Bottom). Firstly, we quantitatively evaluate our method to examine the ability of reactive motion in the simulated environment. In this evaluation, we randomly generate obstacles of different sizes, and the obstacle and a target object are randomly placed on the conveyor belt. We observe that the robot successfully avoids the obstacle and reaches a moving target on the conveyor with a success rate of 93.3% (28/30 trials) thanks to the fast planning of our method. Note that we use a threshold of 3cm and 20 degrees in this experiment, because tight tolerance for reaching a moving target is challenging unless a future state of the target is estimated and used for planning.

In the real-world experiment, the robot starts moving towards the target object with attached AprilTag [41] that is tracked by the third-person camera. For closed-loop planning, the collision predictor takes as input a point cloud for each time step. As illustrated in Fig. 3 Bottom, the robot successfully avoids the moving obstacle to reach and follow the target object.

VI. Conclusion

In this work, we present AMP-LS, a learning-based motion planning approach that generalises to unseen obstacles in complex environments. AMP-LS builds upon LSPP [8] and inherits a number of desirable properties. However, AMP-LS considerably extends LSPP by introducing a collision predictor trained on diverse synthetic scenes to leverage scene embeddings for unseen scene generalisation, and explicit collision checking during planning for safe obstacle avoidance. We demonstrate that AMP-LS successfully generates collision-free paths in both unseen simulated and real-world scenes. The comparison between AMP-LS and several sampling and optimisation-based motion planning baselines shows that our method achieves a commensurate success rate with much improved planning time. Furthermore, our real-world experiments show that AMP-LS can handle both open and closed-loop planning, which significantly broadens the applicability to real-world robotic problems.

Acknowledgment

This work was supported by a UKRI/EPSRC Programme Grant [EP/V000748/1], we would also like to thank the University of Oxford for providing Advanced Research Computing (ARC) facility in carrying out this work (http://dx.doi.org/10.5281/zenodo.22558).

REFERENCES

[1] J. Yamada, Y. Lee, G. Salhotra, K. Pertsch, M. Pflueger, G. S. Sukhatme, J. J. Lim, and P. Englert, "Motion planner augmented reinforcement learning for obstructed environments," in *Conference on Robot Learning*, 2020.

[2] F. Xia, C. Li, R. Martín-Martín, O. Litany, A. Toshev, and S. Savarese, "Relmogen: Leveraging motion generation in reinforcement learning for mobile manipulation," *arXiv preprint arXiv:2008.07792*, 2020.

[3] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Iowa State University, Tech. Rep., 1998.

[4] N. M. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1996.

[5] A. Short, Z. Pan, N. Larkin, and S. Duin, "Recent progress on sampling based dynamic motion planning algorithms," 07 2016, pp. 1305–1311.

[6] A. H. Qureshi, A. Simeonov, M. J. Bency, and M. C. Yip, "Motion planning networks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2118–2124.

[7] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, "Motion planning networks: Bridging the gap between learning-based and classical motion planners," *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 48–66, 2020.

[8] C.-M. Hung, S. Zhong, W. Goodwin, O. P. Jones, M. Engelcke, I. Havoutis, and I. Posner, "Reaching through latent space: From joint statistics to path planning in manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5334–5341, 2022.

[9] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *Technical Report, Univeristé de Montréal*, 01 2009.

[10] A. L. Mitchell, M. Engelcke, O. P. Jones, D. Surovik, S. Gangapurwala, O. Melon, I. Havoutis, and I. Posner, "First steps: Latent-space control with semantic constraints for quadruped locomotion," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5343–5350.

[11] M. Danielczuk, A. Mousavian, C. Eppner, and D. Fox, "Object rearrangement using learned implicit collision functions," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6010–6017.

[12] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[14] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Batch informed trees (bit*): Informed asymptotically optimal anytime search," *The International Journal of Robotics Research*, vol. 39, no. 5, pp. 543–567, 2020.

[15] K. Hauser, "Lazy collision checking in asymptotically-optimal motion planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2951–2957.

[16] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 489–494.

[17] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 4569–4574.

[18] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1985, pp. 500–505.

[19] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 338–345.

[20] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2017.

[21] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7087–7094.

[22] A. H. Qureshi and M. C. Yip, "Deeply informed neural sampling for robot motion planning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6582–6588.

[23] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.

[24] A. Fishman, A. Murali, C. Eppner, B. Peele, B. Boots, and D. Fox, "Motion policy networks," in *6th Annual Conference on Robot Learning*, 2022.

[25] B. Ichter and M. Pavone, "Robot motion planning in learned latent spaces," *IEEE Robotics and Automation Letters*, pp. 2407–2414, 2019.

[26] R. Strudel, R. Garcia, J. Carpentier, J.-P. Laumond, I. Laptev, and C. Schmid, "Learning obstacle representations for neural motion planning," *arXiv preprint arXiv:2008.11174*, 2020.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[28] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," 2014.

[29] Y. Zhou, C. Barnes, L. Jingwan, Y. Jimei, and L. Hao, "On the continuity of rotation representations in neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[30] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations*, 2017.

[31] D. J. Rezende and F. Viola, "Taming vaes," 2018.

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

[33] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3859–3866.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[35] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[36] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings of IEEE International Conference on Robotics and Automation*. IEEE, 2000, pp. 995–1001.

[37] R. Bohlin and L. E. Kavraki, "Path planning using lazy prm," in *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings*, vol. 1. IEEE, 2000, pp. 521–528.

[38] I. A. Şucan and L. E. Kavraki, "Kinodynamic motion planning by interior-exterior cell exploration," in *Algorithmic Foundation of Robotics VIII*. Springer, 2009, pp. 449–464.

[39] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.

[40] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, 1927.

[41] E. Olson, "Apriltag: A robust and flexible visual fiducial system." in *ICRA*. IEEE, 2011.

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).**

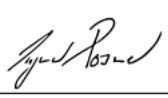
Title of Paper	Leveraging Scene Embeddings for Gradient-Based Motion Planning in Latent Space
Publication Status	✓Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Jun Yamada*, Chia-Man Hung*, Jack Collins, Ioannis Havoutis, Ingmar Posner."Leveraging Scene Embeddings for Gradient-Based Motion Planning in LatentSpace". In: IEEE International Conference on Robotics and Automation (ICRA). June 2023. *Equal contribution.

Student Confirmation

Student Name:	Jun Yamada		
Contribution to the Paper	- Conceived and developed the research idea. - Implemented the main models, including the VAE, GECO, and SceneCollisionNet. - Implemented the main training pipelines - Designed and implemented the real-world experimental setup using a Franka Panda robot. - Conducted both simulated and real-world experiments, primarily leading their execution. - Collected diverse synthetic scenes for training the collision predictor. - Suggested and implemented an alternative end-effector representation (6D - instead of quaternion). - Evaluated the proposed representation in collaboration with a co-author. - Contributed to writing the manuscript together with the co-author.		
Signature		Date	19/09/2025

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: Prof. Ingmar Posner		
Supervisor comments *Jun made a substantial contribution to the publication. The description above is accurate.*		
Signature 	Date	19/09/2025

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

6.1 Limitations and Future Work

While *AMP-LS* employs a collision classifier for collision checking, the classifier occasionally predicts a low collision probability even when the robot arm is in close proximity to an obstacle, which may fail to induce avoidance behaviour. To address this limitation, a differentiable signed-distance function [160] could be incorporated to provide more informative gradients. These gradients encode both the proximity to obstacles and the direction of increasing clearance, thereby inducing smooth repulsive motion that guides the trajectory away from obstacles during optimisation.

To extend *AMP-LS* to more complex scenarios, such as non-tabletop environments, it is essential to collect a diverse set of scenes for training the collision classifier. In addition, as with prior optimisation-based approaches, careful tuning of the loss term coefficients in the *AMP-LS* framework is critical for effective planning. Trajectory smoothness could also be improved by incorporating minimum jerk costs, similar to the prior work [160].

AMP-LS enables a robot to react to moving obstacles or targets by updating the structured latent representation through gradients derived from the decoder and collision classifier, conditioned on new observations. This mechanism allows the planner to adapt reactively to environmental changes. However, the formulation implicitly assumes that the environment remains approximately static within each optimisation step. When dynamic objects undergo substantial motion over that horizon, reactive latent updates based on the current observation become insufficient. In such cases, explicitly modelling environment dynamics and predicting the future motion of dynamic objects would be necessary to enable anticipatory avoidance and tracking.

Finally, a promising direction for future work is to integrate *AMP-LS* with sampling-based methods, using sampled trajectories as initial seeds for gradient-based optimisation. This hybrid approach could not only improve convergence and help avoid local minima but also provide feasible initial seeds for local gradient refinement, leading to shorter and more efficient trajectories.

7

TWIST: Teacher-Student World Model Distillation for Efficient Sim-to-Real Transfer

This chapter addresses the problem of learning a forward dynamics model from data via generative modelling, enabling planning and policy optimisation for contact-rich manipulation tasks. In particular, this chapter introduces *TWIST*, a framework that facilitates sim-to-real transfer for a world model and an associated policy optimised within this model. This approach substantially reduces the reliance on extensive real-world interaction data to train a world model. In contrast to Chapter 6, which focuses on learning a structured latent space for kinematic planning, *TWIST* centres on optimising a control policy on a world model for contact-rich manipulation tasks.

World models [55], which learn the forward dynamics of an environment, provide a powerful foundation for planning and policy optimisation. Model-based RL approaches [14, 126] that exploit such models have shown clear advantages over model-free RL, particularly in sample efficiency and adaptability to novel tasks. However, most prior works on model-based RL focus on simulated experiments [13, 14, 127] or assume access to large-scale offline, real-world datasets [56] to train a world model, a resource not readily available in many real-world scenarios. Instead of training RL policies directly in the real world, sim-to-real transfer has emerged as a common approach, where policies are first trained in simulation

and subsequently deployed in real-world environments. To effectively transfer the policy from simulation to real-world environments, domain randomisation [57] is a key technique, randomising visual appearance and physics parameters in simulation to improve generalisation in the real world. However, most existing work on sim-to-real transfer is applied to model-free RL [193–195], and naively applying domain randomisation to train a world model for model-based RL is sample-inefficient and computationally demanding [58].

To address these challenges, this chapter presents *TWIST* (see Chapter 7, Fig. 1 for an overview), which leverages privileged information readily available in simulation and employs teacher-student distillation to acquire a vision-based student world model for efficient sim-to-real transfer in model-based RL. In particular, *TWIST* utilises two world models, a teacher and a student, to learn the environment dynamics. Firstly, the teacher model and its associated RL policy are trained using low-dimensional state observations, which are privileged information accessible only within the simulator. This design allows the teacher model to remain unaffected by visual appearance changes introduced by domain randomisation and learn compact latent representations that capture task-relevant information of the environment dynamics. While training the teacher from the state observations, a matching dataset of domain-randomised image observations is generated.

To perform teacher–student distillation, *TWIST* exploits the capabilities of generative modelling in world models (see Chapter 7, Alg. 1). In addition to trajectories stored in a replay buffer during teacher model training, the teacher world model generates synthetic trajectories by rolling out the policy within its learnt dynamics, while the student generates corresponding trajectories from the same initial state and action sequence in its compact latent space. The student is then trained to align its latent representations with those of the teacher, ensuring consistent predictive behaviour. This process enables the student to map domain-randomised image observations into the teacher's latent space more effectively, thereby facilitating effective sim-to-real transfer (see Chapter 7, Fig. 6).

After this distillation process, the student world model and the associated RL policy trained on the world model are transferred to real-world environments. Since the RL policy operates on compact latent representations of the world models, it performs consistently across both models after distillation without requiring re-training.

Furthermore, *TWIST* successfully solves a contact-rich non-prehensile pushing task using a Franka Panda robot in real-world settings, demonstrating the effectiveness of the proposed sim-to-real transfer framework for model-based RL (see Chapter 7, Fig. 5). While this work primarily demonstrates policy optimisation, *TWIST* could also be used for planning, such as MPPI [12].

This chapter demonstrates that a world model trained using generative models in simulation can be effectively transferred to real-world environments via teacher–student distillation using imagined trajectories, thereby eliminating the need for large amounts of real-world interaction data. In summary, Chapter 7 presents the following contributions:

1. *TWIST*, a framework for sample-efficient sim-to-real transfer in model-based RL

2. Distillation in imagination aligns the student's latent dynamics with the teacher, using imagined rollouts generated by the world models.

3. *TWIST* substantially outperforms vision-based model-based RL methods using vanilla domain randomisation, as well as strong model-free baselines, in real-world manipulation tasks.

Jun Yamada, Marc Rigter, Jack Collins, Ingmar Posner, (2024). TWIST: Teacher-Student World Model Distillation for Efficient Sim-to-Real Transfer, *IEEE International Conference on Robotics and Automation (ICRA)*

TWIST: Teacher-Student World Model Distillation for Efficient Sim-to-Real Transfer

Jun Yamada, Marc Rigter, Jack Collins, Ingmar Posner

Abstract— **Model-based RL is a promising approach for real-world robotics due to its improved sample efficiency and generalization capabilities compared to model-free RL. However, effective model-based RL solutions for vision-based real-world applications require bridging the sim-to-real gap for any world model learnt. Due to its significant computational cost, standard domain randomisation does not provide an effective solution to this problem. This paper proposes *TWIST* (Teacher-Student World Model Distillation for Sim-to-Real Transfer) to achieve efficient sim-to-real transfer of vision-based model-based RL using distillation. Specifically, *TWIST* leverages state observations as readily accessible, privileged information commonly garnered from a simulator to significantly accelerate sim-to-real transfer. Specifically, a teacher world model is trained efficiently on state information. At the same time, a matching dataset is collected of domain-randomised image observations. The teacher world model then supervises a student world model that takes the domain-randomised image observations as input. By distilling the learned latent dynamics model from the teacher to the student model, *TWIST* achieves efficient and effective sim-to-real transfer for vision-based model-based RL tasks. Experiments in simulated and real robotics tasks demonstrate that our approach outperforms naive domain randomisation and model-free methods in terms of sample efficiency and task performance of sim-to-real transfer.**

I. Introduction

Deep reinforcement learning (RL) has been applied successfully to challenging control problems such as dexterous manipulation [1], locomotion [2], and Atari [3]. A particularly promising approach is *model-based* RL, which learns a *world model* of the environment, and utilises this model for planning or policy optimisation. Compared to *model-free* approaches, model-based RL holds the potential for broader generalisation [4], improved sample efficiency [5], [6], and faster adaptation to new tasks [7], [8]. However, while model-based RL algorithms have been highly successful in simulated environments [9], [10], their application to real-world robots remains limited due to the need for unsafe or costly data collection [11] to train a world model in the real world.

Instead of training an RL agent directly in the real world, *sim-to-real transfer* is a common approach: learning a policy from easily accessible simulated data and deploying it in the real environment. In real-world environments, we often do not have access to accurate state information, and therefore we wish to learn a policy that utilises images as inputs. To overcome the gap between the simulator and the real world, *domain randomisation* (DR) is often employed. DR exposes the policy to a wide range of simulated environments during training to improve generalisation to the real environment. However, a significant drawback of DR is that policy training on randomised environments requires much more data [12]. Therefore, RL with DR can be extremely computationally intensive and may require weeks of computation time for training to converge [1].

Applied AI Lab, Oxford Robotics Institute, University of Oxford
Correspondence to: jyamada@robots.ox.ac.uk
https://twist-sim2real.github.io/

The vast majority of existing work on sim-to-real transfer is applied to model-free RL [13], [14], [12], [15]. In this work, we address the uninvestigated area of sim-to-real transfer for model-based RL trained from images. By leveraging model-based RL algorithms, we benefit from the improved sample efficiency of model-based approaches [5]. However, to address the sim-to-real gap, it is still necessary to apply DR. Similar to applying DR to the model-free case, naïvely applying DR to model-based approaches increases the amount of data required to train a suitable world model, and is therefore computationally very demanding [8].

To address this, we propose **T**eacher-Student **W**orld model **D**istillation for **S**im-to-Real **T**ransfer (*TWIST*). *TWIST* leverages privileged information in a simulator to achieve efficient and robust sim-to-real transfer for model-based RL. In particular, *TWIST* utilises two world models, a *teacher* and a *student*, to learn the environment. The input to the *teacher* is state information that is only accessible within the simulator. The teacher model is therefore unaffected by appearance changes as introduced by DR and can learn to represent the environment dynamics within a compact latent space much more efficiently than a vision-based model. The teacher model then supervises a *student* world model by encouraging it to encode domain-randomised image observations to the same latent representation as the teacher. We demonstrate that *TWIST* provides efficient and effective sim-to-real transfer for model-based RL, outperforming the standard DR-based approach almost by an order of magnitude in terms of success rate when applied to real-world manipulation tasks.

Our general approach of combining world model distillation with DR is applicable to any model-based RL algorithm. In our implementation, we specifically use the DreamerV2 model architecture [9] to learn the world models and associated policies, and apply our approach to a set of simulated and real robotics environments. We show that our approach successfully achieves transfer to real-world environments, and outperforms naïve DR and model-free approaches in terms of sample efficiency and performance. Our work demonstrates empirically, that there is significant potential for sim-to-real transfer of model-based RL, extending its applicability to a wide range of real-world robotics applications.

II. Related Works

The key concepts that *TWIST* builds upon include model-based RL, sim-to-real transfer, and distillation using privileged information. We review the relevant literature of each of these concepts in turn.

Model-based RL has emerged as a promising approach to solving complex control problems by leveraging a learned dynamics model [16], [17], [18]. To achieve the desired behaviour, the dynamics model (or "world" model) can be used for planning [5], [19], [10], [20], or policy optimisation [16], [9], [21], [22]. To handle partially observable environments [23] with high-dimensional observations such as images, a common approach is to employ a recurrent state-space model (RSSM) [24], [16], [17], which predicts transitions in a compact latent space with a recurrent module. Despite considerable success on simulated environments, such as Atari [25] and DMControl [26], applications of vision-based model-based RL to real-world robotics tasks remain limited due to the need for a large number of samples to train the world model [27], [28]. Existing works on model-based RL from images for robotics [27], [28] build upon a suite of Dreamer algorithms [29], [9], [21], which achieves state-of-the-art performance on simulated domains by optimising a policy using only synthetic data generated by the model. DayDreamer [27] relies upon either state information or discretised action-spaces to simplify robotics tasks, and to facilitate learning a model from data collected directly in the real world. Existing approaches to transferring Dreamer from simulation to real robots either require state information [30] or only demonstrate transfer to near-identical real-world environments [28].

Sim-to-real transfer [31] trains a policy using simulated data, and deploys the policy in the real world. Existing approaches to sim-to-real transfer utilise techniques such as domain randomisation (DR) [32], system identification [33], and domain adaptation [34]. DR is a particularly simple, yet effective approach to expose agents to a wide range of instances of the same environment by randomising visual and dynamics parameters. By training policies using DR, agents become more robust to domain mismatches [32]. Previous work on sim-to-real transfer using DR has been primarily applied to model-free RL methods [13], [35], [36] or imitation learning [37].

Compared to sim-to-real transfer of model-free RL algorithms, model-based sim-to-real methods remain relatively unexplored. To our knowledge, [30] is the only work to transfer a model-based method across the sim-to-real gap. The authors accomplish this using a state-based Dreamer model that requires privileged information *in the real world*. Enabling sim-to-real transfer of Dreamer from image observations will help to unleash the potential of model-based RL for real-world applications where state information is not available.

Leveraging **privileged information** to accelerate the training of policies is a common approach. Specifically, [13], [35] utilise information asymmetric actor-critic methods to train the critic faster via access to the privileged information while providing only images for the actor.

Another common technique to make use of the privilege information is **Distillation**, which transfers knowledge about a task from one or multiple teachers to a student. In RL, knowledge transfer is generally achieved via *policy* distillation: training a student policy to imitate a teacher policy [38], [14], [39], [40], [41]. Our work is most closely related in spirit to [38], [14], [41] in that distillation and DR are used to efficiently train a teacher policy from privileged information and distil it into a student policy for sim-to-real transfer. However, for distillation, the prior works focus on model-free RL, which often requires additional trajectories collected by either the teacher or student policy to match the action distribution.

In contrast to these works, we consider model-based RL conditioned on image observations and introduce a novel method for *world model* distillation. Our approach achieves knowledge transfer by supervising a student world model instead of a policy without the need for additional data collection during the distillation. We demonstrate that our approach achieves strong performance for sim-to-real transfer in both simulated and real environments.

III. Preliminaries

In this section, we describe our problem setting and the Dreamer model-based RL algorithm [29], [9], [21]. We implement our approach using Dreamer as it is a commonly used state-of-the-art model-based RL algorithm that demonstrates the capability of our world model distillation approach.

A. Problem Formulation

The real environment is a partially observable Markov Decision Process represented by the tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{I}, r, \gamma, \mathbb{S})$, where: $\mathcal{S}$ is a set of continuous states, $\mathcal{O}$ is a set of image observations, $\mathcal{A}$ is a set of continuous actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the transition function, $\mathcal{I} : \mathcal{O} \times \mathcal{S} \to \mathbb{R}$ is the observation function, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, γ is the discount factor, and $\mathbb{S}$ is the initial state distribution. The goal is to maximise the expected discounted reward $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$.

In our problem setting, we do not have access to the real environment during training. Instead, we have access to a simulator that approximates the real environment. In the simulator, we have direct access to privileged information, $s \in \mathcal{S}$, in addition to randomised image observations $o \in O$.

B. Dreamer

Dreamer [9], [21] is a model-based RL method that learns a world model from pixels or state observations and trains an actor-critic agent by leveraging imagined trajectories from the world model.

a) World Model: Dreamer uses a Recurrent State Space Model (RSSM) [17] to learn the dynamics of environments, consisting of the following modules:

$$\text{RSSM} \begin{cases} \text{Sequence model:} & h_t = f_\phi\left(h_{t-1}, z_{t-1}, a_{t-1}\right) \\ \text{Representation model:} & z_t \sim q_\phi\left(z_t \mid h_t, x_t\right) \\ \text{Dynamics predictor:} & \hat{z}_t \sim p_\phi\left(\hat{z}_t \mid h_t\right) \\ \text{Reward predictor:} & \hat{r}_t \sim p_\phi\left(\hat{r}_t \mid h_t, z_t\right) \\ \text{Decoder:} & \hat{x}_t \sim p_\phi\left(\hat{x}_t \mid h_t, z_t\right) \end{cases} \tag{1}$$

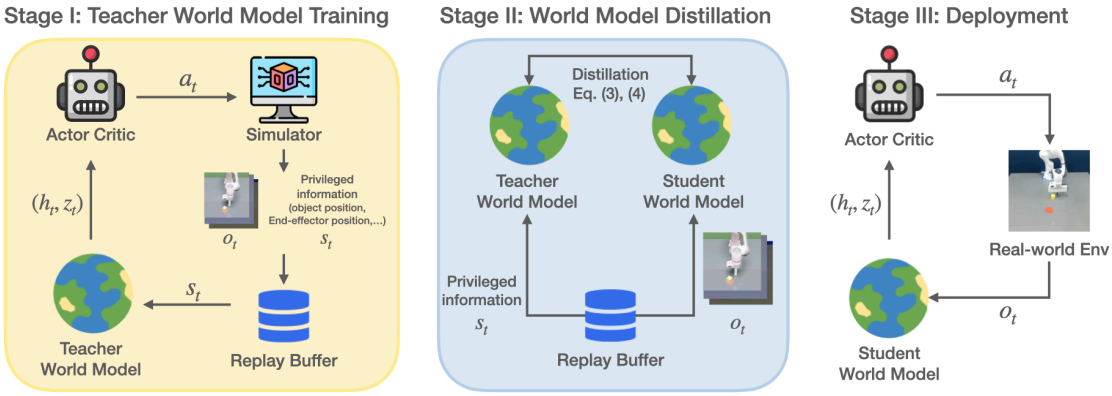


Fig. 1: Overview of *TWIST*. While a teacher world model is trained from privileged information, domain-randomised image observations are collected for distillation. The teacher supervises a student trained from the domain-randomised images to imitate the compact latent states of the teacher. The student world model is then transferred to real-world environments.

All modules are implemented as neural networks parameterised by ϕ. In the RSSM, the state is jointly represented by a recurrent deterministic component, h_t, and a stochastic component represented by a categorical distribution. At each step, the RSSM uses h_t to compute two distributions over the stochastic state: z_t and $\hat{z}$. The stochastic posterior state z_t encodes information about the current input observation x_t, while the prior state $\hat{z}_t$ is a prediction of the posterior state z_t without access to the current input observation. Therefore, by learning to predict $\hat{z}_t$, the model learns to predict the dynamics of the environment. Given the posterior state, the decoder and reward predictor are trained to reconstruct the current input observation x_t and the reward r_t, respectively. These models are jointly learned by minimising the negative variational lower bound [42].

$$\mathcal{L}(\theta) \doteq \mathbb{E}_{q_\theta(s_{1:T}|a_{1:T},x_{1:T})}\Big[\sum_{t=1}^{T}(-\ln p_\theta(x_t \mid h_t, z_t) - \ln p_\theta(r_t \mid h_t, z_t) + \beta KL\left[q_\theta(z_t \mid h_t, x_t)\|p_\theta(\hat{z}_t \mid h_t)\right]\Big] \tag{2}$$

Once the model has been trained, it can be rolled out without access to any input observations by utilising the prior $\hat{z}$ in place of the posterior z. This enables the model to generate unlimited synthetic or *imagined* trajectories of the form: $\{\{h_t, \hat{z}_t, a_t, r_t\}_{t=0}^{t=T}\}$, where T is the time horizon for imagination.

b) Actor-Critic Learning: To learn a policy, Dreamer leverages an actor-critic algorithm that is trained using synthetic data generated by the world model. Given a particular RSSM state $(h_t, \hat{z}_t)$, the critic is trained to predict the total expected reward. The actor (i.e. the policy) is trained to output a distribution over actions, $\pi(a_t|h_t, \hat{z})$, that maximises the total expected reward given the current state.

IV. TWIST: Teacher-Student World Model Distillation for Sim-to-Real Transfer

Dreamer is capable of efficiently solving diverse vision-based continuous control tasks in simulated environments by explicitly learning a task-agnostic world model. To transfer Dreamer to real-world robotics tasks, domain randomisation (DR) is required to bridge the gap between simulation and real-world environments. However, DR dramatically increases the number of samples, and therefore computation time, required for training. To address this issue, we propose *TWIST* (**T**eacher-Student **W**orld Model **D**istillation for **S**im-to-Real **T**ransfer) to efficiently train a world model for vision-based tasks in simulation which readily transfers into real-world environments. In this section, we describe our approach to distilling the teacher to the student world model (see Fig. 1).

A. Overview

A simulator affords access to state information in addition to domain-randomised images. *TWIST* leverages this privileged information in order to accelerate the sim-to-real transfer of model-based RL. Specifically, *TWIST* initially trains a teacher world model and associated policy based on state information. Because the teacher learns from state information, an accurate world model and strong policy can be trained from only a small number of samples.

However, in real-world environments, privileged information is not usually available. To overcome this issue, the teacher is distilled into a vision-based student world model. While training the teacher from the state observations s_t, privileged information easily accessible in simulation, a matching dataset of domain-randomised image observations o_t is generated, denoted as $\mathcal{D} = \{(s_t, o_t, a_t, r_t), ...\}$. The student is trained to imitate the RSSM latent states of the teacher while operating on the corresponding domain-randomised raw pixel inputs o_t from the dataset $\mathcal{D}$. Aligning these representations enables effective knowledge transfer and achieves sample-efficient sim-to-real transfer.

B. World Model Distillation

Given the teacher world model trained on state information, the teacher supervises the student to imitate the dynamics of the environment. Specifically, the student is trained to imitate the prior distribution $p(\hat{z}_t^{\text{teacher}}|h_t^{\text{teacher}})$, posterior distribution $q(z_t^{\text{teacher}}|h_t^{\text{teacher}}, s_t)$, and deterministic representations h_t^{teacher} of the teacher for a trajectory τ

of length L sampled from the dataset, $\mathcal{D}$:

$$\mathcal{L}_{\text{distil}}(\tau) = \mathbb{E}_{\{(a_t,o_t,s_t)\}_{t=k}^{k+L}\sim\mathcal{D}} \sum_{t=k}^{k+L} \Bigg[\underbrace{||h_t^{\text{teacher}} - h_t^{\text{student}}||_2^2}_{\text{Deterministic representation distillation}} + \underbrace{\mathbb{KL}[p_\theta(\hat{z}_t^{\text{student}}|h_t^{\text{student}})||p_\phi(\hat{z}_t^{\text{teacher}}|h_t^{\text{teacher}})]}_{\text{Prior distillation}} + \underbrace{\mathbb{KL}[q_\theta(z_t^{\text{student}}|h_t^{\text{student}}, o_t)||q_\phi(z_t^{\text{teacher}}|h_t^{\text{teacher}}, s_t)]}_{\text{Posterior distillation}} \Bigg] \tag{3}$$

where ϕ and θ represent the parameters of the teacher and student world model, respectively. Note that the parameter of the teacher world model ϕ is frozen during the distillation.

In addition to distilling the two stochastic distributions and deterministic representations, we further derive a training signal for distribution alignment by matching imagined rollouts in both the teacher and the student models. (Algorithm 1). Specifically, a set of initial latent states in each world model is computed by embedding the trajectories τ sampled from the dataset $\mathcal{D}$ (see lines 6 and 7). Starting from the initial states of the teacher, we then generate an imagined rollout $\hat{\tau}^{\text{teacher}} = \{(\hat{z}_i^{\text{teacher}}, h_i^{\text{teacher}}, a_i^{\text{teacher}})\}_{i=t}^{t+H}$ with the time horizon H using the policy π learned with the teacher model (line 8). We also collect an imagined trajectory $\tau^{student}$ in the student world model by replaying the same sequence of actions $\{a_i^{\text{teacher}}\}_{i=1}^{H}$ used for trajectory imagination in the teacher (line 10). Then, we align the prior distribution $p(\hat{z}_t|h_t)$ and deterministic representation h_t in the trajectories generated by the teacher and student world model:

$$\mathcal{L}_{\text{imagined}}(\hat{\tau}^{student}, \hat{\tau}^{\text{teacher}}) = \sum_{i=1}^{H} \Bigg[\underbrace{||h_i^{\text{teacher}} - h_i^{student}||_2^2}_{\text{Deterministic representation distillation}} + \underbrace{\mathbb{KL}[p_\theta(\hat{z}_i^{\text{student}}|h_i^{\text{student}})||p_\phi(\hat{z}_i^{\text{teacher}}|h_i^{\text{teacher}})]}_{\text{Distillation in Imagination}} \Bigg] \tag{4}$$

where $(h_i^{\text{teacher}}, z_i^{\text{teacher}})$ and $(h_i^{\text{student}}, z_i^{\text{student}})$ are the i^{th} entries in $\hat{\tau}^{\text{teacher}}$ and $\hat{\tau}^{\text{student}}$ respectively. To ensure diversity in the imagined trajectories, random noise is added to the action a_t sampled from the policy $\pi(a_t|h_t^{\text{teacher}}, \hat{z}_t^{\text{teacher}})$ when rolling it out in the teacher world model. This bootstraps the trajectories in the dataset $\mathcal{D}$; thus, the student can imitate the prior distribution and deterministic representation of the teacher more accurately. The loss function for world model distillation is therefore $\mathcal{L} = \mathcal{L}_{\text{distill}} + \mathcal{L}_{\text{imagined}}$. Our experimental results demonstrate that, after distillation, an actor trained in the teacher world model successfully transfers to real-world environments as the student world model is trained to imitate the RSSM latent states of the teacher.

V. Implementation Details

Our encoder and decoder of the teacher world model consists of three fully connected hidden layers with 512 units and ELU activation. We use the same architecture for the encoder, decoder, and actor-critic agent of vision-based world models as those used in [9]. For distillation, a trajectory of length $L = 50$ is sampled from the dataset $\mathcal{D}$ (see Eq. 3) and an imagined trajectory of length $H = 15$ is generated (see Eq. 4). All of the agents are trained on a single GeForce RTX 3090 for $500K$ environment steps.

Algorithm 1 TWIST: Teacher-Student World Model Distillation for Sim-To-Real Transfer

1: **Inputs:** Dataset $\mathcal{D} = \{(s_i, o_t, a_t, r_t), ...\}$; Teacher world model W_ϕ^{teacher}; Policy $\pi(a_t|h_t, \hat{z}_t)$
2: **Initialise:** Student world model $W_\theta^{\text{student}}$
3: **while** distilling world model **do**
4: $\quad\tau = \{(a_t, o_t, s_t)\}_{t=k}^{k+L} \sim D$
5: $\quad$Compute $\mathcal{L}_{\text{distill}}$ via Eq. 3 using τ
6: $\quad\mathcal{Z}_\tau^{\text{teacher}} = \{z_t^{\text{teacher}}\}_{t=k}^{k+L} \leftarrow q_\phi(\tau)$
7: $\quad\mathcal{Z}_\tau^{\text{student}} = \{z_t^{\text{student}}\}_{t=k}^{k+L} \leftarrow q_\theta(\tau)$
8: $\quad\hat{\tau}^{teacher}$ = IMAGINE(W_ϕ^{teacher}, Z_τ^{teacher})
9: $\quad A^{\text{teacher}} \leftarrow \{a_i\}_{i=1}^{H}$ in $\hat{\tau}^{\text{teacher}}$
10: $\quad\hat{\tau}^{\text{student}}$ = IMAGINE($W_\theta^{\text{student}}$, Z_τ^{student}, A^{teacher})
11: $\quad$Compute $\mathcal{L}_{\text{imagined}}$ via Eq. 4
12: $\quad\theta \leftarrow \theta - \alpha\nabla_\theta(\mathcal{L}_{\text{distill}} + \mathcal{L}_{\text{imagined}})$
13: **function** IMAGINE(W, Z_{init}, A = None)
14: $\quad$**if** A is None **then** ▷ Imagination in W^{teacher}
15: $\quad\quad\hat{\tau} \leftarrow$ rollout π for H steps from $z \in Z_{\text{init}}$ in W
16: $\quad$**else** ▷ Imagination in W^{student}
17: $\quad\quad\hat{\tau} \leftarrow$ rollout $a \in A$ from $z \in Z$ in W
18: $\quad$**return** $\hat{\tau}$ ▷ $\hat{\tau} = \{(\hat{z}_i, h_i, a_i)\}_{i=1}^{H}$

VI. Experiments

The efficacy of *TWIST* for sim-to-real transfer is evaluated through experiments in both simulated and real-world environments. The experiments aim to answer the following questions: (1) does *TWIST* enable efficient sim-to-real transfer for model-based RL using DR? and (2) does the distillation for imagined trajectories improve the task performance compared to performing distillation only on the original dataset?

A. Baselines

We compare *TWIST* against several competitive baselines, including Dreamer agents with different training methods and model-free RL. *Oracle* is a Dreamer agent trained from privileged information. The performance of the oracle agent is an upper bound on the performance of our method. Since we do not have access to state information in real-world settings, we only provide the performance of the oracle approach in the experiments conducted in simulation environments. *Dreamer w/ DR* is a vision-based Dreamer agent trained with naive DR. *Dreamer w/o DR* is an agent trained without DR. *Dreamer State Recon.* is a vision-based Dreamer agent trained to reconstruct state information from domain-randomised image observations, which is an alternative way of leveraging privileged information. Lastly, *Asymmetric SAC* [13] is a sample-efficient state-of-the-art model-free RL algorithm suitable for DR. While the critic network is trained from privileged information, the policy is trained from domain-randomised image observations.

B. Simulated Results

Firstly, we empirically demonstrate the efficacy of *TWIST* on a set of continuous control tasks in the Distracting Control

Tasks (500K Steps)	Oracle Dreamer	TWIST	Dreamer w/o DR	Dreamer w/ DR	Dreamer State Recon.	Asymmetric SAC
Cup Catch	936.6 ± 0.1	856.6 ± 29.6	150.7 ± 80.3	744.3 ± 93.8	627.0 ± 194.7	873.0 ± 11.1
Cartpole, Balance	992.9 ± 1.7	954.5 ± 37.4	349.3 ± 19.0	590.8 ± 23.0	869.7 ± 40.4	353.1 ± 18.6
Cheetah Run	597.1 ± 24.3	506.0 ± 54.2	206.5 ± 40.3	476.4 ± 61.1	391.6 ± 4.5	222.5 ± 20.4
Hopper Stand	501.1 ± 38.9	483.3 ± 118.9	42.1 ± 11.2	471.8 ± 48.4	358.2 ± 34.4	57.7 ± 86.8
Walker Walk	800.4 ± 53.7	665.8 ± 80.3	182.7 ± 3.4	394.6 ± 25.1	491.0 ± 145.6	439.4 ± 57.0
Finger, Easy Turn	904.4 ± 32.8	798.0 ± 50.3	182.5 ± 26.0	440.7 ± 35.7	553.4 ± 42.2	304.4 ± 22.5

TABLE I: Averaged episodic rewards and standard deviation obtained from 100 trials with 3 seeds in the Distracting Control Suite. The evaluation is conducted using held-out environments.

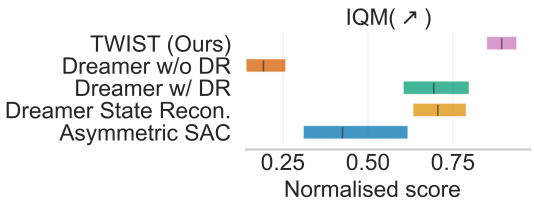


Fig. 2: Aggregated Interquartile Mean (IQM) of normalised episodic rewards with 95% bootstrap CI based on 5 tasks from 100 trials with 3 seeds evaluated using held-out environments in Distracting Control. The lack of overlap with the CIs between TWIST and the baseline methods indicates that the difference is statistically significant.

Suite [43], an extended version of the DMControl [26].

1) Experiment Setup: First, a teacher world model and a policy are trained using Dreamer from ground-truth state information. During training, domain-randomised images are collected by randomizing the background texture used in prior work [44] and the colour of objects every timestep for diverse data acquisition. After training the teacher, we use the domain-randomised image observations to distil the state-based teacher world model into a vision-based student world model. For evaluation, we sample the object colours from the same distribution as training, but the background texture is sampled from a held-out test distribution. Therefore, the distribution of environments for evaluation is different to the training time environments. Note that DR is applied only at the beginning of the episode for the evaluation because the textures are usually consistent at test time.

2) Results: Table I reports the average episodic rewards for six continuous control tasks evaluated on hold-out scenes from the Distracting Control Suite. *TWIST* outperforms the baseline approaches, including model-free RL, often by significant margins. While *Asymmetric SAC* shows comparable performance on the simple *Cup Catch* task, it does not perform well on more complex tasks because the policy struggles to learn task-relevant information efficiently from domain-randomised images due to its visual complexity. *Dreamer State Recon.* and *Dreamer w/ DR* demonstrate better performance among the baselines. However, learning task-relevant information and the actor-critic agent jointly on limited samples is often challenging, resulting in worse performance compared to our approach. *Dreamer w/o DR* does not perform well in any of the six tasks due to the lack of generalisation to unseen scenes.

To assess the statistical significance of our results, Fig. 2 reports Interquartile Mean (IQM) of normalised episodic rewards with 95% bootstrap confidence interval (CI) aggregated across 5 tasks in Distracting Control, computed using [45]. The episodic rewards of each task are normalised by the performance of *Oracle Dreamer* to aggregate the results and validate the efficacy of our method. As shown in Fig. 2, our method is substantially more performant than the baselines. The lack of overlap with the CIs of the baseline method further indicates that this difference is statistically significant.

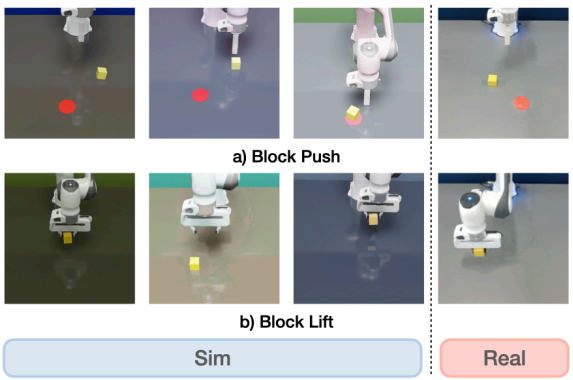


Fig. 3: Sim-to-real manipulation tasks. (a) *Block Push*: A Franka Panda arm pushes the yellow block towards the red goal marker. (b) *Block Lift*: The arm grasps the yellow block and lifts it 10cm above the table

C. Sim-to-Real Transfer for Manipulation Tasks

In this section, we consider sim-to-real transfer for manipulation tasks to verify the effectiveness of *TWIST* in the real world.

1) Experimental setup: In our experiments, a Franka Panda robot is used. In real-world experiments, RGB image observations are taken from a RealSense D435i camera. In the simulation, agents are trained in Omniverse Isaac Orbit [46] powered by Omniverse Isaac Sim [47]. DR is applied to the brightness of the light and texture of the robot body, background, table, and objects every timestep to collect diverse image observations. Further, the friction of objects is randomised in every episode. The action space of the policy is a delta-position of the end-effector in Cartesian coordinates with a maximum delta of 2cm.

2) Tasks: We conduct experiments to showcase the successful sim-to-real transfer capability of *TWIST*, focusing on the *block push* and *block lift* tasks (see Fig. 3). The objective of the *block push* task is to push a $4\text{cm} \times 4\text{cm}$ cube towards a designated red goal marker. If the distance between the centre of the cube and the goal marker is less than 5cm at the end of the episode, then the trial is considered successful. The cube and goal marker positions are uniformly sampled. For the block push task, we replace the robot's hand with a

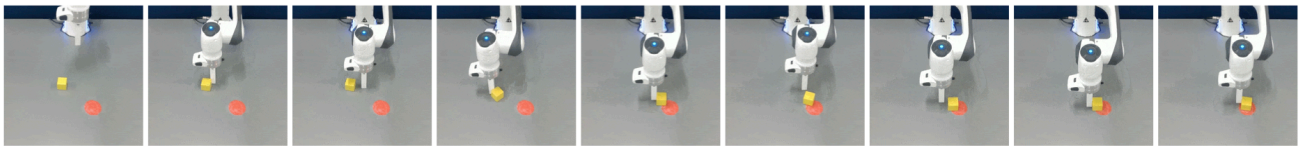


Fig. 4: Example rollouts of the proposed method on the real-world Block Push task. Our method successfully transfers the student world model and solves the block push task in the real world.

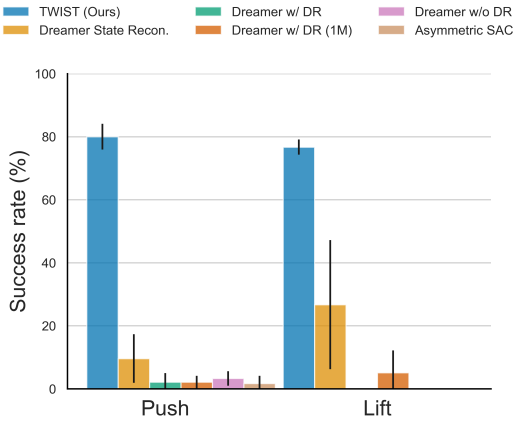


Fig. 5: Success rate on real-world tasks. The success rate and standard deviation are calculated from 20 trials with 3 seeds. *TWIST* significantly outperform baselines including naïve Dreamer with DR and model-free RL.

3D-printed peg to push the block because the original robot's hand often occludes the block from the third-person camera.

The goal of the *block lift* task is to grasp the cube and lift it 10cm above the tabletop by the end of the episode. To train agents in simulation, we define a dense reward function tailored to each task. Privileged information available in these tasks includes end-effector position, object pose, and L2 distance between the object and goal position. Additionally, in the *block lift* task, a grasp state is used to determine whether the object is grasped or not. The episode length of these tasks is 150 timesteps. In real-world experiments, we randomise the camera position and brightness of the scene randomly to ensure robustness of the distilled agents.

3) Results: The success rate for each task across 20 trials averaged over 3 seeds is reported in Fig. 5. Compared to the baselines, including naïve Dreamer with DR and model-free RL, *TWIST* demonstrates significantly better success rates in both *block push* and *lift* tasks. In particular, the block push task requires an accurate dynamics model to successfully push the box towards the goal marker, indicating that our world model is successfully distilled and transferred from simulation to real-world environments. The baseline methods often fail to solve the task, because those methods require more samples to successfully train agents in simulation with DR [8]. *Dreamer State Recon.* shows a better success rate than other baselines. However, it still struggles to learn task-relevant information in image observations effectively while exploring environments for solving manipulation tasks. Although naïve Dreamer agent with DR is also trained from 1M samples (*Dreamer w/ DR* (1M)), its success rate on the *block push* and *block lift* tasks remains low, indicating the sample inefficiency of the naïve DR approach.

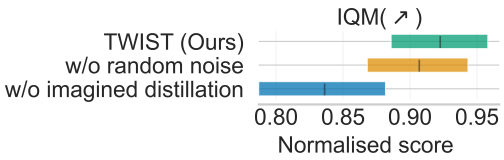


Fig. 6: Interquartile Mean (IQM) of normalised episodic rewards with 95% bootstrap CI to ablate the key components of the proposed distillation method in the Distracting Control Suite. The following variants are compared: (1) the full proposed method, (2) without random noise to actions for imagined distillation, (3) without imagined distillation.

D. Ablation Study

We ablate the distillation for imagined trajectories (*imagined distillation*) (see Eq. 4) and random noise added to actions for the imagined distillation in the Distracting Control Suite. We report normalised aggregated Interquartile Mean (IQM) with a 95% bootstrap confidence interval. As shown in Fig. 6, the CI for *our method* and *our method w/o imagined distillation* do not overlap, indicating that the difference in performance is statistically significant. On the other hand, the gap between *our method* and *our method w/o random noise* is smaller but still notable in practice. These results highlight that distillation using imagined rollouts is particularly important for successful world model distillation.

VII. Conclusion

We propose *TWIST* for efficient sim-to-real transfer of model-based RL. Specifically, a teacher world model trained from privileged information supervises a student world model taking as input domain-randomised image observations to mimic the compact latent states of the teacher. Our experiments demonstrate successful distillation from the teacher world model to the student world model with domain randomisation in simulated environements and further show the efficient and robust sim-to-real transfer for robot manipulation tasks into real-world domains.

TWIST is therefore a significant step towards unlocking the benefits of model-based RL for real-world applications. In future work we will look to explore fine-tuning the distilled world model from few real-world image observations to efficiently acquire new skills in the real world.

ACKNOWLEDGMENT

This work was supported by a UKRI/EPSRC Programme Grant [EP/V000748/1]. We would also like to thank the University of Oxford for providing Advanced Research Computing (ARC) and the SCAN facility in carrying out this work (`http://dx.doi.org/10.5281/zenodo.22558`).

REFERENCES

[1] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[2] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, "Learning to walk via deep reinforcement learning," *Robotics: Science and Systems*, 2019.

[3] 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*, vol. 518, no. 7540, pp. 529–533, 2015.

[4] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn, "Combo: Conservative offline model-based policy optimization," *Advances in neural information processing systems*, vol. 34, pp. 28 954–28 967, 2021.

[5] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *Advances in neural information processing systems*, vol. 31, 2018.

[6] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*, 2011, pp. 465–472.

[7] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak, "Planning to explore via self-supervised world models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8583–8592.

[8] M. Rigter, M. Jiang, and I. Posner, "Reward-free curricula for training robust world models," *arXiv preprint arXiv:2306.09205*, 2023.

[9] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," 2022.

[10] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[11] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[12] S. Salter, D. Rao, M. Wulfmeier, R. Hadsell, and I. Posner, "Attention-privileged reinforcement learning," in *Conference on Robot Learning*. PMLR, 2021, pp. 394–408.

[13] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," 2017.

[14] J. Brosseit, B. Hahner, F. Muratore, M. Gienger, and J. Peters, "Distilled domain randomization," *arXiv preprint arXiv:2112.03149*, 2021.

[15] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," 2019.

[16] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.

[17] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International conference on machine learning*. PMLR, 2019, pp. 2555–2565.

[18] R. S. Sutton, "Dyna, an integrated architecture for learning, planning, and reacting," *ACM Sigart Bulletin*, vol. 2, no. 4, pp. 160–163, 1991.

[19] R. Y. Rubinstein, "Optimization of computer simulation models with rare events," *European Journal of Operational Research*, vol. 99, no. 1, pp. 89–112, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221796003852

[20] G. Williams, A. Aldrich, and E. Theodorou, "Model predictive path integral control using covariance variable importance sampling," 2015.

[21] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[22] M. Rigter, B. Lacerda, and N. Hawes, "RAMBO-RL: Robust adversarial model-based offline reinforcement learning," *Advances in Neural Information Processing Systems*, 2022.

[23] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[24] J. Schmidhuber, "Reinforcement learning in markovian and non-markovian environments," *Advances in neural information processing systems*, vol. 3, 1990.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013.

[26] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. Lillicrap, and M. Riedmiller, "Deepmind control suite," 2018.

[27] P. Wu, A. Escontrela, D. Hafner, K. Goldberg, and P. Abbeel, "Daydreamer: World models for physical robot learning," 2022.

[28] Y. Seo, J. Kim, S. James, K. Lee, J. Shin, and P. Abbeel, "Multi-view masked world models for visual robotic manipulation," *arXiv preprint arXiv:2302.02408*, 2023.

[29] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=S1lOTC4tDS

[30] A. Brunnbauer, L. Berducci, A. Brandstátter, M. Lechner, R. Hasani, D. Rus, and R. Grosu, "Latent imagination facilitates zero-shot transfer in autonomous racing," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7513–7520.

[31] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2020, pp. 737–744.

[32] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[33] M. Lutter, J. Silberbauer, J. Watson, and J. Peters, "Differentiable physics models for real-world offline model-based reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 4163–4170.

[34] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4243–4250.

[35] S. Salter, D. Rao, M. Wulfmeier, R. Hadsell, and I. Posner, "Attention privileged reinforcement learning for domain transfer," 2020. [Online]. Available: https://openreview.net/forum?id=HygW26VYwS

[36] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[37] S. James, A. J. Davison, and E. Johns, "Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task," in *Conference on Robot Learning*. PMLR, 2017, pp. 334–343.

[38] I.-C. A. Liu, S. Uppal, G. S. Sukhatme, J. J. Lim, P. Englert, and Y. Lee, "Distilling motion planner augmented policies into visual control policies for robot manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 641–650.

[39] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell, "Policy distillation," 2016.

[40] W. M. Czarnecki, R. Pascanu, S. Osindero, S. M. Jayakumar, G. Swirszcz, and M. Jaderberg, "Distilling policy distillation," 2019.

[41] T. Chen, J. Xu, and P. Agrawal, "A system for general in-hand object re-orientation," in *Conference on Robot Learning*. PMLR, 2022, pp. 297–307.

[42] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[43] A. Stone, O. Ramirez, K. Konolige, and R. Jonschkowski, "The distracting control suite – a challenging benchmark for reinforcement learning from pixels," 2021.

[44] N. Hansen and X. Wang, "Generalization in reinforcement learning by soft data augmentation," 2021.

[45] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare, "Deep reinforcement learning at the edge of the statistical precipice," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[46] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[47] NVIDIA, "Nvidia isaac sim." [Online]. Available: https://developer.nvidia.com/isaac-sim

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).**

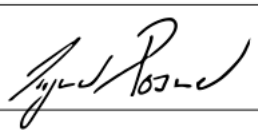
Title of Paper	TWIST: Teacher-Student World Model Distillation for Efficient Sim-to-Real Transfer
Publication Status	■Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Jun Yamada, Marc Rigter, Jack Collins, Ingmar Posner Published at IEEE International Conference on Robotics and Automation (ICRA), 2024

Student Confirmation

Student Name:	Jun Yamada		
Contribution to the Paper	- Proposed the research idea - Created and developed methodologies - Ran all experiments - Created all figures - Paper writing		
Signature		Date	19/09/2025

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: Prof. Ingmar Posner		
Supervisor comments *Jun made a substantial contribution to the publication. The description above is accurate.*		
Signature 	Date	22/09/2025

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

7.1 Limitations and Future Work

TWIST demonstrates the efficacy of sim-to-real transfer in model-based RL by performing teacher-student distillation in imagined trajectories. While *TWIST* optimises a policy on the world model to solve manipulation tasks, it can also be applied for planning, similar to TDMPC [15], given a suitable cost function for real-world deployment. For example, the cost function is defined as the cosine distance between the current and goal latent representations extracted from goal observations.

Although *TWIST* is built on Dreamer [13, 14], it can also be applied to other model-based RL methods. In particular, TDMPC [127, 196] has shown strong performance in high-dimensional control problems, including challenging manipulation tasks. Since distillation in imagined trajectories is compatible with any model-based RL method capable of predicting future trajectories and learning latent representations, integrating this approach into TDMPC is a promising direction.

While *TWIST* is primarily evaluated under controlled visual conditions, real-world industrial environments introduce additional perceptual challenges. Factors such as strong illumination, specular reflections, and glare from highly reflective metal surfaces can significantly degrade observation quality, which may in turn adversely affect sim-to-real transfer due to the domain gap. Although domain randomisation partially accounts for visual variability, it may not sufficiently capture these complex photometric effects.

Future work could therefore explore more targeted strategies, for example, incorporating real-to-sim transfer to better bridge the gap between simulation and real-world environments. Additionally, fine-tuning the world model, originally trained on domain-randomised simulation data, using limited real-world interactions, presents a promising approach for further improving model accuracy. Another potential direction includes integrating the learnt world model with MPC augmented by a value function to solve contact-rich non-prehensile manipulation tasks more safely, particularly in cluttered environments, inspired by *Grasp-MPC* introduced in Chapter 4.

8

D-Cubed: Latent Diffusion Trajectory Optimisation for Dexterous Deformable Manipulation

Chapters 6 and 7 focus on learning dynamics models using generative models to facilitate effective planning and policy optimisation within the learnt dynamics model. In contrast, this chapter employs generative models to learn an action sampler that facilitates more effective exploration, thereby improving the performance of model-based planning in contact-rich manipulation tasks. Rather than learning a dynamics model, this chapter assumes access to a simulated environment for planning. Nevertheless, it can be effectively integrated with a learnt dynamics model, particularly the model presented in Chapter 7.

The ability to manipulate deformable objects is a crucial skill in robotics, extending far beyond common household tasks to include industrial applications such as handling rubber materials in the assembly process. While parallel grippers have been widely used in robotics, they often lack the dexterity and adaptability required to effectively manipulate deformable objects, limiting their applicability in complex manipulation tasks.

On the other hand, dexterous robot hands offer the flexibility and versatility necessary for manipulating a wide range of deformable materials. However, controlling such dexterous robot hands presents significant challenges due to their high

degrees of freedom. Trajectory optimisation methods such as MPPI have shown promising results for rigid object dexterous manipulation tasks [85]. However, they often struggle with deformable object manipulation, where the high DoF of the robot hand, the large number of contacts, and the sparse task information in the cost function make effective exploration particularly difficult.

To address these challenges, this chapter introduces *D-Cubed*, a novel learning-based trajectory optimisation approach that uses a latent diffusion model (LDM) trained on a task-agnostic play dataset of a robot hand as an action sampler, facilitating efficient exploration for trajectory optimisation to solve dexterous deformable manipulation tasks (see Chapter 8 Fig. 1 for an overview). The task-agnostic play dataset, collected within approximately 20 minutes by tracking a human hand and retargeting these motions to a robot hand (see Chapter 8, Fig. 3), encompasses a diverse range of representative hand movements, including closing and opening the hand, as well as moving individual fingers, and does not include any interaction with deformable objects. *D-Cubed* learns a skill-latent space that encodes short-horizon action sequences from the play dataset using a VAE (see Chapter 8, Fig. 2 (1)). An LDM is then trained to compose these skill-latent representations into a skill trajectory that corresponds to the long-horizon motion of the robot hand sampled from the task-agnostic play dataset (see Chapter 8, Fig. 2 (2)). The LDM, which can generate diverse trajectories of meaningful robot hand motions, is used as an action sampler that can effectively search the large solution space for trajectory optimisation. To optimise trajectories for a given task, *D-Cubed* introduces gradient-free guided sampling that leverages a variant of the Cross-Entropy Method (CEM) integrated within the reverse diffusion process (see Chapter 8, Fig. 2 (3)). Initially, random noise samples drawn from a normal distribution are fed to the LDM, generating several candidate action trajectories. These trajectories are evaluated using the provided dynamics model, which computes costs based on the discrepancy between the achieved and desired deformable object shapes for each task. The trajectory with the lowest cost (i.e., best-performing trajectory) is then selected and refined through additional denoising steps within the reverse diffusion process,

while those with higher costs are discarded. This procedure is repeated until the reverse diffusion process is completed, progressively improving the action sequence through iterative refinement. This approach allows the LDM to efficiently explore the large solution space while iteratively refining action sequences to achieve the target object shape by minimising the predefined task-specific cost function.

Through empirical evaluation on DexDeform [89], a publicly available benchmark of dexterous deformable object manipulation tasks, *D-Cubed* achieves substantially higher performance than sampling-based and gradient-based trajectory optimisation methods, as well as competitive baselines such as PPO [31] (see Chapter 8, Tab. 1). *D-Cubed* further demonstrates that a sequence of actions optimised in simulation is successfully transferred to a real-world deformable manipulation task.

D-Cubed improves trajectory optimisation with a generative model, enabling versatile and efficient skill acquisition for complex contact-rich manipulation tasks using a readily collected, task-agnostic play dataset without relying on sample-inefficient policy learning or costly expert demonstrations. In summary, this chapter presents the following contributions:

1. *D-Cubed*, a trajectory optimisation framework using LDMs as an action sampler to tackle dexterous deformable object manipulation tasks.

2. A gradient-free guided sampling method that incorporates CEM into the reverse diffusion process to enable effective trajectory optimisation.

3. Significant performance improvements over competitive baselines, including state-of-the-art RL methods as well as gradient-based and sampling-based trajectory optimisation approaches.

Jun Yamada, Shaohong Zhong, Jack Collins, Ingmar Posner, (2025). D-Cubed: Latent Diffusion Trajectory Optimisation for Dexterous Deformable Manipulation, *Conference on Robot Learning (CoRL)*

D-Cubed: Latent Diffusion Trajectory Optimisation for Dexterous Deformable Manipulation

Jun Yamada, Shaohong Zhong, Jack Collins, Ingmar Posner
Applied AI Lab
Oxford Robotics Institute
University of Oxford

Abstract: Mastering deformable object manipulation often necessitates the use of anthropomorphic, high-degree-of-freedom robot hands capable of precise, contact-rich control. However, current trajectory optimisation methods often struggle in these settings due to the large search space and the sparse task information available from shape-matching cost functions, particularly when contact is absent. In this work, we propose *D-Cubed*, a novel trajectory optimisation method using a latent diffusion model (LDM) trained from a task-agnostic play dataset to solve dexterous deformable object manipulation tasks. *D-Cubed* learns a skill-latent space that encodes short-horizon actions from a play dataset using a VAE and trains a LDM to compose the skill latents into a skill trajectory, representing a long-horizon action trajectory. To optimise a trajectory for a target task, we introduce a novel gradient-free guided sampling method that employs the Cross-Entropy method within the reverse diffusion process. In particular, *D-Cubed* samples a small number of noisy skill trajectories using the LDM for exploration and evaluates the trajectories in simulation. Then *D-Cubed* selects the trajectory with the lowest cost for the subsequent reverse process. This effectively explores promising solution areas and optimises the sampled trajectories towards a target task throughout the reverse diffusion process. Through empirical evaluation on a published benchmark of dexterous deformable object manipulation tasks, we demonstrate that *D-Cubed* outperforms traditional trajectory optimisation and competitive baseline approaches by a significant margin.

Keywords: Trajectory Optimisation, Dexterous Deformable Object Manipulation, Latent Diffusion Model, Gradient-Free Guidance

1 Introduction

The realm of dexterous robot hand manipulation has made remarkable progress in recent years, in part due to advances in learning-based methods [1, 2, 3]. However, past research has focused predominantly on tasks that involve rigid objects [4, 5, 6, 7]. On the other hand, real-world manipulation tasks often present scenarios in which robots need to manipulate deformable objects, such as folding a piece of clothing [8], manipulating soft tissues [9] or shaping dough [10, 11].

One common approach to generating actions for a dexterous robot hand is trajectory optimisation, which optimises an action sequence by minimising a task-informed cost function. However, the application of trajectory optimisation is predominantly limited to rigid object manipulation [7] or relatively simple deformable object manipulation tasks with short horizons [12]. The primary challenges of optimising a trajectory for complex tasks such as those seen in dexterous deformable object manipulation stem from 1) the large search space due to the complexity of the task including the infinite dimensionality of deformable objects and high degrees of freedom (DoF) of the robot hand; 2) the large number of contacts associated with handling the objects; and 3) the limited task infor-

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

mation that the cost function typically provides [13]. Commonly, the cost function to be optimised is defined as the distance between a target shape and the final shape of a deformable object after manipulation [14]. In this scenario, no task-relevant signal is available when no contact is made between the robot and the manipulated object, inhibiting the optimisation of a feasible trajectory.

In this work, we propose *D-Cubed*, Latent **D**iffusion for Trajectory Optimisation in **D**exterous **D**eformable Manipulation (see Fig 1). *D-Cubed* is a novel trajectory optimisation approach that leverages a latent diffusion model (LDM)[15] trained on a task-agnostic play dataset collected using a human hand. This dataset captures diverse representative robot hand motions, such as closing and opening the hand and moving individual fingers without object interaction, enabling reuse across a wide range of tasks. First, *D-Cubed* learns a skill-latent space that encodes short-horizon action sequences from the play dataset using a variational autoencoder (VAE). An LDM is then trained to compose these skill-latent representations into a skill trajectory, replicating motions of the robot hand found in the dataset. By leveraging the task-agnostic play dataset, the LDM, capable of generating diverse trajectories of meaningful robot hand motions, is applicable across diverse manipulation tasks (see Fig. 1). To find a performant action trajectory for a target task defined by a target object shape, given a shape-matching cost, we propose a novel gradient-free guided sampling method that employs a variation of the Cross-Entropy Method (CEM) [16, 17] within the reverse diffusion process. For each denoising step, the LDM generates a small number of noisy skill-latent trajectories to explore the solution space. Since the skill latent space is trained to represent smooth low-level action sequences, each skill in these noisy skill trajectories produces meaningful and consistent action trajectories that facilitate efficient exploration. These skill-latent trajectories are evaluated in a simulator using the shape-matching cost function, and the trajectory with the lowest task cost among the sampled trajectories is selected for further denoising in the reverse diffusion process. With each denoising step guided by the CEM, the noise in the chosen performant trajectory is gradually removed, refining it towards solutions with lower costs.

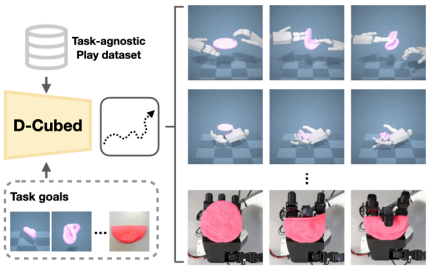


Figure 1: *D-Cubed* leverages a LDM trained from a task-agnostic play dataset to generate action trajectories for long-horizon dexterous deformable object manipulation tasks.

In summary, our contributions are three-fold: (1) we propose *D-Cubed*, a trajectory optimisation method using an LDM to solve challenging long-horizon dexterous manipulation tasks; (2) we introduce a novel gradient-free guided sampling method that employs the CEM within the reverse diffusion process to optimise a trajectory for a target task; and (3) we empirically demonstrate that *D-Cubed* significantly outperforms competitive baselines, including traditional trajectory optimisation methods such as gradient-based and sampling-based approaches.

2 Related Works

Several prior works note the importance of deformable object manipulation and introduce benchmark tasks for evaluating competing methodologies [12, 18, 19, 20, 14]. While most benchmarks focus on deformable object manipulation tasks with point-mass agents or parallel grippers that are incapable of dexterous manipulation, [14] proposes a suite of deformable object manipulation tasks with dexterous robot hands [21] built upon a differentiable physics engine.

Trajectory optimisation, including gradient-based and sampling-based, is a common approach to solving dexterous robot hand manipulation tasks by assuming access to an accurate dynamics model or simplified object geometries (e.g. [4, 5, 6, 7]). Gradient-based approaches directly optimise a task-informed cost function through a learned dynamics model [3, 22, 23] or differentiable simulator [12, 14, 24, 25] to find a performant action sequence. However, their application is limited to relatively simple, short horizon tasks [12] due to convergence to locally optimal solutions caused by a lack of

global task information, exacerbated by nonlinear contacts [14, 26]. Sampling-based methods such as CEM [16, 17] and MPPI [27] offer a simple approach by sampling actions for exploration, with MPPI applied to rigid object manipulation [7]. However, such sampling-based methods tend to be computationally expensive for large solution spaces, requiring many trajectory samples. Although previous work trains policies from expert demonstrations [28] or combine expert demonstrations with trajectory optimisation [14] to alleviate such issues, collecting expert demonstrations for each new task is considered expensive. Instead, in this work, a single task-agnostic play dataset containing representative hand movements is collected to form a structured skill-latent space that is used across a diverse range of tasks.

Diffusion models, a class of generative models, formulate data generation as an iterative denoising process [29, 15, 30]. Classifier guidance [31] is a common technique for using gradients to guide the sampling process of unconditional diffusion models to generate a desired sample, including a trajectory for a target task [32, 33, 34, 35]. However, classifier guidance struggles to guide sampling when gradients are inaccurate [36], such as when gradients are obtained from differentiable physics simulators [13]. In contrast, this work proposes gradient-free guidance that employs a variation of CEM to the reverse diffusion process for trajectory optimisation.

3 Preliminaries

Denoising Diffusion Probabilistic Models (DDPMs) DDPMs [15, 37] are a class of generative models that are trained by denoising a sequence of noise-corrupted inputs. For each training datum $\mathbf{x}_0 \sim q_{data}(\mathbf{x})$, the forward diffusion process constructs a Markov chain $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N$ such that $q(\mathbf{x}_i|\mathbf{x}_{i-1}) = \mathcal{N}(\mathbf{x}_i; \sqrt{1-\beta_i}\mathbf{x}_{i-1}, \beta_i\mathbf{I})$ where β_i denotes a positive noise scale and subscript index i refers to the time step of the diffusion process. Then, the reverse process, which aims to remove noise from the noisy sample x_i, is defined as $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:N}) = p(\mathbf{x}_N)\Pi_{i=1}^{N} p_{\boldsymbol{\theta}}(\mathbf{x}_{i-1}|\mathbf{x}_i)$, where $p(\mathbf{x}_N) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The conditional distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_{i-1}|\mathbf{x}_i)$ is commonly modelled as a Gaussian distribution with mean $\mu_{\boldsymbol{\theta}}(\mathbf{x}_i, i)$ and covariance $\Sigma_\theta(\mathbf{x}_i, i)$:

$$\mu_\theta(\mathbf{x}_i, i) = \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_t}{1-\bar{\alpha}_i}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\mathbf{x}_i, \Sigma_{\boldsymbol{\theta}}(\mathbf{x}_i, i) = \sigma_i^2\mathbf{I} = \tilde{\beta}_i = \frac{1-\overline{\alpha}_i - 1}{1-\overline{\alpha}_i}\beta_i, \quad (1)$$

where $\alpha_i := 1 - \beta_i$ and $\overline{\alpha} := \Pi_{j=1}^{i}\alpha_j$.

Instead of predicting the noise, ϵ_i [15], added to the data, we train a diffusion model $G_\theta(x_i, i)$ to directly predict the clean datapoint, x_0, to simplify the objective [38]:

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), i \sim [1,N]}\left[||\mathbf{x}_0 - G(\mathbf{x}_i, i)||_2^2\right]. \quad (2)$$

Cross-Entropy Method (CEM) The CEM finds solutions to complex problems by iteratively refining a probability distribution, often modelled by a Gaussian distribution, to focus on promising solution areas. The CEM samples a population of solutions from a given distribution, evaluates them using a predefined cost function, and selects the top performing solutions to update the distribution.

4 Latent Diffusion Trajectory Optimisation

Given a representative dynamics model (e.g. a simulator), *D-Cubed* aims to find an action trajectory $\{\mathbf{a}^0, \mathbf{a}^1, \ldots, \mathbf{a}^T\}$, over time horizon T, that enables a dexterous robot hand to manipulate deformable objects to match a pre-defined goal shape. *D-Cubed* consists of three components: (1) a variational autoencoder (VAE) that learns a skill-latent, $\mathbf{z} \in \mathcal{Z}$, by encoding short-horizon action trajectories; (2) a latent diffusion model (LDM) that generates sequences of skill-latents that represent entire trajectories for exploration in the state space; and (3) trajectory optimisation using CEM within the reverse diffusion process for a target task. *D-Cubed* relies on a task-agnostic play dataset of action trajectories that cover a wide range of meaningful robot hand motions to learn a skill-latent space. This section describes the data collection process in Section 4.1, the LDM in Section 4.2, and the sampling method in Section 4.3. An overview of the method can be seen in Fig. 2.

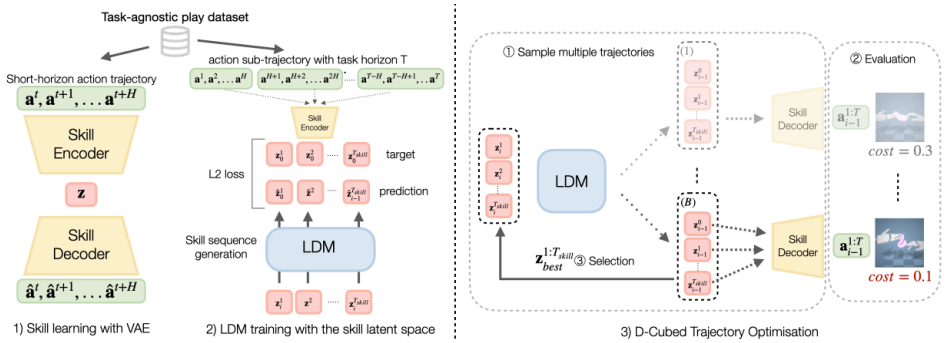


Figure 2: ***D-Cubed* overview.** (1) A VAE learns a skill latent representation $\mathbf{z}$ by reconstructing a short-horizon action sequence $\mathbf{a}^{t:t+H}$ randomly sampled from the task-agnostic play dataset. (2) A LDM learns to compose skills into a skill trajectory, representing a long-horizon action trajectory sampled from the dataset. (3) During trajectory optimisation, the LDM generates B skill trajectories $\{\mathbf{z}_i^{1:T_{skill}}\}^{|B|}$, where $T_{skill} = \frac{T}{H}$ is the length of skill trajectories. These trajectories are evaluated in a simulator, and the best sequence $\mathbf{z}_{best}^{1:T_{skill}}$ that minimises the cost is selected for the subsequent reverse process.

4.1 Data Collection

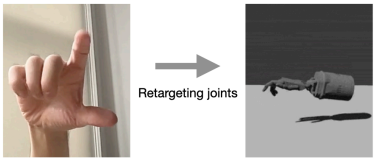


Figure 3: Data collection pipeline. Human hand joints are retargeted to robot hand joints to collect task-agnostic a play dataset designed to span the space of meaningful hand motions that form a skill latent space learnt by a VAE.

Collecting expert demonstrations for every new task is expensive due to the difficulty in manipulating deformable objects through teleoperation systems [14, 28]. To alleviate this issue, a single task-agnostic play dataset of robot hand trajectories $\mathcal{D}_{play}$ is collected per robot platform, without requiring interaction with deformable objects, allowing a human operator to readily collect the dataset without any training and enabling reuse across diverse tasks. This play dataset is designed to span the space of meaningful hand motions that can be performed by the given hardware and thus forming a skill latent space learnt by a VAE. This includes motions such as closing and opening the hand, moving individual fingers, as well as moving and flexing the wrist throughout the robot's workspace. The dataset is collected within 20 minutes by tracking the motion of a human hand and retargeting the human hand pose to a robot hand (see Fig. 3), similar to prior work [39]. This forms a dataset, denoted as D_{play}, comprising sequences of robot actions $\mathbf{a}_t$, where each action represents a relative change in the joint angles. For further details, see Appendix B.

4.2 Latent Diffusion Model as Skill Sampler

Long-horizon dexterous deformable manipulation tasks induce a large solution space due to the task complexity caused by the numerous contacts with deformable objects and the high DoF of a robot hand. As such, sampling low-level actions to find performant solutions is often infeasible because the sampled action trajectory is unlikely to correspond with meaningful robot hand motions that effectively explore the solution space. Instead, *D-Cubed* learns a skill latent space [40] that encodes a short-horizon action trajectory from the play dataset, which plays a significant role in efficiently exploring the search space of tasks. Specifically, a VAE, consisting of an encoder $q_\psi^{enc}(\mathbf{z}|\mathbf{a}^{t:t+H})$ and a decoder $p_\psi^{dec}(\mathbf{a}^{t:t+H}|\mathbf{z})$, is trained to reconstruct short-horizon action trajectories $\mathbf{a}^{t:t+H}$ randomly sampled from the play dataset $\mathcal{D}_{play}$ to learn the skill $\mathbf{z} \in \mathcal{Z}$, by optimising the ELBO objective:

$$\mathcal{L}^{\text{ELBO}} = \mathbb{E}_{\mathbf{z}\sim q_\psi(\mathbf{z}|\mathbf{a}^{t:t+H})} \log p_\psi^{dec}(\mathbf{a}^{t:t+H} \mid \mathbf{z}) - D_{\text{KL}}\left[q_\psi(\mathbf{z} \mid \mathbf{a}^{t:t+H}) \| p(\mathbf{z})\right]$$

where $p(\mathbf{z})$ is a Gaussian prior over the latent representation and H is the length of the short-horizon action sequence ($H = 10$). See Appendix C.1 for further details on the hyperparameters of the VAE.

Since the skill latent representation only encodes short-horizon actions of the robot hand, composing multiple skill latent representations into a long-horizon action trajectory is necessary for efficient exploration (e.g. to make meaningful contacts with an object). To achieve this, an LDM is trained to compose a sequence of skill-latent representations $\mathbf{z}^{1:T_{skill}}$, where $T_{skill} = \frac{T}{H}$ is the length of the skill trajectory, which reconstruct robot hand trajectories from the play dataset. The LDM is trained to optimise the following objective:

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{\mathbf{z}^{1:T_{skill}} \sim \mathcal{D}_{play}, i \sim [1,N]} \left[||\mathbf{z}_0^{1:T_{skill}} - G_\theta(\mathbf{z}_i^{1:T_{skill}}, i)||_2^2 \right]. \tag{3}$$

where N is the number of diffusion steps, $\mathbf{z}_0^{1:T_{skill}}$ is a clean skill-latent trajectory, and $\mathbf{z}_i^{1:T_{skill}}$ is a noisy skill-latet trajectory after i forward diffusion steps. An LDM is chosen as it is a generative model proven to be capable of representing a complex multimodal distribution, like that of the play dataset [41]. In this work, we employ a transformer model as the backbone of the noise prediction model $G_\theta(\cdot, i)$ (see Appendix C.2 for further details).

By leveraging a task-agnostic play dataset that captures meaningful hand motions, the trained LDM can generate sequences of robot hand actions that effectively explore the action space and are reusable across a diverse range of tasks. As a result, the LDM needs to be trained only once from the single play dataset and can be applied to all tasks without retraining.

4.3 Trajectory Optimisation using Gradient-Free Guided Sampling

Traditional trajectory optimisation often struggles to solve dexterous deformable object manipulation tasks due to the large search space. This is further amplified by the limited global task information available from a cost function. Using the capability of the LDM with skill-latent space, the LDM generates diverse skill trajectories that represent long-horizon action trajectories of meaningful robot hand motions, leading to effective exploration of the state space. However, to solve a target task, guidance is required to direct the diffusion sampling process to converge towards high-performing trajectories. While classifier guidance is a common technique for guiding the reverse process using gradients, inaccurate [36] or noisy gradients such as those obtained from differentiable physics simulators [13] are unable to successfully guide the reverse diffusion process (see Section 5.3 for experimental results). To avoid such issues, we propose gradient-free guided sampling that employs the CEM [42] within the reverse diffusion process to optimise a trajectory for a target task. The reverse process can be viewed as analogous to the CEM optimisation steps, as *D-Cubed* evaluates generated action trajectories in a simulator and updates the parameters of a Gaussian distribution based on the trajectory with the lowest cost for each diffusion step (see Fig. 2 and Algorithm 1).

In particular, for each reverse step, a small number of noisy skill trajectories $\mathbf{z}^{1:T_{skill}}$ are sampled from a Gaussian distribution with a mean $\boldsymbol{\mu}_i$ predicted by the LDM (Line 3, 9), where $T_{skill} = \frac{T}{H}$ represents the horizon length of the skill-latent representations. Crucially, during the early stages of the reverse process, *D-Cubed* focuses on exploring the search space. This is because the variance $\Sigma_{\boldsymbol{\theta}}$ of the Gaussian distribution, determined by the noise scheduler (see Equation 1), is large, thereby generating diverse trajectories. During later steps of the reverse process, *D-Cubed* attempts to refine the trajectories for a target task as a result of the small variance of the scheduled distribution. The generated skill trajectories are decoded using the VAE into low-level action trajectories that are then evaluated in the simulator to obtain their respective scores (Line 5). While the generated skill sequences are noisy regarding their composition at early reverse diffusion steps, each short-horizon action sequence decoded from the skill latent representations remains smooth, which effectively promotes meaningful trajectories from the search space for efficient exploration.

Similar to how the CEM updates a Gaussian distribution based on the top-performing samples for each optimisation step (see Section 3), *D-Cubed* also updates a Gaussian distribution to search for more promising solution areas by predicting the mean $\boldsymbol{\mu}_i$ using the LDM given the trajectory with the lowest cost $\mathbf{z}_{best}^{1:T_{skill}}$ as input (see Equation 4 and Line 6):

$$\mu_\theta(\mathbf{z}_{best}^{1:T_{skill}}, i) = \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_t}{1-\bar{\alpha}_i} G_\theta(\mathbf{z}_{best}^{1:T_{skill}}, i) + \frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i} \mathbf{z}_{best}^{1:T_{skill}} \tag{4}$$

Algorithm 1 *D-Cubed* Trajectory Optimisation

1: **Require**: denoising model, $G_{\boldsymbol{\theta}}$; target state of deformable objects, $\mathbf{s}_{\text{target}}$, $T_{skill} = \frac{T}{H}$
2: **Initialise:** $C_{\text{best}} = \infty$, $\boldsymbol{\mu}_{\text{best}} = \text{None}$
3: $\{\mathbf{z}_N^1, \ldots, \mathbf{z}_N^{T_{skill}}\}^{|B|} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample B initial sequences of skill latent representations
4: **for** $i = N, N-1, \ldots, 1$ **do**
5: $\quad \mathbf{z}_{best}^{1:T_{skill}} \leftarrow \text{FINDBESTLATENTS}(\{\mathbf{z}_i^{1:T_s}\}^{|B|})$ ▷ Choose the best sequence of skill latents (Appendix E)
6: $\quad \boldsymbol{\mu}_i \leftarrow \boldsymbol{\mu}_\theta(\mathbf{z}_{best}^{1:T_{skill}})$ ▷ Predict a mean of a Gaussian distribution (see Eq. 4)
7: $\quad \text{cost} = \text{evaluate}(q_\theta^{dec}(\mathbf{a}^{1:T}|\boldsymbol{\mu}_i))$ ▷ Evaluate the predicted mean
8: $\quad$ **if** cost $<$ C_{best} **then** $\boldsymbol{\mu}_{\text{best}} \leftarrow \boldsymbol{\mu}_i$, $\text{C}_{best} \leftarrow$ cost
9: $\quad \{\mathbf{z}_{i-1}^1, \ldots, \mathbf{z}_{i-1}^{T_{skill}}\}^{|B|} \sim \mathcal{N}(\boldsymbol{\mu}_{best}, \sigma_{i-1}^2 \mathbf{I})$ ▷ Sample a batch B of sequences of skill latents
10: **return** $p_\psi^{dec}(\mathbf{a}^{1:T}|\mu_{\mathbf{best}})$

Although CEM normally requires several top-performing samples to compute the mean and variance of the Gaussian distribution, *D-Cubed* needs only a single top-performing sample because the mean at the next diffusion step is determined by the prediction from the LDM (see Line 6). Furthermore, in *D-Cubed*, the variance is updated based on a noise schedule (see Equation 1) for each reverse step. Intuitively, by choosing the lowest-cost trajectory for each diffusion step, the LDM removes noise from performant noisy trajectories for the subsequent reverse process. This leads to further refinement of the trajectory and minimisation of the cost function by exploring more promising solution areas.

Lastly, the mean $\boldsymbol{\mu}_i$ of the Gaussian distribution predicted by the LDM does not necessarily compose a better distribution for the following reverse steps because the LDM may make a poor prediction, leading to trajectory samples that have a higher cost. To address this issue, inspired by a variant of CEM [43, 44], the mean of the distribution is updated only when the current predicted mean $\boldsymbol{\mu}_i$ has a lower cost than the previous best mean $\boldsymbol{\mu}_{\text{best}}$ (see Line 8). This optimisation method, when paired with a LDM is empirically shown in Section 5.3 to generate performant trajectories.

5 Experiments

Our experimental evaluation is designed to answer the following questions: (1) How effective is *D-Cubed* in generating trajectories for dexterous deformable object manipulation? (2) How does our method compare to competitive baselines including traditional trajectory optimisation approaches and other methods that do not require expert demonstrations? (3) How important are the design decisions of *D-Cubed* in generating high-performance trajectories? In addition, we conduct real-world experiments to qualitatively assess whether the optimised action trajectories are executable on real hardware (see Appendix A.2). For further experimental details and resuls, see Appendix 4.3.

5.1 Experimental Setup

Simulated Environments: We evaluate *D-Cubed* on a publicly available benchmark that consists of a suite of six challenging dexterous deformable object manipulation tasks introduced in prior work [14]. The benchmark consists of three single-hand tasks (*Folding*, *Flip*, *Wrap*) and three dual-hand tasks (*Dumpling*, *Bun*, *Rope*). For dual-arm tasks, *D-Cubed* uses the same trained LDM to independently generate action sequences for each arm, producing two coordinated trajectories for each hand simultaneously. The cost function dictated by the benchmark is the Sinkhorn Divergence [45] which measures the difference between the manipulated and the target object shape using point clouds. Our experimental setup closely follows that of prior work [14] in that we evaluate *D-Cubed* and competitive baselines on tasks intended to form 5 different target shapes for each task. For more details on the tasks, see Appendix F and the prior work [14].

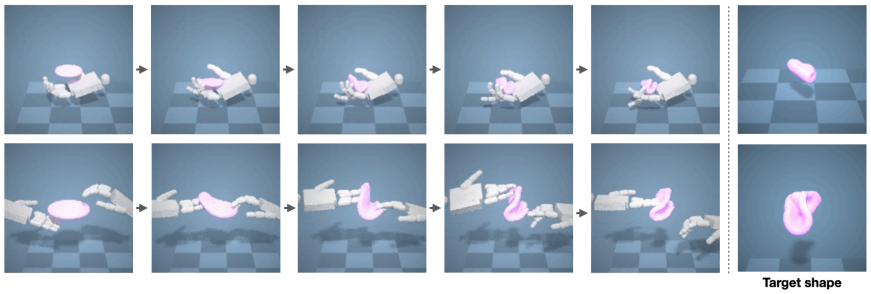


Figure 4: **Qualitative results of *D-Cubed*.** (Top) *Flip* and (Bottom) *Dumpling* task. In *Flip* task, the hand, using primarily the wrist and finger DoFs, is able to fold the plasticine into a configuration that is representative of the goal state. In *Dumpling* task, using two hands to deform the stationary plasticine, *D-Cubed*is able to manipulate the plasticine close to the target shape.

Env	Folding	Rope	Bun	Dumpling	Wrap	Flip
Grad TrajOpt	0.032 ± 0.061	0.079 ± 0.026	0.000 ± 0.000	0.032 ± 0.061	0.079 ± 0.026	0.000 ± 0.000
MPPI	0.002 ± 0.005	0.000 ± 0.000	0.000 ± 0.000	0.021 ± 0.042	0.000 ± 0.000	0.000 ± 0.000
Skill-based MPPI	0.020 ± 0.002	0.000 ± 0.000	0.048 ± 0.012	0.052 ± 0.034	0.000 ± 0.000	0.409 ± 0.001
PPO	0.361 ± 0.173	0.460 ± 0.257	0.069 ± 0.117	0.000 ± 0.000	0.000 ± 0.000	0.223 ± 0.328
LDM w/ Gradient guidance	0.050 ± 0.038	0.001 ± 0.001	0.019 ± 0.018	0.009 ± 0.014	0.016 ± 0.016	0.448 ± 0.080
Diffusion-ES	0.403 ± 0.227	0.192 ± 0.059	0.273 ± 0.092	0.179 ± 0.057	0.305 ± 0.007	0.678 ± 0.032
D-Cubed	$\mathbf{0.871 \pm 0.021}$	$\mathbf{0.741 \pm 0.031}$	$\mathbf{0.704 \pm 0.012}$	$\mathbf{0.699 \pm 0.037}$	$\mathbf{0.512 \pm 0.032}$	$\mathbf{0.909 \pm 0.025}$

Table 1: The averaged normalised improved EMD and standard deviation over 3 seeds are reported for each method. The scores for *Grad TrajOpt* and *PPO* are taken from previous work [14].

Evaluation Metric: Following [14], we report the normalised improvement in Earth-Mover distance (EMD) approximated by the Sinkhorn Divergence, calculated as $d(t) = \frac{d_0 - d_t}{d_0}$ where d_0 and d_t are the initial and current EMD values. When the normalised improvement is 1, the deformable object perfectly matches the target shape. A negative score from a large discrepancy is set to 0.

5.2 Baselines

We compare *D-Cubed* with the following state-of-the-art and competitive baselines that represent competing approaches capable of generating trajectories (for further details, see Appendix D):

- **Grad TrajOpt**: A gradient-based trajectory optimisation [12] that utilises the first-order gradients available from the benchmark simulator.
- **MPPI**: A sampling-based trajectory optimisation method [27] that samples a batch of short-horizon trajectories from a Gaussian distribution and updates the parameters of the distribution based on the top-performing trajectories.
- **Skill-based MPPI**: *Skill-based MPPI* is similar to *MPPI*, but operates in the skill-latent space. In contrast to *D-Cubed*, which uses an LDM to sample meaningful skill compositions, *Skill-based MPPI* must optimise such meaningful compositions by sampling diverse trajectories.
- **PPO**: Proximal Policy Optimisation (PPO) [46] generates closed-loop action sequences from point cloud inputs as an alternative to trajectory optimisation.
- **LDM w/ Gradient Guidance**: Using the learnt LDM to optimise a trajectory through the reverse process with gradient guidance using gradients from the simulator.
- **Diffusion-ES**: A concurrent method [47] that also proposes a gradient-free sampling method based on evolutionary search with a truncated diffusion process.

5.3 Trajectory Optimisation Results

***D-Cubed* is the highest performing method across all tasks.** Table 1 shows the normalised improvement in EMD for each task. *D-Cubed* outperforms the baseline methods by a significant margin in all tasks. These results indicate that *D-Cubed* effectively combines learnt skills to explore

the solution space using the LDM and exploits diverse sampled trajectories through gradient-free guided sampling to minimise a task-informed cost (see Appendix 4.3).

Traditional optimisation methods and RL baselines struggle due to poor exploration. *Grad TrajOpt* rarely obtains useful gradients from the shape-matching cost and struggles to find performant trajectories. *MPPI* and *Skill-based MPPI* focus on local short-horizon optimisation, missing better long-horizon solutions. In contrast, *D-Cubed* optimises the entire trajectory globally, enabling the dexterous hand to discover high-performing trajectories. This highlights the advantage of using an LDM trained to compose meaningful skill sequences, effectively narrowing the search to promising trajectories. Similarly, *PPO* underperforms relative to *D-Cubed* due to RL's difficulty in exploring high-dimensional state spaces, and must also be retrained for each task.

Gradient-guided diffusion and Diffusion-ES suffer from noisy or limited guidance. *LDM w/ Gradient guidance* performs poorly in all tasks, primarily due to noisy gradients from the simulator [13] and the lack of informative task signals, especially in tasks requiring extensive search or involving no object contact. *Diffusion-ES*, while initially generating clean trajectories and perturbing them through short diffusion steps, also struggles to explore the solution space sufficiently. The small perturbations limit its ability to recover when initial trajectories miss object contact.

Skill-level diffusion in *D-Cubed* enables efficient exploration. In contrast, *D-Cubed* explores the solution space more effectively by generating noisy skill trajectories at the beginning of the reverse diffusion process. This higher-level representation allows for broader exploration in the state space compared to low-level action trajectories generated by Diffusion-ES, leading to more efficient and successful planning outcomes.

5.4 Ablation Studies

Number of trajectories sampled during the reverse diffusion process. Fig. 5 shows the mean and Interquartile Mean (IQM) with 95% confidence intervals (CIs) [48] of normalised improvement in EMD averaged across all six tasks with different numbers of trajectories sampled during the reverse diffusion process (Line 9 in Algorithm 1). Fig. 5 indicates that sampling more trajectories during the reverse process significantly improves performance because *D-Cubed* can more effectively search the solution space.

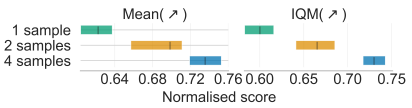


Figure 5: Ablation of the number of trajectories sampled during the reverse diffusion process (line 9 in Algorithm 1).

Efficacy of skill latent representations. Fig. 6 shows the performance of *D-Cubed* with and without the use of a skill latent across the six tasks. It is evident in simpler tasks, such as *Flip* and *Folding*, that the use of a skill-latent space does not significantly improve results. However, the performance of *D-Cubed* on *Rope* and *Wrap*, considerably harder tasks, is substantially better when using a skill-latent space. We reason that this is because *D-Cubed* with a skill-latent space can effectively search the solution space and tackle hard exploration problems.

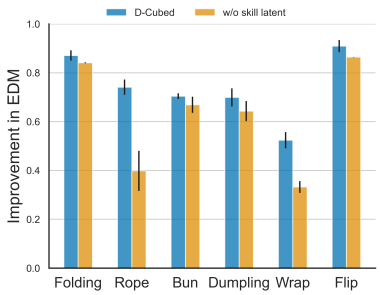


Figure 6: Comparison of *D-Cubed* w/ and w/o skill latent representations.

6 Conclusion

We present *D-Cubed*, a new trajectory optimisation method to solve long-horizon dexterous deformable object manipulation tasks using a latent diffusion model trained from a task-agnostic play dataset. *D-Cubed* leverages a novel gradient-free guided sampling method that adapts the CEM within the reverse diffusion process. The experimental results show that *D-Cubed* outperforms the traditional and competitive trajectory optimisation baselines by a significant margin, showing great promise for other challenging trajectory optimisation tasks.

6.1 Limitations

Experimental studies show that *D-Cubed* can generate a performant action sequence for dexterous deformable object manipulation tasks. However, *D-Cubed* cannot find realistic trajectories for all tasks because some benchmark tasks permit non-physical behaviours, such as table penetration and object floating. As a result, we are unable to conduct real-world experiments for every task, although we successfully demonstrate transfer for Flip (see Appendix A.2). Another limitation is the time required to generate a desired trajectory, which heavily depends on simulation evaluation time. However, using a faster, parallel simulator [49, 50] would greatly improve the speed of optimising a trajectory. Finally, *D-Cubed* is open-loop, making it unable to accommodate for discrepancies observed when executing the trajectory. In the future, we aim to close the loop, potentially by distilling the trajectories into a policy.

References

[1] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[2] T. Chen, J. Xu, and P. Agrawal. A system for general in-hand object re-orientation. In A. Faust, D. Hsu, and G. Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 297–307. PMLR, 08–11 Nov 2022.

[3] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020.

[4] I. Mordatch, Z. Popović, and E. Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, page 137–144, 2012. ISBN 9783905674378.

[5] Y. Bai and C. K. Liu. Dexterous manipulation using both palm and fingers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1560–1565, 2014. doi:10.1109/ICRA.2014.6907059.

[6] B. Sundaralingam and T. Hermans. Relaxed-rigidity constraints: kinematic trajectory optimization and collision avoidance for in-grasp manipulation. *Autonomous Robots*, 43(2):469–483, Feb 2019. doi:10.1007/s10514-018-9772-z.

[7] H. J. Charlesworth and G. Montana. Solving challenging dexterous manipulation tasks with trajectory optimisation and reinforcement learning. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1496–1506. PMLR, 18–24 Jul 2021.

[8] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara. Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation. *Robotics and Autonomous Systems*, 112, 11 2018. doi:10.1016/j.robot.2018.11.004.

[9] A. Pore, E. Tagliabue, M. Piccinelli, D. Dall'Alba, A. Casals, and P. Fiorini. Learning from demonstrations for autonomous soft-tissue retraction. In *2021 International Symposium on Medical Robotics (ISMR)*, pages 1–7. IEEE, 2021.

[10] H. Shi, H. Xu, Z. Huang, Y. Li, and J. Wu. RoboCraft: Learning to See, Simulate, and Shape Elasto-Plastic Objects with Graph Networks. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022. doi:10.15607/RSS.2022.XVIII.008.

[11] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. In J. Tan, M. Toussaint, and K. Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 642–660. PMLR, 06–09 Nov 2023.

[12] Z. Huang, Y. Hu, T. Du, S. Zhou, H. Su, J. B. Tenenbaum, and C. Gan. Plasticinelab: A soft-body manipulation benchmark with differentiable physics. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xCcdBRQEDW.

[13] R. Antonova, J. Yang, K. M. Jatavallabhula, and J. Bohg. Rethinking optimization with differentiable simulation from a global perspective. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=Y_YUEEQMjQK.

[14] S. Li, Z. Huang, T. Chen, T. Du, H. Su, J. B. Tenenbaum, and C. Gan. Dexdeform: Dexterous deformable object manipulation with human demonstrations and differentiable physics. In *The Eleventh International Conference on Learning Representations*, 2023.

[15] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[16] R. Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997. ISSN 0377-2217. doi:https://doi.org/10.1016/S0377-2217(96)00385-2.

[17] M. Kobilarov. Cross-entropy motion planning. *The International Journal of Robotics Research*, 31(7):855–871, 2012. doi:10.1177/0278364912444543. URL https://doi.org/10.1177/0278364912444543.

[18] X. Lin, Y. Wang, J. Olkin, and D. Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, pages 432–448. PMLR, 2021.

[19] S. Chen, Y. Xu, C. Yu, L. Li, X. Ma, Z. Xu, and D. Hsu. Daxbench: Benchmarking deformable object manipulation with differentiable physics. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1NAzMofMnWl.

[20] D. Blanco-Mulero, O. Barbany, G. Alcan, A. Colomé, C. Torras, and V. Kyrki. Benchmarking the sim-to-real gap in cloth manipulation, 2024.

[21] ShadowRobot. Shadowrobot dexterous hand., 2015. URL https://www.shadowrobot.com/products/dexterous-hand/.

[22] V. Kumar, E. Todorov, and S. Levine. Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–383, 2016. doi:10.1109/ICRA.2016.7487156.

[23] J. Yamada, C.-M. Hung, J. Collins, I. Havoutis, and I. Posner. Leveraging scene embeddings for gradient-based motion planning in latent space. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023.

[24] Y. Qiao, J. Liang, V. Koltun, and M. Lin. Differentiable simulation of soft multi-body systems. *Advances in Neural Information Processing Systems*, 34:17123–17135, 2021.

[25] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand. Difftaichi: Differentiable programming for physical simulation. In *International Conference on Learning Representations*, 2019.

[26] X. Lin, Z. Huang, Y. Li, D. Held, J. B. Tenenbaum, and C. Gan. Diffskill: Skill abstraction from differentiable physics for deformable object manipulations with tools. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Kef8cKdHWpP.

[27] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou. Aggressive driving with model predictive path integral control. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1433–1440, 2016. doi:10.1109/ICRA.2016.7487277.

[28] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[29] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265. PMLR, 07–09 Jul 2015.

[30] 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*, 2020.

[31] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[32] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[33] M. Janner, Y. Du, J. Tenenbaum, and S. Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

[34] Z. Liang, Y. Mu, M. Ding, F. Ni, M. Tomizuka, and P. Luo. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. In *International Conference on Machine Learning*, pages 20725–20745. PMLR, 2023.

[35] M. Rigter, J. Yamada, and I. Posner. World models via policy-guided trajectory diffusion. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

[36] C.-H. Chao, W.-F. Sun, B.-W. Cheng, Y.-C. Lo, C.-C. Chang, Y.-L. Liu, Y.-L. Chang, C.-P. Chen, and C.-Y. Lee. Denoising likelihood score matching for conditional score-based data generation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LcF-EEt8cCC.

[37] 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.

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

[39] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020.

[40] K. Pertsch, Y. Lee, and J. Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on robot learning*, pages 188–204. PMLR, 2021.

[41] L. Chen, S. Bahl, and D. Pathak. Playfusion: Skill acquisition via diffusion from language-annotated play. In *Conference on Robot Learning*, pages 2012–2029. PMLR, 2023.

[42] R. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Method. Comput. Appl. Prob.*, 1(2):127–190, sep 1999. ISSN 1387-5841. doi:10.1023/A:1010091220143. URL https://doi.org/10.1023/A:1010091220143.

[43] C. Pinneri, S. Sawant, S. Blaes, J. Achterhold, J. Stueckler, M. Rolinek, and G. Martius. Sample-efficient cross-entropy method for real-time planning. In *Conference on Robot Learning*, pages 1049–1065. PMLR, 2021.

[44] I. Szita and A. Lorincz. Online variants of the cross-entropy method, 2008.

[45] T. Séjourné, J. Feydy, F.-X. Vialard, A. Trouvé, and G. Peyré. Sinkhorn divergences for unbalanced optimal transport. *arXiv preprint arXiv:1910.12958*, 2019.

[46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[47] B. Yang, H. Su, N. Gkanatsios, T.-W. Ke, A. Jain, J. Schneider, and K. Fragkiadaki. Diffusiones: Gradient-free planning with diffusion for autonomous driving and zero-shot instruction following. *arXiv preprint arXiv:2402.06559*, 2024.

[48] R. Agarwal, M. Schwarzer, P. S. Castro, A. Courville, and M. G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 2021.

[49] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State. Isaac gym: High performance GPU based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=fgFBtYgJQX_.

[50] G. Authors. Genesis: A universal and generative physics engine for robotics and beyond, December 2024. URL https://github.com/Genesis-Embodied-AI/Genesis.

[51] K. Shaw, A. Agarwal, and D. Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *Robotics: Science and Systems (RSS)*, 2023.

[52] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[53] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[54] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

[55] A. Karpathy. NanoGPT. https://github.com/karpathy/nanoGPT, 2022.

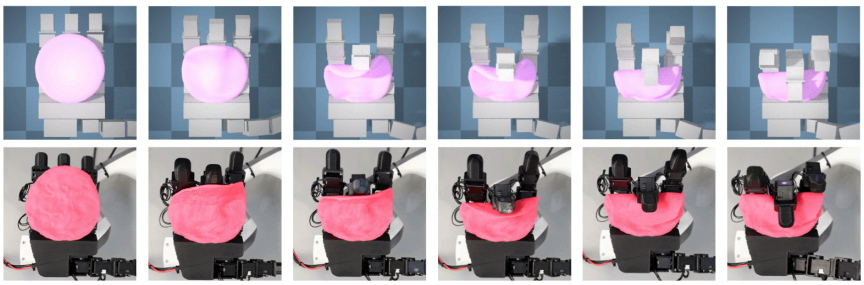


Figure 8: Qualitative results of *D-Cubed* using the LEAP hand in a real-world experiment. The LEAP hand effectively deforms the object, exhibiting similar deformation as observed in the simulation.

A Additional Analysis

A.1 Ablation of Additional Gradient Guidance

The performance disparity when *D-Cubed* also uses gradient guidance with the proposed gradient-free sampling is reported in Fig. 7. As gradient guidance does not demonstrate a statistically significant improvement in the score compared to *D-Cubed* without gradient guidance, the increased time required by the simulator to calculate gradients used for gradient guidance is not warranted. Additionally, this result adds further evidence to the premise that gradients from differentiable simulators are often sparse and uninformative [13].

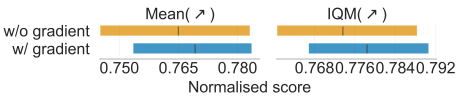


Figure 7: Comparison of performance with and without additional gradient guidance in our method. We report Mean and Interquartile Mean (IQM) of improvement in EDM averaged across all six tasks.

A.2 Qualitative Results in Real-World Environments

We qualitatively investigate whether an optimised trajectory from simulation can be transferred to real-world environments. Due to hardware limitations, we use a LEAP hand, a low-cost dexterous hand [51], instead of the Shadow hand [21]. Thus, the simulator is modified by replacing the Shadow hand with the LEAP hand. We evaluate the trajectory transfer on the *Flip* task, as this benchmark task makes the least number of simplifying assumptions compared to the real world. The other tasks permit object interpenetration with the table and unrealistic floating behaviour of the deformable object, rendering the evaluation impractical. In this experiment, given the known start state of the deformable object, we transfer the sequence of actions optimised in simulation to the real-world environment and control the hand in an open-loop manner. As shown in Fig. 8, the hand successfully *flips* the deformable object so that the object is folded within the hand in the real-world environment.

B Data collection Details

A task-agnostic play dataset of representative robot hand motions, including finger closing and opening and wrist movement, is collected. We use RGB data from a RealSense D435 camera to track human hand motion and re-target the human hand pose to a robot hand in the SAPIEN simulator [52], inspired by prior work [39]. We collect the play data for a duration of only 20 minutes, which corresponds to around $50K$ data points.

C Training Details

C.1 VAE

The VAE encoder and decoder both consist of a 4 layer LSTM [53] with 256 neurons per layer. In this work, we use a subsequence of actions with $H = 10$ to learn the skill-latent space. The VAE is trained using the Adam optimiser [54] with a learning rate of $1\mathrm{e}{-4}$.

C.2 Latent Diffusion Models

We use the transformer architecture used in NanoGPT [55]. We report further hyperparameter details of the transformer denoiser network and diffusion in Table 2 and Table 3.

Table 2: Transformer Denoiser Network Hyperparameters

Parameter	Value
Optimiser	Adam
Learning rate	1e-4
Minibatch size	256
Embedding dimension	312
Batch size	256
Number of layers	6
Self-attention heads	4

Table 3: Diffusion Hyperparameters

Parameter	Value
Number of diffusion timesteps	200
Noise schedule	cosine
Noise schedule parameters s	0.008

C.3 D-Cubed Details

We report hyperparameters of D-Cubed used during trajectory optimisation steps in Table 4.

Table 4: D-Cubed Optimisation Hyperparameters

Parameter	Value
Number of diffusion timesteps	200
Number of samples	5

D Baseline Method Details

As we report the scores for gradient-based trajectory optimisation (TrajOpt) and PPO from prior work [14], we refer the reader to the prior work for further details.

D.1 MPPI

MPPI baseline samples 30 trajectories with a horizon of 15 steps. These parameters are chosen because they result in an optimisation time similar to *D-Cubed*. We report the hyperparameters for

MPPI in Table 5 in detail. In this experiment, we employ the publicly available MPPI implementation[1].

Table 5: MPPI and Skill-based MPPI Hyperparameters

Parameter	Value
planning horizon	15
Number of samples	30
Temperature	1.0
Initial noise mean	0.0
Initial noise std	1.0

D.2 Skill-based MPPI

Skill-based MPPI baseline samples skill-latent representations for effective exploration of the state space. We use the same hyperparameters as those of MPPI, except that the action sampled from a Gaussian distribution is skill-latent representations instead of low-level actions.

D.3 LDM w/ Gradient Guidance

LDM w/ Gradient Guidance baseline leverages gradient guidance [31] to generate a desired trajectory. In particular, first-order gradients from the differentiable physics simulator are used to guide the reverse process of the latent diffusion model. In our experiments, we denoise a noisy trajectory without gradient guidance for the first half of the diffusion steps so that a relatively clean trajectory can be obtained. For the rest of the diffusion steps, the following gradient guidance is applied:

$$\nabla_{\mathbf{x}_i} \log p_{\alpha_i}(\mathbf{x}_i|y) = \nabla_{\mathbf{x}_i} \log p_{\alpha_i}(\mathbf{x}_i) + \gamma \nabla_{\mathbf{x}_i} \log p(y|\mathbf{x}_i). \tag{5}$$

where y is the cost of the trajectory, $\nabla_{\mathbf{x}_i} \log p(y|\mathbf{x}_i)$ corresponds to the first-order gradients obtained from differentiable physics simulators, and γ is the scale of the gradient guidance. In our experiment, we use $\gamma = 1\mathrm{e}{-4}$.

D.4 Diffusion-ES

Diffusion-ES, concurrent research [47], also optimises a trajectory using gradient-free guided sampling with a truncated diffusion process. While the prior work chooses the last trajectory of the optimisation process as output, we observe that it is often worse than the trajectories found in the middle of optimisation iterations. Thus, we report the score of the best trajectory found during the trajectory optimisation process. The hyperparameters used in Diffusion-ES is reported in Table 6. Following the original work [47], initially, the diffusion mutation steps start from 5 and the mutation step is linearly decayed to 1 over 200 search steps.

Table 6: Diffusion ES Hyperparameters

Parameter	Value
Mutation diffusion start steps	5
Mutation diffusion final steps	1
Population	5
Optimisation steps	200

[1] https://github.com/UM-ARM-Lab/pytorch_mppi/tree/master

E Gradient-Free Guided Sampling for Trajectory Optimisation

Algorithm 2 is a complete version of gradient-free guided sampling for trajectory optimisation in *D-Cubed*. To determine the best sequence of skill latent representations $\mathbf{z}_{best}^{1:T_{skill}}$, each skill sequence in a batch of B skill trajectories is evaluated in simulation, and the skill sequence that minimises a cost is selected as the best sequence (Line 5).

Algorithm 2 Gradient-Free Guided Sampling for Trajectory Optimisation in Reverse Diffusion Process

1: **Require**: denoising model, $G_{\boldsymbol{\theta}}$; target state of deformable objects, $\mathbf{s}_{\text{target}}$, $T_{skill} = \frac{T}{H}$
2: **Initialise:** $C_{\text{best}} = \infty$, $\boldsymbol{\mu}_{\text{best}} = \text{None}$
3: $\{\mathbf{z}_N^1, \ldots, \mathbf{z}_N^{T_{skill}}\}^{|B|} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample B initial sequences of latent representations
4: **for** $i = N, N-1, \ldots, 1$ **do**
5: $\quad \mathbf{z}_{best}^{1:T_{skill}} \leftarrow \text{FINDBESTLATENTS}(\{\mathbf{z}_i^{1:T_s}\}^{|B|})$ ▷ Choose the best sequence of skill latents
6: $\quad \boldsymbol{\mu}_i \leftarrow \boldsymbol{\mu}_\theta(\mathbf{z}_{best}^{1:T_{skill}})$ ▷ Predict a mean of a Gaussian distribution (see Eq. 4)
7: $\quad \text{cost} = \text{evaluate}(q_\theta^{dec}(\mathbf{a}^{1:T}|\boldsymbol{\mu}_i))$ ▷ Evaluate the predicted mean
8: $\quad$ **if** cost $<$ C_{best} **then**
9: $\quad\quad \boldsymbol{\mu}_{\text{best}} \leftarrow \boldsymbol{\mu}_i$, $\text{C}_{best} \leftarrow$ cost
10: $\quad \{\mathbf{z}_{i-1}^1, \ldots, \mathbf{z}_{i-1}^{T_{skill}}\}^{|B|} \sim \mathcal{N}(\boldsymbol{\mu}_{best}, \sigma_{i-1}^2 \mathbf{I})$ ▷ Sample a batch B of sequences of skill latents
11: **return** $p_\psi^{dec}(\mathbf{a}^{1:T}|\mu_{\mathbf{best}})$
12: **function** FINDBESTLATENTS($\{\mathbf{z}^1, \ldots, \mathbf{z}^{T_{skill}}\}^{|B|}$)
13: $\quad \text{cost}_{\text{best}}, \mathbf{z}_{\text{best}} = \infty, \text{None}$
14: $\quad$ **for** $\mathbf{z}^{1:T_{skill}} = \{\mathbf{z}^1, \ldots, \mathbf{z}^{T_{skill}}\}^{|B|}$ **do** ▷ Evaluate each sequence of latent representations in the batch
15: $\quad\quad \text{cost}_j = \text{evaluate}(p_\psi^{dec}(\mathbf{a}^{1:T}|\mathbf{z}^{1:T_{skill}}))$
16: $\quad\quad$ **if** cost $<$ cost_{best} **then**
17: $\quad\quad\quad \text{cost}_{\text{best}} \leftarrow$ cost, $\mathbf{z}_{\text{best}} \leftarrow \mathbf{z}^{1:T_{skill}}$
$\quad$ **return** $\mathbf{z}_{\text{best}}$

F Task Details

F.1 Cost Function

The cost function used for trajectory optimisation is defined by Sinkhorn Divegence. Following the prior work [14], the *geomloss* library is used to define the cost function:

```
from geomloss import SamplesLoss
OT_LOSS = SamplesLoss(loss="sinkhorn", p=1, blur=0.0001)
```

F.2 Tasks

For single-hand task, such as *Folding* and *Wrap*, the action dimension is 26 (20 for actuators including finger joints and wrist, and 6 for the base). For in-hand manipulation tasks (*Flip*), a single hand with a fixed base is assumed, resulting in an action dimension of 20. In dual-hand environments, the action dimension is 52, allowing for a movable base for both hands. Since a VAE is trained to encode a single-arm action trajectory in the play dataset, an LDM generates a single-arm skill trajectory. Thus, to handle dual-hand tasks using *D-Cubed*, the LDM generates a trajectory for each arm. In the following, we describe the details of each task.

Folding: The initial position of the robot hand is above the dough, and the hand must fold the dough in four different directions: front, back, left, and right.

Wrap: The robot hand first picks up the plasticine ball and places it onto the dough shaped like a rope. Then, it pinches the side of the rope to wrap the ball inside it.

Flip: The robotic hand tosses the dough in the air to reshape and reposition it.

Bun: The two robotic hands deftly pinch and push the dough to form a bun-shaped object.

Rope: The right-hand grasps the rope on the right, lifts it, and places it above the left rope. Then, the left-hand bends the left rope.

Dumpling: To wrap a dumpling, the right hand first grasps the right side of the dough. While holding the dough with the right hand, the left hand lifts the left side of the dough. Finally, the two hands bring the two sides of the dough together and form it into a dumpling shape.

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).**

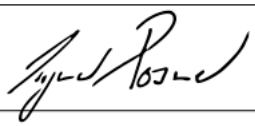
Title of Paper	D-Cubed: Latent Diffusion Trajectory Optimisation for Dexterous Deformable Manipulation
Publication Status	□Published ■ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Jun Yamada, Shaohong Zhong, Jack Collins, Ingmar Posner Accepted for publication at Conference on Robot Learning (CoRL) 2025

Student Confirmation

Student Name:	Jun Yamada		
Contribution to the Paper	- Proposed the research idea - Created and developed methodologies - Ran all experiments - Created all figures - Paper writing		
Signature		Date	19/09/2025

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: Prof. Ingmar Posner			
Supervisor comments *Jun made a substantial contribution to the publication. The description above is accurate.*			
Signature		Date	22/09/2025

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

8.1 Limitations and Future Work

Experimental results demonstrate that *D-Cubed* is capable of generating high-performing action sequences for dexterous deformable object manipulation tasks. Nevertheless, *D-Cubed* does not consistently produce physically realistic trajectories for every benchmark task, as certain tasks allow physically infeasible behaviours, such as object penetration through tables and unnatural object floating, due to the low fidelity of the simulated environments. Consequently, real-world experiments are not feasible for all tasks. Thus, developing physically realistic benchmarks for dexterous deformable manipulation remains an important direction for future work.

Furthermore, *D-Cubed* formulates manipulation as a trajectory optimisation problem whose objective is defined exclusively by the Chamfer distance between the manipulated and target object point clouds. Consequently, the optimisation operates entirely in geometric space and does not explicitly model contact forces, tactile feedback, or contact mechanics. In addition, the generated action sequences are position-based, rather than force- or torque-controlled, and therefore do not directly regulate interaction forces during execution. Contact forces are treated as an implicit consequence of achieving correct geometric alignment under low-level position control, rather than as optimisation variables. While this is effective for tasks in which object shape similarity is the primary success criterion, it is not well-suited to precision-sensitive manipulation scenarios in which task success depends critically on controlled contact forces or sensitivity to applied force magnitudes. Extending the framework to incorporate force-aware objectives or hybrid force–position control would be necessary to address such tasks.

While *D-Cubed* qualitatively demonstrates effective sim-to-real transfer by replaying sequences of optimised actions, it only supports open-loop control and does not incorporate perception input, which significantly limits its applicability in real-world settings. To enable closed-loop control, *D-Cubed* can instead be utilised as a data generator in simulation, providing diverse optimised trajectories that serve as demonstrations for training closed-loop, vision-based policies. More importantly,

since *D-Cubed* is capable of generating solutions under domain randomisation, this approach holds promise for achieving more robust and effective sim-to-real transfer.

Another concern is the computational time required for trajectory optimisation in *D-Cubed*, primarily influenced by simulation evaluation time. In particular, *D-Cubed* (200 diffusion steps) takes around 4.9 seconds, excluding trajectory evaluation. With the inclusion of trajectory evaluation in DexDeform's differentiable physics simulator, the complete optimisation takes approximately 1.5 hours. Thus, *D-Cubed* can achieve order-of-magnitude speedups by using a faster non-differentiable simulator such as MuJoCo [197] and IsaacSim [107], or a learnt dynamics model. Lastly, while *D-Cubed* sometimes generates action trajectories with jerky motions, this issue can be mitigated by incorporating additional cost terms that explicitly penalise trajectory jerks.

9 Discussion and Future Work

9.1 Discussion

This thesis investigates the unification of planning and learning to improve the efficiency, flexibility, safety, and robustness of skill acquisition and execution in contact-rich manipulation tasks. In particular, it explores two complementary themes: (A) planning-guided efficient skill acquisition (Chapter 3, 4, and 5) and (B) improving model-based approaches through the use of generative models (Chapter 6, 7, and 8).

9.1.1 Planning-Guided Efficient Skill Acquisition

Chapter 3 presents a robotic system for sample-efficient and versatile skill acquisition in small-batch manufacturing settings, achieved by combining motion planning with a learnt RL policy for contact-rich insertion tasks in obstructed environments. In this system, motion planning first generates collision-free trajectories that guide the robot toward the target object, and the RL policy subsequently executes the contact-rich manipulation. The system also leverages object-centric generative models for one-shot target object identification, making it effective in small-batch settings. Additionally, a skill transition network facilitates smooth transitions between the terminal states of motion planning and the initiation set of the RL policy.

The system introduced in Chapter 3 has two major drawbacks: (1) online RL training risks damaging hardware due to frequent collisions, and (2) the learnt RL

policy can still collide with the environment and struggle in cluttered settings such as grasping an object on a shelf. To address these limitations, Chapter 4 extends the framework by incorporating a value function learnt from synthetic data into model predictive control (MPC) as a cost function, enabling safe and closed-loop visual grasping that is generalisable to novel objects in cluttered environments. *Grasp-MPC* learns a value function from synthetic grasp trajectories for diverse objects generated in simulation and transfers it directly to real-world environments. Moreover, it ensures safety by optimising action sequences using a combination of the learnt grasp value function and collision cost functions, thereby grasping objects safely even in challenging, cluttered environments with obstacles.

COMBO-Grasp, introduced in Chapter 5, builds upon the principles in Chapters 3 and 4, extending the unified planning and learning framework to bimanual occluded grasping tasks. *COMBO-Grasp* comprises two coordinated policies: a constraint policy, guided by a motion planner, stabilises the target object, while a grasping policy, trained with RL, handles complex non-prehensile manipulation for bimanual occluded grasping. Instead of using object-centric generative models, *COMBO-Grasp* trains a constraint policy from a dataset collected in a self-supervised manner, which predicts a stabilisation pose for one arm to stabilise the object while the other performs the grasp. During RL training of the grasping policy in simulation, the constraint policy predicts a stabilisation pose at the beginning of each episode, and the constraint arm is positioned accordingly. This setup enables the grasping policy to focus on learning effective grasp strategies with the support of the stabilising arm. Moreover, in contrast to the system in Chapter 3, which requires online interaction with the real-world environment, *COMBO-Grasp* trains state-based teacher policies in simulation and distils them into vision-based student policies, enabling effective sim-to-real transfer without requiring real-world data collection. This framework for bimanual coordination effectively accelerates RL training of the grasping policy in simulation and achieves robust transfer to real-world environments. The experiments demonstrate that the unified framework can solve more complex bimanual manipulation tasks while ensuring effective sim-to-real transfer.

Across these three chapters, we demonstrate the effectiveness of unifying planning and learning for efficient skill acquisition in contact-rich manipulation tasks, ranging from single-arm grasping and insertion in cluttered, unstructured environments to complex bimanual manipulation. In particular, planning serves to guide efficient and safe skill acquisition and execution by decomposing and structuring manipulation problems, allowing learning-based components to focus on the most challenging aspects of contact-rich interaction. Starting from the foundational framework in Chapter 3, which integrates motion planning with an RL policy, Chapter 4 enhances safety, generalisability, and adaptability by combining MPC with a value function trained from a large-scale synthetic dataset. This principle of unifying motion planning with learning-based approaches is further extended to bimanual, contact-rich manipulation settings in Chapter 5.

9.1.2 Improving Model-Based Approaches with Generative Models

Chapter 6 proposes gradient-based motion planning in a structured latent space trained using a VAE on readily available robot kinematics data instead of using costly expert demonstrations. *AMP-LS* can be seen as a latent transition model that facilitates planning by capturing feasible kinematic transitions between latent states, which are derived from readily available data of kinematically valid joint configurations. While it does not explicitly model physics-based dynamics, it serves a comparable role for kinematic planning. To navigate around obstacles with diverse and complex shapes, *AMP-LS* integrates a vision-based collision predictor trained on synthetic scenes and uses its gradients as constraints during planning.

In contrast to *AMP-LS*, which is limited to kinematic planning and thus cannot directly solve full manipulation tasks, Chapter 7 introduces a sample-efficient approach for learning dynamics models applicable to manipulation tasks by leveraging the capability of generative modelling. Specifically, *TWIST* addresses the challenge that training dynamics models directly on real-world data is often impractical due to limited data availability and the risk of unsafe interactions during online data collection. To overcome this, *TWIST* learns a transferable dynamics

model and associated policy entirely in simulation. A state-based teacher world model is first trained using low-dimensional state observations, which are readily accessible in simulation. The teacher model supervises a vision-based student model through distillation, leveraging paired low-dimensional states and domain-randomised image observations collected during teacher training to bridge the domain gap. Moreover, using the capabilities of generative models, *TWIST* further leverages imagined trajectories from both teacher and student models to enable more effective distillation. *TWIST* demonstrates that the resulting vision-based dynamics model with the associated policy can be effectively transferred to real-world environments in a zero-shot manner, supporting both policy execution and planning.

In contrast to Chapters 6 and 7, which focus on learning dynamics models, Chapter 8 introduces *D-Cubed*, which learns a generative action sampler and exploits it in combination with the CEM for dexterous deformable manipulation, under the assumption that the dynamics model is available. *D-Cubed* learns a sampling distribution using a latent diffusion model (LDM) for efficient exploration and proposes gradient-free guided sampling for effective exploitation. The LDM is trained on task-agnostic robot hand play data, which includes semantically meaningful motions such as opening, closing, and wrist movements. A VAE first learns skill latent representations by reconstructing short-horizon action sequences sampled from the play dataset. The LDM then composes these skill latents into coherent long-horizon trajectories. To optimise trajectories, *D-Cubed* integrates a variant of the CEM into the reverse diffusion process of the LDM, where the lowest-cost action sequence is selected at each denoising step and iteratively refined in subsequent steps. *D-Cubed* demonstrates that learning a sampling distribution using a generative model facilitates exploration for trajectory optimisation in contact-rich manipulation, enabling the discovery of high-performing trajectories using a readily collected, task-agnostic play dataset instead of costly, task-specific expert demonstrations or sample-inefficient policy learning.

Collectively, these three chapters demonstrate the effectiveness of utilising generative models trained on readily collected datasets to improve model-based decision-

making and overall task performance in contact-rich manipulation. Chapter 6 introduces gradient-based motion planning in a learnt structured latent space that performs gradient-based optimisation within a latent space learnt from kinematically feasible joint states. Rather than focusing on kinematic planning, Chapter 7 develops a sim-to-real transfer framework for model-based RL via teacher-student distillation in imagined trajectories. Finally, Chapter 8 complements these contributions by learning task-agnostic sampling distributions using an LDM to guide trajectory optimisation. Collectively, these contributions demonstrate that generative models can improve model-based approaches by learning dynamics models for planning and policy optimisation, and by acquiring task-agnostic sampling distributions that guide trajectory optimisation, enabling scalable and adaptable solutions for diverse and challenging manipulation tasks.

9.1.3 Connection Between Two Themes

Although the two themes are presented separately, their contributions are complementary and can be combined to further advance the unified planning–learning framework. For example, in *Theme A*, motion planning is extensively employed to guide skill acquisition and structure challenging manipulation tasks in unstructured environments, allowing the learning-based approach to focus on the more difficult aspects of the problem. However, heuristic geometric planning can be slow and lacks the ability to adapt to changes in the environment during execution. On the other hand, *AMP-LS* from *Theme B* offers a direct improvement in this regard, enabling faster, gradient-based planning in a learnt latent space with reactive collision avoidance. Replacing the heuristic motion planner in *Theme A*'s systems with AMP-LS could therefore yield both speed and adaptability gains.

Similarly, in Grasp-MPC (*Theme A*), MPC is combined with a learnt grasp value function to enable safe and generalisable vision-based closed-loop prehensile manipulation in cluttered environments. Because grasping is a prehensile manipulation task, an explicit dynamics model of the environment is not strictly necessary. However, extending such an approach to more complex non-prehensile tasks requires

a dynamics model that captures object dynamics and contact interactions to roll out and evaluate sampled action sequences. *TWIST* from *Theme B* directly addresses this need by learning transferable dynamics models from simulation to real-world settings, enabling safe and sample-efficient acquisition of a world model in tasks where handcrafted models are impractical or real-world data is not readily available.

Furthermore, finding high-performing action sequences in MPC becomes more challenging with the increasing complexity of manipulation tasks. In such cases, *D-Cubed* from *Theme B* offers a promising solution by providing task-agnostic, diffusion-based action samplers that efficiently guide trajectory optimisation toward high-quality solutions. Integrating such sampling strategies into *Grasp-MPC* in *Theme A* could improve search efficiency and solution quality in more challenging contact-rich manipulation tasks.

These examples illustrate that model-based approaches, enhanced by generative modelling in *Theme B*, can serve as drop-in replacements or improvements for components in *Theme A*'s frameworks, boosting planning speed, reactivity, and the ability to handle complex dynamics. Conversely, the planning-guided efficient skill acquisition in *Theme A* provides natural application domains for the models developed in *Theme B*, highlighting their utility in real-world and safety-critical manipulation scenarios.

9.2 Future Work

This thesis has demonstrated the effectiveness of unifying planning and learning for contact-rich manipulation, yet several promising research directions remain to further advance the field and expand the range of real-world applications.

Scaling to long-horizon manipulation tasks with task planning. The approaches presented in this thesis primarily focus on relatively short-horizon manipulation tasks such as insertion and grasping. A natural extension would be to scale these unified approaches to more complex, long-horizon manipulation tasks, such as assembly, that require sequential manipulation of multiple objects. Building

upon assembly benchmarks such as RAMP [68], a work which is not covered in this thesis, future work could investigate how the proposed planning-guided skill acquisition methods can be extended to handle multi-step assembly sequences.

A promising direction involves integrating high-level task planning with learnt manipulation policies developed in this thesis. This hierarchical approach would combine a symbolic task planner that can reason about object relationships and assembly sequences with the low-level learnt policies for contact-rich manipulation. Moreover, to enhance the capability of task planning, a dynamics model can be combined with task planning to estimate future outcomes and determine the next best high-level action, similar to prior work [198]. As presented in Chapter 7, such dynamics models can be learnt from simulated datasets, optionally in addition to a small number of real-world interaction data for finetuning.

Safe and robust approaches to non-prehensile manipulation tasks. *COMBO-Grasp* successfully addresses bimanual occluded grasping tasks, which require non-prehensile manipulation skills such as pushing and reorienting a target object. Nevertheless, the grasping policy trained using RL remains inherently unsafe when obstacles are present or in cluttered settings, for example when objects are located inside a shelf. *Grasp-MPC* can potentially alleviate this issue by incorporating a learnt value function as the cost for MPC, thereby improving safety and robustness while maintaining generalisable manipulation skills for novel objects. However, *Grasp-MPC* is primarily designed for prehensile grasping in cluttered environments and is not directly applicable to non-prehensile manipulation tasks. Such tasks, including deformable object manipulation, require explicit modelling of object and robot dynamics to roll out and evaluate sampled action sequences. *TWIST*, introduced in Chapter 7, can provide such a dynamics model by leveraging readily available synthetic data, thereby enabling *Grasp-MPC* to be applied beyond grasping tasks. Alternatively, another promising direction is to learn a value function conditioned on sequences of actions (i.e., action chunks) [189], which implicitly captures environment dynamics and reduces reliance on explicit rollouts. Together,

these approaches highlight potential pathways for extending safe and robust MPC beyond prehensile grasping to a wider range of non-prehensile manipulation tasks.

Learning accurate dynamics models efficiently for contact-rich manipulation. *TWIST* demonstrates that a dynamics model can be learnt from synthetic data and successfully deployed to real-world contact-rich manipulation tasks. While one of the tasks evaluated in *TWIST* involves non-prehensile manipulation, it remains relatively simple, focusing on a block-pushing scenario in a tabletop environment. To address more complex manipulation problems, it is essential to develop methods for learning accurate dynamics models efficiently from data, particularly for contact-rich interactions. Recent studies have leveraged diffusion models [69, 158] to learn more expressive and accurate dynamics models that can be utilised for planning or policy optimisation. Accordingly, using a diffusion model in *TWIST* to learn a dynamics model represents a promising direction for improving model accuracy. Furthermore, publicly available large-scale robotic datasets [199] enable the pre-training of dynamics models that capture generalisable priors for manipulation tasks, which can then be fine-tuned using a small number of real-world interaction episodes. Lastly, co-training [200], which jointly utilises simulated and real-world data, has also demonstrated promising results for manipulation tasks and could potentially be applied to improve the training of dynamics models.

Trajectory optimisation with learnt dynamics models. *D-Cubed*, presented in Chapter 8, assumes access to a dynamics model; however, such a model, particularly for contact-rich deformable object manipulation, is rarely available in real-world settings. Inspired by *TWIST* (Chapter 7), a world model could be trained entirely in simulation, for example, from point cloud observations, to effectively capture deformable object behaviour and then transferred to real-world environments. To further enhance modelling accuracy, this learnt world model could be fine-tuned using a small set of real-world interactions, enabling *D-Cubed* to identify high-performing trajectories that can be executed effectively in real-world manipulation tasks. Moreover, the optimised sequence of actions can be

utilised as a dataset to train a visuomotor control policy capable of performing the target task in a closed-loop manner.

Unifying planning and learning for foundation models. While this thesis primarily focuses on low-data regimes or leverages readily available data (e.g., from simulation), a recent trend in robot learning is the use of large-scale cross-embodiment demonstration datasets to train vision-language-action (VLA) models [27, 164, 165] for better adaptation and generalisation to novel scenes, objects, and embodiment. However, current VLAs remain difficult to deploy in complex environments with obstacles, unless task-specific demonstrations that include obstacle avoidance behaviour are provided for fine-tuning. To facilitate real-world deplomultiyment, it is more promising to unify planning and learning for safer and more reliable execution of skills. For instance, VLA models can be used to decompose a task into segments that are suitable for motion planning and those requiring fine-grained manipulation. This decomposition enables safe navigation around objects using motion planning, followed by precise manipulation using VLAs. Moreover, inspired by prior work on learning foundation reward models from large-scale robotic data [201], a foundation value function could be trained and integrated with MPC, analogous to *Grasp-MPC*. Lastly, VLA can also serve as an action sampler, which, when combined with a world model, enables the simulation of expected outcomes to select the most suitable next action for manipulation tasks.

10 Conclusions

This thesis presents a comprehensive study on unifying planning and learning for contact-rich manipulation, addressing key challenges that hinder robots from operating effectively in semi-structured and unstructured environments. Across six research works, ranging from planning-guided skill acquisition to the use of generative models for improving model-based approaches, the thesis demonstrates that combining model-based planning with data-driven learning trained on readily available data can effectively overcome the limitations of each when used in isolation.

The central theme of this work, that combining the complementary strengths of planning and learning enables more efficient, safe, and generalisable contact-rich robot manipulation, has been validated through extensive experimental evaluation across both simulated and real-world environments. Planning provides predictive foresight, constraint handling, safety assurance, and navigation to the target, whereas learning contributes by acquiring contact-rich manipulation skills, modelling complex dynamics, enabling adaptation and generalisation to novel objects, and providing priors such as sampling distributions. By unifying these paradigms, the frameworks developed in this thesis have demonstrated significant improvements in sample efficiency, safety, task performance, and adaptability compared to prior approaches.

10.1 Key Contributions and Impact

The first major contribution area, planning-guided efficient skill acquisition, has demonstrated how model-based planning, such as motion planning and trajectory optimisation, can effectively structure or assist learning problems. This approach accelerates skill acquisition and ensures safe execution for complex manipulation tasks in unstructured environments, even with the presence of obstacles. In particular, Chapter 3 presents a system for sample-efficient and versatile skill acquisition in small-batch manufacturing settings, capable of safely performing contact-rich insertion tasks across the robot's entire workspace in obstructed environments by unifying motion planning with an RL policy. This is achieved by leveraging an object-centric generative model that enables one-shot goal identification and facilitates rapid adaptation to novel target objects. Similar to Chapter 3, *Grasp-MPC* (Chapter 4) leverages motion planning to avoid obstacles and reach a predicted pre-grasp pose. In contrast to Chapter 3, which leverages RL for contact-rich manipulation, *Grasp-MPC* employs MPC with a value function learnt from diverse synthetic grasp trajectories as the cost function, enabling safe, robust, and generalisable closed-loop visual grasping in complex, unstructured environments. *COMBO-Grasp* extends this principle to bimanual occluded grasping by employing a motion planner to control one arm for object stabilisation, thereby enabling an RL policy to acquire grasping skills more efficiently through the use of the stabilising arm.

The second major contribution area, improving model-based approaches through generative models, demonstrates how learnt representations and generative sampling can enhance model-based decision-making, particularly by leveraging readily collected data such as synthetic datasets or task-agnostic play data. *AMP-LS* (Chapter 6) demonstrates gradient-based motion planning in a structured latent space learnt from readily available kinematically valid robot states. It generates collision-free trajectories in cluttered scenes with complex object geometries, improving planning efficiency and reactivity to dynamic targets and obstacles. *TWIST* (Chapter 7) presents a teacher-student distillation framework that achieves

effective zero-shot sim-to-real transfer for model-based RL by distilling learnt latent representations in imagined trajectories. *D-Cubed* (Chapter 8) shows that latent diffusion models, trained on task-agnostic play datasets, can generate diverse action sequences that facilitate effective exploration and improve trajectory optimisation for dexterous deformable object manipulation tasks by integrating the Cross-Entropy Method into the reverse diffusion process.

These contributions establish a foundational framework for unified planning and learning for contact-rich manipulation tasks in unstructured settings. The experimental validation across diverse tasks highlights the robustness of the approach and its potential for broad applicability.

10.2 Broader Implications

This thesis has significant implications for the practical deployment of robotics in real-world applications. The demonstrated capability to efficiently acquire complex manipulation skills in unstructured settings addresses key requirements in real-world applications such as small-batch manufacturing or open-world scenarios, where robots must rapidly and safely acquire skills or generalise to novel objects, tasks, and environments using readily available data. Furthermore, this thesis highlights the effectiveness of generative modelling as a component of model-based decision-making, offering tools for both learning dynamics models and guiding optimisation in robotic systems by leveraging readily collected data. Lastly, the successful sim-to-real transfer achieved across multiple frameworks presented in this thesis provides practical pathways for deploying these techniques in real-world environments.

10.3 Final Remarks

This thesis represents a significant step toward safe, efficient, adaptive, and generalisable robotic systems that can operate effectively in contact-rich manipulation tasks in unstructured settings. By demonstrating principled ways to unify planning and learning that leverage the strengths of both paradigms, this thesis contributes to the broader goal of developing capable and practical robotic systems in the

real world. Moreover, enhancing model-based approaches with generative models offers further potential to improve these methods, such as by improving dynamics modelling or serving as action samplers for more efficient trajectory optimisation. The real-world experiments conducted in this thesis provide compelling evidence for the practical potential of unified planning and learning. Although achieving robust performance in semi-structured, unstructured environments remains a challenge, this thesis demonstrates that combining the strengths of planning and learning enables meaningful progress toward this goal. The proposed frameworks establish a strong foundation for future developments that will continue to advance the capabilities of robotic systems in real-world applications. Looking ahead, the unification of planning and learning presented in this thesis represents an important step toward more adaptive, generalisable, efficient, and capable robotic systems.

References

[1] Jan F. Broenink and Martin L.J. Tiernego. “Peg-in-hole assembly using impedance control with a 6 DOF robot”. In: *Simulation in Industry, 8th European Simulation Symposium, ESS’96*. Society for Computer Simulation International, 1996, pp. 504–508.

[2] Hyeonjun Park et al. “Intuitive peg-in-hole assembly strategy with a compliant manipulator”. In: *IEEE ISR 2013*. 2013, pp. 1–5.

[3] Joshua C. Triyonoputro, Weiwei Wan, and Kensuke Harada. “Quickly Inserting Pegs into Uncertain Holes using Multi-view Images and Deep Network Trained on Synthetic Data”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2019.

[4] Haihang Wang et al. “A Novel Soft Robotic Hand Design With Human-Inspired Soft Palm: Achieving a Great Diversity of Grasps”. In: *IEEE Robotics; Automation Magazine* 28.2 (June 2021), pp. 37–49. URL: http://dx.doi.org/10.1109/MRA.2021.3065870.

[5] Yuni Fuchioka et al. “Robotic object insertion with a soft wrist through sim-to-real privileged training”. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2024, pp. 9159–9166.

[6] Mike Lambeta et al. “Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation”. In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 3838–3845.

[7] James J Kuffner and Steven M LaValle. “RRT-connect: An efficient approach to single-query path planning”. In: *IEEE International Conference on Robotics and Automation*. Vol. 2. IEEE. 2000, pp. 995–1001.

[8] Sertac Karaman and Emilio Frazzoli. “Sampling-based algorithms for optimal motion planning”. In: *The International Journal of Robotics Research* 30.7 (2011), pp. 846–894.

[9] Nancy M Amato and Yan Wu. “A randomized roadmap method for path and manipulation planning”. In: *IEEE International Conference on Robotics and Automation*. 1996.

[10] Robert Bohlin and Lydia E Kavraki. “Path planning using lazy PRM”. In: *IEEE International Conference on Robotics and Automation*. Vol. 1. IEEE. 2000, pp. 521–528.

[11] Reuven Rubinstein. “The Cross-Entropy Method for Combinatorial and Continuous Optimization”. In: *Method. Comput. Appl. Prob.* 1.2 (1999), pp. 127–190.

[12] Grady Williams et al. “Aggressive driving with model predictive path integral control”. In: *IEEE International Conference on Robotics and Automation*. 2016, pp. 1433–1440.

[13] Danijar Hafner et al. “Dream to Control: Learning Behaviors by Latent Imagination”. In: *International Conference on Learning Representations*. 2020.

[14] Danijar Hafner et al. *Mastering Atari with Discrete World Models*. 2021.

[15] Nicklas Hansen, Hao Su, and Xiaolong Wang. "TD-MPC2: Scalable, Robust World Models for Continuous Control". In: *International Conference on Learning Representations*. 2024.

[16] Irina Higgins et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *International Conference on Learning Representations*. 2017.

[17] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 6840–6851.

[19] Chia-Man Hung et al. "Reaching Through Latent Space: From Joint Statistics to Path Planning in Manipulation". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 5334–5341.

[20] Volodymyr Mnih et al. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).

[21] Tuomas Haarnoja et al. "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor". In: *International Conference on Machine Learning*. 2018.

[22] Scott Fujimoto, Herke Hoof, and David Meger. "Addressing function approximation error in actor-critic methods". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1587–1596.

[23] Stefan Schaal. "Is imitation learning the route to humanoid robots?" In: *Trends in Cognitive Sciences* 3.6 (1999), pp. 233–242.

[24] A. Billard et al. "Survey: Robot Programming by Demonstration". In: *Springer Handbook of Robotics* (2008), pp. 1371–1394.

[25] Oliver Groth et al. "Goal-Conditioned End-to-End Visuomotor Control for Versatile Skill Primitives". In: *IEEE International Conference on Robotics and Automation*. 2021, pp. 1319–1325.

[26] Tony Z. Zhao et al. "ALOHA Unleashed: A Simple Recipe for Robot Dexterity". In: *Conference on Robot Learning*. 2024.

[27] Kento Kawaharazuka et al. *Vision-Language-Action Models for Robotics: A Review Towards Real-World Applications*. `https://vla-survey.github.io`. 2025.

[28] Jose Barreiros et al. "A careful examination of large behavior models for multitask dexterous manipulation". In: *arXiv preprint arXiv:2507.05331* (2025).

[29] Physical Intelligence et al. "$\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization". In: *arXiv preprint arXiv:2504.16054* (2025).

[30] Steven M. LaValle. *Rapidly-exploring random trees: A new tool for path planning*. Tech. rep. TR 98-11. Computer Science Department, Iowa State University, 1998.

[31] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[32] Jun Yamada et al. “Motion Planner Augmented Reinforcement Learning for Obstructed Environments”. In: *Conference on Robot Learning*. 2020.

[33] I-Chun Arthur Liu et al. “Distilling Motion Planner Augmented Policies into Visual Control Policies for Robot Manipulation”. In: *Conference on Robot Learning*. 2021.

[34] Michelle A Lee et al. “Guided Uncertainty-Aware Policy Optimization: Combining Learning and Model-Based Strategies for Sample-Efficient Policy Learning”. In: *IEEE International Conference on Robotics and Automation* (2020).

[35] Yizhe Wu et al. “APEX: Unsupervised, object-centric scene segmentation and tracking for robot manipulation”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2021, pp. 3375–3382.

[36] Todor Davchev et al. “Residual learning from demonstration: Adapting dmps for contact-rich manipulation”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 4488–4495.

[37] Adithyavairavan Murali et al. “GraspGen: A Diffusion-based Framework for 6-DOF Grasping with On-Generator Training”. In: *arXiv preprint arXiv:2507.13097* (2025).

[38] Wentao Yuan et al. “M2T2: Multi-Task Masked Transformer for Object-centric Pick and Place”. In: *Conference on Robot Learning*. 2023.

[39] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. “6-dof graspnet: Variational grasp generation for object manipulation”. In: *IEEE International Conference on Computer Vision*. 2019, pp. 2901–2910.

[40] Hao-Shu Fang et al. “AnyGrasp: Robust and Efficient Grasp Perception in Spatial and Temporal Domains”. In: *IEEE Transactions on Robotics (T-RO)* (2023).

[41] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. “Acronym: A large-scale grasp dataset based on simulation”. In: *IEEE International Conference on Robotics and Automation*. IEEE. 2021, pp. 6222–6227.

[42] Adithyavairavan Murali et al. “6-dof grasping for target-driven object manipulation in clutter”. In: *IEEE International Conference on Robotics and Automation*. IEEE. 2020, pp. 6232–6238.

[43] Dmitry Kalashnikov et al. “Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation”. In: *Conference on Robot Learning*. 2018, pp. 651–673.

[44] Tyler Ga Wei Lum et al. “DextrAH-G: Pixels-to-Action Dexterous Arm-Hand Grasping with Geometric Fabrics”. In: *Conference on Robot Learning*. PMLR. 2025, pp. 3182–3211.

[45] Shuran Song et al. “Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations”. In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 4978–4985.

[46] Beining Han et al. “FetchBench: A Simulation Benchmark for Robot Fetching”. In: *Conference on Robot Learning*. PMLR. 2025, pp. 3053–3071.

[47] Leia Bagesteiro and Robert Sainburg. “Nondominant Arm Advantages in Load Compensation During Rapid Elbow Joint Movements”. In: *Journal of neurophysiology* 90 (Oct. 2003), pp. 1503–13.

[48] Robert L. Sainburg. "Evidence for a dynamic-dominance hypothesis of handedness". In: *Experimental Brain Research* 142 (2001), pp. 241–258.

[49] Leia B Bagesteiro and Robert L Sainburg. "Handedness: dominant arm advantages in control of limb dynamics". In: *Journal of neurophysiology* 88.5 (2002), pp. 2408–2421.

[50] Prafulla Dhariwal and Alexander Nichol. "Diffusion models beat gans on image synthesis". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 8780–8794.

[51] Alexander Quinn Nichol and Prafulla Dhariwal. "Improved denoising diffusion probabilistic models". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8162–8171.

[52] Adam Fishman et al. "Motion policy networks". In: *Conference on Robot Learning*. PMLR. 2023, pp. 967–977.

[53] Ahmed H Qureshi et al. "Motion planning networks". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2019, pp. 2118–2124.

[54] Ahmed Hussain Qureshi et al. "Motion planning networks: Bridging the gap between learning-based and classical motion planners". In: *IEEE Transactions on Robotics* 37.1 (2020), pp. 48–66.

[55] David Ha and Jürgen Schmidhuber. *World Models*. 2018. URL: `https://zenodo.org/record/1207631`.

[56] Bohan Wu et al. "Example-Driven Model-Based Reinforcement Learning for Solving Long-Horizon Visuomotor Tasks". In: *Conference on Robot Learning*. PMLR. 2022, pp. 1–13.

[57] Josh Tobin et al. "Domain randomization for transferring deep neural networks from simulation to the real world". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2017, pp. 23–30.

[58] Marc Rigter, Minqi Jiang, and Ingmar Posner. "Reward-Free Curricula for Training Robust World Models". In: *International Conference on Learning Representations*. 2024.

[59] Andrei A. Rusu et al. *Policy Distillation*. 2016. arXiv: `1511.06295 [cs.LG]`.

[60] Danijar Hafner et al. "Learning latent dynamics for planning from pixels". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2555–2565.

[61] Robin Rombach et al. "High-resolution image synthesis with latent diffusion models". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695.

[62] Jun Yamada, Jack Collins, and Ingmar Posner. "Efficient skill acquisition for insertion tasks in obstructed environments". In: *Learning for Dynamics and Control Conference*. Ed. by Alessandro Abate et al. Vol. 242. Proceedings of Machine Learning Research. PMLR, 2024, pp. 615–627.

[63] Jun Yamada et al. "Leveraging scene embeddings for gradient-based motion planning in latent space". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2023, pp. 5674–5680.

[64] Jun Yamada et al. “D-Cubed: Latent Diffusion Trajectory Optimisation for Dexterous Deformable Manipulation”. In: *Conference on Robot Learning*. 2025.

[65] Jun Yamada et al. “TWIST: Teacher-Student World Model Distillation for Efficient Sim-to-Real Transfer”. In: *IEEE International Conference on Robotics and Automation*. IEEE. 2024, pp. 9190–9196.

[66] Jun Yamada et al. “COMBO-Grasp: Learning Constraint-Based Manipulation for Bimanual Occluded Grasping”. In: *Conference on Robot Learning*. 2025.

[67] Jun Yamada et al. “Grasp-MPC: Closed-Loop Visual Grasping via Value-Guided Model Predictive Control”. In: *IEEE International Conference on Robotics and Automation* (2026).

[68] Jack Collins et al. “RAMP: A benchmark for evaluating robotic assembly manipulation and planning”. In: *IEEE Robotics and Automation Letters* (2023).

[69] Marc Rigter, Jun Yamada, and Ingmar Posner. “World Models via Policy-Guided Trajectory Diffusion”. In: *Transactions on Machine Learning Research* (2024).

[70] Oiwi Parker Jones et al. “Oscillating latent dynamics in robot systems during walking and reaching”. In: *Scientific Reports* 14.1 (2024), p. 11434.

[71] Alexander Luis Mitchell et al. “From Primates to Robots: Emerging Oscillatory Latent-Space Dynamics for Sensorimotor Control”. In: (2023).

[72] Lydia Kavraki and Jean-Claude Latombe. “Randomized preprocessing of configuration for fast path planning”. In: *IEEE International Conference on Robotics and Automation*. 1994.

[73] Steven M LaValle, James J Kuffner, BR Donald, et al. “Algorithmic and computational robotics: new directions”. In: AK Peters, 2001.
Chap. Rapidly-exploring random trees: Progress and prospects, pp. 293–308.

[74] Jonathan D Gammell, Timothy D Barfoot, and Siddhartha S Srinivasa. “Batch Informed Trees (BIT*): Informed asymptotically optimal anytime search”. In: *The International Journal of Robotics Research* 39.5 (2020), pp. 543–567.

[75] Marlin P Strub and Jonathan D Gammell. “Adaptively Informed Trees (AIT*): Fast asymptotically optimal path planning through adaptive heuristics”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 3191–3198.

[76] Trishant Roy, Anindya Harchowdhury, and Leena Vachhani. “Anytime Planning: A Motion Planner for Dynamic Environment”. In: *arXiv preprint arXiv:1912.11778* (2019).

[77] J. van den Berg, D. Ferguson, and J. Kuffner. “Anytime path planning and replanning in dynamic environments”. In: *IEEE International Conference on Robotics and Automation*. 2006, pp. 2366–2371.

[78] Balakumar Sundaralingam et al. “CuRobo: Parallelized collision-free minimum-jerk robot motion generation”. In: *arXiv preprint arXiv:2310.17274* (2023).

[79] Nathan Ratliff et al. “CHOMP: Gradient optimization techniques for efficient motion planning”. In: *IEEE International Conference on Robotics and Automation*. 2009, pp. 489–494.

[80] Mrinal Kalakrishnan et al. “STOMP: Stochastic trajectory optimization for motion planning”. In: *IEEE International Conference on Robotics and Automation*. 2011, pp. 4569–4574.

[81] John Schulman et al. “Motion planning with sequential convex optimization and convex collision checking”. In: *Int. J. Rob. Res.* 33.9 (2014), pp. 1251–1270.

[82] Igor Mordatch, Zoran Popović, and Emanuel Todorov. “Contact-Invariant Optimization for Hand Manipulation”. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2012, pp. 137–144.

[83] Yunfei Bai and C. Karen Liu. “Dexterous manipulation using both palm and fingers”. In: *IEEE International Conference on Robotics and Automation*. 2014, pp. 1560–1565.

[84] Balakumar Sundaralingam and Tucker Hermans. “Relaxed-rigidity constraints: kinematic trajectory optimization and collision avoidance for in-grasp manipulation”. In: *Autonomous Robots* 43.2 (2019), pp. 469–483.

[85] Henry J Charlesworth and Giovanni Montana. “Solving Challenging Dexterous Manipulation Tasks With Trajectory Optimisation and Reinforcement Learning”. In: *International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 1496–1506.

[86] E. Todorov and Weiwei Li. “A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems”. In: *Proceedings of the 2005, American Control Conference, 2005.* 2005, 300–306 vol. 1.

[87] Yuval Tassa, Tom Erez, and Emanuel Todorov. “Synthesis and stabilization of complex behaviors through online trajectory optimization”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 4906–4913.

[88] Zhiao Huang et al. “PlasticineLab: A Soft-Body Manipulation Benchmark with Differentiable Physics”. In: *International Conference on Learning Representations*. 2021.

[89] Sizhe Li et al. “DexDeform: Dexterous Deformable Object Manipulation with Human Demonstrations and Differentiable Physics”. In: *International Conference on Learning Representations*. 2023.

[90] Yiling Qiao et al. “Differentiable simulation of soft multi-body systems”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 17123–17135.

[91] Yuanming Hu et al. “DiffTaichi: Differentiable Programming for Physical Simulation”. In: *International Conference on Learning Representations*. 2019.

[92] Rika Antonova et al. “Rethinking optimization with differentiable simulation from a global perspective”. In: *Conference on Robot Learning*. PMLR. 2023, pp. 276–286.

[93] D. Whitney. “Historical perspective and state of the art in robot force control”. In: *IEEE International Conference on Robotics and Automation*. Vol. 2. 1985, pp. 262–268.

[94] M. Shimizu and K. Kosuge. “Designing robot admittance for polyhedral parts assembly taking into account grasping uncertainty”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005, pp. 2899–2904.

[95] P. Nguyen and F. Naghdy. "Fuzzy control of automatic peg-in-hole insertion". In: *Proceedings of Third Australian and New Zealand Conference on Intelligent Information Systems. ANZIIS-95*. 1995, pp. 134–139.

[96] Shixiang Gu et al. "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates". In: *IEEE International Conference on Robotics and Automation*. 2017.

[97] Sergey Levine et al. "End-to-End Training of Deep Visuomotor Policies". In: *Journal of Machine Learning Research* (2016).

[98] Wenzhao Lian et al. "Benchmarking off-the-shelf solutions to robotic assembly tasks". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2021, pp. 1046–1053.

[99] Timothy P Lillicrap et al. "Continuous control with deep reinforcement learning". In: *International Conference on Learning Representations* (2016).

[100] Albert Zhan et al. "A Framework for Efficient Robotic Manipulation". In: *arXiv:2012.07975* (2020).

[101] Wenxuan Zhou and David Held. "Learning to grasp the ungraspable with emergent extrinsic dexterity". In: *Conference on Robot Learning*. PMLR. 2023, pp. 150–160.

[102] Yuke Zhu et al. "Reinforcement and Imitation Learning for Diverse Visuomotor Skills". In: *Robotics: Science and Systems*. 2018.

[103] Yevgen Chebotar et al. "Closing the sim-to-real loop: Adapting simulation randomization with real world experience". In: *IEEE International Conference on Robotics and Automation*. 2019, pp. 8973–8979.

[104] Jianlan Luo et al. "Robust multi-modal policies for industrial assembly via reinforcement learning and demonstrations: A large-scale study". In: *Robotics: Science and Systems*. 2021.

[105] John Schulman et al. "Trust region policy optimization". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1889–1897.

[106] John Schulman et al. "High-dimensional continuous control using generalized advantage estimation". In: *arXiv preprint arXiv:1506.02438* (2015).

[107] NVIDIA Corporation. *Isaac Sim*. 2025. URL: `https://developer.nvidia.com/isaac/sim` (visited on 07/25/2025).

[108] Jianlan Luo et al. "Serl: A software suite for sample-efficient robotic reinforcement learning". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2024, pp. 16961–16969.

[109] Jianlan Luo et al. "Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning". In: *arXiv preprint arXiv:2410.21845* (2024).

[110] Sebastian Thrun and Anton Schwartz. "Issues in Using Function Approximation for Reinforcement Learning". In: 1999. URL: `https://api.semanticscholar.org/CorpusID:1115058`.

[111] Philip J Ball et al. "Efficient online reinforcement learning with offline data". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 1577–1594.

[112] Toru Lin et al. “Twisting Lids Off with Two Hands”. In: *Conference on Robot Learning*. PMLR. 2025, pp. 5220–5235.

[113] Rohan Chitnis et al. “Efficient bimanual manipulation using learned task schemas”. In: *IEEE International Conference on Robotics and Automation*. IEEE. 2020, pp. 1149–1155.

[114] Rohan Chitnis et al. “Intrinsic Motivation for Encouraging Synergistic Behavior”. In: *International Conference on Learning Representations*. 2020.

[115] Yunfei Li et al. “Efficient Bimanual Handover and Rearrangement via Symmetry-Aware Actor-Critic Learning”. In: *IEEE International Conference on Robotics and Automation*. 2023, pp. 3867–3874.

[116] Hengyuan Hu, Suvir Mirchandani, and Dorsa Sadigh. *Imitation Bootstrapped Reinforcement Learning*. 2024. arXiv: `2311.02198 [cs.LG]`.

[117] Lars Ankile et al. *From Imitation to Refinement – Residual RL for Precise Assembly*. 2024. arXiv: `2407.16677 [cs.RO]`.

[118] Marius Memmel et al. “ASID: Active Exploration for System Identification in Robotic Manipulation”. In: *International Conference on Learning Representations*. 2024.

[119] Nicholas Pfaff et al. *Scalable Real2Sim: Physics-Aware Asset Generation Via Robotic Pick-and-Place Setups*. 2025. arXiv: `2503.00370 [cs.RO]`.

[120] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. “Gaussian Processes for Data-Efficient Learning in Robotics and Control”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.2 (Feb. 2015), pp. 408–423.

[121] Kurtland Chua et al. “Deep reinforcement learning in a handful of trials using probabilistic dynamics models”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[122] Frederik Ebert et al. *Visual Foresight: Model-Based Deep Reinforcement Learning for Vision-Based Robotic Control*. 2018. arXiv: `1812.00568 [cs.RO]`.

[123] Zhiheng Xi et al. “The rise and potential of large language model based agents: A survey”. In: *Science China Information Sciences* 68.2 (2025), p. 121101.

[124] Yuval Tassa et al. “DeepMind Control Suite”. In: *arXiv preprint arXiv:1801.00690* (2018).

[125] Ashish Kumar Shakya, Gopinatha Pillai, and Sohom Chakrabarty. “Reinforcement learning algorithms: A brief survey”. In: *Expert Systems with Applications* 231 (2023), p. 120495.

[126] Philipp Wu et al. “DayDreamer: World Models for Physical Robot Learning”. In: *Conference on Robot Learning*. 2022.

[127] Nicklas A Hansen, Hao Su, and Xiaolong Wang. “Temporal Difference Learning for Model Predictive Control”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 8387–8406.

[128] Nicklas Hansen et al. “MoDem: Accelerating Visual Model-Based Reinforcement Learning with Demonstrations”. In: *International Conference on Learning Representations*. 2023.

[129] Patrick Lancaster et al. "Modem-v2: Visuo-motor world models for real-world robot manipulation". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 7530–7537.

[130] Younggyo Seo et al. "Masked world models for visual control". In: *Conference on Robot Learning*. PMLR. 2023, pp. 1332–1344.

[131] Shuang Li et al. "Unified Video Action Model". In: *Robotics: Science and Systems*. 2025.

[132] Chuning Zhu et al. "Unified World Models: Coupling Video and Action Diffusion for Pretraining on Large Robotic Datasets". In: *Robotics: Science and Systems*. 2025.

[133] Ashish Vaswani et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems* 30 (2017).

[134] Xue Bin Peng et al. "Advantage-weighted regression: Simple and scalable off-policy reinforcement learning". In: *arXiv preprint arXiv:1910.00177* (2019).

[135] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. "Offline Reinforcement Learning with Implicit Q-Learning". In: *International Conference on Learning Representations*. 2022.

[136] Aviral Kumar et al. "Conservative q-learning for offline reinforcement learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1179–1191.

[137] Seohong Park et al. "Is Value Learning Really the Main Bottleneck in Offline RL?" In: *Advances in Neural Information Processing Systems*. 2024.

[138] Tianhao Zhang et al. "Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation". In: *IEEE International Conference on Robotics and Automation*. 2018, pp. 5628–5635.

[139] Mariusz Bojarski et al. "End to End Learning for Self-Driving Cars". In: *arXiv preprint arXiv:1604.07316* (2016).

[140] Brian D. Ziebart et al. "Maximum Entropy Inverse Reinforcement Learning". In: *National Conference on Artificial Intelligence*. Vol. 3. 2008, pp. 1433–1438.

[141] Pieter Abbeel and Andrew Y. Ng. "Apprenticeship Learning via Inverse Reinforcement Learning". In: *International Conference on Machine Learning*. 2004.

[142] Jonathan Ho and Stefano Ermon. "Generative adversarial imitation learning". In: *Advances in Neural Information Processing Systems* 29 (2016).

[143] Joe Watson, Sandy Huang, and Nicolas Heess. "Coherent soft imitation learning". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 14540–14583.

[144] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. "A reduction of imitation learning and structured prediction to no-regret online learning". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 627–635.

[145] Branton DeMoss et al. "Ditto: Offline imitation learning with world models". In: *arXiv preprint arXiv:2302.03086* (2023).

[146] Tony Z Zhao et al. “Learning fine-grained bimanual manipulation with low-cost hardware”. In: *Robotics: Science and Systems* (2023).

[147] Cheng Chi et al. “Diffusion policy: Visuomotor policy learning via action diffusion”. In: *The International Journal of Robotics Research* (2023), p. 02783649241273668.

[148] Ajay Mandlekar et al. “What Matters in Learning from Offline Human Demonstrations for Robot Manipulation”. In: *Conference on Robot Learning*. PMLR. 2022, pp. 1678–1690.

[149] Philipp Wu et al. “Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2024, pp. 12156–12163.

[150] Aadhithya Iyer et al. “OPEN TEACH: A Versatile Teleoperation System for Robotic Manipulation”. In: *Conference on Robot Learning*. PMLR. 2025, pp. 2372–2395.

[151] Runyu Ding et al. *Bunny-VisionPro: Real-Time Bimanual Dexterous Teleoperation for Imitation Learning*. 2024. arXiv: 2407.03162 [cs.RO].

[152] Yuzhe Qin et al. “AnyTeleop: A General Vision-Based Dexterous Robot Arm-Hand Teleoperation System”. In: *Robotics: Science and Systems*. 2023.

[153] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. “6-dof graspnet: Variational grasp generation for object manipulation”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 2901–2910.

[154] Karl Pertsch, Youngwoon Lee, and Joseph Lim. “Accelerating reinforcement learning with learned skill priors”. In: *Conference on robot learning*. PMLR. 2021, pp. 188–204.

[155] Ziyi Wu et al. “Slotdiffusion: Object-centric generative modeling with diffusion models”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 50932–50958.

[156] Jonathan Ho and Tim Salimans. “Classifier-free diffusion guidance”. In: *arXiv preprint arXiv:2207.12598* (2022).

[157] Yanjie Ze et al. *3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations*. 2024. arXiv: 2403.03954 [cs.RO]. URL: https://arxiv.org/abs/2403.03954.

[158] Zihan Ding et al. “Diffusion world model: Future modeling beyond step-by-step rollout for offline reinforcement learning”. In: *arXiv preprint arXiv:2402.03570* (2024).

[159] Michael Janner et al. “Planning with Diffusion for Flexible Behavior Synthesis”. In: *International Conference on Machine Learning*. 2022.

[160] Joao Carvalho et al. “Motion planning diffusion: Learning and planning of robot motions with diffusion models”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2023, pp. 1916–1923.

[161] Joao Carvalho et al. “Grasp diffusion network: Learning grasp generators from partial point clouds with diffusion models in so (3) xr3”. In: *arXiv preprint arXiv:2412.08398* (2024).

[162] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. "Normalizing flows: An introduction and review of current methods". In: *IEEE transactions on pattern analysis and machine intelligence* 43.11 (2020), pp. 3964–3979.

[163] Jacob Sacks and Byron Boots. "Learning Sampling Distributions for Model Predictive Control". In: *Conference on Robot Learning*. PMLR. 2023, pp. 1733–1742.

[164] Kevin Black et al. "*pi*_0: A Vision-Language-Action Flow Model for General Robot Control". In: *arXiv preprint arXiv:2410.24164* (2024).

[165] Johan Bjorck et al. "Gr00t n1: An open foundation model for generalist humanoid robots". In: *arXiv preprint arXiv:2503.14734* (2025).

[166] David McAllister et al. "Flow Matching Policy Gradients". In: *arXiv preprint arXiv:2507.21053* (2025).

[167] Mark Pfeiffer et al. "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots". In: *IEEE International Conference on Robotics and Automation*. IEEE, 2017.

[168] Brian Ichter, James Harrison, and Marco Pavone. "Learning sampling distributions for robot motion planning". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2018, pp. 7087–7094.

[169] Ahmed H Qureshi and Michael C Yip. "Deeply Informed Neural Sampling for Robot Motion Planning". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2018, pp. 6582–6588.

[170] Murtaza Dalal et al. "Neural MP: A Generalist Neural Motion Planner". In: *arXiv preprint arXiv:2409.05864* (2024).

[171] Brian Ichter and Marco Pavone. "Robot Motion Planning in Learned Latent Spaces". In: *IEEE Robotics and Automation Letters* 4 (2019).

[172] Murtaza Dalal et al. "Plan-Seq-Learn: Language Model Guided RL for Solving Long Horizon Robotics Tasks". In: *International Conference on Learning Representations*. 2024.

[173] Ian Lenz, Ross A. Knepper, and Ashutosh Saxena. "DeepMPC: Learning Deep Latent Features for Model Predictive Control". In: *Robotics: Science and Systems*. 2015.

[174] Chelsea Finn and Sergey Levine. "Deep visual foresight for planning robot motion". In: *IEEE International Conference on Robotics and Automation*. 2017, pp. 2786–2793.

[175] Manuel Watter et al. "Embed to control: A locally linear latent dynamics model for control from raw images". In: *Advances in Neural Information Processing Systems* 28 (2015).

[176] Frederik Ebert et al. "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control". In: *arXiv preprint arXiv:1812.00568* (2018).

[177] Danijar Hafner et al. "Learning Latent Dynamics for Planning from Pixels". In: *International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2555–2565.

[178] Neel Jawale et al. *Dynamic Non-Prehensile Object Transport via Model-Predictive Reinforcement Learning*. 2024. arXiv: 2412.00086 [cs.RO].

[179] Mingyuan Zhong et al. "Value function approximation and model predictive control". In: *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. 2013, pp. 100–107.

[180] Kendall Lowrey et al. "Plan Online, Learn Offline: Efficient Learning and Exploration via Model-Based Control". In: *International Conference on Learning Representations*. 2019.

[181] Yun-Chun Chen et al. "Neural motion fields: Encoding grasp trajectories as implicit value functions". In: *arXiv preprint arXiv:2206.14854* (2022).

[182] Liangjun Zhang and Dinesh Manocha. "An efficient retraction-based RRT planner". In: *IEEE International Conference on Robotics and Automation*. 2008, pp. 3743–3750.

[183] OpenAI et al. "Learning dexterous in-hand manipulation". In: *The International Journal of Robotics Research* 39.1 (2020), pp. 3–20.

[184] Gerrit Schoettler et al. "Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2020.

[185] Alexander Kirillov et al. "Segment anything". In: *IEEE International Conference on Computer Vision*. 2023, pp. 4015–4026.

[186] Nikhila Ravi et al. "SAM 2: Segment Anything in Images and Videos". In: *International Conference on Learning Representations*. 2025.

[187] Bowen Wen et al. "Foundationpose: Unified 6d pose estimation and tracking of novel objects". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 17868–17879.

[188] Matt Deitke et al. "Objaverse: A universe of annotated 3d objects". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 13142–13153.

[189] Qiyang Li, Zhiyuan Zhou, and Sergey Levine. *Reinforcement Learning with Action Chunking*. 2025. arXiv: 2507.07969 [cs.LG].

[190] Zhaole Sun et al. "Learning pregrasp manipulation of objects from ungraspable poses". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2020, pp. 9917–9923.

[191] Ling Yang et al. *Diffusion Models: A Comprehensive Survey of Methods and Applications*. 2024. arXiv: 2209.00796 [cs.LG].

[192] Angel X Chang et al. "Shapenet: An information-rich 3d model repository". In: *arXiv preprint arXiv:1512.03012* (2015).

[193] Lerrel Pinto et al. "Asymmetric actor critic for image-based robot learning". In: *arXiv preprint arXiv:1710.06542* (2017).

[194] Julien Brosseit et al. "Distilled domain randomization". In: *arXiv preprint arXiv:2112.03149* (2021).

[195] Stephen James et al. "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12627–12637.

[196] Nicklas Hansen et al. *Generalizable Robotic Insertion with World Models*. 2025.

[197] Emanuel Todorov, Tom Erez, and Yuval Tassa. "Mujoco: A physics engine for model-based control". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 5026–5033.

[198] Yixuan Huang, Adam Conkey, and Tucker Hermans. "Planning for Multi-Object Manipulation with Graph Neural Network Relational Classifiers". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2023, pp. 1822–1829.

[199] Abby O'Neill et al. "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 6892–6903.

[200] Abhiram Maddukuri et al. "Sim-and-real co-training: A simple recipe for vision-based robotic manipulation". In: *arXiv preprint arXiv:2503.24361* (2025).

[201] Yi-Chen Li et al. *Generalist Reward Models: Found Inside Large Language Models*. 2025. arXiv: 2506.23235 [cs.CL].