

Geometry and Representation Learning in Deep Generative Models

Emile Mathieu

Mansfield College
University of Oxford

*A thesis submitted for the degree of
Doctor of Philosophy*

Trinity 2021

Abstract

Deep generative models have *de facto* emerged as state of the art when it comes to density estimation and sampling high-dimensional and multi-modal data. They combine the abstract yet practical mathematical description of probabilistic modelling with the flexibility and scalability brought by neural networks. They have been successfully applied to a wide spectrum of problems ranging from computer vision and natural language, to the realms of physical sciences. Probabilistic modelling, and in particular Bayesian statistics, have long enabled scientists and practitioners to include prior knowledge that they may have about the data into models. Building well-specified models is beneficial as it leads to improved generalisation capacity, data efficiency and interpretability of the model. In contrast, principled methods to encode such *inductive biases* in deep generative models are still under development.

This thesis presents three pieces of work aimed at addressing this problem. First, we propose a principled approach to introduce inductive biases in variational auto-encoders. Taking a Bayesian perspective, we show that encoding a desired structure into the prior distribution, and applying a proper regularisation, can lead to the desired decomposition in the learnt encodings. We demonstrate this approach on a variety of computer vision datasets and successfully learn representations with sparsity, clustering, and even intricate hierarchical dependency relationships. Next, we introduce an extension of variational auto-encoders to model data with underlying hierarchical structure. As hyperbolic spaces are perfectly suited to embed tree-like data, in contrast to Euclidean geometry, we endow the latent space with hyperbolic geometry. We do so by deriving the necessary methods to work with two main Gaussian generalisations and geometry-aware architectures for the encoder and decoder networks. Finally, we leverage the formalism of Riemannian *geometry* to define flexible distributions for data that are assumed to live on a given manifold. We do so by extending continuous normalising flows and parametrising manifold-valued diffeomorphisms as solutions of ordinary differential equations.

Geometry and Representation Learning in Deep Generative Models



Emile Mathieu
Mansfield College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2021

To Laurène

Acknowledgements

First and foremost, I would like to express my indebtedness appreciation to my departmental supervisor, Professor Yee Whye Teh, for his guidance and belief in me thorough these years. I am grateful to Ryota Tomioka from Microsoft Research and Max Nickel from Facebook Research for their strong support and formative advices. I would like to express my sincere gratitude to Benjamin Bloem-Reddy and Tom Rainforth who have been incredibly generous with their time and expertise, and from whom I have learnt a lot. I am also thankful to Yann Dubois and Laura Ruis for their friendship and continuous support throughout the meanders of life. I would also like to deeply thank the Oxford Computational Statistics and Machine Learning (OxCSML) group, for being an ideal space for pursuing research and meeting passionate people. In particular, I would like to warmly thank Adam Foster, Adam Golinski, Christopher Maddison, Anthony Caterini, Dominic Richard and Michael Hutchinson for their relentless support and strong friendship. I am deeply grateful to the vibrant community of East Oxford, where I met so many exciting people, who helped me grow and find meaning. Finally, I would like to offer my special thanks to partner Simone, my parents and family for their unwavering support.

Abstract

Deep generative models have *de facto* emerged as state of the art when it comes to density estimation and sampling high-dimensional and multi-modal data. They combine the abstract yet practical mathematical description of probabilistic modelling with the flexibility and scalability brought by neural networks. They have been successfully applied to a wide spectrum of problems ranging from computer vision and natural language, to the realms of physical sciences. Probabilistic modelling, and in particular Bayesian statistics, have long enabled scientists and practitioners to include prior knowledge that they may have about the data into models. Building well-specified models is beneficial as it leads to improved generalisation capacity, data efficiency and interpretability of the model. In contrast, principled methods to encode such *inductive biases* in deep generative models are still under development.

This thesis presents three pieces of work aimed at addressing this problem. First, we propose a principled approach to introduce inductive biases in variational auto-encoders. Taking a Bayesian perspective, we show that encoding a desired structure into the prior distribution, and applying a proper regularisation, can lead to the desired decomposition in the learnt encodings. We demonstrate this approach on a variety of computer vision datasets and successfully learn representations with sparsity, clustering, and even intricate hierarchical dependency relationships. Next, we introduce an extension of variational auto-encoders to model data with underlying hierarchical structure. As hyperbolic spaces are perfectly suited to embed tree-like data, in contrast to Euclidean geometry, we endow the latent space with hyperbolic geometry. We do so by deriving the necessary methods to work with two main Gaussian generalisations and geometry-aware architectures for the encoder and decoder networks. Finally, we leverage the formalism of Riemannian *geometry* to define flexible distributions for data that are assumed to live on a given manifold. We do so by extending continuous normalising flows and parametrising manifold-valued diffeomorphisms as solutions of ordinary differential equations.

Contents

1	Introduction and Literature Review	1
1.1	Deep generative models	3
1.1.1	Normalising flows	5
1.1.2	Variational auto-encoders	7
1.2	Representation learning and inductive biases	9
1.2.1	Feature extraction	9
1.2.2	Inductive biases	10
1.3	Geometry for probabilistic modelling	14
1.3.1	Geometric priors	16
1.3.2	Manifold-valued data	17
1.4	Summary	18
1.5	Non-included work	19
2	Disentangling Disentanglement in Variational Autoencoders	23
3	Continuous Hierarchical Representations with Poincaré Variational Autoencoders	45
4	Riemannian Continuous Normalizing Flows	71
5	Conclusion	97
5.1	Contributions and challenges	97
5.1.1	Disentangling disentanglement	97
5.1.2	Poincaré variational auto-encoders	98
5.1.3	Riemannian continuous normalising flows	100
5.2	Broader context	101
	Bibliography	103

Somewhere, something incredible is waiting to be known.

— Carl Sagan, 1977

1

Introduction and Literature Review

Observable phenomena from the surrounding world are driven by complex underlying dynamics; from the motion of astronomical objects, to the diffraction of waves and the diffusion of heat. The school of empiricism aspires to understand observable phenomena and build knowledge through sensorial observation. With the development of numerical sensors, the amount and variety of available *data* is ever increasing. Scientists have been developing theories and *models* supported by these observations. These models are described by the interplay of *variable* quantities of interest along with unknown parameters which are to be inferred, yielding insights and understanding about the observations. Such parametric models approximate phenomena of interest, and provide a mathematical framework in which learning can be formalised.

Physical processes underlying observable phenomena usually result in a variation of possible outcomes. This can be due to physical, or *ontological*, stochasticity inherent to the phenomenon or its environment. This stochasticity is known as *aleatoric uncertainty* of the observational process. Such ‘true randomness’, is for instance commonly posited when modelling atomic decays. Additionally, observations may also exhibit *epistemic uncertainty* due to limitations in the accuracy of the system observation. The numerical precision of sensors and digital data discretisation are common examples. Other instances of epistemic randomness

arise in the study of dynamical systems, such as chaotic dynamics which result in *probabilistic* behaviours despite their fundamental determinism. Regardless of the origin of these random variations, they can often be assumed to be sufficiently regular so that, while an observed process cannot be quantified by any single observation, it can be with a *distribution* of observations. Such is the aim of *probabilistic modelling*; to model an observable process of interest with a probability distribution, so that realisations of that process—referred to as the empirical data distribution—are modelled as exact samples from the model distribution. The observable phenomena is thus seen as a data generating process, implicitly defining a (potentially complex) probability distribution on a space of observations. To build a faithful model of this process, scientists leverage knowledge they have acquired about the phenomenon of interest, and incorporate it into the probabilistic model. This is typically achieved via specific choices of probability distributions, as they can have different supports and properties, but also through a particular structure of statistical (in)dependence between the variables.

Often, the data distribution is highly-multimodal and potentially high dimensional, rendering it challenging to model in a simple parametric form. The vast amount of available data suggests instead that we may try to directly *learn* a flexible parametric probabilistic model from data. Deep generative models have been developed for this purpose, they combine the flexibility of neural networks with the quantified uncertainty of probabilistic modelling. We discuss their howabouts in Section 1.1. As opposed to traditional probabilistic modelling, deep generative models usually have a very simple graphical model, and often assume normally distributed latent variables. This limits the type of assumptions about the data-generating process that can be encoded. In turn, this can be a hindrance for learning good data representations, i.e. data embeddings defined by the model’s posterior distribution. In Section 1.2, we review motivations for learning data representations, what kind of properties make a good representation, and how assumptions on the data can be encoded into deep generative models. In particular,

in Section 1.3 we examine how geometry can be leveraged as an inductive bias to incorporate such data assumptions.

Please find a summary of the thesis structure in Section 1.4.

1.1 Deep generative models

Deep generative models (DGMs) are probabilistic models based on deep neural networks that are trained to approximate complicated, high-dimensional probability distributions using samples from the data. Simply put, they are models that can generate data-points that look ‘similar’ to a given collection of data-points. These models aim to combine the quantified uncertainty offered by *probabilistic modelling*, with the flexibility and scalable learning of *deep neural networks*. They have shown great promise and have in particular become increasingly popular for high-dimensional density estimation tasks as traditional methods such as kernel density estimation (KDE) (Rosenblatt, 1956; Parzen, 1962) are afflicted by the *curse of dimensionality*, i.e. they have a sample complexity that grows exponentially. Traditionally, such methods have principally been developed for images (Aaron Van Oord et al., 2016; van den Oord et al., 2016; Kingma and Dhariwal, 2018), videos (Kalchbrenner et al., 2017), text, and audio (van den Oord et al., 2016; Prenger et al., 2019). Yet, there has recently been a fast growing interest in developing and applying such models to the realms of physical sciences, such as astronomy (Rodríguez et al., 2018), quantum mechanics (Sharir et al., 2020), statistical mechanics (Noé et al., 2019; Wu et al., 2019), particle physics (Louppe et al., 2019; Paganini et al., 2018), and molecular physics (Samanta et al., 2019). DGMs are mostly used for the sole sake of density estimation or generating new data-points, yet they also find applications in dimensionality reduction (Kingma et al., 2014), compression (Townsend et al., 2019; Ho et al., 2020b) and anomaly detection (Pol et al., 2019) to cite a few. This research area is one of the most rapidly evolving fields of statistical machine learning.

At a high level, deep generative models express the observed data as a (potentially noisy) pushforward of some latent random variable along a neural network, which is trained by minimising some statistical divergence to the empirical data distribution. Formally, we have access to an observable dataset $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in \mathcal{X}^n$, forming an empirical measure $P_{\mathcal{D}} = \frac{1}{n} \sum_i^n \delta_{\mathbf{x}_i}$. A deep generative model induces a parametric probability measure $P_{\theta} \in \mathcal{M}_+^1(\mathcal{X})$ ¹ with $\theta \in \Theta$ tunable parameters. Learning amounts to finding the optimal parameters θ that minimise

$$\arg \inf_{\theta \in \Theta} \mathcal{D}(P_{\mathcal{D}} || P_{\theta}) \quad (1.1)$$

with $\mathcal{D}$ a statistical divergence² measuring a notion of ‘distance’ between probability distributions. One such divergence is the Kullback-Leibler divergence defined³ by $\mathcal{D}_{KL}(P || Q) = \int_{\mathcal{X}} \log \left(\frac{dP}{dQ} \right) dP$ which when minimised is equivalent to performing *maximum likelihood*, as replacing $\mathcal{D}_{KL}$ in Equation (1.1) yields

$$\arg \inf_{\theta \in \Theta} \mathcal{D}_{KL}(P_{\mathcal{D}} || P_{\theta}) = \arg \sup_{\theta \in \Theta} \mathbb{E}_{\mathbf{x} \sim P_{\mathcal{D}}} [\log P_{\theta}(\mathbf{x})]. \quad (1.2)$$

The most celebrated deep generative models include *generative adversarial networks (GANs)* (Goodfellow et al., 2014), *autoregressive models* (Neal, 1992; Bengio and Bengio, 1999; Larochelle and Iain Murray, 2011), *variational auto-encoders (VAEs)* (Kingma and Welling, 2014; Rezende et al., 2014), *normalising flows (NFs)* (Rezende and Mohamed, 2015; Tabak and Vanden-Eijnden, 2010; Tabak and Turner, 2013), *energy-based models (EBMs)* (Hinton, 2002; Hinton et al., 2006; Teh et al., 2003; Du and Mordatch, 2019), and *score-based diffusion* models (Song and Ermon, 2019; Song et al., 2021; Ho et al., 2020a). EBMs can model expressive families of probability distributions as they do not place a restriction on the tractability of the normalising constant. However, this renders their training

¹ $\mathcal{M}_+^1(\mathcal{X})$ denotes the set of all probability measures on $\mathcal{X}$, i.e. the set of positive measures with unit volume. We use the terms probability distribution and probability measure interchangeably.

²Formally, a divergence is a function $\mathcal{D}(\cdot || \cdot) : \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{X}) \rightarrow \mathbb{R}_+$ satisfying $\mathcal{D}(p || q) \geq 0$ for all $p, q \in \mathcal{M}_+^1(\mathcal{X})$ and $\mathcal{D}(p || q) = 0$ if and only if $p = q$. Famous examples include the Kullback-Leibler divergence, f-divergences, Bregman divergence, the Wasserstein distance, maximum mean discrepancy (MMD) and integral probability metrics (IPMs).

³It is defined if $P_{\mathcal{D}}$ is *absolutely continuous* with respect to P_{θ} .

particularly difficult (Song and Kingma, 2021), so as evaluating their density. GANs have shown impressive results on image generation, yet they are also difficult to train and do not admit tractable likelihoods. In contrast, methods like NFs and VAEs result in densities that can either be directly computed or estimated in a tractable fashion. In what follows we focus on these two approaches.

1.1.1 Normalising flows

In particular, *normalising flows* (NFs) parametrise a probability measure P_θ by pushing forward a simple probability measure P_0 along an invertible transformation T_θ . Formally, we have $P_\theta = T_{\theta\#}P_0$, meaning that samples $\mathbf{x} \sim P_\theta$ are obtained as $\mathbf{x} = T_\theta(\mathbf{z})$ with $\mathbf{z} \sim P_0$. Informally, this can be thought of as literally moving the mass of P_0 around in the $\mathcal{X}$ space⁴ so that the new *distribution* of the mass matches a given target distribution.⁵ The defining property of NFs is that the transformation T_θ , or *flow*, must be *invertible* and both T_θ and T_θ^{-1} must be *differentiable*. Such functions are called *diffeomorphisms*, and require $\mathbf{x}$ and $\mathbf{z}$ to live in spaces of the same dimension (Milnor and Weaver, 1997). Under these assumptions, we can apply the change-of-variables formula which yields

$$P_\theta(\mathbf{x}) = P_0(\mathbf{z}) \left| \det \frac{\partial T_\theta}{\partial \mathbf{z}}(\mathbf{z}) \right|^{-1} \quad \text{with } \mathbf{z} = T_\theta^{-1}(\mathbf{x}). \quad (1.3)$$

As we can see, Equation (1.3) requires the evaluation of the transformation inverse T_θ^{-1} and the determinant of its Jacobian $\det \left(\frac{\partial T_\theta}{\partial \mathbf{z}} \right)$. Efforts have thus been put into developing transformations for which these operators can *efficiently* be computed, yet still have high expressive power. Several successful approaches have been proposed: linear (Rezende and Mohamed, 2015; Tomczak and Welling, 2017; Van Den Berg et al., 2018), autoregressive (Germain et al., 2015; Aaron Van Oord et al., 2016; Papamakarios et al., 2017; Kingma et al., 2016; Huang et al., 2018; Ho et al., 2019; De Cao et al., 2020; Jaini et al., 2019; Durkan et al., 2019; ?), affine-coupling (Dinh et al., 2017, 2015; Kingma and Dhariwal, 2018), residual (Behrmann et al.,

⁴although P_0 need not be defined on $\mathcal{X}$, only on a space of same dimension.

⁵The seminal work of Gaspar Monge (Monge, 1781) in the field of optimal transport (OT) was interested in finding the optimal way to move a pile of soil or rubble to an excavation or fill.

2019; Chen et al., 2019) and continuous flows (Chen et al., 2018b; Grathwohl et al., 2019). Each of these approaches makes a different assumption on the structure of the transformation, resulting in different yet tractable Jacobian (estimators). Linear flows of the form $\mathbf{x} = W\mathbf{z}$ yield a low-rank Jacobian as W is parametrised as a *low-rank* invertible matrix for tractability purposes of the determinant. Autoregressive flows of the form $p(\mathbf{x}) = \prod_i p(x_i | \mathbf{x}_{< i})$, yield a *lower triangular* Jacobian matrix. Affine-coupling flows yield *sparse lower triangular* Jacobian matrices. Residual flows of the form $\mathbf{x} = \mathbf{z} + g_\theta(\mathbf{z})$ and continuous flow—which are described in the next paragraph—yield an ‘unconstrained’ or *dense* Jacobian. One trade-off for these flows is the lack of general and efficient procedures for computing their Jacobian determinant, necessitating reliance on (unbiased) estimators.

Continuous normalising flows (CNFs) Continuous flows make a *time-continuity* assumption, specifically that there exists a transformation $T_\theta : \mathcal{X} \times \mathbb{R} \rightarrow \mathcal{X}$ which defines a continuous path $\{\mathbf{z}_t = T_\theta(\mathbf{z}, t) : t \in [t_0, t_1]\}$ running from $\mathbf{z} = \mathbf{z}_{t_0}$ to $\mathbf{x} = \mathbf{z}_{t_1}$. In particular, they implicitly parametrise the transformation as the flow generated by a *vector field*, i.e. as a *vector flow*. This flow⁶ is given by the solution of the following *ODE*

$$\frac{d\mathbf{z}_t}{dt} = g_\theta(\mathbf{z}_t, t) \quad (1.4)$$

with g_θ being a vector field that outputs *tangent vectors*, which can be seen as velocities of particles $\mathbf{z}_t$. The transformation T_θ is computed by running the dynamics forward in time, which (in the Euclidean setting) amounts to integrating

$$\mathbf{x} \triangleq \mathbf{z}_{t_1} = T_\theta(\mathbf{z}, t_1) = \mathbf{z} + \int_{t=t_0}^{t_1} g_\theta(\mathbf{z}_t, t) dt. \quad (1.5)$$

In practice, the ODE is usually integrated with a high-order method such as an explicit and adaptive Runge-Kutta integrator of order 4 (Dormand and Prince, 1980). Learning the flow T_θ amounts to learning the parameters θ of the vector field

⁶The terminology *flow* from *normalising flows* takes on its full meaning in this setting as the transformation that is defined by integrating an ordinary differential equation (ODE) is a *flow* in the mathematical sense, that is it has the following group structure: $T(\mathbf{z}, 0) = \mathbf{z}$ and $T(T(\mathbf{z}, t), s) = T(\mathbf{z}, s + t)$ for all $\mathbf{z} \in \mathcal{Z}$ and all $s, t \in \mathbb{R}$.

g_θ . Tackling the computation of the Jacobian determinant explicitly via automatic differentiation would cost $\mathcal{O}(N^3)$, with $N = \dim(\mathcal{X})$, which is prohibitive. CNFs circumvent this bottleneck by leveraging the time continuity of the transformation and express the change in density as

$$\log p_\theta(\mathbf{x}) - \log p_0(\mathbf{z}) = \int_{t_0}^{t_1} \frac{\partial \log p_\theta(\mathbf{z}_t)}{\partial t} dt = - \int_{t_0}^{t_1} \text{div}(g_\theta(\mathbf{z}(t), t)) dt \quad (1.6)$$

The last equality in Equation (1.6) can be obtained by the Liouville equation or equivalently by the Fokker–Planck equation without the diffusion term (Risken, 1989). In the Euclidean setting, we have $\text{div}(g_\theta(\mathbf{z}_t, t)) = \text{tr} \left(\frac{\partial g_\theta(\mathbf{z}_t, t)}{\partial \mathbf{z}_t} \right)$, which can be evaluated with N backward passes, yet that is still prohibitive for high dimensions. Fortunately, the Hutchinson’s trace estimator (Hutchinson, 1990) can be leveraged to obtain an unbiased estimator of the divergence

$$\text{tr} \left(\frac{\partial g_\theta(\mathbf{z}_t, t)}{\partial \mathbf{z}_t} \right) \approx \boldsymbol{\epsilon}^\top \frac{\partial g_\theta(\mathbf{z}_t, t)}{\partial \mathbf{z}_t} \boldsymbol{\epsilon}, \quad (1.7)$$

with $\boldsymbol{\epsilon}$ with centred and normalised random vector.

1.1.2 Variational auto-encoders

In contrast to NFs, variational auto-encoders (VAEs) map latent variables from a relatively small latent space $\mathcal{Z}$ to a higher-dimensional observation space $\mathcal{X}$, through a stochastic transformation f_θ called decoder or generator. This modelling choice is motivated by the *manifold hypothesis* which assumes that real-world high-dimensional data lie on low-dimensional manifolds embedded within the high-dimensional space $\mathcal{X}$. Think for instance of a collection of face pictures from an individual in the same environment. Although the space in which the data is represented is massive (millions of pixels), the true degree of variations is bounded by the number of facial muscles, which is 42. At a high level, VAEs are probabilistic auto-encoders, effectively encoding observable data into a low-dimensional distribution, before stochastically and approximately reconstructing the data from these encodings. VAEs usually assume a standard Gaussian latent

prior $p(\mathbf{z})$, and parametrise the likelihood mean and standard deviation with neural networks $f_\theta^\mu : \mathcal{Z} \rightarrow \mathcal{X}$ and $f_\theta^\sigma : \mathcal{Z} \rightarrow \mathbb{R}_+$ such that $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|f_\theta^\mu(\mathbf{z}), f_\theta^\sigma(\mathbf{z}))$. Maximising the marginal likelihood $p(\mathbf{x}) = \int_{\mathcal{Z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$ directly is intractable due to the non-linearity of the decoder network. Instead, VAEs are trained by maximising a lower bound on the log-likelihood in a similar fashion than expectation-maximisation. Following a standard variational approach it goes as

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \log \int_{\mathcal{Z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \\
&= \log \int_{\mathcal{Z}} \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}d\mathbf{z} \\
&\geq \int_{\mathcal{Z}} \log \left(\frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right) q_\phi(\mathbf{z}|\mathbf{x})d\mathbf{z} \\
&= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \triangleq \mathcal{L}(\theta, \phi) \quad (1.8)
\end{aligned}$$

where we introduced a *variational* distribution $q_\phi(\mathbf{z}|\mathbf{x})$ and applied Jensen inequality using the concavity of the logarithm. The gap between the ‘Evidence Lower Bound’ (ELBO) $\mathcal{L}(\theta, \phi)$ and the marginal log-likelihood is exactly $\mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x}))$, with the true posterior distribution given by *Bayes rule* $p(\mathbf{z}|\mathbf{x}) = p(\mathbf{x}, \mathbf{z})/p(\mathbf{x})$. We consequently have $\mathcal{L}(\theta, \phi) = \log p_\theta(\mathbf{x})$ if and only if $q_\phi(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x})$. In practice, the *variational family* $\mathcal{Q}_\Phi \triangleq \{q_\phi(\cdot|\cdot)|\phi \in \Phi\}$ is often too small to include the true posterior distributions. NFs, introduced above, can be applied on top of $q_\phi(\mathbf{z}|\mathbf{x})$ to increase the size of the variational family $\mathcal{Q}_\Phi$. The variational distribution approximates posterior distributions in an amortized fashion as $q_\phi(\mathbf{z}|\mathbf{x}) = q(\mathbf{z}|g_\phi^\mu(\mathbf{x}), g_\phi^\sigma(\mathbf{x}))$ rendering this method scalable w.r.t. the number of data-points.

VAEs can be seen as (non-linear) *dimensionality reduction* methods, as they learn a low-dimensional encoding, or *representation*, for each data-points via the inference network, $\mathbf{x} \mapsto q_\phi(\mathbf{z}|\mathbf{x})$. They relate observed data-points to hidden latent variables through some stochastic process. This (stochastic) embedding $q_\phi(\mathbf{z}|\mathbf{x})$ *encodes* sufficient features to reconstruct the original data-point; these features *explain* the data. The generative model $p_\theta(\mathbf{x})$ learns what is *common* between all the

data-points’ variability, while latent features $q_\phi(\mathbf{z}|\mathbf{x})$ encode a particular position within the dataset inner variability. With a linear decoder and Gaussian likelihood $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|A\mathbf{z}+b, \sigma^2)$, VAEs are closely related to principal component analysis (PCA) (Karl Pearson, 1901), and to other dimensionality reduction techniques.

1.2 Representation learning and inductive biases

1.2.1 Feature extraction

From this perspective, VAEs are one of many tools that can be leveraged to perform *feature extraction* or *representation learning* (Bengio et al., 2013; Lipton, 2018). These tasks aim to learn data representations that are informative, non-redundant and interpretable by human beings. Here, *informative* is ill-defined, as it needs to be informative w.r.t. something else that matters. The interest often lies in learning a representation $\mathbf{x} \mapsto \mathbf{z}$, of lower-dimensionality, such that the ‘information’ in $\mathbf{x}$ —w.r.t. some properties of interest $\mathbf{y}$ —is preserved, while satisfying other desirable properties—that we discuss in the next paragraph. This idea can be formalised via the concept of mutual information⁷, where one is interested in learning $\mathbf{z}$ with $\dim(\mathbf{z}) < \dim(\mathbf{x})$ such that $\text{MI}(\mathbf{z}; \mathbf{y}) \approx \text{MI}(\mathbf{x}; \mathbf{y})$. In addition, a good representation is generally ‘simple’, as in it is *linearly* dependant with $\mathbf{y}$, or at least more so than the raw data $\mathbf{x}$, i.e. such that $\text{corr}(\mathbf{z}, \mathbf{y}) \geq \text{corr}(\mathbf{x}, \mathbf{y})$. If so, a linear classifier or regressor would be sufficient to solve the task involving the prediction $\mathbf{z} \mapsto \mathbf{y}$. This is implicitly performed in deep neural networks, where all but the last layers learn a highly *non-linear* representation of the data, such that the final layer can successfully *linearly* regress or classify the data.

In an unsupervised setting—when no label is available—learning a representation of the data can be motivated by the sole aim of better understanding and summarising the available data. The knowledge acquired can then, in an iterative fashion, be used to build a better model of the data. Another strong appeal for

⁷defined as $I(\mathbf{x}; \mathbf{y}) = \mathcal{D}_{KL}(p(\mathbf{x}, \mathbf{y}) \| p(\mathbf{x})p(\mathbf{y}))$, which is equal to zero exactly when the joint distribution coincides with the product of the marginals, which is, when $\mathbf{x}$ and $\mathbf{y}$ are independent.

learning good representations is that they can in turn be used as *data-features* to solve downstream tasks. This is particularly useful in a semi-supervised setting, where the encoder can be trained in an unsupervised fashion on a large number of *unlabelled* data-points, before—or while—being used to extract features to tackle downstream tasks (Kingma et al., 2014). One hopes that these learnt features will be better *correlated* with their associated labels than the raw data-points, or at least be decodable by a ‘simpler’ downstream predictor.

Explanatory factors Ideally, the learnt representation would capture the probability distribution of the underlying *explanatory factors* for the observed input. The existence of such lower dimensional features is often referred to as the ‘manifold hypothesis’ (Fefferman et al., 2016) which assumes that the data actually lie on a low-dimensional manifold, whose dimension is referred to as the *intrinsic* data dimension. Think for instance, for a collection of facial pictures, of all the features that explain why a face is different from another, which likely include hair colour, skin colour, the presence or absence of glasses, and so on. In practice, the true underlying explanatory features are *unknown* but for synthetic datasets. For real world data, how can a ‘good’ representation therefore be defined?

1.2.2 Inductive biases

One answer can be given through the lens of prior knowledge that scientists or practitioners may have about the problem or data. Even though the true explanatory factors are likely unknown, properties of the data—may be known. These properties can be encoded as *inductive biases* in the model so as to express assumptions about the data-generating process. They allow some models or representations to be prioritised over others during the learning process.

The motivation for encoding properties of the data is that, if appropriately chosen, this can lead to better (1) performance for unobserved scenarios—*generalization*; (2) better use of the available data—*data-efficiency*; and (3) better human understanding of the phenomenon—*interpretability*. Inductive biases in (latent) probabilistic

modelling can therefore be motivated through both the lenses of representation learning and model specification. One is generally interested in one or the other, although both are obviously strongly connected. A well-specified probabilistic model, when appropriately trained, leads to good generalisation capacity in terms of density estimation and is able to generate realistic samples.

Inductive biases are ubiquitous in machine learning, although they are not always made explicit. They can for instance take the form of *invariances*, such as the time and space translation equivariance of respectively recurrent (Rumelhart et al., 1986; Jordan, 1986) and convolutional neural networks (Lecun et al., 1998). More recently, group and gauge equivariant convolutional neural networks (Cohen and Welling, 2016; Cohen et al., 2019; ?; Weiler et al., 2021) have gained a lot of attention by extending conventional neural networks to larger sets of transformation equivariances and spaces on which data is discretised. Invariances may also be described by conservation laws, such as the conservation of the Hamiltonian in Hamiltonian systems. This can itself be motivated by the celebrated Noether’s theorem (Noether, 1918) which shows that conservation laws are mirrored by (differentiable) symmetries.

Meta priors On the other hand, some properties try to capture very general features about the world and therefore aim to be relevant for a broad set of models. The meta-priors introduced in Bengio et al. (2013) aim to achieve exactly that. These priors can be seen as particular instances of inductive biases that can be used on a large spectrum of problems. In addition to the informativeness and lower-dimensionality criteria previously introduced, below details other such form of ‘meta-priors’.

One such prior is *sparsity*, that is, only a small fraction of the possible factors are expected to be relevant. In practice, this can be represented by features that are often zero, or insensitive to local variations of the input (Olshausen and Field, 1996). Due to their low effective dimensionality, sparse representations are expressive, as in they can capture a big number of possible input configurations with few *active*

dimensions (Bengio, 2009). Also, sparse features have been empirically shown to be able to improve data efficiency (Yip and Sussman, 1997), and provide robustness to noise (Ahmad and Scheinkman, 2019) and adversarial attacks (Gopalakrishnan et al., 2018). Another key prior of ‘good’ representations is the *hierarchical* organisation of explanatory factors. Deep neural networks tend to learn representations that are more and more abstract with respect to the depth. More abstract representations are usually *invariant* to most local changes of the data. Alternatively, one is likely to be interested in *disentangling* the different factors of variations, the assumption being that explanatory factors tend to change independently from each other when the input varies. One hopes to generalise to unseen configurations of the factors. In what follows, we develop on a line of work that aims to learn disentangled representations within VAEs.

Disentanglement Learning data representations with independent features has a long history, with ideas that stem back to traditional methods such as ICA (Hyvärinen and Oja, 2000; Yang and Amari, 1997) and auto-encoders (Schmidhuber, 1992). Independence has long been a property of interest as it is simple yet richer than unconstrained codes. Modern deep learning methods have also been developed to tackle this challenge (Reed et al., 2014; Mathieu et al., 2016; Hjelm et al., 2019; Chen et al., 2016; Achille and Soatto, 2019). There has been in particular, a focus on learning disentangled representation with VAEs (Alemi et al., 2017; Higgins et al., 2017; Siddharth et al., 2017; Kim and Mnih, 2018; Chen et al., 2018a). In this setting, one aims to achieve independence between the different learnt factors, that is, between the different dimensions for a representation $\mathbf{z} \in \mathbb{R}^d$. Within the context of VAEs, the representation is typically defined as the aggregate of the encodings:

$$q_\phi(\mathbf{z}) \triangleq \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} [q(\mathbf{z}|\mathbf{x})] \approx \frac{1}{n} \sum_i^n q(\mathbf{z}|\mathbf{x}_i), \quad (1.9)$$

with $q(\mathbf{z}|\mathbf{x}_i)$ the VAE stochastic encodings. This can be thought as the pushforward distribution for sampling latent representations. The β -VAE model (Higgins et al.,

2017) proposed to scale the KL term in the standard ELBO by a factor β as

$$\mathcal{L}_\beta(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta \mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \quad (1.10)$$

The idea being that a higher value of $\beta > 1$ encourages conditional independence in $q_\phi(\mathbf{z}|\mathbf{x})$, which constrains the capacity of $\mathbf{z}$, and combined with the log-likelihood pressure should encourage efficient representation of the data. Another approach that has been proposed, is to directly enforce independence within the factors by penalising the total correlation on the latent encodings defined as $\text{TC}(\mathbf{z}) = \mathcal{D}_{KL}(q_\phi(\mathbf{z})||\prod_j q_\phi(z_j))$ (Kim and Mnih, 2018; Chen et al., 2018a). Indeed, the total correlation equals zero if and only if the marginals z_j are independent.

We believe that such a notion of *disentanglement* is quite restrictive for complex models where the true generative factors are likely *not* independent. To this end, we develop in Chapter 2 a generalisation of disentanglement that allows for structures richer than independence, such as sparsity and clustering. We take a Bayesian perspective, and impose a desired structure on learnt representations via a choice of prior distribution $p(\mathbf{z})$. This permits disentanglement as a special case, for which the desired structure is that $q_\phi(\mathbf{z})$ factors along its dimensions such that $q_\phi(\mathbf{z}) = \prod_j q_\phi(z_j)$. We additionally regularise a divergence term $\mathcal{D}(q_\phi(\mathbf{z})||p(\mathbf{z}))$ to incorporate the prior’s inductive bias into the learnt representation. We show that β -VAE’s objective is invariant to rotations of the latent representations, and that breaking this invariance with simple manipulations of the prior can yield better disentanglement with little or no detriment to reconstructions. Additionally, we demonstrate how other choices of prior can assist in producing different decompositions such as sparse or clustered representations. One issue however, is that regularising $q_\phi(\mathbf{z})$ towards $p(\mathbf{z})$ can in practice be quite challenging when $p(\mathbf{z})$ is non-Gaussian.

In a follow-up work of Chapter 2 (Miao et al., 2022), we bypass the need to regularise the representation towards the prior distribution. Instead of replacing $p(\mathbf{z})$ directly, we introduce an intermediary set of latent variables $\mathbf{z}' \in \mathcal{Z}'$ with a standard Gaussian prior, $p(\mathbf{z}') = \mathcal{N}(\mathbf{z}'; 0, I)$, and then define $\mathbf{z} = g(\mathbf{z}')$ to be

a pushforward of $\mathbf{z}'$ using a deterministic structured mapping $g : \mathcal{Z}' \rightarrow \mathcal{Z}$. As such, there is no need to compute and regularise $\mathcal{D}(q_\phi(\mathbf{z})||p(\mathbf{z}))$ which can be challenging in practice. First, datapoints are encoded to $\mathbf{z}' \in \mathcal{Z}'$ —which crucially allows to deal with the stochasticity of the encoding process in this well-behaved latent space—but $\mathbf{z}$ is treated as the representation and is used by the decoder $p_\theta(\mathbf{x}|\mathbf{z})$. During generation, we start from the Gaussian variable $\mathbf{z}'$, then map it to $\mathbf{z}$ for further decoding. As such, the mapping g can be viewed as a shared layer between encoding and decoding processes, with the full auto-encoding process given by $\mathcal{X} \rightarrow \mathcal{Z}' \rightarrow \mathcal{Z} \rightarrow \mathcal{X}$. We introduce mappings g to induce multiply-connected, multi-modal, sparse, and hierarchical representations, and demonstrate their effectiveness on synthetic and real data.

Other approaches have been developed to induce richly structured dependencies in the learnt representations from VAEs (Johnson et al., 2016; Bouchacourt et al., 2018; Esmaili et al., 2019).

1.3 Geometry for probabilistic modelling

We now come back to the task of building well-specified deep probabilistic models. In what follows, we develop on how *geometry* can be leveraged as an inductive bias to express assumptions about the data-generating process. We first briefly introduce some background on geometry.

Geometry is one of the oldest branches of mathematics, and is concerned with the study of spaces that formalise concepts such as curves, distances, volumes and angles. For nearly two millennia, the word ‘geometry’ has been synonymous with *Euclidean geometry*, as formalised in Euclid’s *Elements*. In the 19th century, other types of geometries appeared, breaking away from Euclid’s parallel postulate, and eventually diverging into disparate fields. Several mathematicians led this effort, including Lobachevsky, Bolyai, Gauss, and Riemann. A particularly important result was Gauss’ *Theorema Egregium* which roughly asserts that the Gaussian curvature of a surface does not depend on a specific embedding. This has the impactful implication

that surfaces can be studied intrinsically— as stand alone spaces—which eventually expanded into the theory of smooth manifolds and Riemannian geometry.

Smooth manifolds can be seen as hyper-surfaces with no singularity, such as spheres and torii. Given a manifold $\mathcal{M}$, two objects are of particular interest: points which belong to the manifold $\mathbf{z} \in M$ and tangent vectors $\mathbf{v}$ which can be seen as directions in which one tangentially pass through $\mathbf{z}$. For a given point $\mathbf{z} \in M$, the union of all such possible directions forms a vector space referred as the tangent space $\mathcal{T}_{\mathbf{z}}\mathcal{M}$. Then in 1867, Bernhard Riemann sketched what would eventually become Riemannian geometry. He proposed to endow smooth manifolds with a *metric tensor*, that is an inner product defined on the tangent space at each point, $\mathbf{g}(\mathbf{z}) = \langle \cdot, \cdot \rangle_{\mathbf{z}} : \mathcal{T}_{\mathbf{z}}\mathcal{M} \times \mathcal{T}_{\mathbf{z}}\mathcal{M} \rightarrow \mathbb{R}$, and which varies smoothly from point to point. The metric tensor gives a *local* notion of angle, length and volume, from which *global* quantities can be derived by integrating local contributions. We refer the reader to [Lee \(2003\)](#) for an in-depth introduction to these concepts.

There is a rich history of coupling ideas from statistics and differential geometry so as to infer the latent structure of data. In particular, there has been a long focus in the probabilistic machine learning literature on the unsupervised problem of learning manifolds, or (non-linear) dimensionality reduction. This research can be traced back to canonical methods such as principal component analysis (PCA) ([Karl Pearson, 1901](#)) and multi-dimensional scaling ([Kruskal and Wish, 1978](#)). More recent endeavours include isometric feature mapping (ISOMAP) ([Tenenbaum et al., 2000](#)), local linear embedding ([Roweis and Saul, 2000](#)), Laplacian Eigenmaps ([Belkin and Niyogi, 2003](#)), Hessian Eigenmaps [Donoho and Grimes \(2003\)](#), diffusion maps ([Coifman and Lafon, 2006](#)) and t-distributed stochastic neighbour embedding ([van der Maaten and Hinton, 2008](#)).

These methods assume the existence of a lower dimensional manifold $\mathcal{M}$ that is embedded in a higher-dimensional Euclidean space, and on which data-points are (approximately) scattered. They each propose a parametrisation for this submanifold $\mathcal{M}$ and for the generative process $f : \mathcal{M} \mapsto \mathbb{R}^n$. For instance, PCA parametrises

$\mathcal{M}$ as a Euclidean space and f as a linear map, while Hessian Eigenmaps assume $\mathcal{M}$ to be locally isometric to an open, connected subset of Euclidean space $\mathbb{R}^d$ (with $d \leq n$). These dimensionality reduction techniques leverage flexible parametrisations in order to be suited to handle a large class of problems and data.

1.3.1 Geometric priors

Alternatively, one may have prior knowledge about the latent structure of data. In that case, geometry can be leveraged to encode inductive biases about the data, as a *geometric prior*. This can be achieved via formalisms such as the ones of metric space, Lie group, Riemannian manifold and other. A recent development in this line of work has been the trend of endowing VAEs’s latent space with non-Euclidean geometries. For instance, [Falorsi et al. \(2018\)](#) proposed to endow the latent space with an $SO(3)$ group structure. This can be motivated by a setting where one has access to a collection of pictures of a 3D object, which can be assumed to have been generated via a collection of different poses, formally being elements of the $SO(3)$ Lie group. Then, having an $SO(3)$ latent space encodes this assumption about the data generating process. Similarly, [Davidson et al. \(2018\)](#) leveraged a hyperspherical geometry with a reparametrisable von-Mises Fisher distribution.

Recently, hyperbolic spaces have been proposed as an alternative continuous approach to learn hierarchical representations from textual and graph-structured data ([Nickel and Kiela, 2017](#); [Tifrea et al., 2019](#)).

Hierarchical representations It can indeed be argued that in many domains, data should be represented *hierarchically*. For example, in cognitive sciences, it is widely accepted that human beings use a hierarchy to organise object categories (e.g. [Roy et al., 2006](#); [Collins and Quillian, 1969](#); [Keil, 1979](#)). In biology, the theory of evolution ([Darwin, 1859](#)) implies that features of living organisms are related in a hierarchical manner given by the evolutionary tree. Explicitly incorporating hierarchical structure in probabilistic models has unsurprisingly been a long-running research topic (e.g. [Duda et al., 2000](#); [Heller and Ghahramani, 2005](#); [Salakhutdinov](#)

et al., 2011; Teh et al., 2008; Ghahramani et al., 2010; Griffiths et al., 2004). As these approaches leverage trees, which are discrete structures, they are not amenable to deep learning approaches which rely on differentiability.

However, traditional VAEs (see Section 1.1.2) encode data in a Euclidean latent space which cannot efficiently embed *tree-like structures* (Bourgain, 1986). In contrast, *hyperbolic spaces* with negative curvature can (Sarkar, 2012). The Poincaré ball (Beltrami, 1868), a model of hyperbolic geometry, can indeed be thought of as continuous versions of trees, and vice versa. In particular, the exponential growth of the Poincaré volume with respect to its radius is analogous to the exponential growth of the number of leaves in a tree with respect to its depth. Therefore, in Chapter 3 we propose to endow VAEs with a Poincaré ball model of hyperbolic geometry as a latent space. To do so we derive the necessary methods to work with two main Gaussian generalisations on that space, and propose geometry-aware architectures for the encoder and decoder. We empirically show better generalisation to unseen data than the Euclidean counterpart, and that we can qualitatively and quantitatively better recover hierarchical structures.

1.3.2 Manifold-valued data

Most works that we have discussed so far implicitly assume that the data distribution $P_{\mathcal{D}}$ has support on a Euclidean space, i.e. $\mathcal{X} = \mathbb{R}^n$. Whilst this is a reasonable assumption for some settings, many problems are commonly described as objects living on Riemannian manifolds. It is indeed a natural approach to model tasks in many scientific fields ranging from earth and climate science to biology and computer vision. For instance, storm trajectories may be modelled as paths on the sphere (Karpatne et al., 2019), the shape of proteins can be parametrized using tori (Hamelryck et al., 2006), cell developmental processes can be described through paths in hyperbolic space (Klimovskaia et al., 2019), and human actions can be recognized in video using matrix manifolds (Lui, 2012). There has consequently been numerous statistical methods developed that deal with manifold-valued data. These include extensions of PCA (Fletcher et al., 2004; Zhang and Fletcher, 2013),

parametric families of distributions (Pennec, 2006; Debbasch and Chevalier, 2008; Chevallier et al., 2015; Jayasumana et al., 2015; Said et al., 2018), regression methods (Fletcher, 2011; Mallasto and Feragen, 2018) and inference techniques (Pourzanjani et al., 2020; Barp et al., 2019; Liu and Zhu, 2018).

However, conventional deep generative models such as normalising flows are not readily applicable to manifold-valued data since their implicit Euclidean assumption makes them unaware of the underlying geometry or borders of the manifold. As a result, they yield distributions that have some or all of their mass lying outside the manifold, rendering them ill-suited or even misspecified so that statistical divergences $\mathcal{D}(P_{\mathcal{D}}||P_{\theta})$ like the reverse KL divergence would not even be defined. In Chapter 4, we remedy this issue by proposing a principled way to parametrise flexible probability distributions on Riemannian manifolds. In particular, we build on neural ODE methods (Grathwohl et al., 2019; Chen et al., 2018b) and implicitly parametrised normalising flows as solutions to ordinary differential equations. We empirically demonstrate our method on synthetic and real spherical data and empirically show that it converges faster and fits better the data than previous approaches.

1.4 Summary

The remainder of this thesis is an integrated format, and therefore, each chapter is associated to a manuscript. The manuscripts and their associated chapters are summarised as follows.

1. **Chapter 2**: “Disentangling Disentanglement in Variational Autoencoders”, **Emile Mathieu**, Tom Rainforth, N. Siddharth and Yee Whye Teh. In Proceedings of the 36th International Conference on Machine Learning, 2019.
2. **Chapter 3** “Continuous Hierarchical Representations with Poincaré Variational Autoencoders”, **Emile Mathieu**, Charline Le Lan, Chris Maddison, Ryota Tomioka, Yee Whye Teh. Advances in Neural Information Processing Systems 32, 2019.

3. **Chapter 4** “Riemannian Continuous Normalising Flows”, **Emile Mathieu** and Maximilian Nickel. Advances in Neural Information Processing Systems 33, 2020.

1.5 Non-included work

Here we briefly discuss work that has been published or submitted to conferences but that have not been included in this manuscript for the sake of cohesiveness.

Probabilistic programming language with infinite dimensional variables

Let’s start with the paradigm of probabilistic programming language (PPL) (Goodman et al., 2008; Wood et al., 2014; Gordon et al., 2014; Ge et al., 2018) which aims to bring probabilistic modelling and probabilistic reasoning to a broader audience. PPL’s philosophy is to split the model definition from the inference scheme so that practitioners can easily define their model with a given syntax without having to implement any inference scheme. One such generic inference algorithm is *importance sampling*, which is implemented by running forward the model and evaluating the joint likelihood between latent and observed variables, yielding unnormalised importance weights. Running a model, or *program*, amounts to sampling from the prior distribution, which requires to store in memory random variables that have been sampled. One issue that immediately arises is the sampling of infinite dimensional objects such as random probability measures (RPMs). These are building blocks for Bayesian nonparametric models like the Dirichlet (Teh, 2011; Ferguson, 1973) or Pitman-Yor processes (Pitman and Yor, 1997; Ishwaran and James, 2003). We show that *lazily* instantiating RPMs is key. In particular, we prove in Bloem-Reddy et al. (2017) that a lazy initialization of the (prior) measure is *efficient* if and only if it is a *lazy* size-biased version⁸ of the RPM. In contrast, we

⁸This is a particular permutation of the RPM’s weights and associated atoms such that the weights are probabilistically and decreasingly ordered with respect to their value. This corresponds to the stick-breaking process for the Dirichlet Process. An initialisation is *lazy* when each atom is not instantiated in memory until the first time it is actually needed.

also show that a naive sampling scheme for the Pitman-Yor process would possibly require an infinite number of temporary variables. In parallel, we contributed to the open source probabilistic programming language [Turing.jl](#) (Ge et al., 2018) that now supports [RPM based models](#).

Random networks Still within the realm of Bayesian non parametrics, *random graph models* are useful tools for modelling uncertainty over network data. Many real networks are believed to be *sparse* and to feature *heavy-tailed* degree distributions (Newman, 2005; Clauset et al., 2009). Power law degree distributions, in which the proportion of vertices with degree $d \propto d^{-\eta}$, are often used as models for real degree distributions, and serve as a useful analytic tool for characterising the asymptotic properties of random network models. Models based on various forms of exchangeability are able to capture power laws with $\eta < 2$, and admit tractable inference algorithms. We show in Bloem-Reddy et al. (2018) that $\eta > 2$ cannot be generated by the forms of exchangeability used in existing random graph models. We design and implement inference algorithms for a recently proposed class of models, preferential attachment models (Bloem-Reddy and Orbanz, 2017), that generate η of all possible values. We show that although they are not exchangeable, these models have probabilistic structure amenable to inference. We demonstrate the models’ ability to capture the full range of power law exponents and sparsity level for a collection of real networks.

Illegal fishing We previously motivated PPLs as a means to bring probabilistic modelling to a broader audience of scientists. We provide such an example by assessing dynamic trends of the illegal fishing market in Chile with probabilistic modelling tools (Oyanedel et al., 2021). In particular, we build a dynamic microeconomic model of traders’ behaviour—traders being intermediaries between harvesters and consumers—so as to understand incentives traders face to trade legal or illegal products. In order to reflect our uncertainty about the model, we place prior

distributions on the model parameters such as price elasticities or law enforcement efficiency. With empirical data on legal fishing and law enforcement over several years for a small Chilean fishery, we infer the model parameters leveraging approximate Bayesian computation (ABC). We can then infer the amount of illegal fishing being traded on a weekly basis through a yearly cycle. Results show that trade is dominated by illegal units year-round, except towards the end of the year when there is a surplus of quota to selling legal fish. We additionally utilised the model to assess the effect of policy interventions to improve the fishery’s sustainability and explore the trade-offs between ecological, economic, and social goals. In particular, we observe that there exists a negative linear relationship between the fishery’s economic and ecological goals.

Learning representations of functions We now come back to the quest of learning representations from Section 1.2. In particular, learning representations for *stochastic processes* is an emerging problem in machine learning with applications ranging from meta-learning to physical object models and time series. A meta-learning classification task can indeed be formalised as having a collection of K related datasets $\{(x_1, f_k(x_1)), \dots, (x_n, f_k(x_n))\}$ for $k = 1, \dots, K$ associated to realisations $\{f_k\}_{k=1}^K$ sampled from a stochastic process with prior $p(f)$. Typical methods rely on exact reconstruction of observations (Kumar et al., 2019; Garnelo et al., 2018b,a), but this approach breaks down as observations become high-dimensional or noise distributions become complex. To address this, we propose in Mathieu et al. (2021) a unifying framework for learning contrastive representations of stochastic processes that does away with exact reconstruction. Informally, contrastive learning methods learn an encoder outputting representations so that samples $\{(x_1, f(x_1)), \dots, (x_C, f(x_C))\}$ from the same stochastic processes realisation $f \sim p(f)$ have encodings that are close in the representation space. Empirically, we show that our methods are effective for learning representations of periodic functions, 3D objects and dynamical processes.

2

Disentangling Disentanglement in Variational Autoencoders

Disentangling Disentanglement in Variational Autoencoders

Emile Mathieu^{*1} Tom Rainforth^{*1} N. Siddharth^{*2} Yee Whye Teh¹

Abstract

We develop a generalisation of disentanglement in variational autoencoders (VAEs)—*decomposition* of the latent representation—characterising it as the fulfilment of two factors: a) the latent encodings of the data having an appropriate level of overlap, and b) the aggregate encoding of the data conforming to a desired structure, represented through the prior. Decomposition permits disentanglement, i.e. explicit independence between latents, as a special case, but also allows for a much richer class of properties to be imposed on the learnt representation, such as sparsity, clustering, independent subspaces, or even intricate hierarchical dependency relationships. We show that the β -VAE varies from the standard VAE predominantly in its control of latent overlap and that for the standard choice of an isotropic Gaussian prior, its objective is invariant to rotations of the latent representation. Viewed from the decomposition perspective, breaking this invariance with simple manipulations of the prior can yield better disentanglement with little or no detriment to reconstructions. We further demonstrate how other choices of prior can assist in producing different decompositions and introduce an alternative training objective that allows the control of both decomposition factors in a principled manner.

1. Introduction

An oft-stated motivation for learning disentangled representations of data with deep generative models is a desire to achieve interpretability (Bengio et al., 2013; Chen et al., 2017)—particularly the *decomposability* (see §3.2.1 in Lip-ton, 2016) of latent representations to admit intuitive explanations. Most work has focused on capturing purely

independent factors of variation (Alemi et al., 2017; Ansari and Soh, 2019; Burgess et al., 2018; Chen et al., 2018; 2017; Eastwood and Williams, 2018; Esmaeili et al., 2019; Higgins et al., 2016; Kim and Mnih, 2018; Xu and Durrett, 2018; Zhao et al., 2017), typically evaluating this using purpose-built, synthetic data (Eastwood and Williams, 2018; Higgins et al., 2016; Kim and Mnih, 2018), whose generative factors are independent by construction.

This conventional view of disentanglement, as recovering independence, has subsequently motivated the development of formal evaluation metrics for independence (Eastwood and Williams, 2018; Kim and Mnih, 2018), which in turn has driven the development of objectives that target these metrics, often by employing regularisers explicitly encouraging independence in the representations (Eastwood and Williams, 2018; Esmaeili et al., 2019; Kim and Mnih, 2018).

We argue that such an approach is not generalisable, and potentially even harmful, to learning interpretable representations for more complicated problems, where such simplistic representations cannot accurately mimic the generation of high dimensional data from low dimensional latent spaces, and more richly structured dependencies are required.

We posit a generalisation of disentanglement in VAEs—*decomposing* their latent representations—that can help avoid such pitfalls. We characterise decomposition in VAEs as the fulfilment of two factors: a) the latent encodings of data having an appropriate level of overlap, and b) the aggregate encoding of data conforming to a desired structure, represented through the prior. We emphasize that neither of these factors is sufficient in isolation: without an appropriate level of overlap, encodings can degrade to a lookup table where the latents convey little information about data, and without the aggregate encoding of data following a desired structure, the encodings do not decompose as desired.

Disentanglement *implicitly* makes a choice of decomposition: that the latent features are independent of one another. We make this *explicit* and exploit it to both provide improvement to disentanglement through judicious choices of structure in the prior, and to introduce a more general framework flexible enough to capture alternate, more complex, notions of decomposition such as sparsity, clustering, hierarchical structuring, or independent subspaces.

^{*}Equal contribution ¹Department of Statistics ²Department of Engineering, University of Oxford. Correspondence to: Emile Mathieu <emile.mathieu@stats.ox.ac.uk>, Tom Rainforth <rainforth@stats.ox.ac.uk>, N. Siddharth <nsid@robots.ox.ac.uk>.

To connect our framework with existing approaches for encouraging disentanglement, we provide a theoretical analysis of the β -VAE (Alemi et al., 2018; 2017; Higgins et al., 2016), and show that it typically only allows control of latent overlap, the first decomposition factor. We show that it can be interpreted, up to a constant offset, as the standard VAE objective with its prior annealed as $p_\theta(z)^\beta$ and an additional maximum entropy regularization of the encoder that increases the stochasticity of the encodings. Specialising this result for the typical choice of a Gaussian encoder and isotropic Gaussian prior indicates that the β -VAE, up to a scaling of the latent space, is equivalent to the VAE plus a regulariser encouraging higher encoder variance. Moreover, this objective is invariant to rotations of the learned latent representation, meaning that it does not, on its own, encourage the latent variables to take on meaningful representations any more than an arbitrary rotation of them.

We confirm these results empirically, while further using our decomposition framework to show that simple manipulations to the prior can improve disentanglement, and other decompositions, with little or no detriment to the reconstruction accuracy. Further, motivated by our analysis, we propose an alternative objective that takes into account the distinct needs of the two factors of decomposition, and use it to learn clustered and sparse representations as demonstrations of alternative forms of decomposition. An implementation of our experiments and suggested methods is provided at <http://github.com/iffsid/disentangling-disentanglement>.

2. Background and Related Work

2.1. Variational Autoencoders

Let $\mathbf{x}$ be an $\mathcal{X}$ -valued random variable distributed according to an unknown generative process with density $p_{\mathcal{D}}(\mathbf{x})$ and from which we have observations, $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. The aim is to learn a latent-variable model $p_\theta(\mathbf{x}, \mathbf{z})$ that captures this generative process, comprising of a fixed¹ prior over latents $p(\mathbf{z})$ and a parametric likelihood $p_\theta(\mathbf{x}|\mathbf{z})$. Learning proceeds by minimising a divergence between the true data generating distribution and the model w.r.t θ , typically

$$\arg \min_{\theta} \text{KL}(p_{\mathcal{D}}(\mathbf{x}) \parallel p_\theta(\mathbf{x})) = \arg \max_{\theta} \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} [\log p_\theta(\mathbf{x})]$$

where $p_\theta(\mathbf{x}) = \int_{\mathcal{Z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$ is the marginal likelihood, or evidence, of datapoint $\mathbf{x}$ under the model, approximated by averaging over the observations.

However, estimating $p_\theta(\mathbf{x})$ (or its gradients) to any sufficient degree of accuracy is typically infeasible. A common strategy to ameliorate this issue involves the introduction of a parametric inference model $q_\phi(\mathbf{z}|\mathbf{x})$ to construct a varia-

tional evidence lower bound (ELBO) on $\log p_\theta(\mathbf{x})$ as follows

$$\begin{aligned} \mathcal{L}(\mathbf{x}; \theta, \phi) &\triangleq \log p_\theta(\mathbf{x}) - \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x})) \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})). \end{aligned} \quad (1)$$

A variational autoencoder (VAE) (Kingma and Welling, 2014; Rezende et al., 2014) views this objective from the perspective of a deep stochastic autoencoder, taking the inference model $q_\phi(\mathbf{z}|\mathbf{x})$ to be an encoder and the likelihood model $p_\theta(\mathbf{x}|\mathbf{z})$ to be a decoder. Here θ and ϕ are neural network parameters, and learning happens via stochastic gradient ascent (SGA) using unbiased estimates of $\nabla_{\theta, \phi} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\mathbf{x}_i; \theta, \phi)$. Note that when clear from the context, we denote $\mathcal{L}(\mathbf{x}; \theta, \phi)$ as simply $\mathcal{L}(\mathbf{x})$.

2.2. Disentanglement

Disentanglement, as typically employed in literature, refers to independence among features in a representation (Bengio et al., 2013; Eastwood and Williams, 2018; Higgins et al., 2018). Conceptually, however, it has a long history, far longer than we could reasonably do justice here, and is far from specific to VAEs. The idea stems back to traditional methods such as ICA (Hyvärinen and Oja, 2000; Yang and Amari, 1997) and conventional autoencoders (Schmidhuber, 1992), through to a range of modern approaches employing deep learning (Achille and Soatto, 2019; Chen et al., 2016; Cheung et al., 2014; Hjelm et al., 2019; Makhzani et al., 2015; Mathieu et al., 2016; Reed et al., 2014).

Of particular relevance to this work are approaches that explore disentanglement in the context of VAEs (Alemi et al., 2017; Chen et al., 2018; Esmaeili et al., 2019; Higgins et al., 2016; Kim and Mnih, 2018; Siddharth et al., 2017). Here one aims to achieve independence between the dimensions of the aggregate encoding, typically defined as $q_\phi(\mathbf{z}) \triangleq \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} [q(\mathbf{z}|\mathbf{x})] \approx \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}_i)$. The significance of $q_\phi(\mathbf{z})$ is that it is the marginal distribution induced on the latents by sampling a datapoint and then using the encoder to sample an encoding given that datapoint. It can thus informally be thought of as the pushforward distribution for “sampling” representations in the latent space.

Within the disentangled VAEs literature, there is also a distinction between unsupervised approaches, and semi-supervised approaches wherein one has access to the true generative factor values for some subset of data (Bouchacourt et al., 2018; Kingma et al., 2014; Siddharth et al., 2017). Our focus, however, is on the unsupervised setting.

Much of the prior work in the field has either implicitly or explicitly presumed a slightly more ambitious definition of disentanglement than considered above: that it is a measure of how well one captures *true* factors of variation (which happen to be independent by construction for synthetic data), rather than just independent factors. After all, if we wish

¹Learning the prior is possible, but omitted for simplicity.

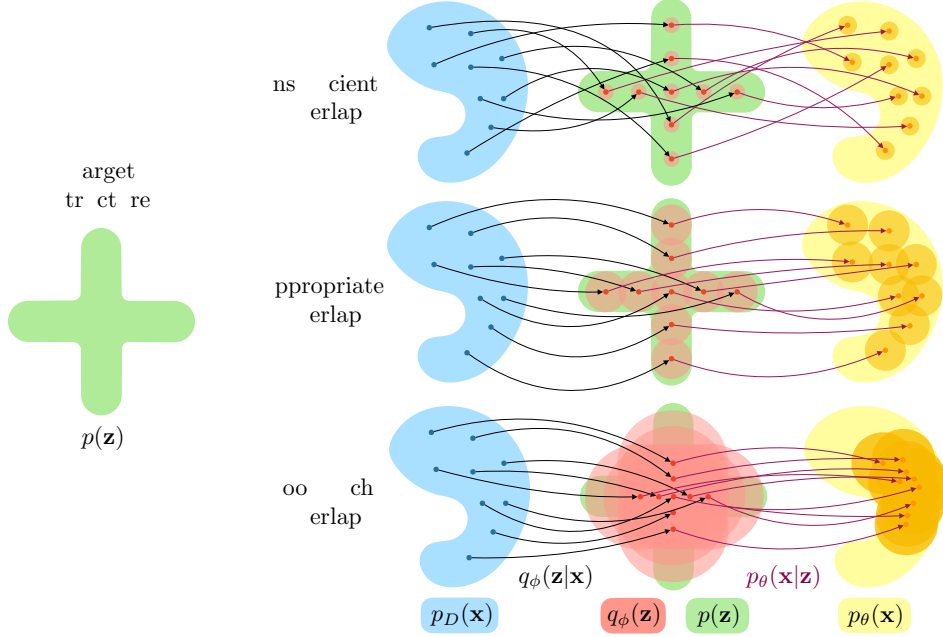


Figure 1. Illustration of decomposition where the desired structure is a cross shape (enforcing *sparsity*), expressed through the prior $p(\mathbf{z})$ as shown on the left. In the scenario where there is insufficient overlap [top], we observe a lookup table behavior: points that are close in the data space are not close in the latent space and so the latent space loses meaning. In the scenario where there is too much overlap [bottom], the latent variable and observed datapoint convey little information about one another, such that the latent space again loses meaning. Note that if the distributional form of the latent distribution does not match that of the prior, as is the case here, this can also prevent the aggregate encoding matching the prior when the level of overlap is large.

for our learned representations to be interpretable, it is necessary for the latent variables to take on clear-cut meaning.

One such definition is given by Eastwood and Williams (2018), who define it as the extent to which a latent dimension $d \in D$ in a representation predicts a true generative factor $k \in K$, with each latent capturing at most one generative factor. This implicitly assumes $D \geq K$, as otherwise the latents are unable to explain all the true generative factors. However, for real data, the association is more likely $D \ll K$, with one learning a low-dimensional abstraction of a complex process involving many factors. Consequently, such simplistic representations cannot, by definition, be found for more complex datasets that require more richly structured dependencies to be able to encode the information required to generate higher dimensional data. Moreover, for complex datasets involving a finite set of datapoints, it might not be reasonable to presume that one could capture the elements of the true generative process—the data itself might not contain sufficient information to recover these and even if it does, the computation required to achieve this through model learning is unlikely to be tractable.

The subsequent need for richly structured dependencies between latent dimensions has been reflected in the motivation for a handful of approaches (Bouchacourt et al., 2018; Esmaili et al., 2019; Johnson et al., 2016; Siddharth

et al., 2017) that explore this through graphical models, although employing mutually-inconsistent, and not generalisable, interpretations of disentanglement. This motivates our development of a decomposition framework as a means of extending beyond the limitations of disentanglement.

3. Decomposition: A Generalisation of Disentanglement

The commonly assumed notion of disentanglement is quite restrictive for complex models where the true generative factors are not independent, very large in number, or where it cannot be reasonably assumed that there is a well-defined set of “true” generative factors (as will be the case for many, if not most, real datasets). To this end, we introduce a generalization of disentanglement, *decomposition*, which at a high-level can be thought of as imposing a desired structure on the learned representations. This permits disentanglement as a special case, for which the desired structure is that $q_\phi(\mathbf{z})$ factors along its dimensions.

We characterise the decomposition of latent spaces in VAEs to be the fulfilment of two factors (as shown in Figure 1):

- An “appropriate” level of overlap in the latent space—ensuring that the range of latent values capable of encoding

ing a particular datapoint is neither too small, nor too large. This is, in general, dictated by the level of stochasticity in the encoder: the noisier the encoding process is, the higher the number of datapoints which can plausibly give rise to a particular encoding.

- b. The aggregate encoding $q_\phi(\mathbf{z})$ matching the prior $p(\mathbf{z})$, where the latter expresses the desired dependency structure between latents.

The overlap factor (a) is perhaps best understood by considering extremes—too little, and the latents effectively become a lookup table; too much, and the data and latents do not convey information about each other. In either case, meaningfulness of the latent encodings is lost. Thus, without the *appropriate* level of overlap—dictated both by noise in the true generative process and dataset size—it is not possible to enforce meaningful structure on the latent space. Though quantitatively formalising overlap in general scenarios is surprisingly challenging (c.f. § 7 and Appendix D), we note for now that when the encoder distribution is unimodal, it is typically well-characterized by the mutual information between the data and the latents $I(\mathbf{x}; \mathbf{z})$.

The regularisation factor (b) enforces a congruence between the (aggregate) latent embeddings of data and the dependency structures expressed in the prior. We posit that such structure is best expressed in the prior, as opposed to explicit independence regularisation of the marginal posterior (Chen et al., 2018; Kim and Mnih, 2018), to enable the *generative* model to express the desired decomposition, and to avoid potentially violating self-consistency between the encoder, decoder, and true data generating distributions. The prior also provides a rich and flexible means of expressing desired structure by defining a generative process that encapsulates dependencies between variables, as with a graphical model.

Critically, *neither factor is sufficient in isolation*. An inappropriate level of overlap in the latent space will impede interpretability, irrespective of quality of regularisation, as the latent space need not be meaningful. Conversely, without the pressure to regularise to the prior, the latent space is under no constraint to exhibit the desired structure.

Decomposition is inherently subjective as we must choose the structure of the prior we regularise to depending on how we intend to use our learned model or what kind of features we would like to uncover from the data. This may at first seem unsatisfactory compared to the seemingly objective adjustments often made to the ELBO by disentanglement methods. However, disentanglement *is itself* a subjective choice for the decomposition. We can embrace this subjective nature through judicious choices of the prior distribution; ignoring this imposes unintended assumptions which can have unwanted effects. For example, as we will later show, the rotational invariance of the standard prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, I)$ can actually hinder disentanglement.

4. Deconstructing the β -VAE

To connect existing approaches to our proposed framework, we now consider, as a case study, the β -VAE (Higgins et al., 2016)—an adaptation of the VAE objective (ELBO) to learn better-disentangled representations. Specifically, it scales the KL term in the standard ELBO by a factor $\beta > 0$ as

$$\mathcal{L}_\beta(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})). \quad (2)$$

Hoffman et al. (2017) showed that the β -VAE target can be viewed as a standard ELBO with the alternative prior $r(\mathbf{z}) \propto q_\phi(\mathbf{z})^{(1-\beta)} p(\mathbf{z})^\beta$, along with terms involving the mutual information and the prior’s normalising constant.

We now introduce an alternate deconstruction as follows

Theorem 1. *The β -VAE target $\mathcal{L}_\beta(\mathbf{x})$ can be interpreted in terms of the standard ELBO, $\mathcal{L}(\mathbf{x}; \pi_{\theta,\beta}, q_\phi)$, for an adjusted target $\pi_{\theta,\beta}(\mathbf{x}, \mathbf{z}) \triangleq p_\theta(\mathbf{x} | \mathbf{z}) f_\beta(\mathbf{z})$ with annealed prior $f_\beta(\mathbf{z}) \triangleq p(\mathbf{z})^\beta / F_\beta$ as*

$$\mathcal{L}_\beta(\mathbf{x}) = \mathcal{L}(\mathbf{x}; \pi_{\theta,\beta}, q_\phi) + (\beta - 1)H_{q_\phi} + \log F_\beta \quad (3)$$

where $F_\beta \triangleq \int_{\mathbf{z}} p(\mathbf{z})^\beta d\mathbf{z}$ is constant given β , and H_{q_ϕ} is the entropy of $q_\phi(\mathbf{z} | \mathbf{x})$.

Proof. All proofs are given in Appendix A. $\square$

Clearly, the second term in (3), enforcing a maximum entropy regulariser on the posterior $q_\phi(\mathbf{z} | \mathbf{x})$, allows the value of β to affect the overlap of encodings in the latent space. We thus see that it provides a means of controlling decomposition factor (a). However, it is itself not sufficient to enforce disentanglement. For example, the entropy of $q_\phi(\mathbf{z} | \mathbf{x})$ is independent of its mean $\mu_\theta(\mathbf{x})$ and is independent to rotations of $\mathbf{z}$, so it is clearly incapable of discouraging certain representations with poor disentanglement. All the same, having the wrong level of regularization can, in turn, lead to an inappropriate level of overlap and undermine the ability to disentangle. Consequently, this term is still important.

Although the precise impact of prior annealing depends on the original form of the prior, the high-level effect is the same—larger values of β cause the effective latent space to collapse towards the modes of the prior. For uni-modal priors, the main effect of annealing is to reduce the scaling of $\mathbf{z}$; indeed this is the only effect for generalized Gaussian distributions. While this would appear not to have any tangible effects, closer inspection suggests otherwise—it ensures that the scaling of the encodings matches that of the prior. Only incorporating the maximum-entropy regularisation will simply cause the scaling of the latent space to increase. The rescaling of the prior now cancels this effect, ensuring the scaling of $q_\phi(\mathbf{z})$ matches that of $p(\mathbf{z})$.

Taken together, this implies that the β -VAE’s ability to encourage disentanglement is predominantly through *direct*

control over the level of overlap. It places no other direct constraint on the latents to disentangle (although in some cases, the annealed prior may inadvertently encourage better disentanglement), but instead helps avoid the pitfalls of inappropriate overlap. Amongst other things, this explains why large β is not universally beneficial for disentanglement, as the level of overlap can be increased too far.

4.1. Special Case – Gaussians

We can gain further insights into the β -VAE in the common use case—assuming a Gaussian prior, $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \Sigma)$, and Gaussian encoder, $q_\phi(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), S_\phi(\mathbf{x}))$. Here it is straightforward to see that annealing simply scales the latent space by $1/\sqrt{\beta}$, i.e. $f_\beta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \Sigma/\beta)$. Given this, it is easy to see that a VAE trained with the adjusted target $\mathcal{L}(\mathbf{x}; \pi_{\theta,\beta}, q_\phi)$, but appropriately scaling the latent space, will behave identically to one trained with the original target $\mathcal{L}(\mathbf{x})$. It will also have an identical ELBO as the expected reconstruction is trivially the same, while the KL between Gaussians is invariant to scaling both equally. More precisely, we have the following result.

Corollary 1. *If $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \Sigma)$ and $q_\phi(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), S_\phi(\mathbf{x}))$, then,*

$$\mathcal{L}_\beta(\mathbf{x}; \theta, \phi) = \mathcal{L}(\mathbf{x}; \theta', \phi') + \frac{(\beta - 1)}{2} \log |S_{\phi'}(\mathbf{x})| + c \quad (4)$$

where θ' and ϕ' represent rescaled networks such that

$$\begin{aligned} p_{\theta'}(\mathbf{x} | \mathbf{z}) &= p_\theta(\mathbf{x} | \mathbf{z}/\sqrt{\beta}), \\ q_{\phi'}(\mathbf{z} | \mathbf{x}) &= \mathcal{N}(\mathbf{z}; \mu_{\phi'}(\mathbf{x}), S_{\phi'}(\mathbf{x})), \\ \mu_{\phi'}(\mathbf{x}) &= \sqrt{\beta} \mu_\phi(\mathbf{x}), \quad S_{\phi'}(\mathbf{x}) = \beta S_\phi(\mathbf{x}), \end{aligned}$$

and $c \triangleq \frac{D(\beta-1)}{2} \left(1 + \log \frac{2\pi}{\beta}\right) + \log F_\beta$ is a constant, with D denoting the dimensionality of $\mathbf{z}$.

Noting that as c is irrelevant to the training process, this indicates an equivalence, up to scaling of the latent space, between training with the β -VAE objective and a maximum-entropy regularised version of the standard ELBO

$$\mathcal{L}_{H,\beta}(\mathbf{x}) \triangleq \mathcal{L}(\mathbf{x}) + \frac{(\beta - 1)}{2} \log |S_\phi(\mathbf{x})|, \quad (5)$$

whenever $p(\mathbf{z})$ and $q_\phi(\mathbf{z} | \mathbf{x})$ are Gaussian. Note that we implicitly presume suitable adjustment of neural-network hyper-parameters and the stochastic gradient scheme to account for the change of scaling in the optimal networks.

Moreover, the stationary points for the two objectives $\mathcal{L}_\beta(\mathbf{x}; \theta, \phi)$ and $\mathcal{L}_{H,\beta}(\mathbf{x}; \theta', \phi')$ are equivalent (c.f. Corollary 2 in Appendix A), indicating that optimising for (5) leads to networks equivalent to those from optimising the β -VAE objective (2), up to scaling the encodings by a factor of

$\sqrt{\beta}$. Under the isotropic Gaussian prior setting, we further have the following result showing that the β -VAE objective is invariant to rotations of the latent space.

Theorem 2. *If $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \sigma I)$ and $q_\phi(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), S_\phi(\mathbf{x}))$, then for all rotation matrices R ,*

$$\mathcal{L}_\beta(\mathbf{x}; \theta, \phi) = \mathcal{L}_\beta(\mathbf{x}; \theta^\dagger(R), \phi^\dagger(R)) \quad (6)$$

where $\theta^\dagger(R)$ and $\phi^\dagger(R)$ are transformed networks such that

$$\begin{aligned} p_{\theta^\dagger}(\mathbf{x} | \mathbf{z}) &= p_\theta(\mathbf{x} | R^T \mathbf{z}), \\ q_{\phi^\dagger}(\mathbf{z} | \mathbf{x}) &= \mathcal{N}(\mathbf{z}; R \mu_\phi(\mathbf{x}), R S_\phi(\mathbf{x}) R^T). \end{aligned}$$

This shows that the β -VAE objective does not directly encourage latent variables to take on meaningful representations when using the standard choice of an isotropic Gaussian prior. In fact, on its own, it encourages latent representations which match the true generative factors no more than it encourages *any arbitrary rotation* of these factors, with such rotations capable of exhibiting strong correlations between latents. This view is further supported by our empirical results (see Figure 2), where we did not observe any gains in disentanglement (using the metric from Kim and Mnih (2018)) from increasing $\beta > 0$ with an isotropic Gaussian prior trained on the 2D Shapes dataset (Matthey et al., 2017). It may also go some way to explaining the extremely high levels of variation we found in the disentanglement-metric scores between different random seeds at train time.

It should be noted, however, that the value of β can indirectly influence the level of disentanglement when using a mean-field assumption for the encoder distribution (i.e. restricting $S_\phi(\mathbf{x})$ to be diagonal). As noted by Rolinek et al. (2018); Stühmer et al. (2019), increasing β can reinforce existing inductive biases, wherein mean-field assumptions encourage representations which reduce dependence between the latent dimensions (Turner and Sahani, 2011).

5. An Objective for Enforcing Decomposition

Given the characterisation set out above, we now develop an objective that incorporates the effect of both factors (a) and (b). Our analysis of the β -VAE tells us that its objective allows direct control over the level of overlap, i.e. factor (a). To incorporate direct control over the regularisation (b) between the marginal posterior and the prior, we add a divergence term $\mathbb{D}(q_\phi(\mathbf{z}), p(\mathbf{z}))$, yielding

$$\begin{aligned} \mathcal{L}_{\alpha,\beta}(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} [\log p_\theta(\mathbf{x} | \mathbf{z})] \\ &\quad - \beta \text{KL}(q_\phi(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z})) - \alpha \mathbb{D}(q_\phi(\mathbf{z}), p(\mathbf{z})) \end{aligned} \quad (7)$$

allowing control over how much factors (a) and (b) are enforced, through appropriate setting of β and α respectively.

Note that such an additional term has been previously considered by Kumar et al. (2017), with $\mathbb{D}(q_\phi(\mathbf{z}), p(\mathbf{z})) =$

$\text{KL}(q_\phi(\mathbf{z}) \parallel p(\mathbf{z}))$, although for the sake of tractability they rely instead on moment matching using covariances. There have also been a number of approaches that decompose the standard VAE objective in different ways (e.g. Dilokthanakul et al., 2019; Esmaili et al., 2019; Hoffman and Johnson, 2016) to expose $\text{KL}(q_\phi(\mathbf{z}) \parallel p(\mathbf{z}))$ as a component, but, as we discuss in Appendix C, this can be difficult to compute correctly in practice, with common approaches leading to highly biased estimates whose practical behaviour is very different than the divergence they are estimating, unless very large batch sizes are used.

Wasserstein Auto-Encoders (Tolstikhin et al., 2018) formulate an objective that includes a general divergence term between the prior and marginal posterior, computed using either maximum mean discrepancy (MMD) or a variational formulation of the Jensen-Shannon divergence (a.k.a GAN loss). However, we find that empirically, choosing the MMD’s kernel and numerically stabilising its U-statistics estimator to be tricky, and designing and learning a GAN to be cumbersome and unstable. Consequently, the problems of choosing an appropriate $\mathbb{D}(q_\phi(\mathbf{z}), p(\mathbf{z}))$ and generating reliable estimates for this choice are tightly coupled, with a general purpose solution remaining an important open problem; see further discussion in Appendix C.

6. Experiments

6.1. Prior for Axis-Aligned Disentanglement

We first show how subtle changes to the prior distribution can yield improvements in disentanglement. The standard choice of an isotropic Gaussian has previously been justified by the correct assertion that the latents are independent under the prior (Higgins et al., 2016). However, as explained in § 4.1, the rotational invariance of this prior means that it does not directly encourage axis-aligned representations. Priors that break this rotational invariance should be better suited for learning disentangled representations. We assess this hypothesis by training a β -VAE (i.e. (7) with $\alpha = 0$) on the *2D Shapes* dataset (Matthey et al., 2017) and evaluating disentanglement using the metric of Kim and Mnih (2018).

Figure 2 demonstrates that notable improvements in disentanglement can be achieved by using non-isotropic priors: for a given reconstruction loss, implicitly fixed by β , non-isotropic Gaussian priors got better disentanglement scores, with further improvement achieved when the prior variance is learnt. With a product of Student-t priors $p_\nu(\mathbf{z})$ (noting $p_\nu(\mathbf{z}) \rightarrow \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ as $\nu \rightarrow \infty$), reducing ν only incurred a minor reconstruction penalty, for improved disentanglement. Interestingly, very low values of ν caused the disentanglement score to drop again (though still giving higher values than the Gaussian). We speculate that this may be related to the effect of heavy tails on the disentanglement metric itself,

rather than being an objectively worse disentanglement. Another interesting result was that for an isotropic Gaussian prior, as per the original β -VAE setup, no gains at all were achieved in disentanglement by increasing β .

6.2. Clustered Prior

We next consider an alternative decomposition one might wish to impose—*clustering* of the latent space. For this, we use the “pinwheels” dataset from (Johnson et al., 2016) and a mixture of four equally-weighted Gaussians as our prior. We then conduct an ablation study to observe the effect of varying α and β in $\mathcal{L}_{\alpha,\beta}(\mathbf{x})$ (as per (7)) on the learned representations, taking the divergence to be $\text{KL}(p(\mathbf{z}) \parallel q_\phi(\mathbf{z}))$ (see Appendix B for details).

We see in Figure 3 that increasing β increases the level of overlap in $q_\phi(\mathbf{z})$, as a consequence of increasing the encoder variance for individual datapoints. When β is too large, the encoding of a datapoint loses meaning. Also, as a single datapoint encodes to a Gaussian distribution, $q_\phi(\mathbf{z}|\mathbf{x})$ is unable to match $p(\mathbf{z})$ exactly. Because $q_\phi(\mathbf{z}|\mathbf{x}) \rightarrow q_\phi(\mathbf{z})$ when $\beta \rightarrow \infty$, this in turn means that overly large values of β actually cause a mismatch between $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$ (see top right of Figure 3). Increasing α , instead always improved the match between $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$. Here, the finiteness of the dataset and the choice of divergence results in an increase in overlap with increasing α , but only up to the level required for a non-negligible overlap between the nearby datapoints: large values of α did not cause the encodings to collapse to a mode.

6.3. Prior for Sparsity

Finally, we consider a commonly desired decomposition—sparsity, which stipulates that only a small fraction of available factors are employed. That is, a *sparse representation* (Olshausen and Field, 1996) can be thought of as one where each embedding has a significant proportion of its dimensions *off*, i.e. close to 0. Sparsity has often been considered for feature-learning (Coates and Ng, 2011; Larochelle and Bengio, 2008) and employed in the probabilistic modelling literature (Lee et al., 2007; Ranzato et al., 2007).

Common ways to achieve sparsity are through a specific penalty (e.g. l_1) or a careful choice of prior (peaked at 0). Concomitant with our overarching desire to encode requisite structure in the prior, we adopt the latter, constructing a sparse prior as $p(\mathbf{z}) = \prod_d (1 - \gamma) \mathcal{N}(z_d; 0, 1) + \gamma \mathcal{N}(z_d; 0, \sigma_0^2)$ with $\sigma_0^2 = 0.05$. This mixture distribution can be interpreted as a mixture of samples being either *off* or *on*, whose proportion is set by the weight parameter γ . We use this prior to learn a VAE for the *Fashion-MNIST* dataset (Xiao et al., 2017) using the objective $\mathcal{L}_{\alpha,\beta}(\mathbf{x})$ (as per (7)), taking the divergence to be an MMD with a kernel that only considers difference between the marginal distri-

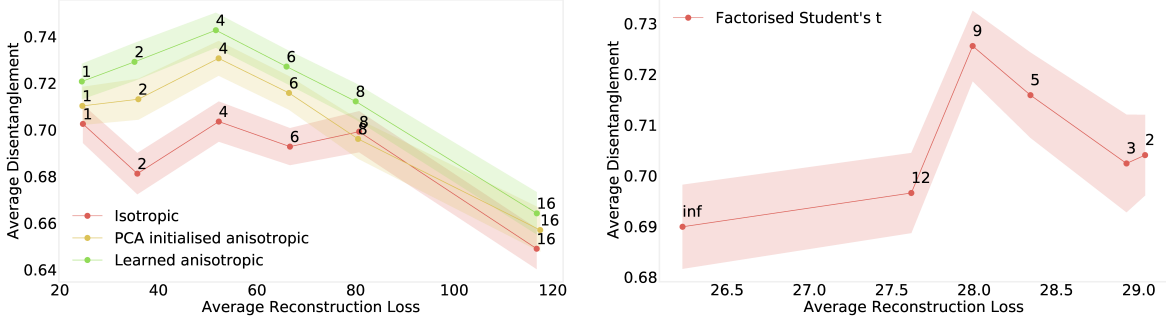


Figure 2. Reconstruction loss vs disentanglement metric of Kim and Mnih (2018). [Left] Using an anisotropic Gaussian with diagonal covariance either learned, or fixed to principal-component values of the dataset. Point labels represent different values of β . [Right] Using $p_\nu(\mathbf{z}) = \prod_d \text{STUDENT-T}(z_d; \nu)$ for different ν with $\beta = 1$. Note the different x-axis scaling. Shaded areas represent ± 2 standard errors for estimated mean disentanglement calculated using 100 separately trained networks. We thus see that the variability on the disentanglement metric is very large, presumably because of stochasticity in whether learned dimensions correspond to true generative factors. The variability in the reconstruction was only negligible and so is not shown. See Appendix B for full experimental details.

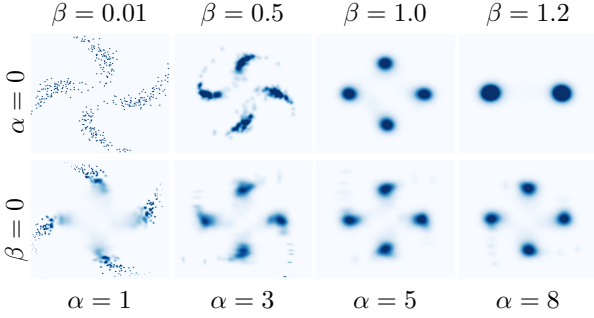


Figure 3. Density of aggregate posterior $q_\phi(\mathbf{z})$ with different α, β for spirals dataset with a mixture of Gaussian prior.

butions (see Appendix B for details).

We measure a representation’s sparsity using the *Hoyer* extrinsic metric (Hurley and Rickard, 2008). For $\mathbf{y} \in \mathbb{R}^d$,

$$\text{Hoyer}(\mathbf{y}) = \frac{\sqrt{d} - \|\mathbf{y}\|_1 / \|\mathbf{y}\|_2}{\sqrt{d} - 1} \in [0, 1],$$

yielding 0 for a fully dense vector and 1 for a fully sparse vector. Rather than employing this metric directly to the mean encoding of each datapoint, we first normalise each dimension to have a standard deviation of 1 under its aggregate distribution, i.e. we use $\bar{z}_d = z_d / \sigma(z_d)$ where $\sigma(z_d)$ is the standard deviation of dimension d of the latent encoding taken over the dataset. This normalisation is important as one could achieve a “sparse” representation simply by having different dimensions vary along different length scales (something the β -VAE encourages through its pruning of dimensions (Stühmer et al., 2019)), whereas we desire a representation where different datapoints “activate” different features. We then compute overall sparsity by averaging over the dataset as $\text{Sparsity} = \frac{1}{n} \sum_i^n \text{Hoyer}(\bar{\mathbf{z}}_i)$. Figure 4

(left) shows that substantial sparsity can be gained by replacing a Gaussian prior ($\gamma = 0$) by a sparse prior ($\gamma = 0.8$). It further shows substantial gains from the inclusion of the aggregate posterior regularization, with $\alpha = 0$ giving far low sparsity than $\alpha > 0$, when using our sparse prior. The use of our sparse prior did not generally harm the reconstruction compared. Large values of α did slightly worsen the reconstruction, but this drop-off was much slower than increases in β (note that α is increased to much higher levels than β). Interestingly, we see that β being either too low or too high also harmed the sparsity.

We explore the qualitative effects of sparsity in Figure 5, using a network trained with $\alpha = 1000, \beta = 1$, and $\gamma = 0.8$, corresponding to one of the models in Figure 4 (left). The top plot shows the average encoding magnitude for data corresponding to 3 of the 10 classes in the *Fashion-MNIST* dataset. It clearly shows that the different classes (trousers, dress, and shirt) predominantly encode information along different sets of dimensions, as expected for sparse representations (c.f. Appendix B for plots for all classes). For each of these classes, we explore the latent space along a particular ‘active’ dimension—one with high average encoding magnitude—to observe if they capture meaningful features in the image. We first identify a suitable ‘active’ dimension for a given instance (top row) from the dimension-wise magnitudes of its encoding, by choosing one, say d , where the magnitude far exceeds σ_0^2 . Given encoding value z_d , we then interpolate along this dimension (keeping all others fixed) in the range $(z_d, z_d + \text{sign}(z_d))$; the sign of z_d indicating the direction of interpolation. Exploring the latent space in such a manner demonstrates a variety of consistent feature transformations in the image, both within class (a, b, c), and across classes (d), indicating that these sparse dimensions do capture meaningful features in the image.

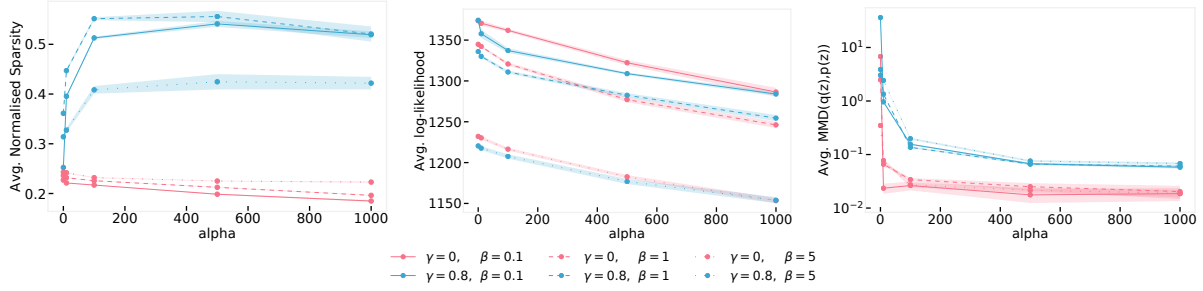


Figure 4. [Left] Sparsity vs regularisation strength α (c.f. (7), high better). [Center] Average reconstruction log-likelihood $\mathbb{E}_{p_D(\mathbf{x})}[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]]$ vs α (higher better). [Right] Divergence (MMD) vs α (lower better). Note here that the different values of γ represent regularizations to different distributions, with regularization to a Gaussian (i.e. $\gamma = 0$) much easier to achieve than the sparse prior, hence the lower divergence. Shaded areas represent ± 2 standard errors in the mean estimate calculated using 8 separately trained networks. See Appendix B for full experimental details.

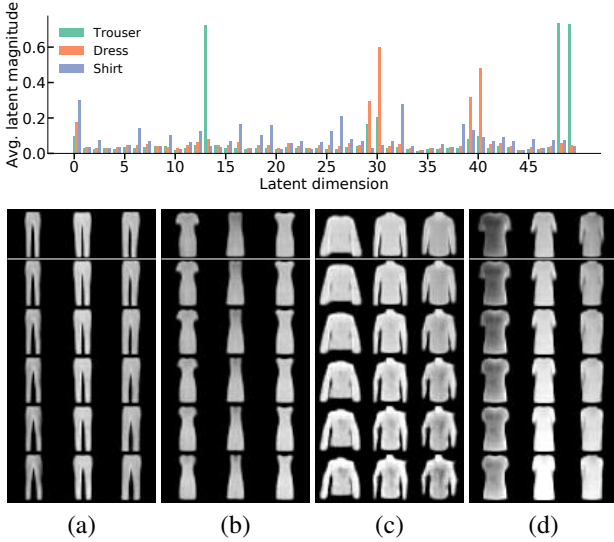


Figure 5. Qualitative evaluation of sparsity. [Top] Average encoding magnitude over data for three example classes in *Fashion-MNIST*. [Bottom] Latent interpolation ($\downarrow$) for different datapoints (top layer) along particular ‘active’ dimensions. (a) Separation between the trouser legs (dim 49). (b) Top/Collar width of dresses (dim 30). (c) Shirt shape (loose/fitted, dim 19). (d) Style of sleeves across different classes—t-shirt, dress, and coat (dim 40).

Concurrent to our work, Tonolini et al. (2019) also considered imposing sparsity in VAEs with a spike-slab prior (such that $\sigma_0 \rightarrow 0$). In contrast to our work, they do not impose a constraint on the aggregate encoder, nor do they evaluate their results with a quantitative sparsity metric that accounts for the varying length scales of different latent dimensions

7. Discussion

Characterising Overlap Precisely formalising what constitutes the level of overlap in the latent space is surprisingly

subtle. Prior work has typically instead considered controlling the level of compression through the mutual information between data and latents $I(\mathbf{x}; \mathbf{z})$ (Alemi et al., 2018; 2017; Hoffman and Johnson, 2016; Phuong et al., 2018), with, for example, (Phuong et al., 2018) going on to discuss how controlling the compression can “explicitly encourage useful representations.” Although $I(\mathbf{x}; \mathbf{z})$ provides a perfectly serviceable characterisation of overlap in a number of cases, the two are not universally equivalent and we argue that it is the latter which is important in achieving useful representations. In particular, if the form of the encoding distribution is not fixed—as when employing normalising flows, for example— $I(\mathbf{x}; \mathbf{z})$ does not necessarily characterise overlap well. We discuss this in greater detail in Appendix D.

However, when the encoder is unimodal with fixed form (in particular the tail behaviour is fixed) and the prior is well-characterised by Euclidean distances, then these factors have a substantially reduced ability to vary for a given $I(\mathbf{x}; \mathbf{z})$, which subsequently becomes a good characterisation of the level of overlap. When $q_\phi(\mathbf{z}|\mathbf{x})$ is Gaussian, controlling the variance of $q_\phi(\mathbf{z}|\mathbf{x})$ (with a fixed $q_\phi(\mathbf{z})$) should similarly provide an effective means of achieving the desired overlap behaviour. As this is the most common use case, we leave the development of more a general definition of overlap to future work, simply noting that this is an important consideration when using flexible encoder distributions.

Can VAEs Uncover True Generative Factors? In concurrently published work, Locatello et al. (2019) question the plausibility of learning unsupervised disentangled representations with meaningful features, based on theoretical analyses showing an equivalence class of generative models, many members of which could be entangled. Though their analysis is sound, we posit a counterargument to their conclusions, based on the *stochastic* nature of the encodings used during training. Namely, that this stochasticity means that they need not give rise to the same ELBO scores (an

important exception is the rotational invariance for isotropic Gaussian priors). Essentially, the encoding noise forces nearby encodings to relate to similar datapoints, while standard choices for the likelihood distribution (e.g. assuming conditional independence) ensure that information is stored in the encodings, not just in the generative network. These restrictions mean that the ELBO prefers smooth representations and, provided the prior is not rotationally invariant, means that there no longer need be a class of different representations with the same ELBO; simpler representations are preferred to more complex ones.

The exact form of the encoding distribution is also important here. For example, imagine we restrict the encoder variance to be isotropic and then use a two dimensional prior where one latent dimension has a much larger variance than the other. It will be possible to store more information in the prior dimension with higher variance (as we can spread points out more relative to the encoder variance). Consequently, that dimension is more likely to correspond to an important factor of the generative process than the other. Of course, this does not imply that this is a true factor of variation in the generative process, but neither is the meaning that can be attributed to each dimension completely arbitrary.

All the same, we agree that an important area for future work is to assess when, and to what extent, one might expect learned representations to mimic the true generative process, and, critically, when it should not. For this reason, we actively avoid including any notion of a true generative process in our definition of decomposition, but note that, analogously to disentanglement, it permits such extension in scenarios where doing so can be shown to be appropriate.

8. Conclusions

In this work, we explored and analysed the fundamental characteristics of learning disentangled representations, and showed how these can be generalised to a more general framework of *decomposition* (Lipton, 2016). We characterised the learning of decomposed latent representation with VAEs in terms of the control of two factors: i) overlap in the latent space between encodings of different datapoints, and ii) regularisation of the aggregate encoding distribution to the given prior, which encodes the structure one would wish for the latent space to have.

Connecting prior work on disentanglement to this framework, we analysed the β -VAE objective to show that its contribution to disentangling is primarily through direct control of the level of overlap between encodings of the data, expressed by maximising the entropy of the encoding distribution. In the commonly encountered case of assuming an isotropic Gaussian prior and an independent Gaussian posterior, we showed that control of overlap is the *only* effect of the β -VAE. Motivated by this observation, we

developed an alternate objective for the ELBO that allows control of the two factors of decomposability through an additional regularisation term. We then conducted empirical evaluations using this objective, targeting alternate forms of decompositions such as clustering and sparsity, and observed the effect of varying the extent of regularisation to the prior on the quality of the resulting clustering and sparseness of the learnt embeddings. The results indicate that we were successful in attaining those decompositions.

Acknowledgements

EM, TR, YWT were supported in part by the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no. 617071. TR research leading to these results also received funding from EPSRC under grant EP/P026753/1. EM was also supported by Microsoft Research through its PhD Scholarship Programme. NS was funded by EPSRC/MURI grant EP/N019474/1.

References

- Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50), 2019.
- Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken ELBO. In *International Conference on Machine Learning*, pages 159–168, 2018.
- Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.
- Abdul Fatir Ansari and Harold Soh. Hyperprior induced unsupervised disentanglement of latent representations. In *AAAI Conference on Artificial Intelligence*, 2019.
- Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, August 2013. ISSN 0162-8828.
- Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *AAAI Conference on Artificial Intelligence*, 2018.
- Christopher P. Burgess, Irina Higgins, Arka Pal, Loïc Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β -vae. *CoRR*, abs/1804.03599, 2018.

- Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018.
- Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.
- Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational Lossy Autoencoder. 2017.
- Brian Cheung, Jesse A Livezey, Arjun K Bansal, and Bruno A Olshausen. Discovering hidden factors of variation in deep networks. *arXiv preprint arXiv:1412.6583*, 2014.
- Adam Coates and Andrew Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. In Lise Getoor and Tobias Scheffer, editors, *ICML*, pages 921–928. Omnipress, 2011.
- Nat Dilokthanakul, Nick Pawlowski, and Murray Shanahan. Explicit information placement on latent variables using auxiliary generative modelling task, 2019. URL <https://openreview.net/forum?id=H1l-SjA5t7>.
- Justin Domke and Daniel Sheldon. Importance weighting and variational inference. In *Advances in Neural Information Processing Systems*, pages 4471–4480, 2018.
- Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.
- Babak Esmaeili, Hao Wu, Sarthak Jain, N Siddharth, Brooks Paige, and Jan-Willem van de Meent. Hierarchical Disentangled Representations. *Artificial Intelligence and Statistics*, 2019.
- Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations*, 2016.
- Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.
- R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.
- Matthew D Hoffman and Matthew J Johnson. ELBO surgery: yet another way to carve up the variational evidence lower bound. In *Workshop on Advances in Approximate Bayesian Inference, NIPS*, pages 1–4, 2016.
- Matthew D Hoffman, Carlos Riquelme, and Matthew J Johnson. The β -VAE’s Implicit Prior. In *Workshop on Bayesian Deep Learning, NIPS*, pages 1–5, 2017.
- Niall P. Hurley and Scott T. Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(4-5):4723–4741, 2008.
- Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.
- Matthew Johnson, David K Duvenaud, Alex Wiltchko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, pages 2946–2954. 2016.
- Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, 2018.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.
- Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, 2014.
- Soheil Kolouri, Phillip E. Pope, Charles E. Martin, and Gustavo K. Rohde. Sliced wasserstein auto-encoders. In *International Conference on Learning Representations*, 2019.
- Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations. *arXiv.org*, November 2017.
- Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *International Conference on Machine Learning*, pages 536–543, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4.

- Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, pages 801–808. MIT Press, 2007.
- Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.
- Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *International Conference on Machine Learning*, 2019.
- Chris J Maddison, John Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Teh. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pages 6573–6583, 2017.
- Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.
- Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048, 2016.
- Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. <https://github.com/deepmind/dsprites-dataset/>, 2017.
- B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.
- Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.
- Mary Phuong, Max Welling, Nate Kushman, Ryota Tomioka, and Sebastian Nowozin. The mutual autoencoder: Controlling information in latent code representations, 2018. URL <https://openreview.net/forum?HkbnWqxCZ>.
- Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016.
- Tom Rainforth, Robert Cornish, Hongseok Yang, Andrew Warrington, and Frank Wood. On nesting monte carlo estimators. In *International Conference on Machine Learning*, pages 4264–4273, 2018a.
- Tom Rainforth, Adam R. Kosiorek, Tuan Anh Le, Chris J. Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. *International Conference on Machine Learning*, 2018b.
- Marc Ranzato, Christopher Poultney, Sumit Chopra, and Yann L. Cun. Efficient learning of sparse representations with an energy-based model. In *Advances in Neural Information Processing Systems*, pages 1137–1144. MIT Press, 2007.
- Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.
- Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *International Conference on Machine Learning*, pages 1431–1439, 2014.
- Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. 2014.
- Michal Rolínek, Dominik Zietlow, and Georg Martius. Variational Autoencoders Pursue PCA Directions (by Accident). *arXiv preprint arXiv:1812.06775*, 2018.
- Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.
- N. Siddharth, T Brooks Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. Learning disentangled representations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, pages 5925–5935, 2017.
- Jan Stühmer, Richard Turner, and Sebastian Nowozin. ISA-VAE: Independent subspace analysis with variational autoencoders, 2019. URL https://openreview.net/forum?id=rJl_NhR9K7.
- Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.
- Francesco Tonolini, Bjorn Sand Jensen, and Roderick Murray-Smith. Variational sparse coding, 2019. URL <https://openreview.net/forum?id=SkeJ6iR9Km>.
- Richard E. Turner and Maneesh Sahani. Two problems with variational expectation maximisation for time-series models. *D. Barber, T. Cemgil, and S. Chiappa (eds.), Bayesian Time series models, chapter 5*, page 109–130, 2011.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Jiacheng Xu and Greg Durrett. Spherical Latent Spaces for Stable Variational Autoencoders. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

Howard Hua Yang and Shun-ichi Amari. Adaptive online learning algorithms for blind separation: maximum entropy and minimum mutual information. *Neural computation*, 9(7):1457–1482, 1997.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *CoRR*, abs/1706.02262, 2017. URL <http://arxiv.org/abs/1706.02262>.

A. Proofs for Disentangling the β -VAE

Theorem 1. The β -VAE target $\mathcal{L}_\beta(\mathbf{x})$ can be interpreted in terms of the standard ELBO, $\mathcal{L}(\mathbf{x}; \pi_{\theta,\beta}, q_\phi)$, for an adjusted target $\pi_{\theta,\beta}(\mathbf{x}, \mathbf{z}) \triangleq p_\theta(\mathbf{x} | \mathbf{z})f_\beta(\mathbf{z})$ with annealed prior $f_\beta(\mathbf{z}) \triangleq p(\mathbf{z})^\beta / F_\beta$ as

$$\mathcal{L}_\beta(\mathbf{x}) = \mathcal{L}(\mathbf{x}; \pi_{\theta,\beta}, q_\phi) + (\beta - 1)H_{q_\phi} + \log F_\beta \quad (3)$$

where $F_\beta \triangleq \int_{\mathbf{z}} p(\mathbf{z})^\beta d\mathbf{z}$ is constant given β , and H_{q_ϕ} is the entropy of $q_\phi(\mathbf{z} | \mathbf{x})$.

Proof. Starting with (2), we have

$$\begin{aligned} \mathcal{L}_\beta(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} [\log p_\theta(\mathbf{x} | \mathbf{z})] + \beta H_{q_\phi} \\ &\quad + \beta \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} [\log p(\mathbf{z})] \\ &= \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} [\log p_\theta(\mathbf{x} | \mathbf{z})] + (\beta - 1)H_{q_\phi} + H_{q_\phi} \\ &\quad + \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} [\log p(\mathbf{z})^\beta - \log F_\beta] + \log F_\beta \\ &= \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} [\log p_\theta(\mathbf{x} | \mathbf{z})] + (\beta - 1)H_{q_\phi} \\ &\quad - \text{KL}(q_\phi(\mathbf{z} | \mathbf{x}) \| f_\beta(\mathbf{z})) + \log F_\beta \\ &= \mathcal{L}(\mathbf{x}; \pi_{\theta,\beta}, q_\phi) + (\beta - 1)H_{q_\phi} + \log F_\beta \end{aligned}$$

as required. $\square$

Corollary 1. If $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \Sigma)$ and $q_\phi(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), S_\phi(\mathbf{x}))$, then,

$$\mathcal{L}_\beta(\mathbf{x}; \theta, \phi) = \mathcal{L}(\mathbf{x}; \theta', \phi') + \frac{(\beta - 1)}{2} \log |S_{\phi'}(\mathbf{x})| + c \quad (4)$$

where θ' and ϕ' represent rescaled networks such that

$$\begin{aligned} p_{\theta'}(\mathbf{x} | \mathbf{z}) &= p_\theta(\mathbf{x} | \mathbf{z}/\sqrt{\beta}), \\ q_{\phi'}(\mathbf{z} | \mathbf{x}) &= \mathcal{N}(\mathbf{z}; \mu_{\phi'}(\mathbf{x}), S_{\phi'}(\mathbf{x})), \\ \mu_{\phi'}(\mathbf{x}) &= \sqrt{\beta} \mu_\phi(\mathbf{x}), \quad S_{\phi'}(\mathbf{x}) = \beta S_\phi(\mathbf{x}), \end{aligned}$$

and $c \triangleq \frac{D(\beta-1)}{2} \left(1 + \log \frac{2\pi}{\beta}\right) + \log F_\beta$ is a constant, with D denoting the dimensionality of $\mathbf{z}$.

Proof. We start by noting that

$$\begin{aligned} \pi_{\theta,\beta}(\mathbf{x}) &= \mathbb{E}_{f_\beta(\mathbf{z})} [p_\theta(\mathbf{x} | \mathbf{z})] = \mathbb{E}_{p(\mathbf{z})} [p_\theta(\mathbf{x} | \mathbf{z}/\sqrt{\beta})] \\ &= \mathbb{E}_{p(\mathbf{z})} [p_{\theta'}(\mathbf{x} | \mathbf{z})] = p_{\theta'}(\mathbf{x}) \end{aligned}$$

Now considering an alternate form of $\mathcal{L}(\mathbf{x}; \pi_{\theta,\beta}, q_\phi)$ in (3),

$$\begin{aligned} \mathcal{L}(\mathbf{x}; \pi_{\theta,\beta}, q_\phi) &= \log \pi_{\theta,\beta}(\mathbf{x}) - \text{KL}(q_\phi(\mathbf{z} | \mathbf{x}) \| \pi_{\theta,\beta}(\mathbf{z} | \mathbf{x})) \\ &= \log p_{\theta'}(\mathbf{x}) - \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} \left[\log \left(\frac{q_\phi(\mathbf{z} | \mathbf{x}) p_{\theta'}(\mathbf{x})}{p_\theta(\mathbf{x} | \mathbf{z}) f_\beta(\mathbf{z})} \right) \right] \\ &= \log p_{\theta'}(\mathbf{x}) \\ &\quad - \mathbb{E}_{q_{\phi'}(\mathbf{z} | \mathbf{x})} \left[\log \left(\frac{q_\phi(\mathbf{z}/\sqrt{\beta} | \mathbf{x}) p_{\theta'}(\mathbf{x})}{p_\theta(\mathbf{x} | \mathbf{z}/\sqrt{\beta}) f_\beta(\mathbf{z}/\sqrt{\beta})} \right) \right]. \quad (8) \end{aligned}$$

We first simplify $f_\beta(\mathbf{z}/\sqrt{\beta})$ as

$$\begin{aligned} f_\beta(\mathbf{z}/\sqrt{\beta}) &= \frac{1}{\sqrt{2\pi}|\Sigma/\beta|} \exp \left(-\frac{1}{2} \mathbf{z}^T \Sigma^{-1} \mathbf{z} \right) \\ &= p(\mathbf{z}) \beta^{(D/2)}. \end{aligned}$$

Further, denoting $\mathbf{z}_\dagger = \mathbf{z} - \sqrt{\beta} \mu_{\phi'}(\mathbf{x})$, and $\mathbf{z}_\ddagger = \mathbf{z}_\dagger / \sqrt{\beta} = \mathbf{z}/\sqrt{\beta} - \mu_{\phi'}(\mathbf{x})$, we have

$$\begin{aligned} q_{\phi'}(\mathbf{z} | \mathbf{x}) &= \frac{1}{\sqrt{2\pi}|S_\phi(\mathbf{x})\beta|} \exp \left(-\frac{1}{2\beta} \mathbf{z}_\dagger^T S_\phi(\mathbf{x})^{-1} \mathbf{z}_\dagger \right), \\ q_\phi \left(\frac{\mathbf{z}}{\sqrt{\beta}} | \mathbf{x} \right) &= \frac{1}{\sqrt{2\pi}|S_\phi(\mathbf{x})|} \exp \left(-\frac{1}{2} \mathbf{z}_\ddagger^T S_\phi(\mathbf{x})^{-1} \mathbf{z}_\ddagger \right) \\ \text{giving } q_\phi \left(\mathbf{z}/\sqrt{\beta} | \mathbf{x} \right) &= q_{\phi'}(\mathbf{z} | \mathbf{x}) \beta^{(D/2)}. \end{aligned}$$

Plugging these back in to (8) while remembering $p_\theta(\mathbf{x} | \mathbf{z}/\sqrt{\beta}) = p_{\theta'}(\mathbf{x} | \mathbf{z})$, we have

$$\begin{aligned} \mathcal{L}(\mathbf{x}; \pi_{\theta,\beta}, q_\phi) &= \log p_{\theta'}(\mathbf{x}) - \mathbb{E}_{q_{\phi'}(\mathbf{z} | \mathbf{x})} \left[\log \left(\frac{q_{\phi'}(\mathbf{z} | \mathbf{x}) p_{\theta'}(\mathbf{x})}{p_{\theta'}(\mathbf{x} | \mathbf{z}) p(\mathbf{z})} \right) \right] \\ &= \mathcal{L}(\mathbf{x}; \theta, \phi), \end{aligned}$$

showing that the ELBOs for the two setups are the same. For the entropy term, we note that

$$\begin{aligned} H_{q_\phi} &= \frac{D}{2} (1 + \log 2\pi) + \frac{1}{2} \log |S_\phi(\mathbf{x})| \\ &= \frac{D}{2} \left(1 + \log \frac{2\pi}{\beta} \right) + \frac{1}{2} \log |S_{\phi'}(\mathbf{x})|. \end{aligned}$$

Finally substituting for H_{q_ϕ} and $\mathcal{L}(\mathbf{x}; \pi_{\theta,\beta}, q_\phi)$ in (3) gives the desired result. $\square$

Corollary 2. Let $[\theta', \phi'] = g_\beta([\theta, \phi])$ represent the transformation required to produce the rescaled networks in Corollary 1. If $0 < |\det \nabla_{\theta,\phi} g([\theta, \phi])| < \infty \forall [\theta, \phi]$, then

$$\nabla_{\theta,\phi} \mathcal{L}_\beta(\mathbf{x}; \theta, \phi) = \mathbf{0} \Leftrightarrow \nabla_{\theta',\phi'} \mathcal{L}_{H,\beta}(\mathbf{x}; \theta', \phi') = \mathbf{0}.$$

Thus $[\theta^*, \phi^*]$ being a stationary point of $\frac{1}{n} \sum_{i=1}^n \mathcal{L}_\beta(\mathbf{x}_i; \theta, \phi)$ indicates that $g_\beta([\theta^*, \phi^*])$ is a stationary point of $\frac{1}{n} \sum_{i=1}^n \mathcal{L}_{H,\beta}(\mathbf{x}_i; \theta', \phi')$ and vice-versa.

Proof. Starting from Corollary 1 we have that

$$\begin{aligned} \nabla_{\theta,\phi} \mathcal{L}_\beta(\mathbf{x}; \theta, \phi) &= \nabla_{\theta,\phi} \mathcal{L}_{H,\beta}(\mathbf{x}; \theta', \phi') \\ &= (\nabla_{\theta,\phi} g_\beta([\theta, \phi])) \nabla_{\theta',\phi'} \mathcal{L}_{H,\beta}(\mathbf{x}; \theta', \phi'), \end{aligned}$$

so $\nabla_{\theta',\phi'} \mathcal{L}_{H,\beta}(\mathbf{x}; \theta', \phi') = \mathbf{0} \Rightarrow \nabla_{\theta,\phi} \mathcal{L}_\beta(\mathbf{x}; \theta, \phi) = \mathbf{0}$ given our assumption that $|\det \nabla_{\theta,\phi} g([\theta, \phi])| < \infty \forall [\theta, \phi]$. Further, as $0 < |\det \nabla_{\theta,\phi} g([\theta, \phi])| \forall [\theta, \phi]$, $(\nabla_{\theta,\phi} g_\beta([\theta, \phi]))^{-1}$ exists and has a finite determinant, so $\nabla_{\theta,\phi} \mathcal{L}_\beta(\mathbf{x}; \theta, \phi) = \mathbf{0}$ also implies $\nabla_{\theta',\phi'} \mathcal{L}_{H,\beta}(\mathbf{x}; \theta', \phi') = \mathbf{0}$. $\square$

Theorem 2. If $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \sigma I)$ and $q_\phi(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), S_\phi(\mathbf{x}))$, then for all rotation matrices R ,

$$\mathcal{L}_\beta(\mathbf{x}; \theta, \phi) = \mathcal{L}_\beta(\mathbf{x}; \theta^\dagger(R), \phi^\dagger(R)) \quad (6)$$

where $\theta^\dagger(R)$ and $\phi^\dagger(R)$ are transformed networks such that

$$\begin{aligned} p_{\theta^\dagger}(\mathbf{x} | \mathbf{z}) &= p_\theta(\mathbf{x} | R^T \mathbf{z}), \\ q_{\phi^\dagger}(\mathbf{z} | \mathbf{x}) &= \mathcal{N}(\mathbf{z}; R\mu_\phi(\mathbf{x}), RS_\phi(\mathbf{x})R^T). \end{aligned}$$

Proof. If $\mathbf{z} \sim q_\phi(\mathbf{z} | \mathbf{x})$ and $\mathbf{y} = R\mathbf{z}$ then, by Petersen et al. (§8.1.4 2008)), we have

$$\mathbf{y} \sim \mathcal{N}(\mathbf{y}; R\mu_\phi(\mathbf{x}), RS_\phi(\mathbf{x})R^T).$$

Consequently, the changes made by the transformed networks cancel to give the same reconstruction error as

$$\begin{aligned} \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})}[\log p_\theta(\mathbf{x} | \mathbf{z})] &= \mathbb{E}_{q_{\phi^\dagger}(\mathbf{z} | \mathbf{x})}[\log p_\theta(\mathbf{x} | R^T \mathbf{z})] \\ &= \mathbb{E}_{q_{\phi^\dagger}(\mathbf{z} | \mathbf{x})}[\log p_{\theta^\dagger}(\mathbf{x} | \mathbf{z})]. \end{aligned}$$

Furthermore, the KL divergence between $q_\phi(\mathbf{z} | \mathbf{x})$ and $p_\theta(\mathbf{z})$ is invariant to rotation, because of the rotational symmetry of the latter, such that $\text{KL}(q_\phi(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z})) = \text{KL}(q_{\phi^\dagger}(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z}))$. The result now follows from noting that the two terms of the β -VAE are equal under rotation. $\square$

B. Experimental Details

Disentanglement - 2d-shapes: The experiments from Section 6 on the impact of the prior in terms of disentanglement are conducted on the **2D Shapes** (Matthey et al., 2017) dataset, comprising of 737,280 binary 64 x 64 images of 2D shapes with ground truth factors [number of values]: shape [3], scale [6], orientation [40], x-position [32], y-position [32]. We use a convolutional neural network for the encoder and a deconvolutional neural network for the decoder, whose architectures are described in Table 1a. We use [0, 1] normalised data as targets for the mean of a Bernoulli distribution and negative cross-entropy for $\log p(\mathbf{x} | \mathbf{z})$. We rely on the Adam optimiser (Kingma and Ba, 2015; Reddi et al., 2018) with learning rate $1e^{-4}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$, to optimise the β -VAE objective from (3).

For $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \text{diag}(\sigma))$, experiments were run with a batch size of 64 and for 20 epochs. For $p(\mathbf{z}) = \prod_d \text{STUDENT-T}(z_d; \nu)$, experiments were run with a batch size of 256 and for 40 epochs. In Figure 2, the *PCA initialised anisotropic* prior is initialised so that its standard deviations are set to be the first D singular values of the data. These are then mapped through a softmax function to ensure that the β regularisation coefficient is not implicitly scaled compared to the isotropic case. For the *learned anisotropic* priors, standard deviations are first initialised as just described, and then learned along with the model through a log-variance parametrisation.

Encoder	Decoder
Input 64 x 64 binary image	Input $\in \mathbb{R}^{10}$
4x4 conv. 32 stride 2 & ReLU	FC. 128 ReLU
4x4 conv. 32 stride 2 & ReLU	FC. 4x4 x 64 ReLU
4x4 conv. 64 stride 2 & ReLU	4x4 upconv. 64 stride 2 & ReLU
4x4 conv. 64 stride 2 & ReLU	4x4 upconv. 64 stride 2 & ReLU
FC. 128	4x4 upconv. 32 stride 2 & ReLU
FC. 2x10	4x4 upconv. 1. stride 2

(a) 2D-shapes dataset.

Encoder	Decoder
Input $\in \mathbb{R}^2$	Input $\in \mathbb{R}^2$
FC. 100. & ReLU	FC. 100 & ReLU
FC. 2x2	FC. 2x2

(b) Pinwheel dataset.

Encoder
Input 32 x 32 x 1 channel image
4x4 conv. 32 stride 2 & BatchNorm2d & LeakyReLU(.2)
4x4 conv. 64 stride 2 & BatchNorm2d & LeakyReLU(.2)
4x4 conv. 128 stride 2 & BatchNorm2d & LeakyReLU(.2)
4x4 conv. 50, 4x4 conv. 50
Decoder
Input $\in \mathbb{R}^{50}$
4x4 upconv. 128 stride 1 pad 0 & BatchNorm2d & ReLU
4x4 upconv. 64 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 upconv. 32 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 upconv. 1 stride 2 pad 1

(c) Fashion-MNIST dataset.

Table 1. Encoder and decoder architectures.

We rely on the metric presented in §4 and Appendix B of Kim and Mnih (2018) as a measure of axis-alignment of the latent encodings with respect to the true (known) generative factors. Confidence intervals in Figure 2 were computed via the assumption of normally distributed samples with unknown mean and variance, with 100 runs of each model.

Clustering - Pinwheel We generated spiral cluster data², with $n = 400$ observations, clustered in 4 spirals, with radial and tangential standard deviations respectively of 0.1 and 0.30, and a rate of 0.25. We use fully-connected neural networks for both the encoder and decoder, whose architectures are described in Table 1b. We minimise the objective from (7), with $\mathbb{D}$ chosen to be the inclusive KL and $q_\phi(\mathbf{z})$ approximated by the aggregate encoding of the full dataset:

$$\begin{aligned} \mathbb{D}(q_\phi(\mathbf{z}), p(\mathbf{z})) &= \text{KL}(p(\mathbf{z}) || q_\phi(\mathbf{z})) \\ &= \mathbb{E}_{p(\mathbf{z})}[\log(p(\mathbf{z})) - \log(\mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}[q_\phi(\mathbf{z} | \mathbf{x})])] \end{aligned}$$

²<http://hips.seas.harvard.edu/content/synthetic-pinwheel-data-matlab>.

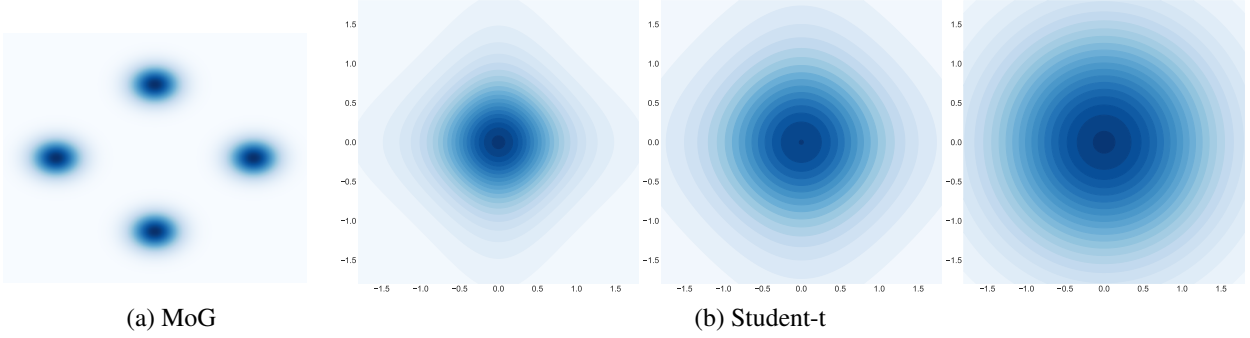


Figure 6. (a) PDF of Gaussian mixture model prior $p(\mathbf{z})$, as per (9). (b) PDF for a 2-dimensional factored Student-t distributions p_ν with degree of freedom $\nu = \{3, 5, 100\}$ (left to right). Note that $p_\nu(\mathbf{z}) \rightarrow \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ as $\nu \rightarrow \infty$.

$$\approx \sum_{j=1}^B \left(\log p(\mathbf{z}_j) - \log \left(\sum_{i=1}^n q_\phi(\mathbf{z}_j | \mathbf{x}_i) \right) \right)$$

with $\mathbf{z}_j \sim p(\mathbf{z})$. A Gaussian likelihood is used for the encoder. We trained the model for 500 epochs using the Adam optimiser (Kingma and Ba, 2015; Reddi et al., 2018), with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and a learning rate of $1e^{-3}$. The batch size is set to $B = n$.

The Gaussian mixture prior (c.f. Figure 6(a)) is defined as

$$\begin{aligned} p(\mathbf{z}) &= \sum_{c=1}^C \pi^c \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}^c, \boldsymbol{\Sigma}^c) \\ &= \sum_{c=1}^C \pi^c \prod_{d=1}^D \mathcal{N}(z_d | \mu_d^c, \sigma_d^c) \end{aligned} \quad (9)$$

with $D = 2$, $C = 4$, $\boldsymbol{\Sigma}^c = 0.03\mathbf{I}_D$, $\pi^c = 1/C$, and $\mu_d^c \in \{0, 1\}$.

Sparsity - Fashion-MNIST The experiments from Section 6 on the latent representation’s sparsity are conducted on the **Fashion-MNIST** (Xiao et al., 2017) dataset, comprising of 70,000 greyscale images resized to 32×32 .

To enforce sparsity, we relied on a prior defined as a factored univariate mixture of a standard and low variance normal distributions:

$$p(\mathbf{z}) = \prod_d (1 - \gamma) \mathcal{N}(z_d; 0, 1) + \gamma \mathcal{N}(z_d; 0, \sigma_0^2)$$

with $\sigma_0^2 = 0.05$. The weight, γ , of the low-variance component indicates how likely samples are to come from that component, hence to be *off*.

We minimised the objective from (7), with $\mathbb{D}(q_\phi(\mathbf{z}), p(\mathbf{z}))$ taken to be a dimension-wise MMD with a sum of *Cauchy* kernels on each dimension. Equivalently, we can think of this as calculating a single MMD using the single kernel

$$k(\mathbf{x}, \mathbf{y}) = \sum_{d=1}^D \sum_{\ell} \frac{\sigma_\ell}{\sigma_{\ell=1} + (x_d - y_d)^2}. \quad (10)$$

where $\sigma_\ell \in \{0.2, 0.4, 1, 2, 4, 10\}$ are a set of length scales.

This dimension-wise kernel only enforces a congruence between the marginal distributions of $\mathbf{x}$ and $\mathbf{y}$ and so, strictly speaking, its MMD does not constitute a valid divergence metric in the sense that we can have $\mathbb{D}(q_\phi(\mathbf{z}), p(\mathbf{z})) = 0$ when $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$ are not identical distributions: it only requires their marginals to match to get zero divergence.

The reasons we chose this approach are twofold. Firstly, we found that conventional kernels based on the Euclidean distance between encodings produced gradients with insurmountably high variances, meaning that effectively minimizing the divergence to get $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$ to match was not possible, even for very large batch sizes and $\alpha \rightarrow \infty$.

Secondly, though just matching the marginal distributions is not sufficient to ensure sparsity—as one could have some points with all dimensions close to the origin and some with all dimensions far away—a combination of the need to achieve good reconstructions and noise in the encoder process should prevent this from occurring. In short, provided the noise from the encoder is properly regulated, there is little information that can be stored in latent dimensions near the origin because of the high level of overlap forced in this region. Therefore, for a datapoint to be effectively encoded, it must have at least some of its latent dimensions outside of this region. Coupled with the need for most of the latent values to be near the origin to match the marginal distributions, this, in turn, enforces a sparse representation. Consequently, the loss in sparsity performance relative to using a hypothetical kernel that is both universal and has stable gradient estimates should only be relatively small, as is borne out in our empirical results. This may, however, be why we see a slight drop in sparsity performance for very large values of α .

We use a convolutional neural network for the encoder and a deconvolutional neural network for the decoder, whose architectures come from the DCGAN model (Radford et al., 2016) and are described in Table 1c. We use $[0, 1]$ normalised data as targets for the mean of a Laplace distribution

with fixed scaling of 0.1. We rely on the Adam optimiser with learning rate $5e^{-4}$, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. The model is then trained (on the training set) for 80 epochs with a batch-size of 500.

As an extrinsic measure of *sparsity*, we use the *Hoyer* metric (Hurley and Rickard, 2008), defined for $\mathbf{y} \in \mathbb{R}^d$ by

$$\text{Hoyer}(\mathbf{y}) = \frac{\sqrt{d} - \|\mathbf{y}\|_1 / \|\mathbf{y}\|_2}{\sqrt{d} - 1} \in [0, 1],$$

yielding 0 for a fully dense vector and 1 for a fully sparse vector. We additionally normalise each dimension to have a standard deviation of 1 under its aggregate distribution, i.e. we use $\bar{z}_d = z_d / \sigma(z_d)$ where $\sigma(z_d)$ is the standard deviation of dimension d of the latent encoding taken over the dataset. Overall sparsity is computed by averaging over the dataset as $\text{Sparsity} = 1/n \sum_i \text{Hoyer}(\bar{z}_i)$.

As discussed in the main text, we use a trained model with $\alpha = 1000$, $\beta = 1$, and $\gamma = 0.8$ to perform a qualitative analysis of sparsity using the *Fashion-MNIST* dataset. Figure 7 shows the per-class average embedding magnitude for this model, a subset of which was shown in the main text. As can be seen clearly, the different classes utilise predominantly different subsets of dimensions to encode the image data, as one might expect for sparse representations.

C. Posterior regularisation

The aggregate posterior regulariser $\mathbb{D}(q(\mathbf{z}), p(\mathbf{z}))$ is a little more subtle to analyse than the entropy regulariser as it involves both the choice of divergence and potential difficulties in estimating that divergence. One possible choice is the exclusive Kullback-Leibler divergence $\text{KL}(q(\mathbf{z}) \parallel p(\mathbf{z}))$, as previously used (without additional entropy regularisation) by (Dilokthanakul et al., 2019; Esmaili et al., 2019), but also implicitly by (Chen et al., 2018), through the use of a total correlation (TC) term. We now highlight a shortfall with this choice of divergence due to difficulties in its empirical estimation.

In short, the approaches used to estimate the $\text{H}[q(\mathbf{z})]$ (noting that $\text{KL}(q(\mathbf{z}) \parallel p(\mathbf{z})) = -\text{H}[q(\mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z})]$, where the latter term can be estimated reliably by a simple Monte Carlo estimate) can exhibit very large biases unless very large batch sizes are used, resulting in quite different effects from what was intended. In fact, our results suggest they will exhibit behaviour similar to the β -VAE if the batch size is too small. These biases arise from the effects of nesting estimators (Rainforth et al., 2018a), where the variance in the nested (inner) estimator for $q(\mathbf{z})$ induces a bias in the overall estimator. Specifically, for any random variable $\hat{Z}$,

$$\mathbb{E}[\log(\hat{Z})] = \log(\mathbb{E}[\hat{Z}]) - \frac{\text{Var}[\hat{Z}]}{2\mathbb{E}[\hat{Z}]^2} + O(\varepsilon)$$

where $O(\varepsilon)$ represents higher-order moments that get dominated asymptotically if $\hat{Z}$ is a Monte-Carlo estimator (see Proposition 1c in Maddison et al. (2017), Theorem 1 in Rainforth et al. (2018b), or Theorem 3 in Domke and Sheldon (2018)). In this setting, $\hat{Z} = \hat{q}(\mathbf{z})$ is the estimate used for $q(\mathbf{z})$. We thus see that if the variance of $\hat{q}(\mathbf{z})$ is large, this will induce a significant bias in our KL estimator.

To make things precise, we consider the estimator used for $\text{H}[q(\mathbf{z})]$ in Chen et al. (2018); Dilokthanakul et al. (2019); Esmaili et al. (2019)

$$\text{H}[q(\mathbf{z})] \approx \hat{\text{H}} \triangleq -\frac{1}{B} \sum_{b=1}^B \log \hat{q}(\mathbf{z}_b), \text{ where} \quad (11a)$$

$$\hat{q}(\mathbf{z}_b) = \frac{q_\phi(\mathbf{z}_b | \mathbf{x}_b)}{n} + \frac{n-1}{n(B-1)} \sum_{b' \neq b} q_\phi(\mathbf{z}_b | \mathbf{x}_{b'}), \quad (11b)$$

$\mathbf{z}_b \sim q_\phi(\mathbf{z} | \mathbf{x}_b)$, and $\{\mathbf{x}_1, \dots, \mathbf{x}_B\}$ is the mini-batch of data used for the current iteration for dataset size n . Esmaili et al. (2019) correctly show that $\mathbb{E}[\hat{q}(\mathbf{z}_b)] = \tilde{q}(\mathbf{z}_b)$, with the first term of (11b) comprising an exact term in $\tilde{q}(\mathbf{z}_b)$ and the second term of (11b) being an unbiased Monte-Carlo estimate for the remaining terms in $\tilde{q}(\mathbf{z}_b)$.

To examine the practical behaviour of this estimator when $B \ll n$, we first note that the second term of (11b) is, in practice, usually very small and dominated by the first term. This is borne out empirically in our own experiments, and also noted in Kim and Mnih (2018). To see why this is the case, consider that given encodings of two independent data points, it is highly unlikely that the two encoding distributions will have any notable overlap (e.g. for a Gaussian encoder, the means will most likely be very many standard deviations apart), presuming a sensible latent space is being learned. Consequently, even though this second term is unbiased and may have an expectation comparable or even larger than the first, it is heavily skewed—it is usually negligible, but occasionally large in the rare instances where there is substantial overlap between encodings.

Let the second term of (11b) be T_2 and the event that this is significant be E_S , such that $\mathbb{E}[T_2 | \neg E_S] \approx 0$. As explained above, $\mathbb{P}(E_S) \ll 1$ typically. We now have

$$\begin{aligned} \mathbb{E}[\hat{\text{H}}] &= \mathbb{P}(E_S) \mathbb{E}[\hat{\text{H}} | E_S] + (1 - \mathbb{P}(E_S)) \mathbb{E}[\hat{\text{H}} | \neg E_S] \\ &= \mathbb{P}(E_S) \mathbb{E}[\hat{\text{H}} | E_S] + (1 - \mathbb{P}(E_S)) \\ &\quad \cdot (\log n - \frac{1}{B} \sum_{b=1}^B \mathbb{E}[\log q_\phi(\mathbf{z}_b | \mathbf{x}_b) | \neg E_S] - \mathbb{E}[T_2 | \neg E_S]) \\ &= \mathbb{P}(E_S) \mathbb{E}[\hat{\text{H}} | E_S] + (1 - \mathbb{P}(E_S)) \\ &\quad \cdot (\log n - \mathbb{E}[\log q_\phi(\mathbf{z}_1 | \mathbf{x}_1) | \neg E_S] - \mathbb{E}[T_2 | \neg E_S]) \\ &\approx \mathbb{P}(E_S) \mathbb{E}[\hat{\text{H}} | E_S] \\ &\quad + (1 - \mathbb{P}(E_S)) (\log n - \mathbb{E}[\log q_\phi(\mathbf{z}_1 | \mathbf{x}_1)]) \end{aligned}$$

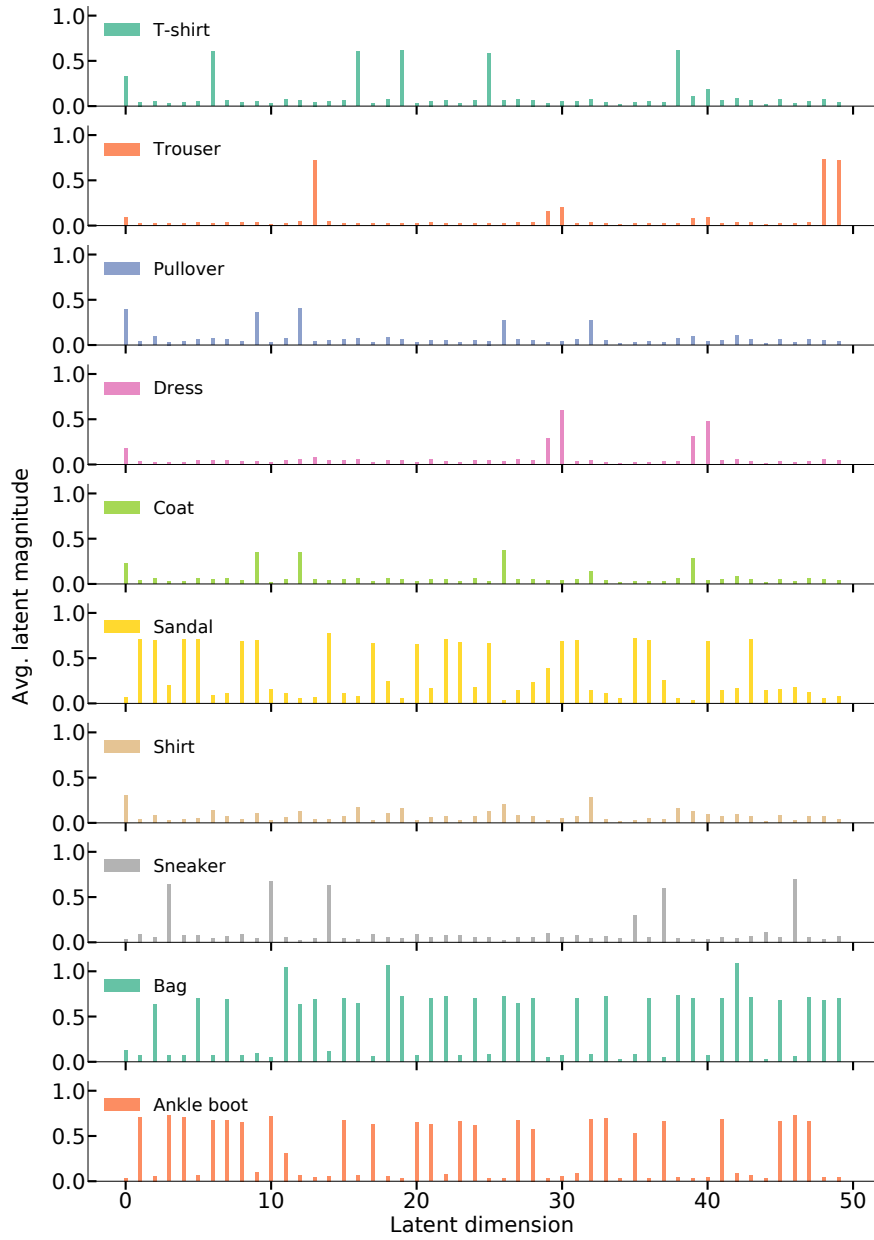


Figure 7. Average encoding magnitude over data for each classes in *Fashion-MNIST*.

where the approximation relies firstly on our previous assumption that $\mathbb{E}[T_2 \mid \neg E_S] \approx 0$ and also that $\mathbb{E}[\log q_\phi(\mathbf{z}_1 \mid \mathbf{x}_1) \mid \neg E_S] \approx \mathbb{E}[\log q_\phi(\mathbf{z}_1 \mid \mathbf{x}_1)]$. This second assumption will also generally hold in practice, firstly because the occurrence of E_S is dominated by whether two similar datapoints are drawn (rather than by the value of $\mathbf{x}_1$) and secondly because $\mathbb{P}(E_S) \ll 1$ implies that

$$\begin{aligned} \mathbb{E}[\log q_\phi(\mathbf{z}_1 \mid \mathbf{x}_1)] &= (1 - \mathbb{P}(E_S)) \mathbb{E}[\log q_\phi(\mathbf{z}_1 \mid \mathbf{x}_1) \mid \neg E_S] \\ &\quad + \mathbb{P}(E_S) \mathbb{E}[\log q_\phi(\mathbf{z}_1 \mid \mathbf{x}_1) \mid E_S] \\ &\approx \mathbb{E}[\log q_\phi(\mathbf{z}_1 \mid \mathbf{x}_1) \mid \neg E_S]. \end{aligned}$$

Characterising $\mathbb{E}[\hat{H} \mid E_S]$ precisely is a little more challenging, but it can safely be assumed to be smaller than $\mathbb{E}[\log q_\phi(\mathbf{z}_1 \mid \mathbf{x}_1)]$, which is approximately what would result from all the $\mathbf{x}'_b$ being the same as $\mathbf{x}_b$. We thus see that even when the event E_S does occur, the resulting estimates will still, at most, be on a comparable scale to when it does not. Consequently, whenever E_S is rare, the $(1 - \mathbb{P}(E_S)) \mathbb{E}[\hat{H} \mid \neg E_S]$ term will dominate and we thus have

$$\begin{aligned} \mathbb{E}[\hat{H}] &\approx \log n - \mathbb{E}[\log q_\phi(\mathbf{z}_1 \mid \mathbf{x}_1)] \\ &= \log n + \mathbb{E}_{p(\mathbf{x})}[\mathbb{H}[q_\phi(\mathbf{z} \mid \mathbf{x})]]. \end{aligned}$$

We now see that the estimator mimics the β -VAE regularisation up to a constant factor $\log n$, as adding the $\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z})]$ back in gives

$$\begin{aligned} -\mathbb{E}[\hat{H}] - \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z})] &\approx \mathbb{E}_{p(\mathbf{x})}[\text{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z}))] - \log n. \end{aligned}$$

We should thus expect to empirically see training with this estimator as a regulariser to behave similarly to the β -VAE with the same regularisation term whenever $B \ll n$. Note that the $\log n$ constant factor will not impact the gradients, but does mean that it is possible, even likely, that negative estimates for KL will be generated, even though we know the true value is positive.

Overcoming the problem can, at least to a certain degree, be overcome by using very large batch sizes B , at an inevitable computational and memory cost. However, the problem is potentially exacerbated in higher dimensional latent spaces and larger datasets, for which one would typically expect the typical overlap of datapoints to decrease.

C.1. Other Divergences

As discussed in the main paper, $\text{KL}(q(\mathbf{z}) \parallel p(\mathbf{z}))$ is far from the only aggregate posterior regulariser one might use. Though we do not analyse them formally, we expect many alternative divergence-estimator pairs to suffer from similar

issues. For example, using Monte Carlo estimators with the inclusive Kullback-Leibler divergence $\text{KL}(p(\mathbf{z}) \parallel q(\mathbf{z}))$ or the sliced Wasserstein distance (Kolouri et al., 2019) both result in nested expectations analogously to $\text{KL}(q(\mathbf{z}) \parallel p(\mathbf{z}))$, and are therefore likely to similarly induce substantial bias without using large batch sizes.

Interestingly, however, MMD and generative adversarial network (GAN) regularisers of the form discussed in (Tolstikhin et al., 2018) do not result in nested expectations and therefore are necessarily not prone to the same issues: they produce unbiased estimates of their respective objectives. Though we experienced practical issues in successfully implementing both of these—we found the signal-to-noise-ratio of the MMD gradient estimates to be very low, particularly in high dimensions, while we experienced training instabilities for the GAN regulariser—their apparent theoretical advantages may indicate that they are preferable approaches, particularly if these issues can be alleviated. The GAN-based approach to estimating the total correlation introduced by Kim and Mnih (2018) similarly allows a nested expectation to be avoided, at the cost of converting a conventional optimization into a minimax problem.

Given the failings of the available existing approaches, we believe that further investigation into divergence-estimator pairs for $\mathbb{D}(q(\mathbf{z}), p(\mathbf{z}))$ in VAEs is an important topic for future work that extends well beyond the context of this paper, or even the general aim of achieving decomposition. In particular, the need for congruence between the posterior (encoder), likelihood (decoder), and marginal likelihood (data distribution) for a generative model, means that ensuring $q(\mathbf{z})$ is close to $p(\mathbf{z})$ is a generally important endeavour for training VAEs. For example, mismatch between $q(\mathbf{z})$ and $p(\mathbf{z})$ will cause samples drawn from the learned generative model to mismatch the true data-generating distribution, regardless of the fidelity of our encoder and decoder.

D. Characterising Overlap

Reiterating the argument from the main text, although the mutual information $I(\mathbf{x}; \mathbf{z})$ between data and latents provides a perfectly serviceable characterisation of overlap in a number of cases, the two are not universally equivalent and we argue that it is overlap which is important in achieving useful representations. In particular, if the form of the encoding distribution is not fixed—as when employing normalising flows, for example— $I(\mathbf{x}; \mathbf{z})$ does not necessarily characterise overlap well.

Consider, for example, an encoding distribution that is a mixture between the prior and a uniform distribution on a tiny ϵ -ball around the mean encoding $\mu_\phi(\mathbf{x})$, i.e. $q_\phi(\mathbf{z} \mid \mathbf{x}) = \lambda \cdot \text{Uniform}(\|\mu_\phi(\mathbf{x}) - \mathbf{z}\|_2 < \epsilon) + (1 - \lambda) \cdot p(\mathbf{z})$. If the encoder and decoder are sufficiently flexible to learn

arbitrary representations, one now could arrive at *any* value for mutual information simply by an appropriate choice of λ . However, enforcing structuring of the latent space will be effectively impossible due to the lack of any pressure (other than a potentially small amount from internal regularization in the encoder network itself) for similar encodings to correspond to similar datapoints; the overlap between any two encodings is the same unless they are within ϵ of each other.

While this example is a bit contrived, it highlights a key feature of overlap that $I(\mathbf{x}; \mathbf{z})$ fails to capture: $I(\mathbf{x}; \mathbf{z})$ does not distinguish between large overlap with a small number of other datapoints and small overlap with a large number of other datapoints. This distinction is important because we are particularly interested in *how many* other datapoints one datapoint’s encoding overlaps with when imposing structure—the example setup fails because each datapoint has the same level of overlap with all the other datapoints.

Another feature that $I(\mathbf{x}; \mathbf{z})$ can fail to account for is a notion of *locality* in the latent space. Imagine a scenario where the encoding distributions are extremely multimodal with similar sized modes spread throughout the latent space, such as $q(\mathbf{z}|\mathbf{x}) = \sum_{i=1}^{1000} \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}) + \mathbf{m}_i, \sigma I)$ for some constant scalar σ , and vectors $\mathbf{m}_i$. Again we can achieve almost any value for $I(\mathbf{x}; \mathbf{z})$ by adjusting σ , but it is difficult to impose meaningful structure regardless as each datapoint can be encoded to many different regions of the latent space.

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	Disentangling disentanglement in variational autoencoders
Publication Status	☒ Published
Publication Details	Emile Mathieu, Tom Rainforth, N. Siddharth and Yee Whye Teh (2019). Proceedings of the 36th International Conference on Machine Learning, 4402–4412, 97, Proceedings of Machine Learning Research.

Student Confirmation

Student Name:	Emile Mathieu		
Contribution to the Paper	Performed all experiments, with some help from N. Siddharth for the implementation. Wrote a first draft for sections 2, 5 and 6, and assisted writing final version alongside T. Rainforth and N. Siddharth.		
Signature		Date	18/06/2021

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: Professor Yee Whye Teh		
Supervisor comments: I do certify that the candidate made a substantial contribution to the publication, and the description above is accurate.		
Signature		Date 30 June 2021

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

3

Continuous Hierarchical Representations with Poincaré Variational Autoencoders

Continuous Hierarchical Representations with Poincaré Variational Auto-Encoders

Emile Mathieu[†]

emile.mathieu@stats.ox.ac.uk

Charline Le Lan[†]

charline.lelan@stats.ox.ac.uk

Chris J. Maddison^{†*}

cmaddis@stats.ox.ac.uk

Ryota Tomioka[‡]

ryoto@microsoft.com

Yee Whye Teh^{†*}

y.w.teh@stats.ox.ac.uk

[†] Department of Statistics, University of Oxford, United Kingdom

* DeepMind, London, United Kingdom

[‡] Microsoft Research, Cambridge, United Kingdom

Abstract

The variational auto-encoder (VAE) is a popular method for learning a generative model and embeddings of the data. Many real datasets are hierarchically structured. However, traditional VAEs map data in a Euclidean latent space which cannot efficiently embed tree-like structures. Hyperbolic spaces with negative curvature can. We therefore endow VAEs with a Poincaré ball model of hyperbolic geometry as a latent space and rigorously derive the necessary methods to work with two main Gaussian generalisations on that space. We empirically show better generalisation to unseen data than the Euclidean counterpart, and can qualitatively and quantitatively better recover hierarchical structures.

1 Introduction

Learning useful representations from unlabelled raw sensory observations, which are often high-dimensional, is a problem of significant importance in machine learning. Variational auto-encoders (VAEs) (Kingma and Welling, 2014; Rezende et al., 2014) are a popular approach to this: they are probabilistic generative models composed of an *encoder* stochastically embedding observations in a low dimensional latent space $\mathcal{Z}$, and a *decoder* generating observations $\mathbf{x} \in \mathcal{X}$ from encodings $\mathbf{z} \in \mathcal{Z}$. After training, the encodings constitute a low-dimensional representation of the original raw observations, which can be used as features for a downstream task (e.g. Huang and LeCun, 2006; Coates et al., 2011) or be interpretable for their own sake. VAEs are therefore of interest for representation learning (Bengio et al., 2013), a field which aims to learn *good representations*, e.g. interpretable representations, ones yielding better generalisation, or ones useful for downstream tasks.

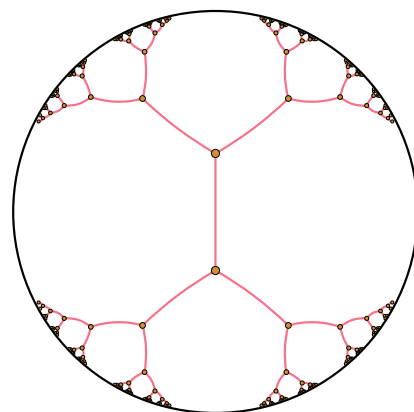


Figure 1: A regular tree isometrically embedded in the Poincaré disc. Red curves are same length *geodesics*, i.e. "straight lines".

It can be argued that in many domains data should be represented hierarchically. For example, in cognitive science, it is widely accepted that human beings use a hierarchy to organise object categories (e.g. Roy et al., 2006; Collins and Quillian, 1969; Keil, 1979). In biology, the theory of evolution (Darwin, 1859) implies that features of living organisms are related in a hierarchical manner given by the evolutionary tree. Explicitly incorporating hierarchical structure in probabilistic models has unsurprisingly been a long-running research topic (e.g. Duda et al., 2000; Heller and Ghahramani, 2005).

Earlier work in this direction tended to use trees as data structures to represent hierarchies. Recently, hyperbolic spaces have been proposed as an alternative continuous approach to learn hierarchical representations from textual and graph-structured data (Nickel and Kiela, 2017; Tifrea et al., 2019). Hyperbolic spaces can be thought of as continuous versions of trees, and vice versa, as illustrated in Figure 1. Trees can be embedded with arbitrarily low error into the Poincaré disc model of hyperbolic geometry (Sarkar, 2012). The exponential growth of the Poincaré surface area with respect to its radius is analogous to the exponential growth of the number of leaves in a tree with respect to its depth. Further, these spaces are smooth, enabling the use of deep learning approaches which rely on differentiability.

We show that replacing VAEs latent space components, which traditionally assume a Euclidean metric over the latent space, by their hyperbolic generalisation helps to represent and discover hierarchies. Our goals are twofold: (a) learn a latent representation that is interpretable in terms of hierarchical relationships among the observations, (b) learn a more efficient representation which generalises better to unseen data that is hierarchically structured. Our main contributions are as follows:

1. We propose efficient and reparametrisable sampling schemes, and calculate the probability density functions, for two canonical Gaussian generalisations defined on the Poincaré ball, namely the maximum-entropy and wrapped normal distributions. These are the ingredients required to train our VAEs.
2. We introduce a decoder architecture that explicitly takes into account the hyperbolic geometry, which we empirically show to be crucial.
3. We empirically demonstrate that endowing a VAE with a Poincaré ball latent space can be beneficial in terms of model generalisation and can yield more interpretable representations.

Our work fits well with a surge of interest in combining hyperbolic geometry and VAEs. Of these, it relates most strongly to the concurrent works of Ovinnikov (2018); Grattarola et al. (2019); Nagano et al. (2019). In contrast to these approaches, we introduce a decoder that takes into account the geometry of the hyperbolic latent space. Along with the *wrapped normal* generalisation used in the latter two articles, we give a thorough treatment of the *maximum entropy normal* generalisation and a rigorous analysis of the difference between the two. Additionally, we train our model by maximising a lower bound on the marginal likelihood, as opposed to Ovinnikov (2018); Grattarola et al. (2019) which consider a Wasserstein and an adversarial auto-encoder setting, respectively. We discuss these works in more detail in Section 4.

2 The Poincaré Ball model of hyperbolic geometry

2.1 Review of Riemannian geometry

Throughout the paper we denote the Euclidean norm and inner product by $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ respectively. A real, smooth *manifold* $\mathcal{M}$ is a set of points $\mathbf{z}$, which is "locally similar" to a linear space. For every point $\mathbf{z}$ of the manifold $\mathcal{M}$ is attached a real vector space of the same dimensionality as $\mathcal{M}$ called the *tangent space* $\mathcal{T}_{\mathbf{z}}\mathcal{M}$. Intuitively, it contains all the possible directions in which one can tangentially pass through $\mathbf{z}$. For each point $\mathbf{z}$ of the manifold, the *metric tensor* $\mathbf{g}(\mathbf{z})$ defines an inner product on the associated tangent space : $\mathbf{g}(\mathbf{z}) = \langle \cdot, \cdot \rangle_{\mathbf{z}} : \mathcal{T}_{\mathbf{z}}\mathcal{M} \times \mathcal{T}_{\mathbf{z}}\mathcal{M} \rightarrow \mathbb{R}$. The *matrix representation of the Riemannian metric* $\mathbf{G}(\mathbf{z})$, is defined such that $\forall \mathbf{u}, \mathbf{v} \in \mathcal{T}_{\mathbf{z}}\mathcal{M}, \langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{z}} = \mathbf{g}(\mathbf{z})(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{G}(\mathbf{z}) \mathbf{v}$. A *Riemannian manifold* is then defined as a tuple $(\mathcal{M}, \mathbf{g})$ (Petersen, 2006). The metric tensor gives a *local* notion of angle, length of curves, surface area and volume, from which *global* quantities can be derived by integrating local contributions. A norm is induced by the inner product on $\mathcal{T}_{\mathbf{z}}\mathcal{M}$: $\|\cdot\|_{\mathbf{z}} = \sqrt{\langle \cdot, \cdot \rangle_{\mathbf{z}}}$. An infinitesimal volume element is induced on each tangent space $\mathcal{T}_{\mathbf{z}}\mathcal{M}$, and thus a measure $d\mathcal{M}(\mathbf{z}) = \sqrt{|\mathbf{G}(\mathbf{z})|}d\mathbf{z}$ on the manifold, with $d\mathbf{z}$ being the Lebesgue measure.

3. Continuous Hierarchical Representations with Poincaré Variational Autoencoders

The length of a curve $\gamma : t \mapsto \gamma(t) \in \mathcal{M}$ is given by $L(\gamma) = \int_0^1 \|\gamma'(t)\|_{\gamma(t)}^{1/2} dt$. The concept of straight lines can then be generalised to *geodesics*, which are constant speed curves giving the shortest path between pairs of points $\mathbf{z}, \mathbf{y}$ of the manifold: $\gamma^* = \arg \min L(\gamma)$ with $\gamma(0) = \mathbf{z}$, $\gamma(1) = \mathbf{y}$ and $\|\gamma'(t)\|_{\gamma(t)} = 1$. A *global* distance is thus induced on $\mathcal{M}$ given by $d_{\mathcal{M}}(\mathbf{z}, \mathbf{y}) = \inf L(\gamma)$. Endowing $\mathcal{M}$ with that distance consequently defines a metric space $(\mathcal{M}, d_{\mathcal{M}})$. The concept of moving along a "straight" curve with constant velocity is given by the *exponential map*. In particular, there is a unique unit speed *geodesic* γ satisfying $\gamma(0) = \mathbf{z}$ with initial tangent vector $\gamma'(0) = \mathbf{v}$. The corresponding exponential map is then defined by $\exp_{\mathbf{z}}(\mathbf{v}) = \gamma(1)$, as illustrated on Figure 2. The *logarithm map* is the inverse $\log_{\mathbf{z}} = \exp_{\mathbf{z}}^{-1} : \mathcal{M} \rightarrow \mathcal{T}_{\mathbf{z}}\mathcal{M}$. For geodesically complete manifolds, such as the Poincaré ball, $\exp_{\mathbf{z}}$ is well-defined on the full tangent space $\mathcal{T}_{\mathbf{z}}\mathcal{M}$ for all $\mathbf{z} \in \mathcal{M}$.

2.2 The Poincaré ball model of hyperbolic geometry

A d -dimensional hyperbolic space, denoted $\mathbb{H}^d$, is a complete, simply connected, d -dimensional Riemannian manifold with constant negative curvature c . In contrast with the Euclidean space $\mathbb{R}^d$, $\mathbb{H}^d$ can be constructed using various isomorphic models (none of which is prevalent), including the hyperboloid model, the Beltrami-Klein model, the Poincaré half-plane model and the Poincaré ball $\mathcal{B}_c^d$ (Beltrami, 1868). The Poincaré ball model is formally defined as the Riemannian manifold $\mathbb{B}_c^d = (\mathcal{B}_c^d, \mathfrak{g}_p^c)$, where $\mathcal{B}_c^d$ is the open ball of radius $1/\sqrt{c}$, and $\mathfrak{g}_p^c$ its *metric tensor*, which along with its induced distance are given by

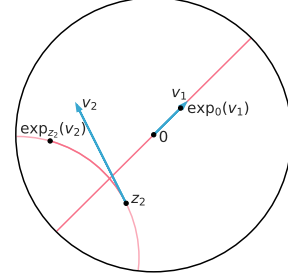


Figure 2: Geodesics and exponential maps in the Poincaré disc.

$$\mathfrak{g}_p^c(\mathbf{z}) = (\lambda_{\mathbf{z}}^c)^2 \mathfrak{g}_e(\mathbf{z}), \quad d_p^c(\mathbf{z}, \mathbf{y}) = \frac{1}{\sqrt{c}} \cosh^{-1} \left(1 + 2c \frac{\|\mathbf{z} - \mathbf{y}\|^2}{(1 - c\|\mathbf{z}\|^2)(1 - c\|\mathbf{y}\|^2)} \right),$$

where $\lambda_{\mathbf{z}}^c = \frac{2}{1 - c\|\mathbf{z}\|^2}$ and $\mathfrak{g}_e$ denotes the Euclidean metric tensor, i.e. the usual dot product.

The *Möbius addition* (Ungar, 2008) of $\mathbf{z}$ and $\mathbf{y}$ in $\mathbb{B}_c^d$ is defined as

$$\mathbf{z} \oplus_c \mathbf{y} = \frac{(1 + 2c \langle \mathbf{z}, \mathbf{y} \rangle + c\|\mathbf{y}\|^2)\mathbf{z} + (1 - c\|\mathbf{z}\|^2)\mathbf{y}}{1 + 2c \langle \mathbf{z}, \mathbf{y} \rangle + c^2\|\mathbf{z}\|^2\|\mathbf{y}\|^2}.$$

One recovers the Euclidean addition of two vectors in $\mathbb{R}^d$ as $c \rightarrow 0$. Building on that framework, Ganea et al. (2018) derived closed-form formulations for the *exponential map* (illustrated in Figure 2)

$$\exp_{\mathbf{z}}^c(\mathbf{v}) = \mathbf{z} \oplus_c \left(\tanh \left(\sqrt{c} \frac{\lambda_{\mathbf{z}}^c \|\mathbf{v}\|}{2} \right) \frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|} \right)$$

and its inverse, the *logarithm map*

$$\log_{\mathbf{z}}^c(\mathbf{y}) = \frac{2}{\sqrt{c}\lambda_{\mathbf{z}}^c} \tanh^{-1}(\sqrt{c}\|\mathbf{y} - \mathbf{z} \oplus_c \mathbf{y}\|) \frac{-\mathbf{z} \oplus_c \mathbf{y}}{\|\mathbf{y} - \mathbf{z} \oplus_c \mathbf{y}\|}.$$

3 The Poincaré VAE

We consider the problem of mapping an empirical distribution of observations to a lower dimensional Poincaré ball $\mathbb{B}_c^d$, as well as learning a map from this latent space $\mathcal{Z} = \mathbb{B}_c^d$ to the observation space $\mathcal{X}$. Building on the VAE framework, this *Poincaré-VAE* model, or $\mathcal{P}^c$ -VAE for short, differs by the choice of prior and posterior distributions being defined on $\mathbb{B}_c^d$, and by the encoder g_ϕ and decoder f_θ maps which take into account the latent space geometry. Their parameters $\{\theta, \phi\}$ are learned by maximising the *evidence lower bound* (ELBO). Our model can be seen as a generalisation of a classical Euclidean VAE (Kingma and Welling, 2014; Rezende et al., 2014) that we denote by $\mathcal{N}$ -VAE, i.e. $\mathcal{P}^c\text{-VAE} \xrightarrow{c \rightarrow 0} \mathcal{N}\text{-VAE}$.

3.1 Prior and variational posterior distributions

In order to parametrise distributions on the Poincaré ball, we consider two canonical generalisations of normal distributions on that space. A more detailed review of Gaussian generalisations on manifolds can be found in Appendix B.1.

Riemannian normal One generalisation is the distribution maximising entropy given an expectation and variance (Said et al., 2014; Pennec, 2006; Hauberg, 2018), often called the *Riemannian normal* distribution, which has a density w.r.t. the metric induced measure $d\mathcal{M}$ given by

$$\mathcal{N}_{\mathbb{B}_c^d}^R(\mathbf{z}|\boldsymbol{\mu}, \sigma^2) = \frac{d\nu^R(\mathbf{z}|\boldsymbol{\mu}, \sigma^2)}{d\mathcal{M}(\mathbf{z})} = \frac{1}{Z^R} \exp\left(-\frac{d_p^c(\boldsymbol{\mu}, \mathbf{z})^2}{2\sigma^2}\right), \quad (1)$$

where $\sigma > 0$ is a dispersion parameter, $\boldsymbol{\mu} \in \mathbb{B}_c^d$ is the Fréchet mean, and Z^R is the normalising constant derived in Appendix B.4.3.

Wrapped normal An alternative is to consider the push-forward measure obtained by mapping a normal distribution along the *exponential map* $\exp_{\boldsymbol{\mu}}$. That probability measure is often referred to as the *wrapped normal* distribution, and has been used in auto-encoder frameworks with other manifolds (Grattarola et al., 2019; Nagano et al., 2019; Falorsi et al., 2018). Samples $\mathbf{z} \in \mathbb{B}_c^d$ are obtained as $\mathbf{z} = \exp_{\boldsymbol{\mu}}^c(\mathbf{v}/\lambda_{\boldsymbol{\mu}}^c)$ with $\mathbf{v} \sim \mathcal{N}(\cdot|\mathbf{0}, \Sigma)$ and its density is given by (details given in Appendix B.3)

$$\begin{aligned} \mathcal{N}_{\mathbb{B}_c^d}^W(\mathbf{z}|\boldsymbol{\mu}, \Sigma) &= \frac{d\nu^W(\mathbf{z}|\boldsymbol{\mu}, \Sigma)}{d\mathcal{M}(\mathbf{z})} \\ &= \mathcal{N}(\lambda_{\boldsymbol{\mu}}^c \log_{\boldsymbol{\mu}}(\mathbf{z})|\mathbf{0}, \Sigma) \left(\frac{\sqrt{c} d_p^c(\boldsymbol{\mu}, \mathbf{z})}{\sinh(\sqrt{c} d_p^c(\boldsymbol{\mu}, \mathbf{z}))} \right)^{d-1}. \end{aligned} \quad (2)$$

The (usual) normal distribution is recovered for both generalisations as $c \rightarrow 0$. We discuss the benefits and drawbacks of those two distributions in Appendix B.1. We refer to both as *hyperbolic normal* distributions with pdf $\mathcal{N}_{\mathbb{B}_c^d}(\mathbf{z}|\boldsymbol{\mu}, \sigma^2)$. Figure 8 shows several probability densities for both distributions.

The prior distribution defined on $\mathcal{Z}$ is chosen to be a hyperbolic normal distribution with mean zero, $p(\mathbf{z}) = \mathcal{N}_{\mathbb{B}_c^d}(\cdot|\mathbf{0}, \sigma_0^2)$, and the variational family is chosen to be parametrised as $\mathcal{Q} = \{\mathcal{N}_{\mathbb{B}_c^d}(\cdot|\boldsymbol{\mu}, \sigma^2) \mid \boldsymbol{\mu} \in \mathbb{B}_c^d, \sigma \in \mathbb{R}_*^+\}$.

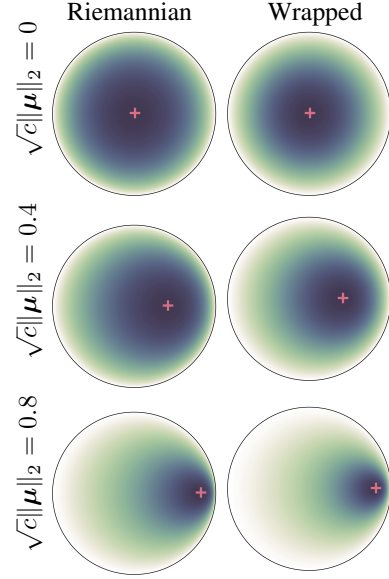


Figure 3: Hyperbolic normal probability density for different Fréchet mean, same standard deviation and $c = 10$. The *Riemannian* hyperbolic radius has a slightly larger mode.

3.2 Encoder and decoder

We make use of two neural networks, a *decoder* f_{θ} and an *encoder* g_{ϕ} , to parametrise the likelihood $p(\cdot|f_{\theta}(\mathbf{z}))$ and the variational posterior $q(\cdot|g_{\phi}(\mathbf{x}))$ respectively. The input of f_{θ} and the output of g_{ϕ} need to respect the hyperbolic geometry of $\mathcal{Z}$. In the following we describe appropriate choices for the first layer of the decoder and the last layer of the encoder.

Decoder In the Euclidean case, an affine transformation can be written in the form $f_{\mathbf{a}, \mathbf{p}}(\mathbf{z}) = \langle \mathbf{a}, \mathbf{z} - \mathbf{p} \rangle$, with orientation and offset parameters $\mathbf{a}, \mathbf{p} \in \mathbb{R}^d$. This can be rewritten in the form

$$f_{\mathbf{a}, \mathbf{p}}(\mathbf{z}) = \text{sign}(\langle \mathbf{a}, \mathbf{z} - \mathbf{p} \rangle) \|\mathbf{a}\| d_E(\mathbf{z}, H_{\mathbf{a}, \mathbf{p}}^c)$$

where $H_{\mathbf{a}, \mathbf{p}} = \{\mathbf{z} \in \mathbb{R}^p \mid \langle \mathbf{a}, \mathbf{z} - \mathbf{p} \rangle = 0\} = \mathbf{p} + \{\mathbf{a}\}^{\perp}$ is the decision hyperplane. The third term is the distance between $\mathbf{z}$ and the decision hyperplane $H_{\mathbf{a}, \mathbf{p}}^c$ and the first term refers to the side of $H_{\mathbf{a}, \mathbf{p}}^c$ where $\mathbf{z}$ lies. Ganea et al. (2018) analogously introduced an operator $f_{\mathbf{a}, \mathbf{p}}^c : \mathbb{B}_c^d \rightarrow \mathbb{R}^p$ on the Poincaré ball,

$$f_{\mathbf{a}, \mathbf{p}}^c(\mathbf{z}) = \text{sign}(\langle \mathbf{a}, \log_{\mathbf{p}}^c(\mathbf{z}) \rangle_{\mathbf{p}}) \|\mathbf{a}\|_{\mathbf{p}} d_p^c(\mathbf{z}, H_{\mathbf{a}, \mathbf{p}}^c)$$

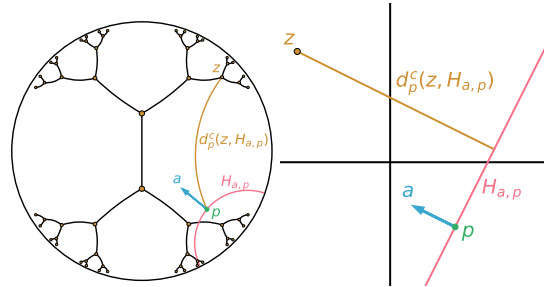


Figure 4: Illustration of an orthogonal projection on a hyperplane in a Poincaré disc (Left) and an Euclidean plane (Right).

3. Continuous Hierarchical Representations with Poincaré Variational Autoencoders

with $H_{a,p}^c = \{z \in \mathbb{B}_c^d \mid \langle a, \log_p^c(z) \rangle = 0\} = \exp_p^c(\{a\}^\perp)$. A closed-form expression for the distance $d_p^c(z, H_{a,p}^c)$ was also derived, $d_p^c(z, H_{a,p}^c) = \frac{1}{\sqrt{c}} \sinh^{-1} \left(\frac{2\sqrt{c} | \langle -p \oplus_c z, a \rangle |}{(1-c\| -p \oplus_c z \|^2) \|a\|} \right)$. The hyperplane decision boundary $H_{a,p}^c$ is called *gyroplane* and is a semi-hypersphere orthogonal to the Poincaré ball's boundary as illustrated on Figure 4. The decoder's first layer, called *gyroplane* layer, is chosen to be a concatenation of such operators, which are then composed with a standard feed-forward neural network.

Encoder The encoder g_ϕ outputs a Fréchet mean $\mu \in \mathbb{B}_c^d$ and a distortion $\sigma \in \mathbb{R}_*^+$ which parametrise the hyperbolic variational posterior. The Fréchet mean μ is obtained as the image of the exponential map $\exp_0^c$, and the distortion σ through a *softplus* function.

3.3 Training

We follow a standard variational approach by deriving a lower bound on the marginal likelihood. The ELBO is optimised via an unbiased Monte Carlo (MC) estimator thanks to the reparametrisable sampling schemes that we introduce for both hyperbolic normal distributions.

Objective The *evidence lower bound* (ELBO) can readily be extended to Riemannian latent spaces by applying Jensen's inequality w.r.t. $d\mathcal{M}$ (see Appendix A)

$$\log p(x) \geq \mathcal{L}_{\mathcal{M}}(x; \theta, \phi) \triangleq \int_{\mathcal{M}} \ln \left(\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} \right) q_\phi(z|x) d\mathcal{M}(z).$$

Densities have been introduced earlier in Equations 1 and 2.

Reparametrisation In the Euclidean setting, by working in polar coordinates, an isotropic normal distribution centred at μ can be described by a directional vector α uniformly distributed on the hypersphere and a univariate radius $r = d_E(\mu, z)$ following a χ -distribution. In the Poincaré ball we can rely on a similar representation, through a *hyperbolic polar* change of coordinates, given by

$$z = \exp_\mu^c \left(G(\mu)^{-\frac{1}{2}} v \right) = \exp_\mu^c \left(\frac{r}{\lambda_\mu^c} \alpha \right) \quad (3)$$

with $v = r\alpha$ and $r = d_p^c(\mu, z)$. The direction α is still uniformly distributed on the hypersphere and for the *wrapped normal*, the radius r is still χ -distributed, while for the *Riemannian normal* its density $\rho^R(r)$ is given by (derived in Appendix B.4.1)

$$\rho^W(r) \propto \mathbb{1}_{\mathbb{R}_+}(r) e^{-\frac{r^2}{2\sigma^2}} r^{d-1}, \quad \rho^R(r) \propto \mathbb{1}_{\mathbb{R}_+}(r) e^{-\frac{r^2}{2\sigma^2}} \left(\frac{\sinh(\sqrt{c}r)}{\sqrt{c}} \right)^{d-1}.$$

The latter density $\rho^R(r)$ can efficiently be sampled via rejection sampling with a piecewise exponential distribution proposal. This makes use of its log-concavity. The *Riemannian normal* sampling scheme is not directly affected by dimensionality since the radius is a *one-dimensional* random variable. Full sampling schemes are described in Algorithm 1, and in Appendices B.4.1 and B.4.2.

Gradients Gradients $\nabla_\mu z$ can straightforwardly be computed thanks to the exponential map reparametrisation (Eq 3), and gradients w.r.t. the dispersion $\nabla_\sigma z$ are readily available for the *wrapped normal*. For the *Riemannian normal*, we additionally rely on an implicit reparametrisation (Figurnov et al., 2018) of ρ^R via its cdf $F^R(r; \sigma)$.

Optimisation Parameters of the model living in the Poincaré ball are parametrised via the exponential mapping: $\phi_i = \exp_0^c(\phi_i^0)$ with $\phi_i^0 \in \mathbb{R}^m$, so we can make use of usual optimisation schemes. Alternatively, one could directly optimise such manifold parameters with manifold gradient descent schemes (Bonnabel, 2013).

Algorithm 1 Hyperbolic normal sampling scheme

Require: μ, σ^2 , dimension d , curvature c
if Wrapped normal **then** $v \sim \mathcal{N}(\mathbf{0}_d, \sigma^2)$
else if Riemannian normal **then**
 Let g be a piecewise exponential proposal
 while sample r not accepted **do**
 Propose $r \sim g(\cdot)$, $u \sim \mathcal{U}([0, 1])$
 if $u < \frac{\rho^R(r)}{g(r)}$ **then** Accept sample r
 Sample direction $\alpha \sim \mathcal{U}(\mathbb{S}^{d-1})$
 $v \leftarrow r\alpha$
Return $z = \exp_\mu^c(v/\lambda_\mu^c)$

4 Related work

Hierarchical models The Bayesian Nonparametric literature has a rich history of explicitly modelling the hierarchical structure of data (Teh et al., 2008; Heller and Ghahramani, 2005; Griffiths et al., 2004; Ghahramani et al., 2010; Larsen et al., 2001; Salakhutdinov et al., 2011). The discrete nature of trees used in such models makes learning difficult, whereas performing optimisation in a continuous hyperbolic space is an attractive alternative. Such an approach has been empirically and theoretically shown to be useful for graphs and word embeddings (Nickel and Kiela, 2017, 2018; Chamberlain et al., 2017; Sala et al., 2018; Tifrea et al., 2019).

Distributions on manifold Probability measures defined on manifolds are of interest to model uncertainty of data living (either intrinsically or assumed to) on such spaces, e.g. directional statistics (Ley and Verdebout, 2017; Mardia and Jupp, 2009). Pennec (2006) introduced a maximum entropy generalisation of the normal distribution, often referred to as *Riemannian normal*, which has been used for maximum likelihood estimation in the Poincaré half-plane (Said et al., 2014) and on the hypersphere (Hauberg, 2018). Another class of manifold probability measures are *wrapped* distributions, i.e. push-forward of distributions defined on a tangent space, often along the exponential map. They have recently been used in auto-encoder frameworks on the *hyperboloid* model (of hyperbolic geometry) (Grattarola et al., 2019; Nagano et al., 2019) and on Lie groups (Falorsi et al., 2018). Rey et al. (2019); Li et al. (2019) proposed to parametrise a variational family through a Brownian motion on manifolds such as spheres, tori, projective spaces and $SO(3)$.

VAEs with Riemannian latent manifold VAEs with non Euclidean latent space have been recently introduced, such as Davidson et al. (2018) making use of hyperspherical geometry and Falorsi et al. (2018) endowing the latent space with a $SO(3)$ group structure. Concurrent work considers endowing auto-encoders (AEs) with a hyperbolic latent space. Grattarola et al. (2019) introduces a constant curvature manifold (CCM) (i.e. hyperspherical, Euclidean and hyperboloid) latent space within an adversarial auto-encoder framework. However, the encoder and decoder are not designed to explicitly take into account the latent space geometry. Ovinnikov (2018) recently proposed to endow a VAE latent space with a Poincaré ball model. They choose a Wasserstein Auto-Encoder framework (Tolstikhin et al., 2018) because they could not derive a closed-form solution of the ELBO’s entropy term. We instead rely on a MC estimate of the ELBO by introducing a novel reparametrisation of the Riemannian normal. They discuss the *Riemannian* normal distribution, yet they make a number of heuristic approximations for sampling and reparametrisation. Also, Nagano et al. (2019) propose using a *wrapped* normal distribution to model uncertainty on the *hyperboloid* model of hyperbolic space. They derive its density and a reparametrisable sampling scheme, allowing such a distribution to be used in a variational learning framework. They apply this *wrapped* normal distribution to stochastically embed graphs and to parametrise the variational family in VAEs. Ovinnikov (2018) and Nagano et al. (2019) rely on a standard feed-forward decoder architecture, which does not take into account the hyperbolic geometry.

5 Experiments

We implemented our model and ran our experiments within the automatic differentiation framework PyTorch (Paszke et al., 2017). We open-source our code for reproducibility and to benefit the community¹. Experimental details are fully described in Appendix C.

5.1 Branching diffusion process

We assess our modelling assumption on data generated from a branching diffusion process which explicitly incorporate hierarchical structure. Nodes $y_i \in \mathbb{R}^n$ are normally distributed with mean given by their parent and with unit variance. Models are trained on a noisy vector representations $(x_1, \dots, x_N)$, hence do not have access to the true hierarchical representation. We train several $\mathcal{P}^c$ -VAEs with increasing curvatures, along with a vanilla $\mathcal{N}$ -VAE as a baseline. Table 1 shows that the $\mathcal{P}^c$ -VAE outperforms its Euclidean counterpart in terms of test marginal likelihood. As expected, we observe that the performance of the $\mathcal{N}$ -VAE is recovered as the curvature c tends to zero. Also, we notice that increasing the prior distribution distortion σ_0 helps embeddings lie closer to the border,

¹<https://github.com/emilemathieu/pvae>

3. Continuous Hierarchical Representations with Poincaré Variational Autoencoders

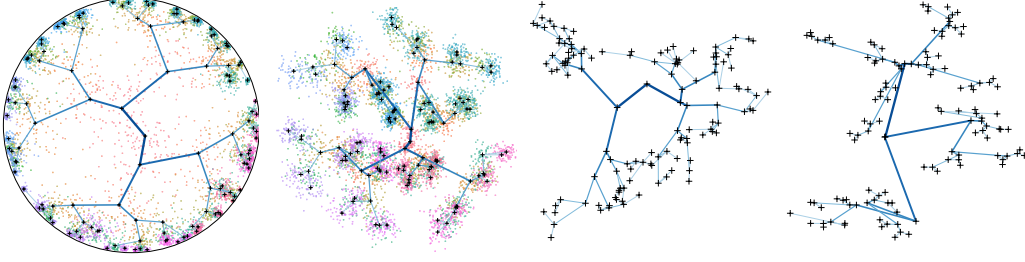


Figure 5: Latent representations learned by $\mathcal{P}^1$ -VAE (Leftmost), $\mathcal{N}$ -VAE (Center-Left), PCA (Center-Right) and GPLVM (Rightmost) trained on synthetic dataset. Embeddings are represented by black crosses, and colour dots are posterior samples. Blue lines represent true hierarchy.

and as a consequence improved generalisation performance. Figure 5 represents latent embeddings for $\mathcal{P}^1$ -VAE and $\mathcal{N}$ -VAE, along with two embedding baselines: principal component analysis (PCA) and a Gaussian process latent variable model (GPLVM). A hierarchical structure is somewhat learned by all models, yet $\mathcal{P}^c$ -VAE’s latent representation is the least distorted.

Table 1: Negative test marginal likelihood estimates $\mathcal{L}_{\text{IWAE}}$ (Burda et al., 2015) (computed with 5000 samples) on the synthetic dataset. 95% confidence intervals are computed over 20 trainings.

	σ_0	Models					
		$\mathcal{N}$ -VAE	$\mathcal{P}^{0.1}$ -VAE	$\mathcal{P}^{0.3}$ -VAE	$\mathcal{P}^{0.8}$ -VAE	$\mathcal{P}^{1.0}$ -VAE	$\mathcal{P}^{1.2}$ -VAE
$\mathcal{L}_{\text{IWAE}}$	1	57.1 $\pm$ 0.2	57.1 $\pm$ 0.2	57.2 $\pm$ 0.2	56.9 $\pm$ 0.2	56.7 $\pm$ 0.2	56.6 $\pm$ 0.2
$\mathcal{L}_{\text{IWAE}}$	1.7	57.0 $\pm$ 0.2	56.8 $\pm$ 0.2	56.6 $\pm$ 0.2	55.9 $\pm$ 0.2	55.7 $\pm$ 0.2	55.6 $\pm$ 0.2

5.2 Mnist digits

The MNIST (LeCun and Cortes, 2010) dataset has been used in the literature for hierarchical modelling (Salakhutdinov et al., 2011; Saha et al., 2018). One can view the natural clustering in MNIST images as a hierarchy with each of the 10 classes being internal nodes of the hierarchy. We empirically assess whether our model can take advantage of such simple underlying hierarchical structure, first by measuring its generalisation capacity via the test marginal log-likelihood. Table 2 shows that our model outperforms its Euclidean counterpart, especially for low latent dimension. This can be interpreted through an information bottleneck perspective; as the latent dimensionality increases, the pressure on the embeddings quality decreases, hence the gain from the hyperbolic geometry is reduced (as observed by Nickel and Kiela (2017)). Also, by using the *Riemannian normal* distribution, we achieve slightly better results than with the *wrapped normal*.

Table 2: Negative test marginal likelihood estimates computed with 5000 samples. 95% confidence intervals are computed over 10 runs. * indicates numerically unstable settings.

	c	Dimensionality			
		2	5	10	20
$\mathcal{N}$ -VAE	(0)	144.5 $\pm$ 0.4	114.7 $\pm$ 0.1	100.2 $\pm$ 0.1	97.6 $\pm$ 0.1
$\mathcal{P}$ -VAE (Wrapped)	0.1	143.9 $\pm$ 0.5	115.5 $\pm$ 0.3	100.2 $\pm$ 0.1	97.2 $\pm$ 0.1
	0.2	144.2 $\pm$ 0.5	115.3 $\pm$ 0.3	100.0 $\pm$ 0.1	97.1 $\pm$ 0.1
	0.7	143.8 $\pm$ 0.6	115.1 $\pm$ 0.3	100.2 $\pm$ 0.1	97.5 $\pm$ 0.1
	1.4	144.0 $\pm$ 0.6	114.7 $\pm$ 0.1	100.7 $\pm$ 0.1	98.0 $\pm$ 0.1
$\mathcal{P}$ -VAE (Riemannian)	0.1	143.7 $\pm$ 0.6	115.2 $\pm$ 0.2	99.9 $\pm$ 0.1	97.0 $\pm$ 0.1
	0.2	143.8 $\pm$ 0.4	114.7 $\pm$ 0.3	99.7 $\pm$ 0.1	97.4 $\pm$ 0.1
	0.7	143.1 $\pm$ 0.4	114.1 $\pm$ 0.2	101.2 $\pm$ 0.2	*
	1.4	142.5 $\pm$ 0.4	115.5 $\pm$ 0.3	*	*

We conduct an ablation study to assess the usefulness of the *gyroplane* layer introduced in Section 3.2. To do so we estimate the test marginal log-likelihood for different choices of decoder. We select a multi-layer perceptron (MLP) to be the baseline decoder. We additionally compare to a MLP pre-composed by $\log_0$, which can be seen as a linearisation of the space around the centre of the ball. Figure 6 shows the relative performance improvement of decoders over the MLP baseline w.r.t. the latent space dimension. We observe that *linearising* the input of a MLP through the logarithm map slightly improves generalisation, and that using a *gyroplane* layer as the first layer of the decoder additionally improves generalisation. Yet, these performance gains appear to decrease as the latent dimensionality increases.

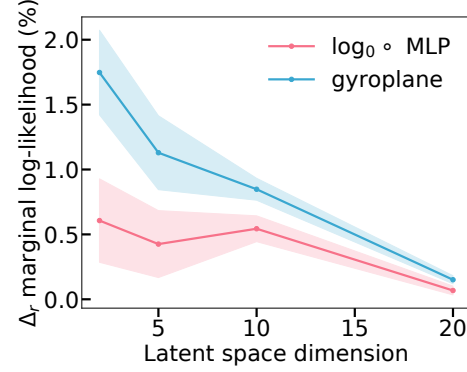


Figure 6: Decoder ablation study on MNIST with *wrapped* normal $\mathcal{P}^1$ -VAE. Baseline decoder is a MLP.

Second, we explore the learned latent representations of the trained $\mathcal{P}$ -VAE and $\mathcal{N}$ -VAE models shown in Figure 7. Qualitatively our $\mathcal{P}$ -VAE produces a clearer partitioning of the digits, in groupings of $\{4, 7, 9\}$, $\{0, 6\}$, $\{2, 3, 5, 8\}$ and $\{1\}$, with right-slanting $\{5, 8\}$ being placed separately from the non-slanting ones. Recall that distances increase towards the edge of the Poincaré ball. We quantitatively assess the quality of the embeddings by training a classifier predicting labels. Table 3 shows that the embeddings learned by our $\mathcal{P}$ -VAE model yield on average an 2% increase in accuracy over the digits. The full confusion matrices are shown in Figure 12 in Appendix.

Table 3: Per digit accuracy of a classifier trained on the learned latent 2-d embeddings. Results are averaged over 10 sets of embeddings and 5 classifier trainings.

Digits	0	1	2	3	4	5	6	7	8	9	Avg
$\mathcal{N}$ -VAE	89	97	81	75	59	43	89	78	68	57	73.6
$\mathcal{P}^{1.4}$ -VAE	94	97	82	79	69	47	90	77	68	53	75.6

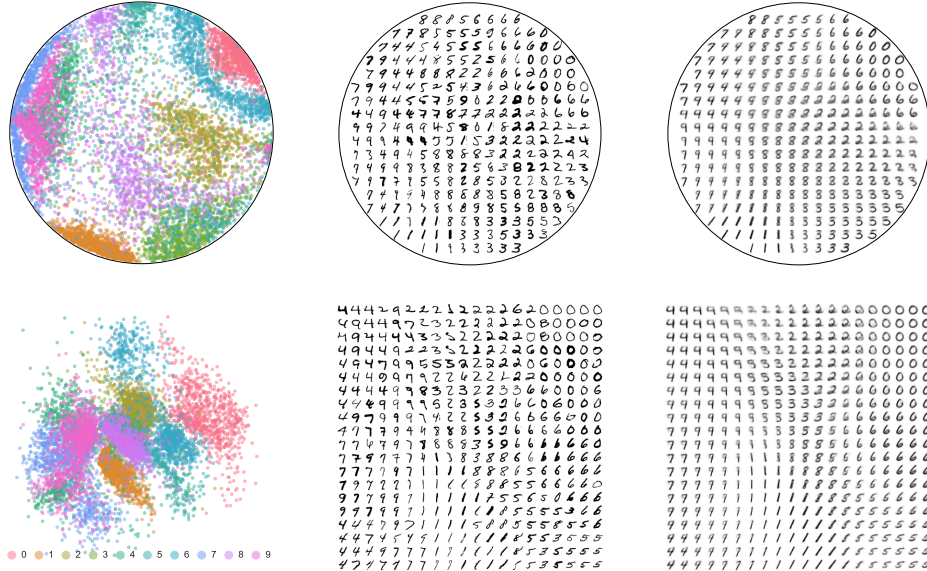


Figure 7: MNIST Posteriors mean (Left) sub-sample of digit images associated with posteriors mean (Middle) Model samples (Right) – for $\mathcal{P}^{1.4}$ -VAE (Top) and $\mathcal{N}$ -VAE (Bottom).

5.3 Graph embeddings

We evaluate the performance of a variational graph auto-encoder (VGAE) (Kipf and Welling, 2016) with Poincaré ball latent space for link prediction in networks. Edges in complex networks can typically be explained by a latent hierarchy over the nodes (Clauset et al., 2008). We believe the Poincaré ball latent space should help in terms of generalisation. We demonstrate these capabilities on three network datasets: a graph of Ph.D. advisor-advisee relationships (Nooy et al., 2011), a phylogenetic tree expressing genetic heritage (Hofbauer et al., 2016; Sanderson and Eriksson, 1994) and a biological set representing disease relationships (Goh et al., 2007; Rossi and Ahmed, 2015).

We follow the VGAE model, which maps the adjacency matrix $\mathbf{A}$ to node embeddings $\mathbf{Z}$ through a graph convolutional network (GCN), and reconstructs $\mathbf{A}$ by predicting edge probabilities from the node embeddings. In order to take into account the latent space geometry, we parametrise the probability of an edge by $p(\mathbf{A}_{ij} = 1 | \mathbf{z}_i, \mathbf{z}_j) = 1 - \tanh(d_{\mathcal{M}}(\mathbf{z}_i, \mathbf{z}_j)) \in (0, 1]$ with $d_{\mathcal{M}}$ the latent geodesic metric.

We set the latent dimension to 5. We follow the training and evaluation procedures introduced in Kipf and Welling (2016). Models are trained on an incomplete adjacency matrix where some of the edges have randomly been removed. A test set is formed from previously removed edges and an equal number of randomly sampled pairs of unconnected nodes. We report in Table 4 the *area under the ROC curve* (AUC) and *average precision* (AP) evaluated on the test set. It can be observed that the $\mathcal{P}$ -VAE performs better than its Euclidean counterpart in terms of generalisation to unseen edges.

Table 4: Results on network link prediction. 95% confidence intervals are computed over 40 runs.

	Phylogenetic		CS PhDs		Diseases	
	AUC	AP	AUC	AP	AUC	AP
$\mathcal{N}$ -VAE	54.2 $\pm$ 2.2	54.0 $\pm$ 2.1	56.5 $\pm$ 1.1	56.4 $\pm$ 1.1	89.8 $\pm$ 0.7	91.8 $\pm$ 0.7
$\mathcal{P}$ -VAE	59.0$\pm$1.9	55.5 $\pm$ 1.6	59.8$\pm$1.2	56.7 $\pm$ 1.2	92.3$\pm$0.7	93.6$\pm$0.5

6 Conclusion

In this paper we have explored VAEs with a Poincaré ball latent space. We gave a thorough treatment of two canonical – *wrapped* and *maximum entropy* – normal generalisations on that space, and a rigorous analysis of the difference between the two. We derived the necessary ingredients for training such VAEs, namely efficient and reparametrisable sampling schemes, along with probability density functions for these two distributions. We introduced a decoder architecture explicitly taking into account the hyperbolic geometry, and empirically showed that it is crucial for the hyperbolic latent space to be useful. We empirically demonstrated that endowing a VAE with a Poincaré ball latent space can be beneficial in terms of model generalisation and can yield more interpretable representations if the data has hierarchical structure.

There are a number of interesting future directions. There are many models of hyperbolic geometry, and several have been considered in a gradient-based setting. Yet, it is still unclear which models should be preferred and which of their properties matter. Also, it would be useful to consider principled ways of assessing whether a given dataset has an underlying hierarchical structure, in the same way that topological data analysis (Pascucci et al., 2011) attempts to discover the topologies that underlie datasets.

Acknowledgments We are extremely grateful to Adam Foster, Phillipe Gagnon and Emmanuel Chevallier for their help. EM, YWT’s research leading to these results received funding from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 617071 and they acknowledge Microsoft Research and EPSRC for funding EM’s studentship, and EPSRC grant agreement no. EP/N509711/1 for funding CL’s studentship.

References

- Beltrami, E. (1868). *Teoria fondamentale degli spazii di curvatura costante: memoria*. F. Zanetti.
- Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828.
- Bonnabel, S. (2013). Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229.
- Box, G. E. P. and Muller, M. E. (1958). A note on the generation of random normal deviates. *Ann. Math. Statist.*, 29(2):610–611.
- Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Importance Weighted Autoencoders. *arXiv.org*.
- Chamberlain, B. P., Clough, J., and Deisenroth, M. P. (2017). Neural Embeddings of Graphs in Hyperbolic Space. *arXiv.org*.
- Chevallier, E., Barbaresco, F., and Angulo, J. (2015). Probability density estimation on the hyperbolic space applied to radar processing. In Nielsen, F. and Barbaresco, F., editors, *Geometric Science of Information*, pages 753–761, Cham. Springer International Publishing.
- Clauset, A., Moore, C., and Newman, M. E. J. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101.
- Coates, A., Lee, H., and Ng, A. (2011). An analysis of single-layer networks in unsupervised feature learning. In Gordon, G., Dunson, D., and Dudík, M., editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *JMLR Workshop and Conference Proceedings*, pages 215–223. JMLR W&CP.
- Collins, A. and Quillian, M. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8:240–248.
- Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection*. Murray, London. or the Preservation of Favored Races in the Struggle for Life.
- Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. (2018). Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*.
- Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification (2Nd Edition)*. Wiley-Interscience, New York, NY, USA.
- Falorsi, L., de Haan, P., Davidson, T. R., De Cao, N., Weiler, M., Forré, P., and Cohen, T. S. (2018). Explorations in Homeomorphic Variational Auto-Encoding. *arXiv e-prints*, page arXiv:1807.04689.
- Figurnov, M., Mohamed, S., and Mnih, A. (2018). Implicit reparameterization gradients. In *International Conference on Neural Information Processing Systems*, pages 439–450.
- Ganea, O.-E., Bécigneul, G., and Hofmann, T. (2018). Hyperbolic neural networks. In *International Conference on Neural Information Processing Systems*, pages 5350–5360.
- Ghahramani, Z., Jordan, M. I., and Adams, R. P. (2010). Tree-structured stick breaking for hierarchical data. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 19–27. Curran Associates, Inc.
- Goh, K.-I., Cusick, M. E., Valle, D., Childs, B., Vidal, M., and Barabási, A.-L. (2007). The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690.

- Grattarola, D., Livi, L., and Alippi, C. (2019). Adversarial autoencoders with constant-curvature latent manifolds. *Appl. Soft Comput.*, 81.
- Griffiths, T. L., Jordan, M. I., Tenenbaum, J. B., and Blei, D. M. (2004). Hierarchical topic models and the nested chinese restaurant process. In Thrun, S., Saul, L. K., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*, pages 17–24. MIT Press.
- Hauberg, S. (2018). Directional statistics with the spherical normal distribution. In *Proceedings of 2018 21st International Conference on Information Fusion, FUSION 2018*, pages 704–711. IEEE.
- Heller, K. A. and Ghahramani, Z. (2005). Bayesian hierarchical clustering. In *International Conference on Machine Learning, ICML '05*, pages 297–304, New York, NY, USA. ACM.
- Hofbauer, W., Forrest, L., M. Hollingsworth, P., and Hart, M. (2016). Preliminary insights from dna barcoding into the diversity of mosses colonising modern building surfaces. *Bryophyte Diversity and Evolution*, 38:1.
- Hsu, E. P. (2008). A brief introduction to brownian motion on a riemannian manifold.
- Huang, F. J. and LeCun, Y. (2006). Large-scale learning with SVM and convolutional for generic object categorization. In *CVPR (1)*, pages 284–291. IEEE Computer Society.
- Keil, F. (1979). *Semantic and Conceptual Development: An Ontological Perspective*. Cognitive science series. Harvard University Press.
- Kingma, D. P. and Ba, J. (2016). Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Kipf, T. N. and Welling, M. (2016). Variational graph auto-encoders. *Workshop on Bayesian Deep Learning, NIPS*.
- Larsen, J., Szymkowiak, A., and Hansen, L. K. (2001). Probabilistic hierarchical clustering with labeled and unlabeled data.
- LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.
- Ley, C. and Verdebout, T. (2017). *Modern Directional Statistics*. New York: Chapman and Hall/CRC.
- Li, H., Lindenbaum, O., Cheng, X., and Cloninger, A. (2019). Variational Random Walk Autoencoders. *arXiv.org*.
- Mardia, K. and Jupp, P. (2009). *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley.
- Mardia, K. V. (1975). Statistics of directional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 37(3):349–371.
- Nagano, Y., Yamaguchi, S., Fujita, Y., and Koyama, M. (2019). A Differentiable Gaussian-like Distribution on Hyperbolic Space for Gradient-Based Learning. In *International Conference on Machine Learning (ICML)*.
- Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pages 6341–6350.
- Nickel, M. and Kiela, D. (2018). Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. In *International Conference on Machine Learning (ICML)*.
- Nooy, W. D., Mrvar, A., and Batagelj, V. (2011). *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, New York, NY, USA.
- Ovinnikov, I. (2018). Poincaré Wasserstein Autoencoder. *NeurIPS Workshop on Bayesian Deep Learning*, pages 1–8.

- Paeng, S.-H. (2011). Brownian motion on manifolds with time-dependent metrics and stochastic completeness. *Journal of Geometry and Physics*, 61(5):940 – 946.
- Pascucci, V., Tricoche, X., Hagen, H., and Tierny, J. (2011). *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*. Springer Publishing Company, Incorporated, 1st edition.
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS-W*.
- Pennec, X. (2006). Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127.
- Petersen, P. (2006). *Riemannian Geometry*. Springer-Verlag New York.
- R. Gilks, W. and Wild, P. (1992). Adaptive rejection sampling for gibbs sampling. 41:337–348.
- Rey, L. A. P., Menkovski, V., and Portegies, J. W. (2019). Diffusion Variational Autoencoders. *CoRR*.
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conference on Machine Learning (ICML)*.
- Rossi, R. A. and Ahmed, N. K. (2015). The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI’15, pages 4292–4293. AAAI Press.
- Roy, D. M., Kemp, C., Mansinghka, V. K., and Tenenbaum, J. B. (2006). Learning annotated hierarchies from relational data. In *NIPS*, pages 1185–1192. MIT Press.
- Saha, S., Varma, G., and Jawahar, C. V. (2018). Class2str: End to end latent hierarchy learning. *International Conference on Pattern Recognition (ICPR)*, pages 1000–1005.
- Said, S., Bombrun, L., and Berthoumieu, Y. (2014). New riemannian priors on the univariate normal model. *Entropy*, 16(7):4015–4031.
- Sala, F., De Sa, C., Gu, A., and Re, C. (2018). Representation tradeoffs for hyperbolic embeddings. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4460–4469, Stockholmsmässan, Stockholm Sweden. PMLR.
- Salakhutdinov, R. R., Tenenbaum, J. B., and Torralba, A. (2011). One-shot learning with a hierarchical nonparametric bayesian model. In *ICML Unsupervised and Transfer Learning*.
- Sanderson, M. J., M. J. D. W. P. and Eriksson, T. (1994). Treebase: a prototype database of phylogenetic analyses and an interactive tool for browsing the phylogeny of life. *American Journal of Botany*.
- Sarkar, R. (2012). Low distortion delaunay embedding of trees in hyperbolic plane. In van Kreveld, M. and Speckmann, B., editors, *Graph Drawing*, pages 355–366, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Teh, Y. W., Daume III, H., and Roy, D. M. (2008). Bayesian agglomerative clustering with coalescents. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20*, pages 1473–1480. Curran Associates, Inc.
- Tifrea, A., Becigneul, G., and Ganea, O.-E. (2019). Poincare glove: Hyperbolic word embeddings. In *International Conference on Learning Representations (ICLR)*.
- Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. (2018). Wasserstein auto-encoders. In *International Conference on Learning Representations*.
- Ungar, A. A. (2008). A gyrovector space approach to hyperbolic geometry. *Synthesis Lectures on Mathematics and Statistics*, 1(1):1–194.

A Evidence Lower Bound

The ELBO can readily be extended for Riemannian latent spaces by applying Jensen’s inequality w.r.t. the metric induced measure $d\mathcal{M}$ which yield

$$\begin{aligned}
\ln p(\mathbf{x}) &= \ln \int_{\mathcal{Z}=\mathcal{M}} p_\theta(\mathbf{x}, \mathbf{z}) d\mathcal{M}(\mathbf{z}) = \ln \int_{\mathcal{M}} p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathcal{M}(\mathbf{z}) \\
&= \ln \int_{\mathcal{M}} \frac{p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} q_\phi(\mathbf{z}|\mathbf{x}) d\mathcal{M}(\mathbf{z}) \\
&\geq \int_{\mathcal{M}} \ln \frac{p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} q_\phi(\mathbf{z}|\mathbf{x}) d\mathcal{M}(\mathbf{z}) \\
&= \int_{\mathcal{M}} [\ln p_\theta(\mathbf{x}|\mathbf{z}) - \ln p(\mathbf{z}) - \ln q_\phi(\mathbf{z}|\mathbf{x})] q_\phi(\mathbf{z}|\mathbf{x}) d\mathcal{M}(\mathbf{z}) \\
&= \mathbb{E}_{\mathbf{z} \sim q_\phi(\cdot|\mathbf{x})} [\ln p_\theta(\mathbf{x}|\mathbf{z}) + \ln p(\mathbf{z}) - \ln q_\phi(\mathbf{z}|\mathbf{x})] \\
&\triangleq \mathcal{L}_{\mathcal{M}}(\mathbf{x}; \theta, \phi) \\
&\approx \sum_k \ln p_\theta(\mathbf{x}|\mathbf{z}^k) + \ln p(\mathbf{z}^k) - \ln q_\phi(\mathbf{z}^k|\mathbf{x}), \quad \mathbf{z}^k \sim q_\phi(\cdot|\mathbf{x}) \sqrt{|G(\cdot)|}
\end{aligned}$$

B Hyperbolic normal distributions

In this section, we first review some canonical generalisation of the normal distributions to Riemannian manifolds, and then introduce in more details the *Riemannian* and *wrapped* normal distributions on the Poincaré ball. Finally, we give architecture and training details about the conducted experiments.

B.1 Probability measures on Riemannian manifolds

Probability measures and random vectors can intrinsically be defined on Riemannian manifolds so as to model uncertainty on non-flat spaces (Pennec, 2006). The Riemannian metric $G(\mathbf{z})$ induces an infinitesimal volume element on each tangent space $\mathcal{T}_{\mathbf{z}}\mathcal{M}$, and thus a measure on the manifold,

$$d\mathcal{M}(\mathbf{z}) = \sqrt{|G(\mathbf{z})|} d\mathbf{z}, \quad (4)$$

with $d\mathbf{z}$ being the Lebesgue measure. Random variables $\mathbf{z} \in \mathcal{M}$ would naturally be characterised by the Radon-Nikodym derivative of a measure ν w.r.t. the Riemannian measure $d\mathcal{M}(\cdot)$ (assuming absolute continuity)

$$f(\mathbf{z}) = \frac{d\nu(\mathbf{z})}{d\mathcal{M}(\mathbf{z})}.$$

Since the normal distribution plays such a canonical role in statistics, generalising it to manifold is of interest. Given a Fréchet expectation $\boldsymbol{\mu} \in \mathcal{M}$ – defined as minimisers of $\int_{\mathcal{M}} d_{\mathcal{M}}(\boldsymbol{\mu}, \mathbf{z})^2 p(\mathbf{z}) d\mathcal{M}(\mathbf{z})$ – and a dispersion parameter $\sigma > 0$ (generally not equal to the standard deviation), several properties ought to be verified by such generalised normal distributions. Such a distribution should tend towards a delta function at $\boldsymbol{\mu}$ when $\sigma \rightarrow 0$ and to an (improper for non-compact) uniform distribution when $\sigma \rightarrow \infty$. Also, as the curvature tends to 0, one should recover the vanilla normal distribution. Hereby, we review canonical generalisations of the normal distribution, which have different theoretical and computational advantages.

Maximum entropy normal The property that Pennec (2006) takes for granted is the maximization of the entropy given a mean and a covariance matrix, yielding in the isotropic setting

$$\frac{d\nu^R(\mathbf{z}|\boldsymbol{\mu}, \sigma^2)}{d\mathcal{M}(\mathbf{z})} = \mathcal{N}_{\mathcal{M}}^R(\mathbf{z}|\boldsymbol{\mu}, \sigma^2) = \frac{1}{Z^R} \exp\left(-\frac{d_{\mathcal{M}}(\boldsymbol{\mu}, \mathbf{z})^2}{2\sigma^2}\right), \quad (5)$$

with $d_{\mathcal{M}}$ being the Riemannian distance on the manifold induced by the tensor metric. Such a formulation – sometimes referred as *Riemannian Normal* distribution – is used by Said et al. (2014) in the Poincaré half-plane, or by Hauberg (2018) in the hypersphere $\mathbb{S}^d$. Sampling from such distributions and computing the normalising constant – especially in the anisotropic setting – is usually challenging.

Wrapped normal Another generalisation is defined by taking the image by the exponential map of a Gaussian distribution on the tangent space centered at the mean value. Such a distribution has been referred in literature as *wrapped*, *push-forward*, *exp-map* or *tangential* normal distribution. Sampling is therefore straightforward. The pdf is then readily available through the change of variable formula if one can compute the Jacobian of the exponential map (or its inverse). Hence such a distribution is attractive from a computational perspective. Grattarola et al. (2019) and Nagano et al. (2019) rely on such a distribution defined on the hyperboloid model. Wrapped distributions are often encountered in the *directional* statistics (Ley and Verdebout, 2017; Hauberg, 2018).

Restricted normal What is more, for sub-manifolds of $\mathbb{R}^n$, one can consider the restriction of a normal distribution pdf to the manifold. This yields the Von Mises distribution on $\mathbb{S}^1$ and the Von Mises-Fisher distribution on $\mathbb{S}^d$ (Hauberg, 2018) and the Stiefel manifold. It is the maximum entropy distribution but with respect to the ambient euclidean metric (Mardia, 1975).

Diffusion normal Yet another generalisation arises by defining the normal pdf through the *heat kernel*, or fundamental solution of the heat equation, $K : \mathbb{R}^+ \times \mathcal{M} \times \mathcal{M} \rightarrow \mathcal{M}$,

$$\mathcal{N}_{\mathcal{M}}^{\Delta}(z|\mu, \sigma^2) = K(\sigma^2/2, \mu, z). \quad (6)$$

See for instance Hsu (2008) for an introduction of Brownian motion on Riemannian manifolds and Paeng (2011) for conditions on existence and uniqueness of the kernel. Sampling amounts to simulating a Brownian motion, which may be challenging for non sub-manifolds of $\mathbb{R}^n$. Closed form solutions of the heat kernel is available for some manifolds such as spheres or flat tori, otherwise numerical approximations can be used. Such a distribution has been used in a VAE setting (Rey et al., 2019; Li et al., 2019).

Other than normal distributions Of course one needs not to restrict itself to generalisations of the normal distribution. For instance, one could consider a wrapped spherical Student-t as $z \sim \exp_{\mu\#} S_t(0, \nu)$ or a Riemannian Student-t with density proportional to $(1 + d_{\mathcal{M}}(z, \mu)^2/\nu)^{(-\nu+1)/2}$ (by making sure that this density is $d\mathcal{M}$ -integrable).

B.2 Hyperbolic polar coordinates

In this subsection, we review the hyperbolic polar change of coordinates allowing us to reparametrise hyperbolic normal distributions in a similar fashion than the Box–Muller transform (Box and Muller, 1958).

Polar coordinates Euclidean polar coordinates, express points $z \in \mathbb{R}^d$ through a radius $r \geq 0$ and a direction $\alpha \in \mathbb{S}^{d-1}$ such that $z = r\alpha$. Yet, one could choose another *pole* (or *reference point*) $\mu \neq \mathbf{0}$ such that $z = \mu + r\alpha$. Consequently, $r = d_E(\mu, z)$. An analogous change of variables can also be constructed in Riemannian manifolds relying on the exponential map instead of the addition operator. Given a *pole* $\mu \in \mathbb{B}_c^d$, the point of hyperbolic polar coordinates $z = (r, \alpha)$ is defined as $z = \gamma(r)$, with $r = d_p^c(\mu, z)$ and $\gamma : \mathbb{R}^+ \rightarrow \mathbb{B}_c^d$ a curve such that $\gamma'(0) = \alpha \in \mathbb{S}^{d-1}$. Hence $z = \exp_{\mu}^c\left(\frac{r}{\lambda_{\mu}^c}\alpha\right)$ since $d_p^c(\mu, z) = \|\ln_{\mu}^c(x)\|_{\mu} = \|\frac{r}{\lambda_{\mu}^c}\alpha\|_{\mu} = r$.

Tensor metric We derive below the expression of the Poincaré ball metric in such hyperbolic polar coordinate, for the specific setting where $\mu = \mathbf{0}$: $z = \exp_{\mathbf{0}}^c(\frac{r}{2}\alpha)$. Switching to Euclidean polar coordinate we get

$$\begin{aligned} ds_{\mathbb{B}_c^d}^2 &= (\lambda_z^c)^2(dz_1^2 + \dots + dz_d^2) = \frac{4}{(1 - c\|x\|^2)^2} dz^2 \\ &= \frac{4}{(1 - c\rho^2)^2} (d\rho^2 + \rho^2 ds_{\mathbb{S}^{d-1}}^2). \end{aligned} \quad (7)$$

Let's define $r = d_p^c(\mathbf{0}, z) = L(\gamma)$, with γ being the geodesic joining $\mathbf{0}$ and z . Since such a geodesic is the segment $[\mathbf{0}, z]$, we have

$$r = \int_0^{\rho} \lambda_t^c dt = \int_0^{\rho} \frac{2}{1 - ct^2} dt = \int_0^{\sqrt{c}\rho} \frac{2}{1 - t^2} \frac{dt}{\sqrt{c}} = \frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c}\rho).$$

3. Continuous Hierarchical Representations with Poincaré Variational Autoencoders

Plugging $\rho = \frac{1}{\sqrt{c}} \tanh(\sqrt{c}\frac{r}{2})$ (and $d\rho = (1 - c\rho^2)/(2dr)$) into Eq 7 yields

$$\begin{aligned} ds_{\mathbb{B}_c^d}^2 &= \frac{4}{(1 - c\rho^2)^2} \frac{1}{4} (1 - c\rho^2)^2 dr^2 + \left(2 \frac{\rho}{1 - c\rho^2} \right)^2 ds_{\mathbb{S}^{d-1}}^2 \\ &= dr^2 + \left(2 \frac{\frac{1}{\sqrt{c}} \tanh(\sqrt{c}\frac{r}{2})}{1 - c \left(\frac{1}{\sqrt{c}} \tanh(\sqrt{c}\frac{r}{2}) \right)^2} \right)^2 ds_{\mathbb{S}^{d-1}}^2 \\ &= dr^2 + \left(\frac{1}{\sqrt{c}} \sinh(\sqrt{c}r) \right)^2 ds_{\mathbb{S}^{d-1}}^2. \end{aligned} \quad (8)$$

The Euclidean line element is recovered when $c \rightarrow 0$

$$ds_{\mathbb{R}^d}^2 = dr^2 + r^2 ds_{\mathbb{S}^{d-1}}^2. \quad (9)$$

In an appropriate orthonormal basis of $\mathcal{T}_{\mu}\mathbb{B}_c^d$, the hyperbolic polar coordinate leads to the following expression of the matrix of the metric

$$G(\mathbf{z}) = \begin{pmatrix} 1 & 0 \\ 0 & \left(\frac{\sinh(\sqrt{c}r)}{\sqrt{c}r} \right)^2 \mathbf{I}_{d-1} \end{pmatrix}. \quad (10)$$

Hence, the density of the Riemannian measure with respect to the image of the Lebesgue measure of $\mathcal{T}_{\mu}\mathbb{B}_c^d$ by $\exp_{\mu}^c$ is given by

$$\sqrt{|G(\mathbf{z})|} = \left(\frac{\sinh(\sqrt{c}r)}{\sqrt{c}r} \right)^{d-1}. \quad (11)$$

This result holds for any *reference point* $\mu \in \mathbb{B}_c^d$, with $r = d_p^c(\mu, \mathbf{z})$, since the metric induced measure is invariant under the isometries of the manifold (i.e. Möbius transformations). This result can also be found in Chevallier et al. (2015); Said et al. (2014). Also, the fact that the line element $ds_{\mathbb{B}_c^d}^2$ and equivalently the metric G only depends on the radius in hyperbolic polar coordinate, is a consequence of the hyperbolic space's isotropy.

Integration We now make use of the aforementioned hyperbolic polar coordinates to integrate functions following Said et al. (2014). The integral of a function $f : \mathbb{B}_c^d \rightarrow \mathbb{R}$ can be computed by using polar coordinates,

$$\begin{aligned} \int_{\mathbb{B}_c^d} f(\mathbf{z}) d\mathcal{M}(\mathbf{z}) &= \int_{\mathbb{B}_c^d} f(\mathbf{z}) \sqrt{|G(\mathbf{z})|} d\mathbf{z} \\ &= \int_{\mathcal{T}_{\mu}\mathbb{B}_c^d \cong \mathbb{R}^d} f(\mathbf{v}) \sqrt{|G(\mathbf{v})|} d\mathbf{v} \end{aligned} \quad (12)$$

$$\begin{aligned} &= \int_{\mathbb{R}_+} \int_{\mathbb{S}^{d-1}} f(r) \sqrt{|G(r)|} dr r^{d-1} ds_{\mathbb{S}^{d-1}} \\ &= \int_{\mathbb{R}_+} \int_{\mathbb{S}^{d-1}} f(r) \left(\frac{\sinh(\sqrt{c}r)}{\sqrt{c}r} \right)^{d-1} dr r^{d-1} ds_{\mathbb{S}^{d-1}} \\ &= \int_{\mathbb{R}_+} \int_{\mathbb{S}^{d-1}} f(r) \left(\frac{\sinh(\sqrt{c}r)}{\sqrt{c}} \right)^{d-1} dr ds_{\mathbb{S}^{d-1}}. \end{aligned} \quad (13)$$

B.3 Wrapped hyperbolic normal distribution on $\mathbb{B}_c^d$

Anisotropic The *wrapped normal* distribution considers a normal distribution in the tangent space $\mathcal{T}_{\mu}\mathbb{B}_c^d$ being pushed-forward along the exponential map. One can obtain sampled as follow

$$\mathbf{z} = \exp_{\mu}^c \left(G(\mu)^{-\frac{1}{2}} \mathbf{v} \right) = \exp_{\mu}^c \left(\frac{\mathbf{v}}{\lambda_{\mu}^c} \right), \text{ with } \mathbf{v} \sim \mathcal{N}(\cdot | \mathbf{0}, \Sigma). \quad (14)$$

Then, its density is given by

$$\begin{aligned}\mathcal{N}_{\mathbb{B}_c^d}^{\mathbf{W}}(\mathbf{z}|\boldsymbol{\mu}, \Sigma) &= \mathcal{N}\left(G(\boldsymbol{\mu})^{1/2} \log_{\boldsymbol{\mu}}(\mathbf{z}) \mid \mathbf{0}, \Sigma\right) \left(\frac{\sqrt{c} d_p^c(\boldsymbol{\mu}, \mathbf{z})}{\sinh(\sqrt{c} d_p^c(\boldsymbol{\mu}, \mathbf{z}))}\right)^{d-1} \\ &= \mathcal{N}\left(\lambda_{\boldsymbol{\mu}}^c \log_{\boldsymbol{\mu}}(\mathbf{z}) \mid \mathbf{0}, \Sigma\right) \left(\frac{\sqrt{c} d_p^c(\boldsymbol{\mu}, \mathbf{z})}{\sinh(\sqrt{c} d_p^c(\boldsymbol{\mu}, \mathbf{z}))}\right)^{d-1}\end{aligned}\quad (15)$$

with $G(\boldsymbol{\mu})^{1/2}$ the unique square-root matrix of $G(\boldsymbol{\mu})$ (thanks to the positive definiteness of the metric tensor). This can be shown by plugging this density as f in Equation (12) with $\mathbf{v} = r\boldsymbol{\alpha} = \lambda_{\boldsymbol{\mu}}^c \log_{\boldsymbol{\mu}}(\mathbf{z})$, we get

$$\begin{aligned}\int_{\mathbb{B}_c^d} \mathcal{N}_{\mathbb{B}_c^d}^{\mathbf{W}}(\mathbf{z}|\boldsymbol{\mu}, \Sigma) d\mathcal{M}(\mathbf{z}) &= \int_{\mathcal{T}_{\boldsymbol{\mu}} \mathbb{B}_c^d \cong \mathbb{R}^d} \mathcal{N}(\mathbf{v} \mid \mathbf{0}, \Sigma) \left(\frac{\sqrt{c} \|\mathbf{v}\|_2}{\sinh(\sqrt{c} \|\mathbf{v}\|_2)}\right)^{d-1} \sqrt{|G(\mathbf{v})|} d\mathbf{v} \\ &= \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{v} \mid \mathbf{0}, \Sigma) \left(\frac{\sqrt{c} \|\mathbf{v}\|_2}{\sinh(\sqrt{c} \|\mathbf{v}\|_2)}\right)^{d-1} \left(\frac{\sinh(\sqrt{c} \|\mathbf{v}\|_2)}{\sqrt{c} \|\mathbf{v}\|_2}\right)^{d-1} d\mathbf{v} \\ &= \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{v} \mid \mathbf{0}, \Sigma) d\mathbf{v}.\end{aligned}$$

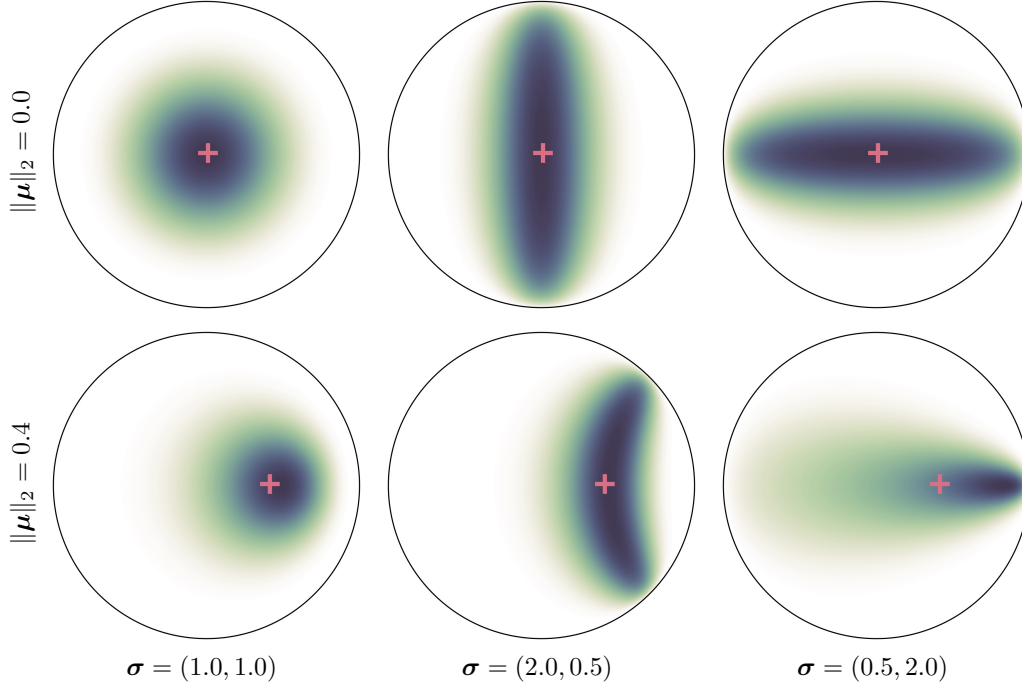


Figure 8: Anisotropic wrapped normal probability measures for Fréchet means $\boldsymbol{\mu}$ (red +), concentrations $\Sigma = \text{diag}(\boldsymbol{\sigma})$ and $c = 1$.

Isotropic In the isotropic setting, we therefore get

$$\int_{\mathbb{B}_c^d} \mathcal{N}_{\mathbb{B}_c^d}^{\mathbf{W}}(\mathbf{z}|\boldsymbol{\mu}, \sigma^2) d\mathcal{M}(\mathbf{z}) = \int_{\mathbb{R}_+} \int_{\mathbb{S}^{d-1}} \frac{1}{Z^{\mathbb{R}}} e^{-\frac{r^2}{2\sigma^2}} r^{d-1} dr d\mathbb{S}^{d-1}. \quad (16)$$

The hyperbolic radius $r = d_p^c(\boldsymbol{\mu}, \mathbf{z})$ consequently follows the usual χ distribution with density

$$\rho^{\mathbf{W}}(r) \propto \mathbb{1}_{\mathbb{R}_+}(r) e^{-\frac{r^2}{2\sigma^2}} r^{d-1}, \quad (17)$$

and the density of the *wrapped* normal given by

$$\mathcal{N}_{\mathbb{B}_c^d}^{\mathbf{W}}(\mathbf{z}|\boldsymbol{\mu}, \sigma^2) = \frac{d\nu^{\mathbf{W}}(\mathbf{z}|\boldsymbol{\mu}, \sigma^2)}{d\mathcal{M}(\mathbf{z})} = (2\pi\sigma^2)^{-d/2} \exp\left(-\frac{d_p^c(\boldsymbol{\mu}, \mathbf{z})^2}{2\sigma^2}\right) \left(\frac{\sqrt{c} d_p^c(\boldsymbol{\mu}, \mathbf{z})}{\sinh(\sqrt{c} d_p^c(\boldsymbol{\mu}, \mathbf{z}))}\right)^{d-1}.$$

B.4 Maximum entropy hyperbolic normal distribution on $\mathbb{B}_c^d$

Alternatively, by considering the maximum entropy generalisation of the normal distribution one gets (Pennec, 2006)

$$\mathcal{N}_{\mathbb{B}_c^d}^R(\mathbf{z}|\boldsymbol{\mu}, \sigma^2) = \frac{d\nu^R(\mathbf{z}|\boldsymbol{\mu}, \sigma^2)}{d\mathcal{M}(\mathbf{z})} = \frac{1}{Z^R} \exp\left(-\frac{d_p^c(\boldsymbol{\mu}, \mathbf{z})^2}{2\sigma^2}\right). \quad (18)$$

Such a pdf can be computed pointwise once Z^R is known, which we derive in Appendix B.4.3. Also, we observe that as c and σ get smaller (resp. bigger), the *Riemannian* normal pdf gets closer (resp. further) to the *wrapped* normal pdf.

B.4.1 Reparametrisation

Plugging the *Riemannian* normal density as f in Equation (13), with $r = d_p^c(\boldsymbol{\mu}, \mathbf{z})$, we have

$$\begin{aligned} \int_{\mathbb{B}_c^d} \mathcal{N}_{\mathbb{B}_c^d}^R(\mathbf{z}|\boldsymbol{\mu}, \sigma^2) d\mathcal{M}(\mathbf{z}) &= \int_{\mathbb{R}_+} \int_{\mathbb{S}^{d-1}} \frac{1}{Z^R} e^{-\frac{r^2}{2\sigma^2}} \left(\frac{\sinh(\sqrt{c}r)}{\sqrt{c}}\right)^{d-1} dr d\mathbb{S}^{d-1} \\ &= \frac{1}{Z^R} \left(\int_{\mathbb{R}_+} e^{-\frac{r^2}{2\sigma^2}} \left(\frac{\sinh(\sqrt{c}r)}{\sqrt{c}}\right)^{d-1} dr \right) \left(\int_{\mathbb{S}^{d-1}} d\mathbb{S}^{d-1} \right) \end{aligned} \quad (19)$$

Hence, samples $\mathbf{z} \sim \mathcal{N}_{\mathbb{B}_c^d}^R(\mathbf{z}|\boldsymbol{\mu}, \sigma^2)d\mathcal{M}(\mathbf{z})$ can be reparametrised as

$$\mathbf{z} = \exp_{\boldsymbol{\mu}}^c\left(\frac{r}{\lambda_{\boldsymbol{\mu}}^c}\boldsymbol{\alpha}\right) \quad (20)$$

with the direction $\boldsymbol{\alpha}$ being uniformly distributed on the hypersphere $\mathbb{S}^{d-1}$, i.e.

$$\boldsymbol{\alpha} \sim \mathcal{U}(\mathbb{S}^{d-1})$$

and the hyperbolic radius $r = d_p^c(\boldsymbol{\mu}, \mathbf{z})$ distributed according to the following density (w.r.t the Lebesgue measure)

$$\rho^R(r) = \frac{\mathbb{1}_{\mathbb{R}_+}(r)}{Z_r^R} e^{-\frac{r^2}{2\sigma^2}} \left(\frac{\sinh(\sqrt{c}r)}{\sqrt{c}}\right)^{d-1}. \quad (21)$$

Developed expression By expanding the sinh term using the binomial formula, we get

$$\begin{aligned} \rho^R(r) &= \frac{\mathbb{1}_{\mathbb{R}_+}(r)}{Z_r^R} e^{-\frac{r^2}{2\sigma^2}} \left(\frac{\sinh(\sqrt{c}r)}{\sqrt{c}}\right)^{d-1} \\ &= \frac{\mathbb{1}_{\mathbb{R}_+}(r)}{Z_r^R} e^{-\frac{r^2}{2\sigma^2}} \left(\frac{e^{\sqrt{c}r} - e^{-\sqrt{c}r}}{2\sqrt{c}}\right)^{d-1} \\ &= \frac{\mathbb{1}_{\mathbb{R}_+}(r)}{Z_r^R} e^{-\frac{r^2}{2\sigma^2}} \frac{1}{(2\sqrt{c})^{d-1}} \sum_{k=0}^{d-1} \binom{d-1}{k} (e^{\sqrt{c}r})^{d-1-k} (-e^{-\sqrt{c}r})^k \\ &= \frac{\mathbb{1}_{\mathbb{R}_+}(r)}{Z_r^R} \frac{1}{(2\sqrt{c})^{d-1}} e^{-\frac{r^2}{2\sigma^2}} \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} e^{(d-1-2k)\sqrt{c}r} \\ &= \frac{\mathbb{1}_{\mathbb{R}_+}(r)}{Z_r^R} \frac{1}{(2\sqrt{c})^{d-1}} \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} e^{-\frac{r^2}{2\sigma^2} + (d-1-2k)\sqrt{c}r} \\ &= \frac{\mathbb{1}_{\mathbb{R}_+}(r)}{Z_r^R} \frac{1}{(2\sqrt{c})^{d-1}} \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} e^{\frac{(d-1-2k)^2}{2} c \sigma^2} e^{-\frac{1}{2\sigma^2} [r - (d-1-2k)\sqrt{c}\sigma^2]^2}. \end{aligned} \quad (22)$$

B.4.2 Sampling

In this section we detail the sampling scheme that we use for the Riemannian normal distribution $\mathcal{N}_{\mathbb{B}_c^d}^R(\cdot | \mu, \sigma^2)$, along with a reparametrisation which allows to compute gradients with respect to the parameters μ and σ .

Sampling challenges due to the hyperbolic geometry Several properties of the Euclidean space do not generalise to the hyperbolic setting, unfortunately hardening the task of obtaining samples from *Riemannian* normal distributions. First, one can factorise a normal density through the space’s dimensions – thanks to the Pythagorean theorem – hence allowing to divide the task on several subspaces and then concatenate the samples. Such a property does not extend to the hyperbolic geometry, thus seemingly preventing us from focusing on 2-dimensional samples. Second, in Euclidean geometry, the polar radius r is distributed according to $\rho^W(r) = \frac{\mathbb{1}_{\mathbb{R}_+}(r)}{Z_r} e^{-\frac{r^2}{2\sigma^2}} r^{d-1}$, making it easy by a linear change of variable to take into account different scaling values. The non-linearity of $\sinh$ prevent us from using such a simple change of variable.

Computing gradients with respect to parameters So as to compute gradients of samples z with respect to the parameters μ and σ of samples of a hyperbolic distributions, we respectively rely on the reparametrisation given by Eq 20 for $\nabla_\mu z$, and on an implicit reparametrisation (Figurnov et al., 2018) of r for $\nabla_\sigma z$. We have $z = \exp_\mu^c\left(\frac{r}{\lambda_\mu^c}\alpha\right)$ with $\alpha \sim \mathcal{U}(\mathbb{S}^{d-1})$ and $r \sim \rho^R(\cdot)$. Hence,

$$\nabla_\mu z = \nabla_\mu \exp_\mu^c(u), \quad (23)$$

with $u = \frac{r}{\lambda_\mu^c}\alpha$ (actually) independent of μ , and

$$\nabla_\sigma z = \nabla_\sigma \exp_\mu^c(u) = \nabla_u \exp_\mu^c(u) \frac{\alpha}{\lambda_\mu^c} \nabla_\sigma r, \quad (24)$$

with $\nabla_\sigma(r)$ computed via the implicit reparametrisation given by

$$\begin{aligned} \nabla_\sigma(r) &= -(\nabla_r F^R(r, \sigma))^{-1} \nabla_\sigma F^R(r, \sigma) \\ &= -(\rho^R(r; \sigma))^{-1} \nabla_\sigma F^R(r, \sigma). \end{aligned} \quad (25)$$

Sampling hyperbolic radii Unfortunately the density of the hyperbolic radius $\rho^R(r)$ is not a well-known distribution and its cumulative density function does not seem analytically invertible. We therefore rely on rejection sampling methods.

Adaptive Rejection Sampling By making use of the log-concavity of ρ^R , we can rely on a piecewise exponential distribution proposal from adaptive rejection sampling (ARS) (R. Gilks and Wild, 1992). Such a proposal automatically adapt itself with respect to the parameters σ , c and d . Even though $\mathcal{N}_{\mathbb{B}_c^d}$ is defined on a d -dimensional manifold, ρ^R is a univariate distribution hence the sampling scheme is not directly affected by dimensionality. The difficulty in ARS is to choose the initial set of points to construct the piecewise exponential proposal. To do so, we first compute the mean $m = \mathbb{E}_{r \sim \rho^R}[r]$ and standard deviation $s = \mathbb{V}_{r \sim \rho^R}[r]^{1/2}$ of the targeted distribution. Then we choose a grid $\eta = (\eta_1, \dots, \eta_K) = (\text{linspace}(\eta_{max}, \eta_{min}, K/2), \text{linspace}(\eta_{min}, \eta_{max}, K/2))$. Eventually, we set the initial points $(x_1, \dots, x_K)$ to $x_k = m + \eta_k * \min(s, 0.95 * m/\eta_{max})$. For our experiments we chose $\eta_{min} = .1, \eta_{max} = 3, K = 20$. We do not adapt the proposal within the rejection sampling since we empirically found it unnecessary.

Alternatively, we derived below two non-adaptive proposal distributions along with their rejection rate constants. Yet, we observe that these rates do not scale well the dimensionality d and distortion σ , making them ill-suited for practical purposes.

Rejection Sampling with truncated Normal proposal The developed expression of $\rho(r)^R$ from Eq (22) highlights the fact that the density can immediately be upper bounded by a truncated normal

density:

$$\begin{aligned}\rho(r)^R &= \frac{\mathbb{1}_{R_+}(r)}{Z_r^R} \frac{1}{(2\sqrt{c})^{d-1}} \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} e^{\frac{(d-1-2k)^2}{2} c \sigma^2} e^{-\frac{1}{2\sigma^2} [r - (d-1-2k)\sqrt{c}\sigma^2]^2} \\ &\leq \frac{\mathbb{1}_{R_+}(r)}{Z_r^R} \frac{1}{(2\sqrt{c})^{d-1}} \sum_{2k=0}^{d-1} \binom{d-1}{2k} e^{\frac{(d-1-4k)^2}{2} c \sigma^2} e^{-\frac{1}{2\sigma^2} [r - (d-1-4k)\sqrt{c}\sigma^2]^2}.\end{aligned}$$

Then we choose our proposal g to be the truncated normal distribution associated with $k = 0$, i.e. with mean $(d-1)\sqrt{c}\sigma^2$ and variance σ^2

$$\begin{aligned}g(r) &= \frac{\mathbb{1}_{r>0}}{\sigma \left(1 - \Phi\left(-\frac{(d-1)\sqrt{c}\sigma^2}{\sigma}\right)\right)} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2} (r - (d-1)\sqrt{c}\sigma^2)^2} \\ &= \frac{1}{\sqrt{2\pi}} \frac{\mathbb{1}_{r>0}}{\sigma \left(1 - \frac{1}{2} - \frac{1}{2} \operatorname{erf}\left(-\frac{(d-1)\sqrt{c}\sigma}{\sqrt{2}}\right)\right)} e^{-\frac{1}{2\sigma^2} (r - (d-1)\sqrt{c}\sigma^2)^2} \\ &= \sqrt{\frac{2}{\pi}} \frac{\mathbb{1}_{r>0}}{\sigma \left(1 + \operatorname{erf}\left(\frac{(d-1)\sqrt{c}\sigma}{\sqrt{2}}\right)\right)} e^{-\frac{1}{2\sigma^2} (r - (d-1)\sqrt{c}\sigma^2)^2} \\ &= \frac{\mathbb{1}_{r>0}}{Z_g(\sigma)} e^{-\frac{1}{2\sigma^2} (r - (d-1)\sqrt{c}\sigma^2)^2}\end{aligned}\tag{26}$$

with

$$Z_g = \sqrt{\frac{\pi}{2}} \sigma \left(1 + \operatorname{erf}\left(\frac{(d-1)\sqrt{c}\sigma}{\sqrt{2}}\right)\right).\tag{27}$$

Computing the ratio of the densities yield

$$\begin{aligned}\frac{\rho(r)^R}{g(r)} &= \frac{Z_g(\sigma)}{Z_r^R} \frac{1}{(2\sqrt{c})^{d-1}} \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} e^{\frac{(d-1-2k)^2}{2} c \sigma^2} e^{-\frac{1}{2\sigma^2} [r - (d-1-2k)\sqrt{c}\sigma^2]^2} e^{\frac{1}{2\sigma^2} [r - (d-1)\sqrt{c}\sigma^2]^2} \\ &= \frac{Z_g(\sigma)}{Z_r^R} \frac{1}{(2\sqrt{c})^{d-1}} \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} e^{\frac{(d-1-2k)^2}{2} c \sigma^2} e^{2k\sqrt{c}((d-1-k)\sqrt{c}\sigma^2 - r)}.\end{aligned}$$

Hence

$$\rho(r)^R/g(r) \leq M \triangleq \frac{Z_g(\sigma)}{Z_r^R} \frac{1}{(2\sqrt{c})^{d-1}} e^{\frac{(d-1)^2 c \sigma^2}{2}}.\tag{28}$$

Rejection Sampling with Gamma proposal Now let's consider the following Gamma(2, σ) density:

$$g(r) = \frac{\mathbb{1}_{r>0}}{Z_g(\sigma)} r e^{-\frac{r}{\sigma}}$$

with

$$Z_g(\sigma) = \Gamma(2)\sigma^2.$$

Then log ratio of the densities can be upper bounded as following:

$$\begin{aligned}\ln\left(\frac{\rho(r)^R}{g(r)}\right) &= \ln\left(\frac{Z_g(\sigma)}{Z_r^R}\right) - \frac{r^2}{2\sigma^2} + (d-1) \ln(e^{\sqrt{c}r} - e^{-\sqrt{c}r}) - (d-1) \ln 2 - \ln r + \frac{r}{\sigma} \\ &= \ln\left(\frac{Z_g(\sigma)}{Z_r^R}\right) - (d-1) \ln 2 - \underbrace{\frac{r^2}{2\sigma^2} + \left((d-1)\sqrt{c} + \frac{1}{\sigma}\right)r}_{\leq \frac{((d-1)\sqrt{c}\sigma + 1)^2}{2}} + \underbrace{(d-1) \ln\left(\frac{1 - e^{-2\sqrt{c}r}}{r}\right)}_{\leq (d-1) \ln(2\sqrt{c})} \\ &\leq \ln\left(\frac{Z_g(\sigma)}{Z_r^R}\right) + \frac{((d-1)\sqrt{c}\sigma + 1)^2}{2} + (d-1) \ln \sqrt{c}.\end{aligned}$$

Hence

$$\rho(r)^R/g(r) \leq M \triangleq \frac{Z_g(\sigma)}{Z_r^R} c^{\frac{d-1}{2}} e^{\frac{((d-1)\sqrt{c}\sigma + 1)^2}{2}}.\tag{29}$$

B.4.3 Normalisation constant

In order to evaluate the density of the *Riemannian* normal distribution, we need to compute the normalisation constant, which we derive in this subsection.

Cumulative density function First let's derive the cumulative density function of the hyperbolic radius. Integrating the expended density of Eq (22) yields

$$\begin{aligned}
 F_r^R(r) &= \int_{-\infty}^r \rho^R(r) dr \\
 &= \frac{1}{Z_r^R} \frac{1}{(2\sqrt{c})^{d-1}} \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} e^{\frac{(d-1-2k)^2}{2} c \sigma^2} \\
 &\quad \times \int_0^r e^{-\frac{1}{2\sigma^2} [r - (d-1-2k)\sqrt{c}\sigma^2]^2} dr \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} e^{\frac{(d-1-2k)^2}{2} c \sigma^2} \\
 &\quad \times \left[\operatorname{erf} \left(\frac{r - (d-1-2k)\sqrt{c}\sigma^2}{\sqrt{2}\sigma} \right) \operatorname{erf} \left(\frac{(d-1-2k)\sqrt{c}\sigma}{\sqrt{2}} \right) \right] \quad (30)
 \end{aligned}$$

with $\Phi : x \mapsto \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{x}{\sqrt{2}} \right) \right)$, the cumulative distribution function of a standard normal distribution.

Taking the limit $F_r^R(r) \xrightarrow{r \rightarrow \infty} 1$ in Eq (30) yield

$$Z_r^R = \sqrt{\frac{\pi}{2}} \sigma \frac{1}{(2\sqrt{c})^{d-1}} \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} e^{\frac{(d-1-2k)^2}{2} c \sigma^2} \left[1 + \operatorname{erf} \left(\frac{(d-1-2k)\sqrt{c}\sigma}{\sqrt{2}} \right) \right]. \quad (31)$$

Note that by the antisymmetry of erf, one can simplify Eq (31) with a sum over $\lceil d/2 \rceil$ terms (as done in Hauberg (2018)). Also, computing such a sum is much more stable by relying on the *log sum exp* trick. Integrating Equation (19) of Appendix B.2 gives

$$Z^R = Z_r^R Z_\alpha \quad (32)$$

As a reminder, the surface area of the $d - 1$ -dimensional hypersphere with radius 1 is given by

$$Z_\alpha = A_{\mathbb{S}^{d-1}} = \frac{2\pi^{d/2}}{\Gamma(d/2)}.$$

For the special case of $c = 1$ and $d = 2$ we recover the formula given in Said et al. (2014)

$$Z_r^R = \sqrt{\frac{\pi}{2}} \sigma e^{\frac{\sigma^2}{2}} \operatorname{erf} \left(\frac{\sigma}{\sqrt{2}} \right).$$

B.4.4 Expectation of hyperbolic radii

Computing the expectation of the hyperbolic radius $r \sim \rho^R$ is of use to choose the initial set of points to construct the piecewise exponential proposal. By integrating the expended density of Eq (22), we

get

$$\begin{aligned}
\mathbb{E}[r] &= \int_{-\infty}^{\infty} r \rho^R(r) dr \\
&= \frac{1}{Z_r^R} \frac{1}{(2\sqrt{c})^{d-1}} \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} e^{\frac{(d-1-2k)^2}{2} c \sigma^2} \int_0^{\infty} r e^{-\frac{1}{2\sigma^2} [r - (d-1-2k)\sqrt{c}\sigma]^2} dr \\
&= \frac{1}{Z_r^R} \frac{1}{(2\sqrt{c})^{d-1}} \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} e^{\frac{(d-1-2k)^2}{2} c \sigma^2} \\
&\quad \times \left[\sqrt{\frac{\pi}{2}} (d-1-2k) \sqrt{c} \sigma^2 \left(1 + \operatorname{erf} \left(\frac{(d-1-2k)\sqrt{c}\sigma}{\sqrt{2}} \right) \right) + \sigma^2 e^{-\frac{(d-1-2k)^2}{2} c \sigma^2} \right] \\
&= \frac{1}{Z_r^R} \sqrt{\frac{\pi}{2}} \sigma \frac{1}{(2\sqrt{c})^{d-1}} \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} \\
&\quad \times \left[e^{\frac{(d-1-2k)^2}{2} c \sigma^2} (d-1-2k) \sqrt{c} \sigma^2 \left(1 + \operatorname{erf} \left(\frac{(d-1-2k)\sqrt{c}\sigma}{\sqrt{2}} \right) \right) + \sigma \sqrt{\frac{2}{\pi}} \right]
\end{aligned} \tag{33}$$

C Experimental details

In this section we give more details on the datasets, architecture designs and optimisation schemes used for the experimental results given in Section 5.

C.1 Synthetic Branching Diffusion Process

Generation Nodes $(\mathbf{y}_1, \dots, \mathbf{y}_N) \in \mathbb{R}^n$ of the branching diffusion process are sampled as follow

$$\mathbf{y}_i \sim \mathcal{N}(\cdot | \mathbf{y}_{\pi(i)}, \sigma_0^2) \quad \forall i \in 1, \dots, N$$

with $\pi(i)$ being the index of the i th node’s ancestor and $d(i)$ its depth. Then, noisy observations are sampled for each node $\mathbf{x}_i$,

$$\mathbf{x}_{i,j} = \mathbf{y}_i + \boldsymbol{\epsilon}_{i,j}, \quad \boldsymbol{\epsilon}_{i,j} \sim \mathcal{N}(\cdot | \mathbf{0}, \sigma_j^2) \quad \forall i, j.$$

The root x_0 is set to $\mathbf{0}$ for simplicity. The observation dimension is set to $n = 50$. The dataset $(\mathbf{x}_{i,j})_{i,j}$ is centered and normalised to have unit variance. Thus, the choice of variance σ_0^2 does not matter and it is set to $\sigma_0 = 1$. The number of noisy observations is set to $J = 5$, and its variance to $\sigma_j^2 = \sigma_0^2/5 = 1/5$. The depth is set to 6 and the branching factor to 2.

Architectures Both $\mathcal{N}$ -VAE and $\mathcal{P}^c$ -VAE decoders parametrise the mean of the unit variance Gaussian likelihood $\mathcal{N}(\cdot | f_{\theta}(\mathbf{z}), 1)$. Their encoders parametrise the mean and the log-variance of respectively an isotropic normal distribution $\mathcal{N}(\cdot | g_{\phi}(\mathbf{z}))$ and an isotropic hyperbolic normal distribution $\mathcal{N}_{\mathbb{B}^d}(\cdot | g_{\phi}(\mathbf{z}))$. The $\mathcal{N}$ -VAE’s encoder and decoder are composed of 2 Fully-Connected layers with a ReLU activation in between, as summed up in Tables 5 and 6. The $\mathcal{P}^c$ -VAE’s design is similar, the differences being that the decoder’s output is mapped to manifold via the exponential map $\exp_0^c$, and the decoder’s first layer is made of *gyroplane units* presented in Section 3.2, as summarised in Tables 7 and 8. Observations live in $\mathcal{X} = \mathbb{R}^{50}$ and the latent space dimensionality d is set to $d = 2$.

Table 5: Encoder network for $\mathcal{N}$ -VAE

Layer	Output dim	Activation
Input	50	Identity
FC	200	ReLU
FC	2, 1	Identity

Table 6: Decoder network for $\mathcal{N}$ -VAE

Layer	Output dim	Activation
Input	2	Identity
FC	200	ReLU
FC	50	Identity

Table 7: Encoder network for $\mathcal{P}^c$ -VAE

Layer	Output dim	Activation
Input	50	Identity
FC	200	ReLU
FC	2, 1	$\exp_0^c$, Identity

Table 8: Decoder network for $\mathcal{P}^c$ -VAE

Layer	Output dim	Activation
Input	2	Identity
Gyropplane	200	ReLU
FC	50	Identity

The synthetic datasets are generated as described in Section 5, then centred and normalised to unit variance. There are then randomly split into training and testing datasets with a proportion 0.7.

Optimisation Gyroplane offset $\mathbf{p} \in \mathbb{B}_c^d$ are only *implicitly* parametrised to live in the manifold, by projecting a real vector $\mathbf{p} = \exp_0^c(\mathbf{p}')$. Hence, all parameters $\{\theta, \phi\}$ of the model explicitly live in Euclidean spaces which means that usual optimisation schemes can be applied. We therefore rely on Adam optimiser (Kingma and Ba, 2016) with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a constant learning rate set to $1e - 3$. Models are trained with mini-batches of size 64 for 1000 epochs. The ELBO is approximated with a MC estimate with $K = 1$.

Baselines The principal component analysis (PCA) embeddings are obtained via a singular-value decomposition (SVD) by projecting the dataset on the basis associated with the two highest singular values. The Gaussian process latent variable model (GPLVM) embeddings are obtained by maximising the marginal likelihood of a (non-Bayesian) GPLVM with RBF kernel, and whose latent variables are initialised with PCA.

C.2 MNIST digits

The MNIST dataset (LeCun and Cortes, 2010) contains 60,000 training and 10,000 test images of ten handwritten digits (zero to nine), with 28x28 pixels.

Architectures The architectures used for the encoder and the decoder for Mnist are similar to the ones used for the Synthetic Branching Diffusion Process. They differ by the dimensions of the observation space ($\mathcal{X} = \mathbb{R}^{28 \times 28}$) and hidden space. The output of the first fully connected layer is here equal to 600. The latent space dimensionality d is set to 2, 5, 10 and 20 respectively. The bias of the decoder’s last layer is set to the average value of digits (for each pixel). The architectures used for the classifier are similar than the decoder architectures, the only difference being the output dimensionality (10 labels). We initialise the classifier’s first layer with decoder’s first layer weights. Then the classifier is trained to minimise the cross entropy for 5 epochs, with mini-batches of size 64 and a constant learning rate of $1e-3$.

Optimisation We use $[0, 1]$ normalised data as targets for the mean of a Bernoulli distribution, using negative cross-entropy for $\log p(\mathbf{x}|\mathbf{z})$. We set the prior distribution’s distortion to $\sigma = 1$. We rely on Adam optimiser with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a constant learning rate of $5e^{-4}$. Models are trained with mini-batches of size 128 for 80 epochs.

C.3 Graph embeddings

The PhD advisor-advisee relationships graph (Nooy et al., 2011) contains 344 nodes and 343 edges. The phylogenetic tree expressing genetic heritage (Hofbauer et al., 2016; Sanderson and Eriksson, 1994) contains 1025 nodes and 1043 edges. The biological set representing disease relationships (Goh et al., 2007; Rossi and Ahmed, 2015) contains 516 nodes and 1188 edges. We follow the training and evaluation procedure introduced in Kipf and Welling (2016).

Architectures We also follow the featureless architecture introduced in Kipf and Welling (2016), namely a two-layer GCN with 32 hidden dimensions to parametrise the variational posteriors, and a likelihood which factorises along edges $p(\mathbf{A}|\mathbf{Z}) = \prod_{i=1}^N \prod_{j=1}^N p(A_{ij}|\mathbf{z}_i, \mathbf{z}_j)$, with $\mathbf{A}$ being the adjacency matrix. The probability of an edge is defined through the latent metric by $p(A_{ij} = 1|\mathbf{z}_i, \mathbf{z}_j) = 1 - \tanh(d_{\mathcal{M}}(\mathbf{z}_i, \mathbf{z}_j))$. For the Poincaré ball latent space, the encoder output is projected

on the manifold: $\mu = \exp_0(\text{GCN}_\mu(\mathbf{A}))$. The latent dimension is set to 5 for the experiments. We use a *Wrapped* Gaussian prior and variational posterior.

Optimisation We use the adjacency matrix $\mathbf{A}$ as target for the mean of a Bernoulli distribution, using negative cross-entropy for $\log p(\mathbf{A}|\mathbf{Z})$. We rely on Adam optimiser with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a constant learning rate of $1e^{-2}$. We perform full-batch gradient descent for 800 epochs and make use of the reparametrisation trick for training.

D More experimental qualitative results

Figure 9 shows latent representations of $\mathcal{P}^c$ -VAEs with different curvatures. With "small" curvatures, we observe that embeddings lie close the center of the ball, where the geometry is close to be Euclidean. Similarly as Figure 9, Figure 10 illustrates the learned latent representations of $\mathcal{P}^c$ -VAE with decreasing curvatures c , by highlighting the leaned *gyroplanes* of the decoder.

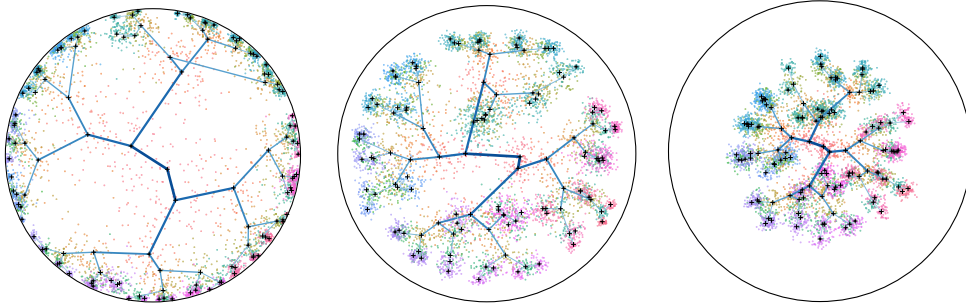


Figure 9: Branching diffusion process latent representations of $\mathcal{P}^c$ -VAE with decreasing curvatures $c = 1.2, 0.3, 0.1$ (Left to Right).

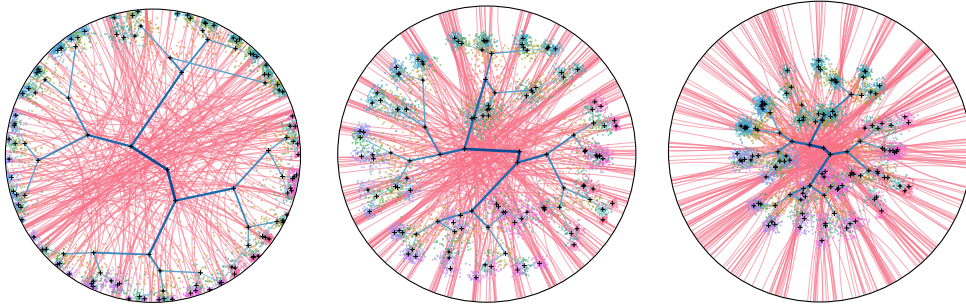


Figure 10: Branching diffusion process latent representations of $\mathcal{P}^c$ -VAE with decreasing curvatures $c = 1.2, 0.3, 0.1$ (Left to Right).

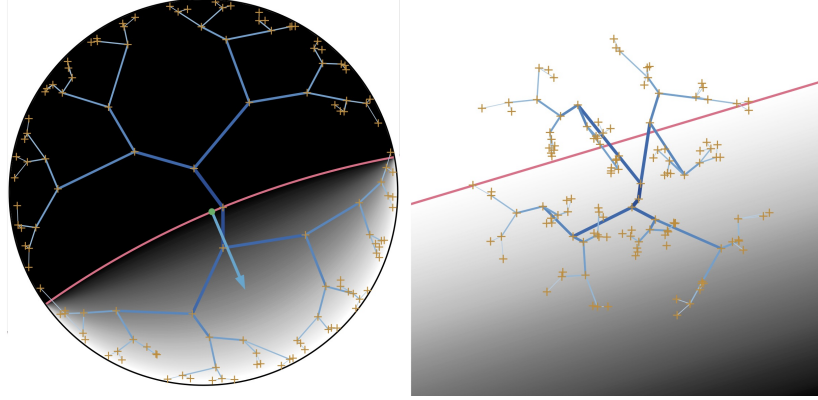


Figure 11: Branching diffusion process latent representation of $\mathcal{P}^1$ -VAE (Left) and $\mathcal{N}$ -VAE (Right) with heatmap of the log distance to the hyperplane (in pink).

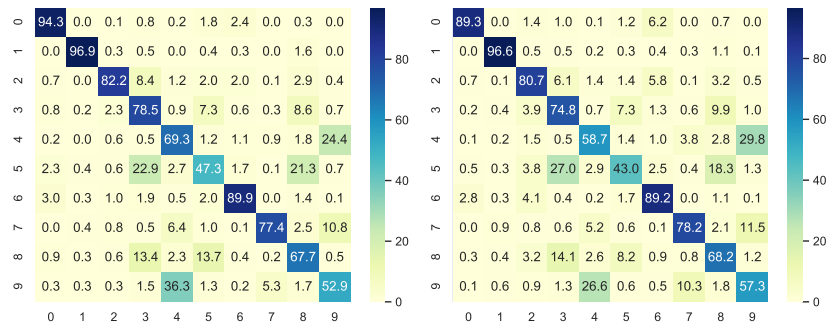


Figure 12: MNIST average confusion matrices of the classifiers trained on embeddings from the $\mathcal{P}^{1.4}$ -VAE (Left) and $\mathcal{N}$ -VAE (Right) models.

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**).

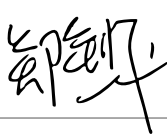
Title of Paper	Continuous Hierarchical Representations with Poincaré Variational Auto-Encoders
Publication Status	☒ Published
Publication Details	Emile Mathieu, Charline Le Lan, Chris Maddison, Ryota Tomioka, Yee Whye Teh (2019). Advances in Neural Information Processing Systems 32, 12565--12576, Curran Associates, Inc.

Student Confirmation

Student Name:	Emile Mathieu		
Contribution to the Paper	Developed the core methodology, with the help of fruitful discussions with Chris Maddison. Developed most of the model implementation, and conducted the graph embeddings experiments. Wrote most of the manuscript, assisted by Y.W. Teh and Charline Le Lan for writing the final version.		
Signature		Date	18/06/2021

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: Professor Yee Whye Teh		
Supervisor comments: I do certify that the candidate made a substantial contribution to the publication, and the description above is accurate.		
Signature		Date
		30 June 2021

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

4

Riemannian Continuous Normalizing Flows

Riemannian Continuous Normalizing Flows

Emile Mathieu^{†*}, Maximilian Nickel[‡]

emile.mathieu@stats.ox.ac.uk, maxn@fb.com

[†] Department of Statistics, University of Oxford, UK

[‡] Facebook Artificial Intelligence Research, New York, USA

Abstract

Normalizing flows have shown great promise for modelling flexible probability distributions in a computationally tractable way. However, whilst data is often naturally described on Riemannian manifolds such as spheres, tori, and hyperbolic spaces, most normalizing flows implicitly assume a flat geometry, making them either misspecified or ill-suited in these situations. To overcome this problem, we introduce *Riemannian continuous normalizing flows*, a model which admits the parametrization of flexible probability measures on smooth manifolds by defining flows as the solution to ordinary differential equations. We show that this approach can lead to substantial improvements on both synthetic and real-world data when compared to standard flows or previously introduced projected flows.

1 Introduction

Learning well-specified probabilistic models is at the heart of many problems in machine learning and statistics. Much focus has therefore been placed on developing methods for modelling and inferring expressive probability distributions. Normalizing flows (Rezende and Mohamed, 2015) have shown great promise for this task as they provide a general and extensible framework for modelling highly complex and multimodal distributions (Papamakarios et al., 2019).

An orthogonal but equally important aspect of well-specified models is to correctly characterize the geometry which describes the proximity of data points. Riemannian manifolds provide a general framework for this purpose and are a natural approach to model tasks in many scientific fields ranging from earth and climate science to biology and computer vision. For instance, storm trajectories may be modelled as paths on the sphere (Karpatne et al., 2019), the shape of proteins can be parametrized using tori (Hamelryck et al., 2006), cell developmental processes can be described through paths in hyperbolic space (Klimovskaia et al., 2020), and human actions can be recognized in video using matrix manifolds (Lui, 2012). If appropriately chosen, manifold-informed methods can lead to improved sample complexity and generalization, improved fit in the low parameter regime, and guide inference methods to interpretable models. They can also be understood as a geometric prior that encodes a practitioner’s assumption about the data and imposes an inductive bias.

However, conventional normalizing flows are not readily applicable to such manifold-valued data since their implicit Euclidean assumption makes them unaware of the underlying geometry or borders of the manifold. As a result they would yield distributions having some or all of their mass lying

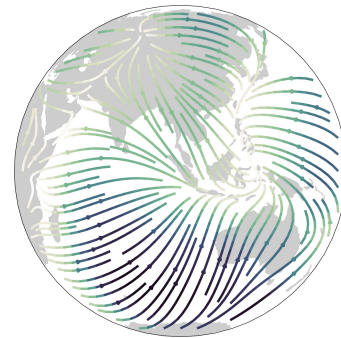


Figure 1: Trajectories generated on the sphere to model volcano eruptions. Note that these converge to the known *Ring of Fire*.

*Work done while at Facebook AI research.

outside the manifold, rendering them ill-suited or even misspecified so that central concepts like the reverse Kullback-Leibler (KL) divergence would not even be defined.

In this work, we propose a principled way to combine both of these aspects and parametrize flexible probability distributions on Riemannian manifolds. Specifically, we introduce *Riemannian continuous normalizing flows* in which flows are defined via vector fields on manifolds and computed as the solution to the associated ordinary differential equation (ODE) (see Figure 1 for an illustration). Intuitively, our method operates by first parametrizing a vector field on the manifold with a neural network, then sampling particles from a base distribution, and finally approximating their flow along the vector field using a numerical solver. Both the neural network and the solver are aware of the underlying geometry which ensures that the flow is always located on the manifold – yielding a *Riemannian* method.

This approach allows us to combine multiple important advantages: One major challenge of normalizing flows lies in designing transformations that enable efficient sampling and density computation. By basing our approach on *continuous normalizing flows* (CNFs) (Chen et al., 2018; Grathwohl et al., 2019; Salman et al., 2018) we avoid strong structural constraints to be imposed on the flow, as is the case for most *discrete* normalizing flows. Such unconstrained *free-form* flows have empirically been shown to be highly expressive (Chen et al., 2019; Grathwohl et al., 2019). Moreover, *projected* methods require a differentiable mapping from a Euclidean space to the manifold, yet such a function cannot be bijective, which in turn leads to numerical challenges. By taking a Riemannian approach, our method is more versatile since it does not rely on an ad-hoc projection map and simultaneously reduces numerical artefacts that interfere with training. To the best of our knowledge, our method is the first to combine these properties as existing methods for normalizing flows on manifolds are either discrete (Bose et al., 2020; Rezende et al., 2020), projected (Gemici et al., 2016; Falorsi et al., 2019; Bose et al., 2020) or manifold-specific (Sei, 2011; Bose et al., 2020; Rezende et al., 2020).

We empirically demonstrate the advantages of our method on constant curvature manifolds – i.e., the Poincaré disk and the sphere – and show the benefits of the proposed approach compared to non-Riemannian and projected methods for maximum likelihood estimation and reverse KL minimization. We also apply our method to density estimation on earth-sciences data (e.g., locations of earthquakes, floods and wildfires) and show that it yields better generalization performance and faster convergence.

2 Continuous Normalizing Flows on Riemannian Manifolds

Normalizing flows operate by pushing a simple base distribution through a series of parametrized invertible maps, referred as the *flow*. This can yield a highly complex and multimodal distribution which is typically assumed to live in a Euclidean vector space. Here, we propose a principled approach to extend normalizing flows to manifold-valued data, i.e. *Riemannian continuous normalizing flows* (RCNFs). Following CNFs (Chen et al., 2018; Grathwohl et al., 2019; Salman et al., 2018) we define manifold flows as the solutions to ODEs. The high-level idea is to parametrize flows through the time-evolution of manifold-valued particles z – in particular via their velocity $\dot{z}(t) = f_\theta(z(t), t)$ where f_θ denotes a *vector field*. Particles are first sampled from a simple base distribution, and then their evolution is integrated by a manifold-aware numerical solver, yielding a new complex multimodal distribution of the particles. This *Riemannian and continuous* approach has the advantages of allowing almost *free-form* neural networks and of not requiring any mapping from a Euclidean space which would potentially lead to numerical challenges.

For practical purposes, we focus our theoretical and experimental discussion on *constant curvature manifolds* (see Table 1). In addition to being widely used in the literature (Nickel and Kiela, 2017; Davidson et al., 2018; Mardia and Jupp, 2000; Hasnat et al., 2017), these manifolds are convenient to work with since most related geometrical quantities are available in closed-form. However, our

Table 1: Summary of d-dimensional continuous constant (sectional) curvature manifolds.

Geometry	Model	Curvature	Coordinates	$\sqrt{\det g} = d \text{ Vol} / d \text{ Leb}_{\mathbb{R}^d}$	Compact	
Euclidean	$\mathbb{R}^d$	Real vector space	$K = 0$	Cartesian z	1	No
Hyperbolic	$\mathbb{B}_K^d$	Poincaré ball	$K < 0$	Cartesian z	$\left(2 / (1 + K \ z\ ^2)\right)^d$	No
Elliptic	$\mathbb{S}_K^d$	Hypersphere	$K > 0$	n-spherical φ	$K^{-\frac{d-1}{2}} \prod_{i=1}^{d-2} \sin(\varphi_i)^{d-i-1}$	Yes

proposed approach is generic and could be used on a broad class of manifolds such as product and matrix manifolds like tori and Grassmanians. For a brief overview of relevant concepts in Riemannian geometry please see Appendix A.1 or Lee (2003) for a more thorough introduction.

In the following, we develop the key components which allow us to define continuous normalizing flows that are aware of the underlying Riemannian geometry: flow, likelihood, and vector field.

Vector flows Flows in conventional normalizing flows are defined as smooth mappings $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^d$ which transform a base distribution $z \sim P_0$ into a complex distribution P_θ . For normalizing flows to be well-behaved and convenient to work with, the flow is required to be *bijective* and *differentiable* which introduces significant structural constraints on ϕ . Continuous normalizing flows overcome this issue by defining the flow $\phi : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$ generated by an ordinary differential equation, allowing for unrestricted neural network architectures. Here we show how *vector fields* can be used to define similar flows $\phi : \mathcal{M} \times \mathbb{R} \rightarrow \mathcal{M}$ on general *Riemannian manifolds*.

Consider the temporal evolution of a particle $z(t)$ lying on a d -dimensional manifold $\mathcal{M}$, whose velocity is given by a *vector field* $f_\theta(z(t), t)$. Intuitively, $f_\theta(z(t), t)$ indicates the direction and speed along which the particle is moving on the *manifold's surface*. Classic examples for such vector fields include weathercocks giving wind direction and compasses pointing toward the magnetic north pole of the earth. Formally, let $\mathcal{T}_z \mathcal{M}$ denote the *tangent space* at z and $\mathcal{T}\mathcal{M} = \bigcap_{z \in \mathcal{M}} \mathcal{T}_z \mathcal{M}$ the associated *tangent bundle*. Furthermore, let $f_\theta : \mathcal{M} \times \mathbb{R} \mapsto \mathcal{T}\mathcal{M}$ denote a vector field on $\mathcal{M}$. The particle's time-evolution according to f_θ is then given by the following ODE

$$\frac{dz(t)}{dt} = f_\theta(z(t), t). \quad (1)$$

To transform a base distribution using this vector field, we are then interested in a particle's position after time t . When starting at an initial position $z(0) = z_0$, the *flow operator* $\phi : \mathcal{M} \times \mathbb{R} \mapsto \mathcal{M}$ gives the particle's position at any time t as $z(t) = \phi(z_0, t)$. Leveraging the fundamental theorem of flows (Lee, 2003), we can show that under mild conditions, this flow is *bijective* and *differentiable*. We write C^1 for the set of differentiable functions whose derivative are continuous.

Proposition 1 (Vector flows). *Let $\mathcal{M}$ be a smooth complete manifold. Furthermore, let f_θ be a C^1 -bounded time-dependent vector field. Then there exists a global flow $\phi : \mathcal{M} \times \mathbb{R} \mapsto \mathcal{M}$ such that for each $t \in \mathbb{R}$, the map $\phi(\cdot, t) : \mathcal{M} \mapsto \mathcal{M}$ is a C^1 -diffeomorphism (i.e. C^1 bijection with C^1 inverse).*

Proof. See Appendix E.1 for a detailed derivation. $\square$

Note that scaling the vector field as $f_\theta^\alpha \triangleq \alpha f_\theta$ results in a time-scaled flow $\phi^\alpha(z, t) = \phi(z, \alpha t)$. The integration duration t is therefore arbitrary. Without loss of generality we set $t = 1$ and write $\phi \triangleq \phi(\cdot, 1)$. Concerning the evaluation of the flow ϕ , it generally does not accept a closed-form solution and thus requires to be approximated numerically. To this extent we rely on an explicit and adaptive Runge-Kutta (RK) integrator of order 4 (Dormand and Prince, 1980). However, standard integrators used in CNFs generally do not preserve manifold constraints (Hairer, 2006). To overcome this issue we rely on a *projective solver* (Hairer, 2011). This solver works by conveniently solving the ODE in the ambient Cartesian coordinates and projecting each step onto the manifold. Projections onto $\mathbb{S}^d$ are computationally cheap since they amount to l^2 norm divisions. No projection is required for the Poincaré ball.

Likelihood Having a flow at hand, we are now interested in evaluating the likelihood of our *pushforward* model $P_\theta = \phi_\# P_0$. Here, the *pushforward* operator $\#$ indicates that one obtains samples $z \sim \phi_\# P_0$ as $z = \phi(z_0)$ with $z_0 \sim P_0$. For this purpose, we derive in the following the change in density in terms of the geometry of the manifold and show how to efficiently estimate the likelihood.

Change in density In normalizing flows, we can compute the likelihood of a sample via the change in density from the base distribution to the pushforward. Applying the chain rule we get

$$\log p_\theta(z) - \log p_0(z_0) = \log \left| \det \frac{\partial \phi^{-1}(z)}{\partial z} \right| = -\log \left| \det \frac{\partial \phi(z_0)}{\partial z} \right|. \quad (2)$$

In general, computing the Jacobian's determinant of the flow is challenging since it requires d reverse-mode automatic differentiations to obtain the full Jacobian matrix, and $O(d^3)$ operations to

compute its determinant. CNFs side step direct computation of the determinant by leveraging the time-continuity of the flow and re-expressing Equation 2 as the integral of the *instantaneous change* in log density $\int_0^t \frac{d \log p_\theta(z(t))}{dt} dt$. However, standard CNFs make an implicit Euclidean assumption to compute this quantity which is violated for general Riemannian manifolds. To overcome this issue we express the instantaneous change in log-density in terms of the *Riemannian metric*. In particular, let $G(z)$ denote the *matrix representation of the Riemannian metric* for a given manifold $\mathcal{M}$, then $G(z)$ endows tangent spaces $\mathcal{T}_z \mathcal{M}$ with an inner product. For instance in the Poincaré ball $\mathbb{B}^d$, it holds that $G(z) = (2 / (1 + K \|z\|^2)) I_d$, while in Euclidean space $\mathbb{R}^d$ we have $G(z) = I_d$, where I_d denotes the identity matrix. Using the Liouville equation, we can then show that the instantaneous change in variable is defined as follows.

Proposition 2 (Instantaneous change of variables). *Let $z(t)$ be a continuous manifold-valued random variable given in local coordinates, which is described by the ODE from Equation 1 with probability density $p_\theta(z(t))$. The change in log-probability then also follows a differential equation given by*

$$\frac{d \log p_\theta(z(t))}{dt} = -\operatorname{div}(f_\theta(z(t), t)) = -|G(z(t))|^{-\frac{1}{2}} \operatorname{tr} \left(\frac{\partial \sqrt{|G(z(t))|} f_\theta(z(t), t)}{\partial z} \right) \quad (3)$$

$$= -\operatorname{tr} \left(\frac{\partial f_\theta(z(t), t)}{\partial z} \right) - |G(z(t))|^{-\frac{1}{2}} \left\langle f_\theta(z(t), t), \frac{\partial}{\partial z} \sqrt{|G(z(t))|} \right\rangle. \quad (4)$$

Proof. For a detailed derivation of Equation 3 see Appendix C. $\square$

Note that in the Euclidean setting $\sqrt{|G(z)|} = 1$ thus the second term of Equation 4 vanishes and we recover the formula from Grathwohl et al. (2019); Chen et al. (2018).

Estimating the divergence Even though the determinant of Equation 2 has been replaced in Equation 3 by a trace operator with lower computational complexity, we still need to compute the full Jacobian matrix of f_θ . Similarly to Grathwohl et al. (2019); Salman et al. (2018), we make use of Hutchinson’s trace estimator to compute the Jacobian efficiently. In particular, Hutchinson (1990) showed that $\operatorname{tr}(A) = \mathbb{E}_{p(\epsilon)}[\epsilon^\top A \epsilon]$ with $p(\epsilon)$ being a d -dimensional random vector such that $\mathbb{E}[\epsilon] = 0$ and $\operatorname{Cov}(\epsilon) = I_d$. Leveraging this trace estimator to approximate the divergence in Equation 3 yields

$$\operatorname{div}(f_\theta(z(t), t)) = |G(z(t))|^{-\frac{1}{2}} \mathbb{E}_{p(\epsilon)} \left[\epsilon^\top \frac{\partial \sqrt{|G(z(t))|} f_\theta(z(t), t)}{\partial z} \epsilon \right]. \quad (5)$$

We note that the variance of this estimator can potentially be high since it scales with the inverse of the determinant term $\sqrt{|G(z(t))|}$ (see Appendix D.2). By integrating Equation 3 over time with the stochastic divergence estimator from Equation 5, we get the following total change in log-density between the manifold-valued random variables z and z_0

$$\log \left(\frac{p_\theta(z)}{p_0(z_0)} \right) = -\int_0^1 \operatorname{div}(f_\theta(z(t), t)) dt = -\mathbb{E}_{p(\epsilon)} \left[\int_0^1 |G(z(t))|^{-\frac{1}{2}} \epsilon^\top \frac{\partial \sqrt{|G(z(t))|} f_\theta(z(t), t)}{\partial z} \epsilon dt \right]. \quad (6)$$

It can be seen that Equation 6 accounts again for the underlying geometry through the metric $G(z(t))$. Table 1 lists closed-form solutions of its determinant for constant curvature manifolds. Furthermore, the vector-Jacobian product can be computed through backward auto-differentiation with linear complexity, avoiding the quadratic cost of computing the full Jacobian matrix. Additionally, the integral is approximated via the discretization of the flow returned by the solver.

Choice of base distribution P_0 The closer the initial base distribution is to the target distribution, the easier the learning task should be. However, it is challenging in practice to incorporate such prior knowledge. We consequently use a uniform distribution on $\mathbb{S}^d$ since it is the most "uncertain" distribution. For the Poincaré ball $\mathbb{B}^d$, we rely on a standard wrapped Gaussian distribution $\mathcal{N}^W$ (Nagano et al., 2019; Mathieu et al., 2019) because it is convenient to work with.

Vector field Finally, we discuss the form of the *vector field* $f_\theta : \mathcal{M} \times \mathbb{R} \rightarrow \mathcal{T}\mathcal{M}$ which generates the flow ϕ used to pushforward samples. We parametrize f_θ via a feed-forward neural network which takes as input manifold-valued particles, and outputs their velocities. The architecture of the vector field has a direct impact on the expressiveness of the distribution and is thus crucially important. In order to take into account these geometrical properties we make use of specific input and output layers that we describe below. The rest of the architecture is based on a multilayer perceptron.

Input layer To inform the neural network about the geometry of the manifold $\mathcal{M}$, we use as first layer a *geodesic distance layer* (Ganea et al., 2018; Mathieu et al., 2019) which generalizes *linear layers* to manifolds, and can be seen as computing distances to *decision boundaries* on $\mathcal{M}$. These boundaries are parametrized by *geodesic hyperplanes* H_w , and the associated *neurons* $h_w(z) \propto d_{\mathcal{M}}(z, H_w)$, with $d_{\mathcal{M}}$ being the geodesic distance. Horizontally stacking several of these *neurons* makes a *geodesic distance layer*. We refer to Appendix E.2 for more details.

Output layer To constrain the neural net to $\mathcal{T}\mathcal{M}$, we output vectors in $\mathbb{R}^{d+1}$ when $\mathcal{M} = \mathbb{S}^d$, before projecting them to the tangent space i.e. $\mathbf{f}_{\theta}(z) = \text{proj}_{\mathcal{T}_z\mathcal{M}} \text{neural_net}(z)$. This is not necessary in $\mathbb{B}^d$ since the ambient space is of equal dimension. Yet, velocities scale as $\|\mathbf{f}_{\theta}(z)\|_z = |G(z)|^{1/2} \|\mathbf{f}_{\theta}(z)\|_2$, hence we scale the `neural_net` by $|G(z)|^{-1/2}$ s.t. $\|\mathbf{f}_{\theta}(z)\|_z = \|\text{neural_net}(z)\|_2$.

Regularity For the flow to be bijective, the vector field $\mathbf{f}_{\theta}$ is required to be C^1 and bounded (cf Proposition 1). The boundness and smoothness conditions can be satisfied by relying on bounded smooth non-linearities in $\mathbf{f}_{\theta}$ such as $\tanh$, along with bounded weight and bias at the last layer.

Training In density estimation and inference tasks, one aims to learn a model P_{θ} with parameters θ by minimising a *divergence* $\mathcal{L}(\theta) = D(P_{\mathcal{D}} \parallel P_{\theta})$ w.r.t. a target distribution $P_{\mathcal{D}}$. In our case, the parameters θ refer to the parameters of the vector field $\mathbf{f}_{\theta}$. We minimize the loss $\mathcal{L}(\theta)$ using first-order stochastic optimization, which requires Monte Carlo estimates of loss gradients $\nabla_{\theta} \mathcal{L}(\theta)$. We back-propagate gradients through the explicit solver with $O(1/h)$ memory cost, h being the step size. When the loss $\mathcal{L}(\theta)$ is expressed as an expectation over the model P_{θ} , as in the *reverse* KL divergence, we rely on the *reparametrization trick* (Kingma and Welling, 2014; Rezende et al., 2014). In our experiments we will consider both the negative log-likelihood and reverse KL objectives

$$\mathcal{L}^{\text{Like}}(\theta) = -\mathbb{E}_{z \sim P_{\mathcal{D}}} [\log p_{\theta}(z)] \text{ and } \mathcal{L}^{\text{KL}}(\theta) = D_{\text{KL}}(P_{\theta} \parallel P_{\mathcal{D}}) = \mathbb{E}_{z \sim P_{\theta}} [\log p_{\theta}(z) - \log p_{\mathcal{D}}(z)]. \quad (7)$$

Additionally, regularization terms can be added in the hope of improving training and generalization. See Appendix D for a discussion and connections to the dynamical formulation of optimal transport.

3 Related work

Here we discuss previous work that introduced normalizing flows on manifolds. For clarity we split these into *projected* vs *Riemannian* methods which we describe below.

Projected methods These methods consist in parametrizing a normalizing flow on $\mathbb{R}^d$ and then pushing-forward the resulting distribution along an invertible map $\psi : \mathbb{R}^d \rightarrow \mathcal{M}$. Yet, the existence of such an *invertible* map is equivalent to $\mathcal{M}$ being homeomorphic to $\mathbb{R}^d$ (e.g. being "flat"), hence limiting the scope of that approach. Moreover there is no principled way to choose such a map, and different choices lead to different numerical or computational challenges which we discuss below.

Exponential map The first generic *projected* map that comes to mind in this setting is the exponential map $\exp_{\mu} : T_{\mu}\mathcal{M} \cong \mathbb{R}^d \rightarrow \mathcal{M}$, which parameterizes geodesics starting from μ with velocity $v \in T_{\mu}\mathcal{M}$. This leads to so called *wrapped* distributions $P_{\theta}^{\text{W}} = \exp_{\mu_{\theta}} P$, with P a probability measure on $\mathbb{R}^d$. This approach has been taken by Falorsi et al. (2019) to parametrize probability distributions on Lie groups. Yet, in compact manifolds – such as spheres or the $\text{SO}(3)$ group – computing the density of *wrapped* distributions requires an infinite summation, which in practice needs to be truncated. This is not the case however on hyperbolic spaces (like the Poincaré ball) since the exponential map is bijective on these manifolds. This approach has been proposed in Bose et al. (2020) where they extend Real-NVP (Dinh et al., 2017) to the hyperboloid model of hyperbolic geometry. In addition to this *wrapped* Real-NVP, they also introduced a hybrid coupling model which is empirically shown to be more expressive. We note however that the exponential map is believed to be "badly behaved" away from the origin (Dooley and Wildberger, 1993; Al-Mohy and Higham, 2010).

Stereographic map Alternatively to the exponential map, Gemici et al. (2016) proposed to parametrize probability distributions on $\mathbb{S}^d$ via the *stereographic projection* defined as $\rho(z) = z_{2:d} / (1 + z_1)$ with *projection point* $-\{\mu_0\} = (-1, 0, \dots, 0)$. Gemici et al. then push a probability measure P defined on $\mathbb{R}^d$ along the inverse of the *stereographic* map ρ , yielding $P_{\theta}^{\text{S}} = \rho_{\#}^{-1} P$.

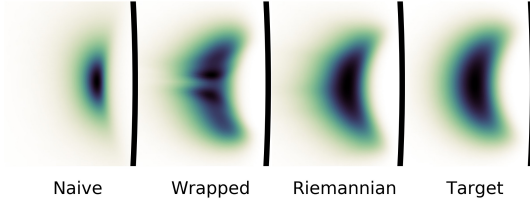


Figure 2: Probability densities on $\mathbb{B}^2$. Models have been trained by maximum likelihood to fit $\mathcal{N}^W(\exp_0(2 \partial x), \Sigma)$. The black semi-circle indicates the disk’s border. The best run out of twelve trainings is shown for each model.

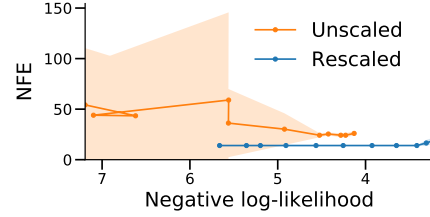


Figure 3: Ablation study of the vector field architecture for the *Riemannian* model. Models have been trained to fit a $\mathcal{N}^W(\exp_0(\partial x), \Sigma)$.

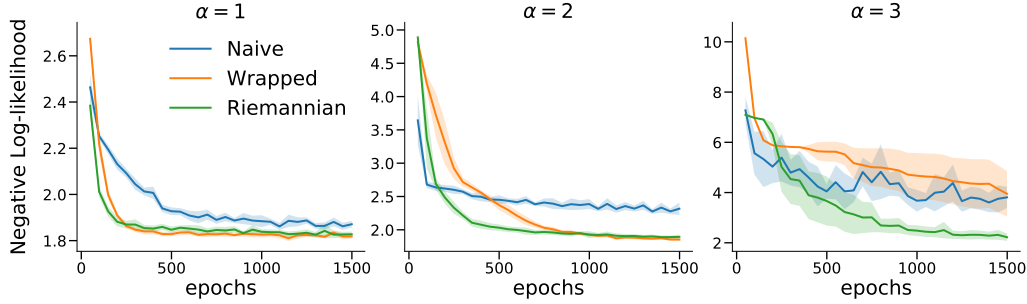


Figure 4: Negative Log-likelihood of CNFs trained to fit a $\mathcal{N}^W(\exp_0(\alpha \partial x), \Sigma)$ target on $\mathbb{B}^2$.

However, the stereographic map ρ is not injective, and projects $-\mu_0$ to ∞ . This implies that spherical points close to the projection point $-\{\mu_0\}$ are mapped far away from the origin of the plane. Modelling probability distributions with mass close to $\{-\mu_0\}$ may consequently be numerically challenging since the norm of the Euclidean flow would explode. Similarly, Rezende et al. (2020) introduced flows on hyperspheres and tori by using the inverse tangent function. Although this method is empirically shown to perform well, it similarly suffers from numerical instabilities near singularity points.

Riemannian methods In contrast to *projected* methods which rely on mapping the manifold to a Euclidean space, *Riemannian* methods do not. As a consequence they side-step any artefact or numerical instability arising from the manifold’s projection. Early work (Sei, 2011) proposed transformations along geodesics on the hypersphere by evaluating the exponential map at the gradient of a scalar manifold function. Recently, Rezende et al. (2020) introduced ad-hoc *discrete Riemannian* flows for hyperspheres and tori based on Möbius transformations and spherical splines. We contribute to this line of work by introducing *continuous* flows on general Riemannian manifolds. In contrast to *discrete* flows (e.g. Bose et al., 2020; Rezende et al., 2020), *time-continuous* flows as ours alleviate strong structural constraints on the flow by implicitly parametrizing it as the solution to an ODE (Grathwohl et al., 2019). Additionally, recent and concurrent work (Lou et al., 2020; Falorsi and Forré, 2020) proposed to extend neural ODEs to smooth manifolds.

4 Experimental results

We evaluate the empirical performance of the above-mentioned models on hyperbolic and spherical geometry. We will first discuss experiments on two synthetic datasets where we highlight specific pathologies of the naive and projected methods via unimodal distributions at the point (or the limit) of the pathology. This removes additional modelling artefacts that would be introduced through more complex distributions and allows to demonstrate advantages of our approach on the respective manifolds. We further show that these advantages also translate to substantial gains on highly multi-modal real world datasets.

For all *projected* models (e.g. stereographic and wrapped cf Section 3), the vector field’s architecture is chosen to be a multilayer perceptron as in Grathwohl et al. (2019), whilst the architecture described in Section 2 is used for our *Riemannian* (continuous normalizing flow) model. For fair comparisons,

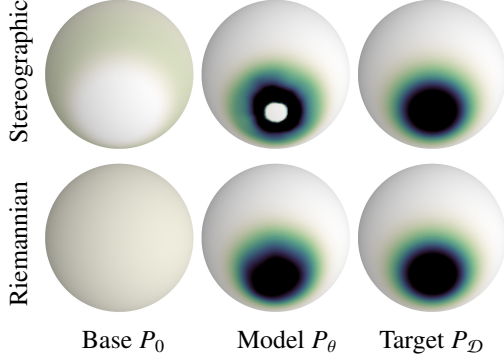


Figure 5: Probability distributions on $\mathbb{S}^2$. Models trained to fit a $\text{vMF}(\mu = -\mu_0, \kappa = 10)$.

	model	Stereographic	Riemannian
Loss	κ		
$\mathcal{L}^{\text{Like}}$	100	$63.60_{\pm 3.56}$	$-1.78_{\pm 0.01}$
	50	$32.68_{\pm 3.15}$	$-1.09_{\pm 0.01}$
	10	$6.45_{\pm 2.42}$	$0.52_{\pm 0.01}$
$\mathcal{L}^{\text{KL}}$	100	$1.56_{\pm 0.34}$	$0.04_{\pm 0.02}$
	50	$0.68_{\pm 0.16}$	$0.03_{\pm 0.02}$
	10	$0.12_{\pm 0.01}$	$0.01_{\pm 0.00}$

Table 2: Performance of continuous flows on $\mathbb{S}^2$ with $\text{vMF}(\mu = -\mu_0, \kappa)$ targets (the smaller the better). When models perfectly fit the target P_D , then $\mathcal{L}^{\text{Like}} = \mathbb{H}[P_D]$ which decreases with κ , explaining $\mathcal{L}^{\text{Like}}$'s results for the *Riemannian* model.

we also parametrize *projected* models with a CNF. Also, all models are chosen to have approximately the same number of parameters. All models were implemented in PyTorch (Paszke et al., 2017) and trained by stochastic optimization with Adam (Kingma and Ba, 2015). All 95% confidence intervals are computed over 12 runs. Please refer to Appendix G for full experimental details.

Hyperbolic geometry and limits of conventional and wrapped methods First, we aim to show that conventional normalizing flows are ill-suited for modelling target manifold distributions. These are blind to the geometry, so we expect them to behave poorly when the target is located where the manifold behaves most differently from a Euclidean space. We refer to such models as *naive* and discuss their properties in more detail in Appendix B.1. Second, we wish to inspect the behaviour of *wrapped* models (see Section 3) when the target is away from the exponential map origin.

To this extent we parametrize a wrapped Gaussian target distribution $\mathcal{N}^{\text{W}}(\exp_0(\alpha \partial x), \Sigma) = \exp_{\mu_0} \mathcal{N}(\alpha \partial x, \Sigma)$ defined on the Poincaré disk $\mathbb{B}^2$ (Nagano et al., 2019; Mathieu et al., 2019). The scalar parameter α allows us to locate the target closer or further away from the origin of the disk. We put three CNFs models on the benchmark; our *Riemannian* (from Section 2), a conventional *naive* and a *wrapped* model. The base distribution P_0 is a standard Gaussian for the *naive* and *wrapped* models, and a standard wrapped Gaussian for the *Riemannian* model. Models are trained by maximum likelihood until convergence. Throughout training, the Hutchinson’s estimator is used to approximate the divergence as in Equation 5. It can be seen from Figure 4 that the *Riemannian* model indeed outperforms the *naive* and *wrapped* models as we increase the values of α – i.e., the closer we move to the boundary of the disk. Figure 2 shows that qualitatively the *naive* and *wrapped* models seem to indeed fail to properly fit the target when it is located far from the origin. Additionally, we assess the architectural choice of the vector field used in our *Riemannian* model. In particular, we conduct an ablation study on the rescaling of the output layer, by training for 10 iterations a *rescaled* and an *unscaled* version of our model. Figure 3 shows that the number of function evaluations (NFE) tends to be large and sometimes even dramatically diverges when the vector field’s output is *unscaled*. In addition to increasing the computational cost, this in turns appears to worsen the convergence’s speed of the model. This further illustrates the benefits of our vector field parameterization.

Spherical geometry and limits of the stereographic projection model Next, we evaluate the ability of our model and the stereographic projection model from Section 3 to approximate distributions on the sphere which are located around the projection point $-\mu_0$. We empirically assess this phenomenon by choosing the target distribution to be a Von-Mises Fisher (Downs, 1972) distribution $\text{vMF}(\mu, \kappa)$ located at $\mu = -\mu_0$, and with concentration κ (which decreases with the variance). Along with the *stereographic* projection method, we also consider our *Riemannian* model from Section 2. We neither included the *naive* model since it is misspecified here (leading to an undefined reverse KL divergence), nor the *wrapped* model as computing its density requires an infinite summation (see Section 3). The base distribution P_0 is chosen to be a standard Gaussian on $\mathbb{R}^2$ for the *stereographic* model and a uniform distribution on $\mathbb{S}^2$ for the *Riemannian* model. Models are trained by computing the exact divergence. The performance of these two models are quantitatively assessed on both the negative log-likelihood and reverse KL criteria.

Table 3: Negative test log-likelihood of continuous normalizing flows on $\mathbb{S}^2$ datasets.

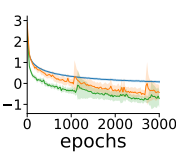
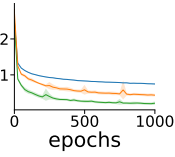
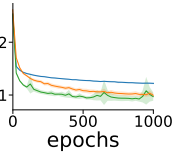
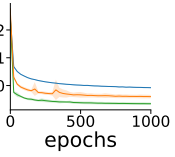
	Volcano	Earthquake	Flood	Fire
Mixture vMF ■	$-0.31_{\pm 0.07}$	$0.59_{\pm 0.01}$	$1.09_{\pm 0.01}$	$-0.23_{\pm 0.02}$
Stereographic ■	$-0.64_{\pm 0.20}$	$0.43_{\pm 0.04}$	$0.99_{\pm 0.04}$	$-0.40_{\pm 0.06}$
Riemannian ■	$-0.97_{\pm 0.15}$	$0.19_{\pm 0.04}$	$0.90_{\pm 0.03}$	$-0.66_{\pm 0.05}$
Learning curves				
Data size	829	6124	4877	12810

Figure 5 shows densities of the target distribution along with the base and learned distributions. We observe that the stereographic model fails to push mass close enough to the singularity point $-\mu_0$, as opposed to the Riemannian model which perfectly fits the target. Table 2 shows the negative log-likelihood and reverse KL losses of both models when varying the concentration parameter κ of the vMF target. The larger the concentration κ is, the closer to the singularity point $-\mu_0$ the target’s mass gets. We observe that the *Riemannian* model outperforms the *stereographic* one to fit the target for both objectives, although this performance gap shrinks as the concentration gets smaller. Also, we believe that the gap in performance is particularly large for the log-likelihood objective because it heavily penalizes models that fail to cover the support of the target. When the vMF target is located away from the singularity point, we noted that both models were performing similarly well.

Density estimation of spherical data Finally, we aim to measure the expressiveness and modelling capabilities of our method on real world datasets. To this extent, we gathered four earth location datasets, representing respectively volcano eruptions (NOAA, 2020b), earthquakes (NOAA, 2020a), floods (Brakenridge, 2017) and wild fires (EOSDIS, 2020). We approximate the earth’s surface (and thus also these data points) as a perfect sphere. Along our *Riemannian* CNF, we also assess the fitting capacity of a mixture of von Mises-Fisher (vMF) distributions and a *stereographic* projected CNF. The locations of the vMF components are learned via stochastic Riemannian optimization (Bonnabel, 2013; Becigneul and Ganea, 2019). The learning rate and number of components are selected by hyperparameter grid search. In our experiments, we split datasets randomly into training and testing datasets, and fit the models by maximum likelihood estimation on the training dataset. CNF models are trained by computing the exact divergence.

We observe from Table 3 that for all datasets, the *Riemannian* model outperforms its *stereographic* counterpart and the mixture of vMF distributions by a large margin. It can also be seen from the learning curves that the *Riemannian* model converges faster. Figure 6 shows the learned spherical distributions along with the training and testing datasets. We note that qualitatively the *stereographic* distribution is generally more diffuse than its *Riemannian* counterpart. It also appears to allocate some of its mass outside the target support, and to cover less of the data points. Additional figures are shown in Appendix H.

Limitations In the spherical setting, the stochastic estimator to approximate the divergence from Equation 5 exhibits high variance. Its variance scales with the inverse of $\sqrt{|G(z)|} = \sin(\theta)$, which becomes too large around the north pole and thus requires the use of the exact estimator. On large-scale datasets, where the stochastic estimator has important runtime advantages, this issue could be alleviated by choosing a different vector field basis than the one induced by some local coordinates (e.g. Falorsi and Forré, 2020). For the Poincaré ball, no such variance behavior of the stochastic estimator exists and it can readily be applied to large-scale data.

Compared to a well-optimized linear layer, the use of the geodesic distance layer (see Section 2) induces an extra computational cost as shown in Figure 7. Empirically, the geodesic layer helps to improve performance in the hyperbolic setting but had less of an effect in the spherical setting. As

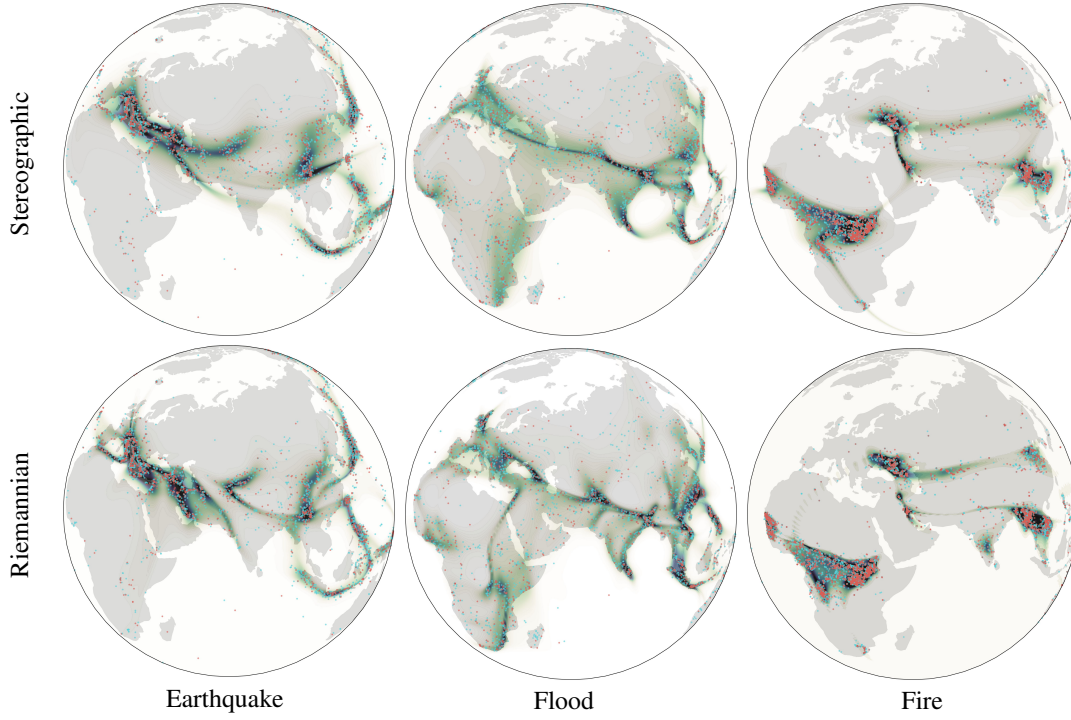


Figure 6: Density estimation for earth sciences data. Blue and red dots represent training and testing datapoints, respectively. Heatmaps depict the log-likelihood of the trained models.

such, the geodesic layer can be regarded as an optional component that can improve the quality of the model at an additional computational cost.

5 Discussion

In this paper we proposed a principled way to parametrize expressive probability distributions on Riemannian manifolds. Specifically, we introduced *Riemannian continuous normalizing flows* in which flows are defined via vector fields on manifolds and computed as the solution to the associated ODE. We empirically demonstrated that this method can yield substantial improvements when modelling data on constant curvature manifolds compared to conventional or projected flows.

Broader impact

The work presented in this paper focuses on the learning of well-specified probabilistic models for manifold-valued data. Consequently, its applications are especially promising to advance scientific understanding in fields such as earth and climate science, computational biology, and computer vision. As a foundational method, our work inherits the broader ethical aspects and future societal consequences of machine learning in general.

Acknowledgments

We are grateful to Adam Foster, Yann Dubois, Laura Ruis, Anthony Caterini, Adam Golinski, Chris Maddison, Salem Said, Alessandro Barp, Tom Rainforth and Yee Whye Teh for fruitful discussions and support. EM research leading to these results received funding from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007- 2013) ERC grant agreement no. 617071 and he acknowledges Microsoft Research and EPSRC for funding EM’s studentship.

References

- Al-Mohy, A. H. and Higham, N. J. (2010). A New Scaling and Squaring Algorithm for the Matrix Exponential. *SIAM Journal on Matrix Analysis and Applications*, 31(3):970–989.
- Ambrosio, L. (2003). Optimal transport maps in Monge-Kantorovich problem. *arXiv:math/0304389 [math]*.
- Avron, H. and Toledo, S. (2011). Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM*, 58(2).
- Becigneul, G. and Ganeva, O.-E. (2019). Riemannian adaptive optimization methods. In *International Conference on Learning Representations*.
- Benamou, J.-D. and Brenier, Y. (2000). A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393.
- Blumenson, L. E. (1960). A derivation of n-Dimensional spherical coordinates. *The American Mathematical Monthly*, 67(1):63–66.
- Bonnabel, S. (2013). Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229.
- Bose, A. J., Smofsky, A., Liao, R., Panangaden, P., and Hamilton, W. L. (2020). Latent variable modelling with hyperbolic normalizing flows. *Proceedings of the 37th International Conference on Machine Learning*.
- Brakenridge, G. (2017). Global active archive of large flood events. <http://floodobservatory.colorado.edu/Archives/index.html>. Dartmouth Flood Observatory, University of Colorado,.
- Chen, R. T. Q., Behrmann, J., Duvenaud, D. K., and Jacobsen, J.-H. (2019). Residual flows for invertible generative modeling. In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 9916–9926. Curran Associates, Inc.
- Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 6571–6583. Curran Associates, Inc.
- Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. (2018). Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*.

- Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using Real NVP. In *International Conference on Learning Representations*.
- Dooley, A. and Wildberger, N. (1993). Harmonic analysis and the global exponential map for compact Lie groups. *Functional Analysis and Its Applications*, 27(1):21–27.
- Dormand, R. J. and Prince, J. P. (1980). A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, pages 19–26.
- Downs (1972). Orientational statistics. *Biometrika*, 59.
- Drucker, H. and Cun, Y. L. (1992). Improving Generalization Performance Using Double Backpropagation. *IEEE Transactions on Neural Networks and Learning Systems*, 3(6):991–997.
- EOSDIS (2020). Active fire data. <https://earthdata.nasa.gov/earth-observation-data/near-real-time/firms/active-fire-data>. Land, Atmosphere Near real-time Capability for EOS (LANCE) system operated by NASA’s Earth Science Data and Information System (ESDIS).
- Falorsi, L., de Haan, P., Davidson, T. R., and Forré, P. (2019). Reparameterizing distributions on lie groups. volume 89 of *Proceedings of Machine Learning Research*, pages 3244–3253. PMLR.
- Falorsi, L. and Forré, P. (2020). Neural Ordinary Differential Equations on Manifolds. *arXiv:2006.06663 [cs, stat]*.
- Finlay, C., Jacobsen, J.-H., Nurbekyan, L., and Oberman, A. M. (2020). How to train your neural ODE. *Proceedings of the 37th International Conference on Machine Learning*.
- Ganea, O., Becigneul, G., and Hofmann, T. (2018). Hyperbolic neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 5345–5355. Curran Associates, Inc.
- Gemici, M. C., Rezende, D., and Mohamed, S. (2016). Normalizing Flows on Riemannian Manifolds. *arXiv:1611.02304 [cs, math, stat]*.
- Grathwohl, W., Chen, R. T. Q., Bettencourt, J., and Duvenaud, D. (2019). Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*.
- Hairer, E. (2006). *Geometric Numerical Integration : Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer Series in Computational Mathematics ; 31. Springer, Berlin, 2nd ed. edition.
- Hairer, E. (2011). Solving Differential Equations on Manifolds. page 55.
- Hamelryck, T., Kent, J. T., and Krogh, A. (2006). Sampling Realistic Protein Conformations Using Local Structural Bias. *PLoS Computational Biology*, 2(9).
- Hasnat, M. A., Bohné, J., Milgram, J., Gentric, S., and Chen, L. (2017). Von Mises-Fisher Mixture Model-based Deep learning: Application to Face Verification. *arXiv:1706.04264 [cs]*.
- Hutchinson (1990). A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450.
- Karpatne, A., Ebert-Uphoff, I., Ravela, S., Babaie, H. A., and Kumar, V. (2019). Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1544–1554.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

- Klimovskaia, A., Lopez-Paz, D., Bottou, L., and Nickel, M. (2020). Poincaré maps for analyzing complex hierarchies in single-cell data. *Nat Commun.*, 11:2966.
- Lee, J. M. (2003). *Introduction to Smooth Manifolds*. Number 218 in Graduate Texts in Mathematics. Springer, New York.
- Lou, A., Lim, D., Katsman, I., Huang, L., Jiang, Q., Lim, S.-N., and De Sa, C. (2020). Neural manifold ordinary differential equations.
- Lui, Y. M. (2012). Advances in matrix manifolds for computer vision. *Image and Vision Computing*, 30(6):380–388.
- Mardia, K. V. and Jupp, P. E. (2000). *Directional Statistics*. Wiley Series in Probability and Statistics. J. Wiley, Chichester ; New York.
- Mathieu, E., Le Lan, C., Maddison, C. J., Tomioka, R., and Teh, Y. W. (2019). Continuous hierarchical representations with poincaré variational auto-encoders. In Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 12565–12576. Curran Associates, Inc.
- Nagano, Y., Yamaguchi, S., Fujita, Y., and Koyama, M. (2019). A Differentiable Gaussian-like Distribution on Hyperbolic Space for Gradient-Based Learning. In *International Conference on Machine Learning (ICML)*.
- Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pages 6341–6350.
- NOAA (2020a). Global significant earthquake database. <https://data.nodc.noaa.gov/cgi-bin/iso?id=gov.noaa.ngdc.mgg.hazards:G012153>. National Geophysical Data Center / World Data Service (NGDC/WDS): NCEI/WDS Global Significant Earthquake Database. NOAA National Centers for Environmental Information.
- NOAA (2020b). Global significant volcanic eruptions database. <https://data.nodc.noaa.gov/cgi-bin/iso?id=gov.noaa.ngdc.mgg.hazards:G10147>. National Geophysical Data Center / World Data Service (NGDC/WDS): NCEI/WDS Global Significant Volcanic Eruptions Database. NOAA National Centers for Environmental Information.
- Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. (2018). Sensitivity and generalization in neural networks: an empirical study. In *International Conference on Learning Representations*.
- Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2019). Normalizing Flows for Probabilistic Modeling and Inference. *arXiv:1912.02762 [cs, stat]*.
- Papamakarios, G., Pavlakou, T., and Murray, I. (2017). Masked autoregressive flow for density estimation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 2338–2347. Curran Associates, Inc.
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NIPS-W*.
- Pennec, X. (2006). Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154.
- Petersen, P. (2006). *Riemannian Geometry*. Springer-Verlag New York.
- Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Beijing, China. PMLR.

- Rezende, D. J., Papamakarios, G., Racanière, S., Albergo, M. S., Kanwar, G., Shanahan, P. E., and Cranmer, K. (2020). Normalizing Flows on Tori and Spheres. *Proceedings of the 37th International Conference on Machine Learning*.
- Salman, H., Yadollahpour, P., Fletcher, T., and Batmanghelich, K. (2018). Deep Diffeomorphic Normalizing Flows. *arXiv:1810.03256 [cs, stat]*.
- Sei, T. (2011). A Jacobian Inequality for Gradient Maps on the Sphere and Its Application to Directional Statistics. *Communications in Statistics - Theory and Methods*, 42(14):2525–2542.
- Skopek, O., Ganea, O.-E., and Bécigneul, G. (2020). Mixed-curvature variational autoencoders. In *International Conference on Learning Representations*.
- Ungar, A. A. (2008). A gyrovector space approach to hyperbolic geometry. *Synthesis Lectures on Mathematics and Statistics*, 1(1):1–194.

Appendix for

Riemannian Continuous Normalizing Flows

A Constant curvature manifolds

In the following, we provide a brief overview of Riemannian geometry and constant curvature manifolds, specifically the Poincaré ball and the hypersphere models. We will use $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ to denote the Euclidean norm and inner product. For norms and inner products on tangent spaces $\mathcal{T}_z\mathcal{M}$, we write $\|\cdot\|_z$ and $\langle \cdot, \cdot \rangle_z$ where $z \in \mathcal{M}$.

A.1 Review of Riemannian geometry

A real, smooth *manifold* $\mathcal{M}$ is a set of points z , which is "locally similar" to a linear space. For every point z of the manifold $\mathcal{M}$ is attached a real vector space of the same dimensionality as $\mathcal{M}$ called the *tangent space* $\mathcal{T}_z\mathcal{M}$. Intuitively, it contains all the possible directions in which one can tangentially pass through z . Taking the disjoint union of all tangent spaces yields the *tangent bundle* $\mathcal{TM} = \bigcup_{z \in \mathcal{M}} \mathcal{T}_z\mathcal{M}$. For each point z of the manifold, the *metric tensor* $g(z)$ defines an inner product on the associated tangent space as $g(z) = \langle \cdot, \cdot \rangle_z : \mathcal{T}_z\mathcal{M} \times \mathcal{T}_z\mathcal{M} \rightarrow \mathbb{R}$. The *matrix representation of the Riemannian metric* $G(z)$, is defined such that

$$\forall u, v \in \mathcal{T}_z\mathcal{M} \times \mathcal{T}_z\mathcal{M}, \langle u, v \rangle_z = g(z)(u, v) = u^T G(z) v.$$

A *Riemannian manifold* is then given as a tuple $(\mathcal{M}, g)$ (Petersen, 2006). The metric tensor gives a *local* notion of angle, length of curves, surface area and volume, from which *global* quantities can be derived by integrating local contributions. A norm is induced by the inner product on $\mathcal{T}_z\mathcal{M}$: $\|\cdot\|_z = \sqrt{\langle \cdot, \cdot \rangle_z}$. An infinitesimal volume element is induced on each tangent space $\mathcal{T}_z\mathcal{M}$, and thus a measure $d\text{Vol}(z) = \sqrt{|G(z)|} d\text{Leb}(z)$ on the manifold, with $\text{Leb}(z)$ being the Lebesgue measure. The length of a curve $\gamma : t \mapsto \gamma(t) \in \mathcal{M}$ is given by $L(\gamma) = \int_0^1 \|\gamma'(t)\|_{\gamma(t)} dt$. The concept of straight lines can then be generalized to *geodesics*, which are constant speed curves giving the shortest path between pairs of points z, y of the manifold: $\gamma^* = \arg \min L(\gamma)$ with $\gamma(0) = z$, $\gamma(1) = y$ and $\|\gamma'(t)\|_{\gamma(t)} = 1$. A *global* distance is thus induced on $\mathcal{M}$ given by

$$d_{\mathcal{M}}(z, y) = \inf L(\gamma).$$

Endowing $\mathcal{M}$ with that distance consequently defines a metric space $(\mathcal{M}, d_{\mathcal{M}})$. The concept of moving along a "straight" curve with constant velocity is given by the *exponential map*. In particular, there is a unique unit speed *geodesic* γ satisfying $\gamma(0) = z$ with initial tangent vector $\gamma'(0) = v$. The corresponding exponential map is then defined by $\exp_z(v) = \gamma(1)$. The *logarithm map* is the inverse $\log_z = \exp_z^{-1} : \mathcal{M} \rightarrow \mathcal{T}_z\mathcal{M}$. The $\exp_z$ map is well-defined on the full tangent space $\mathcal{T}_z\mathcal{M}$ for all $z \in \mathcal{M}$ if and only if $\mathcal{M}$ is geodesically complete, i.e. if all geodesics can "run" indefinitely. This is the case for the Poincaré ball and hypersphere.

A.2 The Poincaré ball model of hyperbolic geometry

In the following, we provide a brief overview of key concepts related to hyperbolic geometry. A d -dimensional *hyperbolic* space is a complete, simply connected, d -dimensional Riemannian manifold with *constant negative curvature* K . The *Poincaré ball* is one model of this geometry, and is formally defined as the Riemannian manifold $\mathbb{B}_K^d = (\mathcal{B}_K^d, g_K)$. Here $\mathcal{B}_K^d$ denotes the open ball of radius $1/\sqrt{|K|}$, and g_K the *metric tensor* $g_K(z) = (\lambda_z^K)^2 g^e(z)$, where $\lambda_z^K = \frac{2}{1+K\|z\|^2}$ and g^e denotes the Euclidean metric tensor, i.e. the usual dot product. The induced *invariant measure* Vol is absolutely continuous with respect to the Lebesgue measure Leb , and its density is given by $\frac{d\text{Vol}}{d\text{Leb}}(z) = \sqrt{|G(z)|} = (\lambda_z^K)^d$ for all $z \in \mathbb{B}_K^d$. As motivated by Skopek et al. (2020), the Poincaré ball $\mathbb{B}_K^d$ can conveniently be described through the formalism of *gyrovector spaces* (Ungar, 2008). These can be seen as an analogy to the way vector spaces are used in Euclidean geometry, but in the non-Euclidean geometry setting. In

particular, the *Möbius addition* $\oplus_K$ of $\mathbf{z}, \mathbf{y}$ in $\mathbb{B}_K^d$ is defined as

$$\mathbf{z} \oplus_K \mathbf{y} = \frac{(1 - 2K \langle \mathbf{z}, \mathbf{y} \rangle - K \|\mathbf{y}\|^2) \mathbf{z} + (1 + K \|\mathbf{z}\|^2) \mathbf{y}}{1 - 2K \langle \mathbf{z}, \mathbf{y} \rangle + K^2 \|\mathbf{z}\|^2 \|\mathbf{y}\|^2}.$$

Then the *exponential map* can be expressed via this *Möbius addition* as

$$\exp_z^K(\mathbf{v}) = \mathbf{z} \oplus_K \left(\tanh \left(\sqrt{-K} \frac{\lambda_z^K \|\mathbf{v}\|}{2} \right) \frac{\mathbf{v}}{\sqrt{-K} \|\mathbf{v}\|} \right)$$

where $\mathbf{x} = -\mathbf{z} \oplus_K \mathbf{y}$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{B}_K^d$.

A.3 The hypersphere model of elliptic geometry

In the following, we discuss key concepts related to positively curved spaces known as *elliptic* spaces, and in particular to the *hypersphere* model. The d -sphere, or *hypersphere*, is a compact submanifold of $\mathbb{R}^{d+1}$ with positive constant curvature K whose support is defined by $\mathcal{S}_K^d = \{\mathbf{z} \in \mathbb{R}^{d+1} \mid \langle \mathbf{z}, \mathbf{z} \rangle = 1/K\}$. It is endowed with the pull-back metric of the ambient Euclidean space.

Sphere In the two-dimensional setting $d = 2$, we rely on polar coordinates to parametrize the sphere $\mathbb{S}^2$. These coordinates consist of polar $\theta \in [0, \pi]$ and azimuth $\varphi \in [0, 2\pi)$ angles. The ambient Cartesian coordinates are then given by $\mathbf{r}(\theta, \varphi) = (\sin(\theta) \cos(\varphi), \sin(\theta) \sin(\varphi), \cos(\theta))$. We have $\sqrt{|G(\theta, \varphi)|} = \sin(\theta)$. Applying the generic divergence formula (see Equation 11) yields the celebrated spherical divergence formula

$$\text{div}(\mathbf{g}) = \frac{1}{\sin(\theta)} \frac{\partial}{\partial \theta} (\sin(\theta) g^\theta(\theta, \varphi)) + \frac{1}{\sin(\theta)} \frac{\partial}{\partial \varphi} (g^\varphi(\theta, \varphi)).$$

Hypersphere For higher dimensions, we can rely on the n -spherical coordinate system in which the coordinates consist of $d - 1$ angular coordinates $\varphi_1, \dots, \varphi_{d-2} \in [0, \pi]$ and $\varphi_{d-1} \in [0, 2\pi)$ (Blumenson, 1960). Then we have $\sqrt{|G(\varphi)|} = \sin^{d-2}(\varphi_1) \sin^{d-3}(\varphi_2) \dots \sin(\varphi_{d-2})$.

Using the ambient cartesian coordinates, the *exponential map* is given by

$$\exp_\mu^c(\mathbf{v}) = \cos(\sqrt{K} \|\mathbf{v}\|) \boldsymbol{\mu} + \sin(\sqrt{K} \|\mathbf{v}\|) \frac{\mathbf{v}}{\sqrt{K} \|\mathbf{v}\|}$$

for all $\mathbf{z} \in \mathbb{S}_K^d$ and $\mathbf{v} \in \mathcal{T}_z \mathbb{S}_K^d$.

B Probability measures on Riemannian manifolds

In what follows, we discuss core concepts of probability measures on Riemannian manifolds and show how naive methods lead to ill- and mis-specified models on manifolds.

Probability measures and random vectors can intrinsically be defined on Riemannian manifolds so as to model uncertainty on non-flat spaces (Pennec, 2006). The Riemannian metric $G(\mathbf{z})$ induces an infinitesimal volume element on each tangent space $\mathcal{T}_z \mathcal{M}$, and thus a measure on the manifold,

$$d \text{Vol}(\mathbf{z}) = \sqrt{|G(\mathbf{z})|} d \text{Leb},$$

with Leb the Lebesgue measure. Manifold-valued random variables would naturally be characterized by the *Radon-Nikodym derivative* of a measure ν w.r.t. the Riemannian measure Vol (assuming absolute continuity)

$$p(\mathbf{z}) = \frac{d\nu}{d \text{Vol}}(\mathbf{z}).$$

B.1 Ambient Euclidean probability distributions

Unfortunately, conventional probabilistic models implicitly assume a *flat* geometry. This in turn cause these models to either be misspecified or ill-suited to fit manifold distributions. Below we discuss the reasons why.

Let $P_{\mathcal{D}}$ be a target probability measure that we aim to approximate, and which is defined on a d -dimensional manifold $\mathcal{M} \subseteq \mathbb{R}^D$. Furthermore, we assume it admits a Radon-Nikodym derivative $p_{\mathcal{D}}$ with respect to the manifold invariant measure Vol , denoting $P_{\mathcal{D}} \ll \text{Vol}$ with $\ll$ denoting absolute continuity. Conventional normalizing flows implicitly assume the parametrized probability measure P_{θ} to have support on the ambient space $\mathbb{R}^D$ and to be absolutely continuous with respect to the Lebesgue measure $\text{Leb}_{\mathbb{R}^D}$. We denote its density by p_{θ} .

Next, assume $D = d$, such as for $\mathcal{M} = \mathbb{B}^d \subseteq \mathbb{R}^d$. With z the d -dimensional Cartesian coordinates, we have $\frac{d\text{Vol}}{d\text{Leb}_{\mathbb{R}^d}}(z) = \sqrt{|G(z)|}$. One could then see the manifold-valued target $P_{\mathcal{D}}$ as being a probability measure on $\mathbb{R}^d$ with a density w.r.t. the Lebesgue measure given by

$$\frac{dP_{\mathcal{D}}}{d\text{Leb}_{\mathbb{R}^d}}(z) = p_{\mathcal{D}}(z) \sqrt{|G(z)|} \triangleq \tilde{p}_{\mathcal{D}}(z).$$

In general $P_{\mathcal{D}} \ll P_{\theta}$ which implies that the *forward* Kullback-Leibler divergence, or *negative log-likelihood* up to constants, is defined and given by

$$\mathcal{L}^{\text{Like}}(\theta) = D_{\text{KL}}(P_{\mathcal{D}} \parallel P_{\theta}) + \mathbb{H}(P_{\mathcal{D}}) = \mathbb{E}_{P_{\mathcal{D}}} \left[\log \left(\frac{\tilde{p}_{\mathcal{D}}(z)}{p_{\theta}(z)} \right) \right] + \mathbb{H}(P_{\mathcal{D}}) = -\mathbb{E}_{P_{\mathcal{D}}} \left[\log \frac{p_{\theta}(z)}{\sqrt{|G(z)|}} \right].$$

Minimising $\mathcal{L}^{\text{Like}}(\theta)$ amounts to pushing-forward P_{θ} 's mass so that empirical observations $z_i \sim P_{\mathcal{D}}$ have a positive likelihood under P_{θ} . Yet, in general the model (P_{θ}) has (most of his) mass outside the manifold's support which may cause such a *naive* approach to be ill-suited. More crucially it implies that in general the model's mass is *not* covering the full target's support. In that case, the *reverse* Kullback-Leibler divergence $D_{\text{KL}}(P_{\theta} \parallel P_{\mathcal{D}}) = \mathcal{L}^{\text{KL}}(\theta)$ is not even defined.

Next, consider the case where $\mathcal{M}$ is a submanifold embedded in $\mathbb{R}^D$ with $D > d$, such as $\mathcal{M} = \mathbb{S}^d$ where $D = d + 1$. In this setting the *naive* model P_{θ} is even *misspecified* since it is defined on a different probability space than the target. In the limit $\text{supp}(P_{\theta}) \rightarrow \mathcal{M}$, P_{θ} is not defined because we have that $\int_{\mathbb{R}^D} P_{\theta} \rightarrow \infty$. The target does consequently not belong to the model's class.

C Instantaneous change of variable

In the following we derive the *instantaneous change of density* that a manifold-valued random variable induces when its dynamics are governed by an ODE. We show that in the Riemannian setting this *instantaneous change of density* can be expressed in terms of the manifold's metric tensor.

Proof of Proposition 2

Proof. For a time dependant particles $z(t)$, whose dynamics are given by the following ODE

$$\frac{dz(t)}{dt} = f(z(t), t)$$

the change in density is given by the Liouville equation (or Fokker-Planck equation without the diffusion term); $\forall z \in \mathcal{M}, \forall t \in [0, T]$

$$\begin{aligned} \frac{\partial}{\partial t} p(z, t) &= -\text{div}(p(z, t) f(z, t)) \\ &= -\left\langle \frac{\partial}{\partial z} p(z, t), f(z, t) \right\rangle_z - p(z, t) \text{div}(f(z, t)) \end{aligned}$$

where the last step was obtained by applying the divergence product rule. By introducing the time dependence in $z(t)$ and differentiating with respect to time we get

$$\begin{aligned} \frac{d}{dt} p(z(t), t) &= \left\langle \frac{\partial}{\partial z} p(z(t), t), \frac{\partial}{\partial t} z(t) \right\rangle_{z(t)} + \frac{\partial}{\partial t} p(z(t), t) \\ &= \left\langle \frac{\partial}{\partial z} p(z(t), t), f(z(t), t) \right\rangle_{z(t)} - \left\langle \frac{\partial}{\partial z} p(z(t), t), f(z(t), t) \right\rangle_{z(t)} - p(z(t), t) \text{div}(f(z(t), t)) \\ &= -p(z(t), t) \text{div}(f(z(t), t)) \end{aligned}$$

Hence the evolution of the log density is given by

$$\frac{d}{dt} \log p(z(t), t) = -\text{div}(f(z(t), t)). \quad (8)$$

□

Divergence computation For a Riemannian manifold $(\mathcal{M}, g)$, with local coordinates $\mathbf{z}$, the divergence of a vector field $\mathbf{f}$ is given by

$$\operatorname{div}(\mathbf{f}(\mathbf{z}, t)) = \frac{1}{\sqrt{|G(\mathbf{z})|}} \sum_{i=1}^d \frac{\partial}{\partial z^i} \left(\sqrt{|G(\mathbf{z})|} f^i(\mathbf{z}, t) \right) \quad (9)$$

$$\begin{aligned} &= \frac{1}{\sqrt{|G(\mathbf{z})|}} \sum_{i=1}^d \left(\sqrt{|G(\mathbf{z})|} \frac{\partial}{\partial z^i} f^i(\mathbf{z}, t) + f^i(\mathbf{z}, t) \frac{\partial}{\partial z^i} \sqrt{|G(\mathbf{z})|} \right) \\ &= \sum_{i=1}^d \frac{\partial}{\partial z^i} f^i(\mathbf{z}, t) + \frac{1}{\sqrt{|G(\mathbf{z})|}} \sum_{i=1}^d f^i(\mathbf{z}, t) \frac{\partial}{\partial z^i} \sqrt{|G(\mathbf{z})|} \\ &= \operatorname{tr} \left(\frac{\partial}{\partial \mathbf{z}} \mathbf{f}(\mathbf{z}, t) \right) + \frac{1}{\sqrt{|G(\mathbf{z})|}} \left\langle \mathbf{f}(\mathbf{z}, t), \frac{\partial}{\partial \mathbf{z}} \sqrt{|G(\mathbf{z})|} \right\rangle. \end{aligned} \quad (10)$$

We note that in Equation 9, f_i are the components of the vector field $\mathbf{f}$ with respect to the local unnormalized covariant basis $(\mathbf{e}_i)_{i=1}^d = \left(\left(\frac{\partial}{\partial z^i} \right)_{\mathbf{z}} \right)_{i=1}^d$. However it is convenient to work with local basis having unit length vectors. If we write $\hat{\mathbf{e}}_i$ for this normalized basis, and $\hat{f}^i$ for the components of $\mathbf{f}$ with respect to this normalized basis, we have that

$$\mathbf{f} = \sum_i f^i \mathbf{e}_i = \sum_i f^i \|\mathbf{e}_i\| \frac{\mathbf{e}_i}{\|\mathbf{e}_i\|} = \sum_i f^i \sqrt{G_{ii}} \frac{\mathbf{e}_i}{\|\mathbf{e}_i\|} = \sum_i \hat{f}^i \hat{\mathbf{e}}_i$$

using one of the properties of the metric tensor. By dotting both sides of the last equality with the contravariant element $\hat{\mathbf{e}}_i$ we get that $\hat{f}^i = f^i \sqrt{G_{ii}}$. Substituting in Equation 9 yields

$$\operatorname{div}(\hat{\mathbf{f}}(\mathbf{z}, t)) = \frac{1}{\sqrt{|G(\mathbf{z})|}} \sum_{i=1}^d \frac{\partial}{\partial z^i} \left(\sqrt{\frac{|G(\mathbf{z})|}{G_{ii}(\mathbf{z})}} \hat{f}^i(\mathbf{z}, t) \right). \quad (11)$$

Combining Equations 8 and 11 and we finally get

$$\frac{\partial \log p(\mathbf{z}(t), t)}{\partial t} = - \frac{1}{\sqrt{|G(\mathbf{z})|}} \sum_{i=1}^d \frac{\partial}{\partial z^i} \left(\sqrt{\frac{|G(\mathbf{z})|}{G_{ii}(\mathbf{z})}} \hat{f}^i(\mathbf{z}, t) \right). \quad (12)$$

We rely on this Equation 12 for practical numerical experiments.

D regularization

D.1 ℓ^2 -norm

Henceforth we motivate the use of an ℓ^2 norm regularization in the context of continuous normalizing flows. We do so by highlighting a connection with the dynamical formulation of optimal transport, and by proving that this formulation still holds in the manifold setting.

Monge-Kantorovich mass transfer problem Let $(\mathcal{M}, d_{\mathcal{M}})$ be a metric space, and $c : \mathcal{M} \times \mathcal{M} \rightarrow [0, +\infty)$ a measurable map. Given probability measures p_0 and p_T on $\mathcal{M}$, Monge's formulation of the optimal transportation problem is to find a transport map $\phi^* : \mathcal{M} \rightarrow \mathcal{M}$ that realizes the infimum

$$\inf_{\phi} \int_{\mathcal{M}} c(\phi(\mathbf{z}), \mathbf{z}) p_0(d\mathbf{z}) \text{ s.t. } p_T \# = p_0.$$

It can be shown that this yields a metric on probability measures, and for $c = d_{\mathcal{M}}^2$, it is called the L^2 Kantorovich (or Wasserstein) distance

$$d_{W^2}(p_0, p_T)^2 = \inf_{\phi} \int_{\mathcal{M}} d_{\mathcal{M}}(\phi(\mathbf{z}), \mathbf{z})^2 p_0(d\mathbf{z}). \quad (13)$$

By reintroducing the time variable in the L^2 Monge-Kantorovich mass transfer problem, the optimal transport map ϕ^* can be reformulated as the generated flow from an optimal vector field $\mathbf{f}$.

Proposition 3 (Dynamical formulation from (Benamou and Brenier, 2000)). *Indeed we have*

$$d_{W^2}(p_0, p_T)^2 = \inf \frac{1}{T} \int_0^T \|\mathbf{f}\|_{p_t}^2 dt = \inf \frac{1}{T} \int_0^T \int_{\mathcal{M}} \langle \mathbf{f}(z, t), \mathbf{f}(z, t) \rangle_z p_t(dz) dt \quad (14)$$

where the infimum is taken among all weakly continuous distributional solutions of the continuity equation $\frac{\partial}{\partial t} p_t = -\text{div}(p_t \mathbf{f})$ such that $p(0) = p_0$ and $p(T) = p_T$. Writing $\phi_t^* = \phi^*(\cdot, t)$ the flow generated by the optimal ODE, then the optimal transport map is given by $\phi^* = \phi_T^*$.

The RHS of Equation (14) can then be approximated with no extra-cost with a Monte Carlo estimator given samples from $p_t = \phi_t \# p_0$.

Manifold Setting Let's now focus on the setting where $\mathcal{M}$ is a Riemannian manifold.

Proposition 4 (Optimal map (Ambrosio, 2003)). *Assume that $\mathcal{M}$ is a C^3 , complete Riemannian manifold with no boundary and $d_{\mathcal{M}}$ is the Riemannian distance. If p_0, p_T have finite second order moments and p_0 is absolutely continuous with respect to $\text{vol}_{\mathcal{M}}$, then there exists a unique optimal transport map ϕ for the Monge-Kantorovich problem with cost $c = d_{\mathcal{M}}^2$. Moreover there exists a potential $h : \mathcal{M} \mapsto \mathbb{R}$ such that*

$$\phi^*(z) = \exp_z(-\nabla h(z)) \quad \text{vol}_{\mathcal{M}} - a.e..$$

Proposition 3 has been stated and proved for the case $\mathcal{M} = \mathbb{R}^d$. Below we extend the proof given by Benamou and Brenier (2000) for the manifold setting.

Proof of Proposition 3. We follow the same reasoning as the one developed for the Euclidean setting. Let's first upper bound the Wasserstein distance, and then state the optimal flow which yields equality. We have

$$\begin{aligned} \frac{1}{T} \int_0^T \int_{\mathcal{M}} \|\mathbf{f}(z, t)\|_z^2 p_t(dz) dt &= \frac{1}{T} \int_0^T \int_{\mathcal{M}} \|\mathbf{f}(\phi(z, t), t)\|_z^2 p_0(dz) dt \\ &= \frac{1}{T} \int_0^T \int_{\mathcal{M}} \left\| \frac{\partial}{\partial t} \phi(z, t) \right\|_z^2 p_0(dz) dt \\ &\geq \int_{\mathcal{M}} d_{\mathcal{M}}(\phi(z, T), \phi(z, 0))^2 p_0(dz) dt \\ &= \int_{\mathcal{M}} d_{\mathcal{M}}(\phi(z, T), z)^2 p_0(dz) dt \\ &\geq \int_{\mathcal{M}} d_{\mathcal{M}}(\phi(z), z)^2 p_0(dz) dt \\ &= d_{W^2}(p_0, p_T)^2. \end{aligned}$$

Thus, the optimal choice of flow ϕ is given by

$$\phi(z, t) = \exp_z \left(\frac{t}{T} \log_z(\phi^*(z)) \right), \quad (15)$$

since $\phi(z, 0) = z$, $\phi(z, T) = \phi^*(z)$ and

$$\left\| \frac{\partial}{\partial t} \phi(z, t) \right\|_z = \left\| \frac{\partial}{\partial t} \phi(z, t=0) \right\|_z = \|\log_z(\phi^*(z))\|_z = d_{\mathcal{M}}(\phi^*(z), z).$$

□

Note that the optimal flow from Equation 15 yields integral paths $\gamma(t) = \phi(z, t)$ that are *geodesics* and have constant *velocity*.

Motivation Regularizing the vector field with the RHS of Equation 14 would hence tend to make the generated flow ϕ_T closer to the optimal map ψ^* . By doing so, one hopes to increase smoothness of $\mathbf{f}$ and consequently lower the solver NFE given a fixed tolerance.

This has been observed in the Euclidean setting by Finlay et al. (2020). They empirically showed that regularizing the loss of a CNF with the vector field's l^2 norm improves training speed. Motivated by the successful use of gradient regularization (Novak et al., 2018; Drucker and Cun, 1992), they showed that additionally regularizing the Frobenius norm of the vector field's Jacobian helps. In the following subsection we remind that this regularization term can also be motivated from an estimator's variance perspective.

D.2 Frobenius norm

Hutchinson’s estimator Hutchinson’s estimator (Hutchinson, 1990) is a simple way to obtain a stochastic estimate of the trace of a matrix. Given a d -dimensional random vector $\epsilon \sim p$ such that $\mathbb{E}[\epsilon] = 0$ and $\text{Cov}(\epsilon) = I_d$, we have

$$\text{tr}(A) = \mathbb{E}_{\epsilon \sim p}[\epsilon^T A \epsilon].$$

Rademacher and Gaussian distributions have been used in practice. For a Rademacher, the variance is given by (Avron and Toledo, 2011)

$$\mathbb{V}_{\epsilon \sim p}[\epsilon^T A \epsilon] = 2 \|A\|_F^2 - 2 \sum_i A_{ii}^2,$$

whereas for a Gaussian it is given by

$$\mathbb{V}_{\epsilon \sim p}[\epsilon^T A \epsilon] = 2 \|A\|_F^2.$$

Divergence computation As reminded in Appendix C by Equation 10, computing the vector field divergence $\text{div}(\mathbf{f}(\mathbf{z}, t))$ involves the computation of the trace of vector field’s Jacobian $\text{tr}\left(\frac{\partial}{\partial \mathbf{z}} \mathbf{f}(\mathbf{z}, t)\right)$. As highlighted in Grathwohl et al. (2019); Salman et al. (2018), one can rely on the Hutchinson’s estimator to estimate this trace with $A = \frac{\partial}{\partial \mathbf{z}} \mathbf{f}(\mathbf{z}, t)$.

The variance of this estimator thus depends on the Frobenius norm of the vector’s field Jacobian $\|\frac{\partial}{\partial \mathbf{z}} \mathbf{f}(\mathbf{z}, t)\|_F$, as noted in Grathwohl et al. (2019). Regularizing this Jacobian should then improve training by reducing the variance of the divergence estimator.

E Vector flows and neural architecture

Hereafter we discuss about flows generated by vector fields, and neural architectural choices that we make for their parametrization. Properties of vector fields have direct consequences on the properties of the generated flow and in turn on the associated pushforward probability distributions. In particular we derive sufficient conditions on the flow so that it is *global*, i.e. is a bijection mapping the manifold to itself.

E.1 Existence and uniqueness of a global flow

We start by discussing about vector flows and sufficient conditions on their uniqueness and existence.

Local flow First we remind the Fundamental theorem of flows (Lee, 2003) which gives the existence and uniqueness of a smooth *local flow*.

Proposition 5 (Fundamental theorem of flows). *Let $\mathcal{M}$ be a smooth complete manifold with local coordinates $\mathbf{z}$. Let $f_\theta : \mathcal{M} \times \mathbb{R} \mapsto \mathcal{TM}$ a C^1 time-dependent vector field and $\mathbf{z}_0 \in \mathcal{M}$. Then there exists an open interval I with $0 \in I$, an open subset $U \subseteq \mathcal{M}$ containing $\mathbf{z}_0$, and a unique smooth map $\phi : I \times U \mapsto \mathcal{M}$ called local flow which satisfies the following properties. We write $\phi_t(\mathbf{z}) = \phi(\mathbf{z}, t)$.*

1. $\frac{\partial}{\partial t} \phi(\mathbf{z}, t) = f_\theta(\phi(\mathbf{z}, t), t)$ for all $\mathbf{z}, t \in U \times I$, and $\phi_0 = \text{id}_{\mathcal{M}}$.
2. For each $t \in I$, the map $\phi_t : U \mapsto \mathcal{M}$ is a local C^1 -diffeomorphism.

Note that with such assumptions, the existence and uniqueness of flows ϕ_t are only *local*.

Global flow We would like the flow ϕ to be defined for all times and on the whole manifold, i.e. a *global flow* $\phi : \mathcal{M} \times \mathbb{R} \mapsto \mathcal{M}$. Fortunately, if $\mathcal{M}$ is *compact* (such as n -spheres and torii), then the flow is global (Lee, 2003). We show below that another sufficient condition for the flow to be global is that the vector field be bounded.

Proposition 6 (Global Flow). *Let $\mathcal{M}$ be a smooth complete manifold. Let $f_\theta : \mathcal{M} \times \mathbb{R} \mapsto \mathcal{TM}$ be a C^1 bounded time-dependent vector field. Then the domain of the flow ϕ is $\mathbb{R} \times \mathcal{M}$, i.e. the flow is global.*

Corollary 6.1. *For each $t \in \mathbb{R}$, the map $\phi_t : \mathcal{M} \mapsto \mathcal{M}$ is a C^1 -diffeomorphism.*

Proof of Proposition 6. Let $c > 0$ s.t. $\|f\| < c$, and $z_0 \in \mathcal{M}$ be an initial point. Proposition 5 gives the existence of an open interval $I = (a, b)$, a neighbourhood U of z_0 and a local flow $\phi : (a, b) \times U \mapsto \mathcal{M}$. We write $\gamma = \phi(z_0, \cdot)$. The maximal interval of γ is (a, b) , which means that γ cannot be extended outside (a, b) . Suppose that $b < \infty$.

The integral path γ is Lipschitz continuous on (a, b) since we have

$$d_{\mathcal{M}}(\gamma(t), \gamma(s)) \leq \int_s^t \|\gamma'(t)\| dt = \int_s^t \|f(\gamma(t), t)\| dt \leq c |t - s| \quad (16)$$

for all $s < t \in (a, b)$.

Let (t_n) be a sequence in (a, b) that converges to b . Then since (t_n) is a convergent sequence, it must also be a Cauchy sequence. Then $\gamma(t_n)$ is also a Cauchy sequence by Equation 16. Since $\mathcal{M}$ is geodesically complete, it follows by Hopf-Rinow theorem that $(\mathcal{M}, d_{\mathcal{M}})$ is complete, hence that $\gamma(t_n)$ converges to a point $p \in \mathcal{M}$.

Now suppose that (s_n) is another sequence in (a, b) that converges to b . Then by Equation 16 $\lim_{n \rightarrow \infty} d(\gamma(s_n), \gamma(t_n)) = 0$, thus $\gamma(s_n)$ also converges to $\lim_{n \rightarrow \infty} \gamma(t_n) = p$. So for every sequence (t_n) in (a, b) that converges to b , we have that $(\gamma(t_n))$ converges to p . Therefore by the sequential criterion for limits, we have that γ has the limit p at the point b . Therefore, define $\gamma(b) = p$ and so γ is continuous at b which is a contradiction. $\square$

E.2 Geodesic distance layer

The expressiveness of CNFs directly depends on the expressiveness of the vector field and consequently on its architecture. Below we detail and motivate the use of a *geodesic distance* layer, as an input layer for the vector field neural architecture.

Linear layer A linear layer with one *neuron* can be written in the form $h_{a,p}(z) = \langle a, z - p \rangle$, with orientation and offset parameters $a, p \in \mathbb{R}^d$. Stacking l such neurons h yields a linear layer with width l . This neuron can be rewritten in the form

$$h_{a,p}(z) = \text{sign}(\langle a, z - p \rangle) \|a\| d_E(z, H_{a,p}^K)$$

where $H_{a,p} = \{z \in \mathbb{R}^p \mid \langle a, z - p \rangle = 0\} = p + \{a\}^\perp$ is the decision hyperplane. The third term is the distance between z and the decision hyperplane $H_{a,p}^K$ and the first term refers to the side of $H_{a,p}^K$ where z lies.

Poincaré ball Ganea et al. (2018) analogously introduced a neuron $f_{a,p}^K : \mathbb{B}_K^d \rightarrow \mathbb{R}^p$ on the Poincaré ball,

$$h_{a,p}^K(z) = \text{sign}\left(\left\langle a, \log_p^K(z) \right\rangle_p\right) \|a\|_p d^K(z, H_{a,p}^K) \quad (17)$$

with $H_{a,p}^K = \{z \in \mathbb{B}_K^d \mid \langle a, \log_p^K(z) \rangle_p = 0\} = \exp_p^K(\{a\}^\perp)$. A closed-form expression for the distance $d^K(z, H_{a,p}^K)$ was also derived, $d^K(z, H_{a,p}^K) = \frac{1}{\sqrt{|K|}} \sinh^{-1} \left(\frac{2\sqrt{|K|} \langle -p \oplus_K z, a \rangle}{(1+K\| -p \oplus_K z \|^2) \|a\|} \right)$ in the Poincaré ball. To avoid an over-parametrization of the hyperplane, we set $p = \exp_0(a_0)$, and $a = \Gamma_{0 \rightarrow p}(a_0)$ with Γ parallel transport (under Levi-Civita connection). We observed that the term $\|a\|_p$ from Equation 17 was sometimes causing numerical instabilities, and that when it was not it also did not improve performance. We consequently removed this scaling term. The hyperplane decision boundary $H_{a,p}^K$ is called *gyroplane* and is a semi-hypersphere orthogonal to the Poincaré ball's boundary.

Hypersphere In hyperspherical geometry, geodesics are great circles which can be parametrized by a vector $w \in \mathbb{R}^{d+1}$ as $H_w = \{z \in \mathbb{S}^d \mid \langle w, z \rangle = 0\}$. The geodesic distance between $z \in \mathbb{S}^d$ and the hyperplane H_w is then given by

$$d(z, H_w) = \left| \sin^{-1} \left(\frac{\langle w, z \rangle}{\sqrt{\langle w, w \rangle}} \right) \right|.$$

In a similar fashion, a neuron is now defined by

$$h_w(z) = \|w\|_2 \sin^{-1} \left(\frac{\langle w, z \rangle}{\sqrt{\langle w, w \rangle}} \right).$$

Geodesic distance layer One can then horizontally-stack l neurons to make a *geodesic distance* layer $g : \mathcal{M} \mapsto \mathbb{R}^l$ (Mathieu et al., 2019). Any standard feed-forward neural network can then be vertically-stacked on top of this layer.

F Extensions

F.1 Product of manifolds

Having described CNFs for complete smooth manifolds in Section 2, we extend these for product manifolds $\mathcal{M} = \mathcal{M}_1 \times \dots \times \mathcal{M}_k$. For instance a d -dimensional torus is defined as $\mathbb{T}^d = \underbrace{\mathbb{S}^1 \times \dots \times \mathbb{S}^1}_d$.

Any density $p_\theta(z_1, \dots, z_K)$ can be decomposed via the chain rule of probability as

$$p_\theta(z_1, \dots, z_K) = \prod_k p_{\theta_k}(z_k | z_1, \dots, z_{k-1})$$

where each conditional $p_{\theta_k}(z_k | z_1, \dots, z_{k-1})$ is a density on $\mathcal{M}_k$. As suggested in Rezende et al. (2020), each conditional density can be implemented via a flow $\phi_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ generated by a vector field $\mathbf{f}_k$, whose parameters θ_k are a function of $(z_1, \dots, z_{k-1})$. Such a flow $\phi = \phi_1 \circ \dots \circ \phi_k$ is called *autoregressive* (Papamakarios et al., 2017) and conveniently has a lower triangular Jacobian, which determinant can be computed efficiently as the product of the diagonal term.

G Experimental details

Below we fully describe the experimental settings used to generate results introduced in Section 4.

Architecture The architecture of the *vector field* $\mathbf{f}_\theta$ is given by a multilayer perceptron (MLP) with 3 hidden layers and 64 hidden units – as in (Grathwohl et al., 2019) – for *projected* (e.g. stereographic and wrapped cf Section 3) and *naive* (cf Appendix B.1) models. We rely on tanh activation. For our Riemannian continuous normalizing flow (RCNF), the input layer of the MLP is replaced by a *geodesic distance* layer (Ganea et al., 2018; Mathieu et al., 2019) (see Appendix E.2).

Objectives We consider two objectives, a Monte Carlo (MC) estimator of the negative log-likelihood

$$\hat{\mathcal{L}}^{\text{Like}}(\theta) = - \sum_{i=1}^B \log p_\theta(z_i) \text{ with } z_i \sim P_{\mathcal{D}}$$

and a MC estimator of the reverse KL divergence

$$\hat{\mathcal{L}}^{\text{KL}}(\theta) = \sum_{i=1}^B \log p_\theta(h_\theta(\epsilon_i)) - \log p_{\mathcal{D}}(h_\theta(\epsilon_i))$$

with $z_i \sim P_\theta$ being reparametrized as $z_i = h_\theta(\epsilon_i)$ and $\epsilon_i \sim P$.

Optimization All models are trained by the stochastic optimizer Adam (Kingma and Ba, 2015) with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, batch-size of 400 data-points and a learning rate set to $1e^{-3}$.

Training We rely on the Dormand-Prince solver (Dormand and Prince, 1980), an adaptive Runge-Kutta 4(5) solver, with absolute and relative tolerance of $1e - 5$ to compute approximate numerical solutions of the ODE. Each solver step is projected onto the manifold. Models are trained on a cluster of GeForce RTX 2080 Ti GPU cards.

G.1 Hyperbolic geometry and limits of conventional and wrapped methods

In this experiment the target is set to be a wrapped normal on $\mathbb{B}^2$ (Nagano et al., 2019; Mathieu et al., 2019) with density $\mathcal{N}^{\text{W}}(\exp_0(\alpha \partial x), \Sigma) = \exp_{\mu_\#} \mathcal{N}(\alpha \partial x, \Sigma)$ with $\Sigma = \text{diag}(0.3, 1.0)$. The scalar parameter α allows us to locate the target closer or further away from the origin of the disk. Through this experiment we consider three CNFs:

- *Naive*: $P_\theta^N = \phi_\#^{\mathbb{R}^2} \mathcal{N}(0, 1)$
- *Wrapped*: $P_\theta^W = (\exp_0 \circ \phi^{\mathbb{R}^2})_\# \mathcal{N}(0, 1)$
- *Riemannian*: $P_\theta^R = \phi_\#^{\mathbb{B}^2} \mathcal{N}^W(0, 1)$

with $\phi^{\mathbb{R}^2}$ a conventional CNF on $\mathbb{R}^2$, $\phi^{\mathbb{B}^2}$ our RCNF introduced in Section 2, $\mathcal{N}(0, 1)$ the standard Gaussian and $\mathcal{N}^W(0, 1)$ the standard wrapped normal. For the RCNF we scale the vector field as

$$f_\theta(z) = |G(z)|^{-1/2} \text{neural_net}(z) = \left(\frac{1 - \|z\|^2}{2} \right)^2 \text{neural_net}(z).$$

These three models are trained for 1500 iterations, by minimizing the negative log-likelihood (see Figure 4). The reported results are averaged over 12 runs. When training, the divergence is approximated by the (Hutchinson) stochastic estimator from Equation 5.

G.2 Spherical geometry

Through the following spherical experiments we consider the two following models

- *Stereographic*: $P_\theta^S = (\rho^{-1} \circ \phi^{\mathbb{R}^2})_\# \mathcal{N}(0, 1)$
- *Riemannian*: $P_\theta^R = \phi_\#^{\mathbb{S}^2} \mathcal{U}(\mathbb{S}^2)$

with ρ^{-1} the inverse of the stereographic projection, $\phi^{\mathbb{R}^2}$ a conventional CNF on $\mathbb{R}^2$, $\phi^{\mathbb{S}^2}$ our RCNF, $\mathcal{N}(0, 1)$ the standard Gaussian and $\mathcal{U}(\mathbb{S}^2)$ the uniform distribution on $\mathbb{S}^2$. For the RCNF we project the output layer of the vector field as

$$f_\theta(z) = \text{proj}_{\mathcal{T}_z \mathbb{S}^2} \text{neural_net}(z) = \frac{\text{neural_net}(z)}{\|\text{neural_net}(z)\|^2}$$

so as to enforce output vectors to be *tangent*. All spherical experiments were performed using the exact divergence estimator.

Limits of the stereographic projection model In this experiment the target is chosen to be a vMF(μ, κ) located at $\mu = -\mu_0$ with $-\{\mu_0\} = (-1, 0, \dots, 0)$. Both models are trained for 3000 iterations by minimizing the negative log-likelihood and the reverse KL divergence. The reported results are averaged over 4 runs.

Density estimation of spherical data Finally we consider four earth location datasets, representing respectively volcano eruptions (NOAA, 2020b), earthquakes (NOAA, 2020a), floods (Brakenridge, 2017) and wild fires (EOSDIS, 2020). The reported results are averaged over 12 runs.

Concerning the CNFs, these models are trained by minimizing the negative log-likelihood for 1000 epochs, except for the volcano eruption dataset where 3000 epochs are required for convergence. We observed that training models with a solver's tolerance of $1e-5$ was computationally intensive so we lowered the training tolerance to $1e-3$ for the volcano eruptions, earthquakes and wild fires datasets, and $1e-4$ for the floods dataset, while keeping it to $1e-5$ during evaluation. We additionally did not use the geodesic layer (presented in Appendix E.2) to lower the computational time. We also observed that annealing the learning rate such that $\alpha(t) = 0.98^{(t/300)} \alpha_0$ with $\alpha_0 = 1e^{-3}$ helped stabilizing training convergence.

Concerning the mixture of von Mises-Fisher distributions, the parameters are learned by minimizing the negative log-likelihood with Riemannian Adam (Becigneul and Ganeva, 2019). The number of epochs is set to 10000 for all datasets but for the volcano eruption one where 30000 epochs are required for the vMF model to converge. The learning rate and number of mixture components are selected by performing a hyperparameter grid search, over the following range: learning rate $\in \{1e-1, 5e-2, 1e-2\}$ and number of components $\in \{50, 100, 150, 200\}$.

H Additional figures

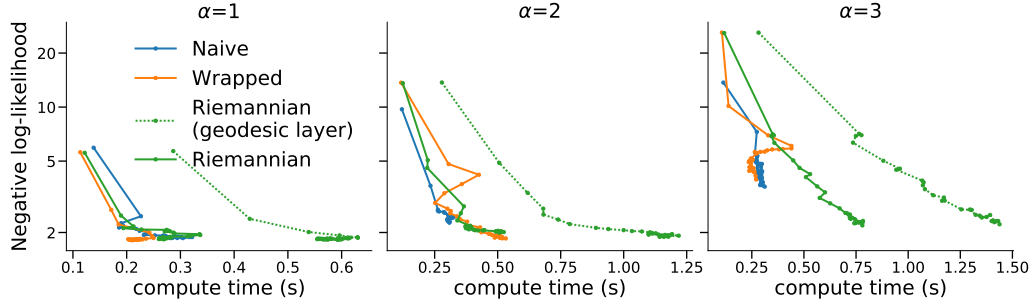


Figure 7: Ablation study of the geodesic layer computational impact for the *Riemannian* model. Negative Log-likelihood of *Riemannian* CNFs trained to fit a $N^W(\exp_0(\alpha \partial x), \Sigma)$ target on $\mathbb{B}^2$.

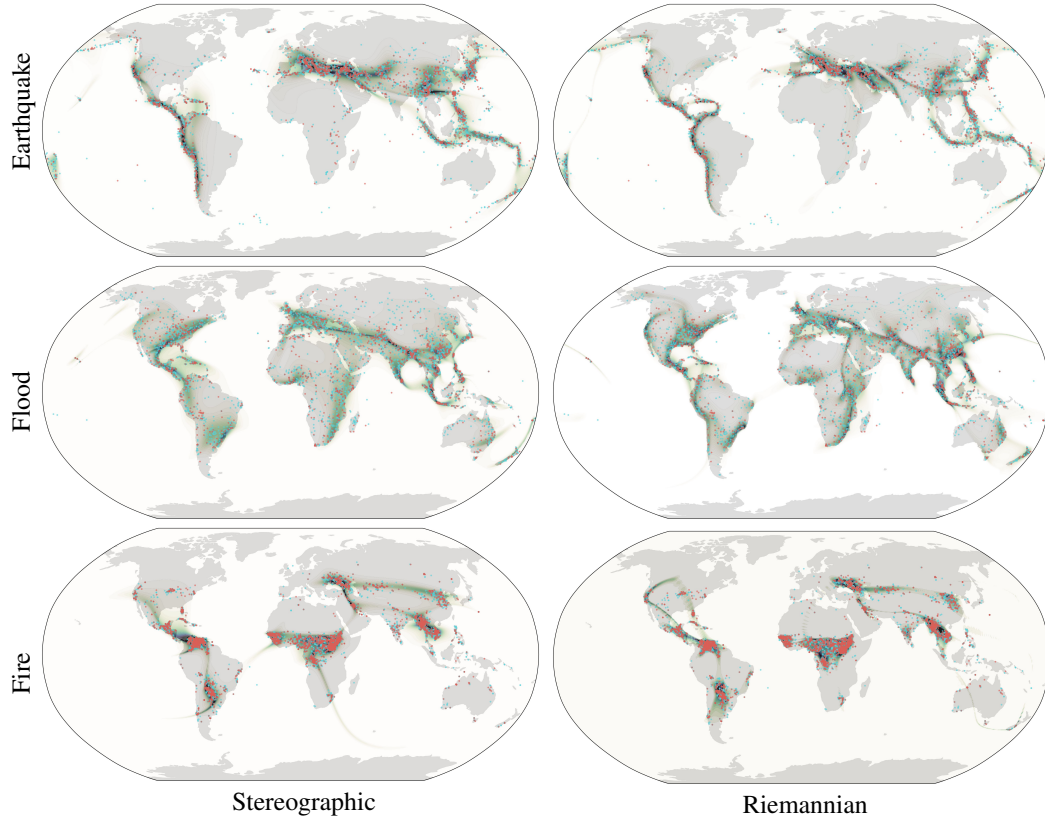


Figure 8: Density estimation for earth sciences data with Robinson projection. Blue and red dots represent training and testing datapoints, respectively. Heatmaps depict the log-likelihood of the trained models.

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	Riemannian Continuous Normalizing Flows
Publication Status	☒ Published
Publication Details	Emile Mathieu and Maximilian Nickel (2020). Advances in Neural Information Processing Systems 33, 2503--2515, Curran Associates, Inc.

Student Confirmation

Student Name:	Emile Mathieu		
Contribution to the Paper	Developed the methodology. Implemented the model and performed all experiments. Wrote the first draft of the manuscript and was helped by M. Nickel for writing the final version.		
Signature		Date	18/06/2021

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: Professor Yee Whye Teh		
Supervisor comments: I do certify that the candidate made a substantial contribution to the publication, and the description above is accurate.		
Signature		Date
		30 June 2021

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

Le savant n'est pas l'homme qui fournit les vraies réponses, c'est celui qui pose les vraies questions.

The scientist is not a person who gives the right answers, he's one who asks the right questions.

— Claude Lévi-Strauss, *Le Cru et le cuit*, 1964

5

Conclusion

This thesis has presented three pieces of work aimed at addressing problems of defining and learning *well-specified* probabilistic models and symmetrically good data representations by encoding prior knowledge with inductive biases. In the following section we summarise these results, briefly explore some of the challenges and drawbacks and eventually suggest directions for future work.

5.1 Contributions and challenges

5.1.1 Disentangling disentanglement

In Chapter 2, we addressed the problem of learning representations with desired properties within VAEs. Taking a Bayesian perspective, we characterised the successful *decomposition* of the latent representation as the fulfilment of two factors: a) the latent encodings of the data having an appropriate level of overlap, and b) the aggregate encoding of the data conforming to a desired structure, represented through the prior. Decomposition allows for a much richer class of properties to be imposed on the learnt representation than independence between latents, such as sparsity, clustering, independent subspaces, or even intricate hierarchical dependency relationships. We showed that the β -VAE model allows control over the level of latent overlap but, that its objective is invariant to rotations of the

latent representation, hence it is unable to achieve disentanglement on its own. We demonstrated how particular choices of prior can assist in producing different decompositions and introduce an alternative training objective that allows the control of both decomposition factors in a principled manner.

This objective contains an additional regularisation term between the aggregate posterior and the prior distributions. In practice, we found that this regulariser needs to be tailored for each problem as each prior brings different challenges with regards to efficiently and accurately estimating statistical divergences. This limitation motivated follow-up work (Miao et al., 2022), where the need for such regularisation is bypassed. The proposed method relies on an intermediary set of latent variables defined as the pushforward of Gaussian latent variables along a deterministic structured map. Properties of the data can then be encoded via specific designs of the map. We introduce a few structured maps that induce multiply-connected, multi-modal, sparse, and hierarchical representations. Albeit effective, this approach requires to handcraft a map for each property of interest. Future work is therefore needed to develop a more generic framework to encode properties into the map design.

5.1.2 Poincaré variational auto-encoders

In Chapter 3, we considered the problem of learning deep generative models for data with underlying hierarchical structure. Many real datasets are hierarchically structured, such as gene expression patterns encoded within our RNA or features of living organisms. Focusing on VAEs, we noticed that they traditionally map data in a Euclidean latent space which cannot efficiently embed tree-like structures. In contrast, hyperbolic spaces with negative curvature can. We therefore proposed to endow VAEs with the Poincaré ball model of hyperbolic geometry as latent space. To do so, we derived the necessary methods to work with two main Gaussian generalisations on that space, and proposed geometry-aware architectures for the encoder and decoder. We empirically showed better generalisation to unseen data

than the Euclidean counterpart, and could qualitatively and quantitatively better recover hierarchical structures.

One such Gaussian generalisation satisfies the maximum entropy property (Said et al., 2014; Chevallier, 2017; Pennec, 2006)—called the Riemannian Gaussian—while the other is defined as the pushforward of a Gaussian distribution along the exponential map—often referred to as the exp-wrapped Gaussian. Even though the Riemannian Gaussian may appear more principled, it is in practice harder to work with, and did not empirically bring any significant improvement compared to the exp-wrapped distribution which additionally allows for full covariance matrices (Nagano et al., 2019). Both are nonetheless strongly restricted in their number of parameters and therefore in expressiveness. Richer variational families lead to tighter ELBO gaps, and in particular normalising flows have been shown to improve generalisation performance (Rezende and Mohamed, 2015; Kingma et al., 2016). The development of normalising flows defined on manifolds is consequently of importance, motivating the method introduced in Chapter 4.

As shown in our ablation studies, the encoder and decoder architectures are also key to achieving good generalisation. We tried the fully connected layer proposed in prior work (Ganea et al., 2018) for the decoder network but did not obtain successful results. This is likely due to decision boundaries being parametrised as Euclidean hyperplanes projected onto the manifold. We proposed instead an architecture based on the parametrisation of decision boundaries as hyperbolic hyperplanes, i.e. geodesics orthogonal to the layer’s directions and containing the offset. Other work has been investigating and developing generalisations of multilayer perceptrons (MLPs) for hyperbolic space (Shimizu et al., 2021). A principled approach for extending MLPs to general Riemannian manifolds is still lacking, whilst gauge equivariance has resulted in a promising theory for convolutional networks (Weiler et al., 2021).

5.1.3 Riemannian continuous normalising flows

In Chapter 4, we tackled the problem of parametrising and learning flexible probability distributions for manifold valued-data. Data is often naturally described on Riemannian manifolds such as spheres, tori, and hyperbolic spaces, yet most generative models implicitly assume a flat geometry, making them either misspecified or ill-suited in these situations. We built on neural ODE methods and implicitly parametrised normalising flows as the solution to ordinary differential equations. We proposed a parametrisation of the vector field based on a choice of local coordinates aka coordinates vector fields, and showed that under mild assumption on the vector field, the generated flow is a diffeomorphism. We derived the instantaneous change of variables from this implicit flow and showed how to estimate it. We empirically demonstrated, on synthetic and real spherical data, that this model converges faster and fits the data better than previous methods.

We focused on extending continuous normalising flows (CNFs) since ODEs are straightforwardly defined on manifolds, and evaluating the density yields the same computational cost as conventional CNFs. In contrast, generalising discrete flows, such as residual flows, is trickier as the linearity of the addition is key to obtaining a tractable Jacobian determinant. Moreover, CNFs usually require dramatically less parameters than discrete flows for similar expressiveness. However, CNFs are known to be computationally intensive to train because an ODE needs to be solved with a numerical integrator at each iteration. Several works proposed to regularise the vector field to avoid solving potentially stiff ODEs in order to speed up numerical integration (Grathwohl et al., 2019; Kelly et al., 2020; Finlay et al., 2020). A theoretical contribution of Chapter 4 (in Appendix D.1) is the generalisation of the dynamical formulation of optimal transport (Benamou and Brenier, 2000) to the Riemannian setting. This formulation links the Wasserstein distance to the transport map defined by the vector field with smallest l_2 -norm. Regularising the vector field would hence yield a flow that is closer to the optimal transport map. Even though we empirically observed an increase in stability and training speed, we

also noticed that the fitting capacities were reduced. This may be due to the choice of ODE integrator or the flow parametrisation. The Riemannian convex potential maps introduced in [Cohen et al. \(2021\)](#) are an attractive alternative as they take the form of such optimal transport maps. Additionally, [Falorsi \(2021\)](#) proposed other parametrisations for vector flows which are simpler to work with for a broader class of manifolds. It is still unclear which normalising flow is better suited for modelling manifold-valued random variables, as further empirical evaluation is required to assess the expressiveness vs computational time trade-off of each method.

5.2 Broader context

In this thesis we have discussed and investigated different manners of encoding prior knowledge as inductive biases in modern machine learning models. Prior knowledge can take different forms, as well as the mathematical frameworks that are used to encode them. Domain knowledge is strongly leveraged during the entire data preprocessing journey, from data collection and feature selection to data augmentation along standard preprocessing techniques. We have mainly been focusing in this thesis on the formalism of Riemannian manifolds but also on concepts such as rotational invariance. Yet, other relevant mathematical concepts include symmetries, conservation laws, but also invariance under a broader class of transformations like (Lie) groups and gauges. These notions are strongly interconnected as symmetries are transformations leaving invariant a physical system or a manifold in geometry, but also since (differentiable) symmetries are mirrored by conservation laws ([Noether, 1918](#)). We also had a focus on probabilistic modelling, where prior knowledge is traditionally encoded via distributions, which can have different supports (e.g. integers, positive reals, set of covariance matrices etc) and different properties (e.g. multi-modality, sparsity, high entropy etc), but also through statistical (in)dependence with the formalism of graphical models. Similarly, neural networks (NNs)'s architecture and layers have been designed and constrained to encode inductive biases. Traditional instances include recurrent NNs ([Rumelhart et al., 1986](#); [Jordan, 1986](#)) and convolutional NNs ([Lecun et al., 1998](#)),

but broader classes of equivariance have more recently been developed, such as group or gauge equivariance (Cohen and Welling, 2016; Cohen et al., 2019; ?; Weiler et al., 2021; Finzi et al., 2021). Permutation invariance has also been at the center of graph neural networks (Zhou et al., 2020; de Haan et al., 2020) and deep sets (Zaheer et al., 2017). Other instances take the form of standard tools for modelling physical processes such as differential equations (Raissi et al., 2019; Rackauckas et al., 2020) and Hamiltonian systems (Greydanus et al., 2019) while learning their functional form with neural networks. As the concepts of symmetry and invariance play a very central role in the physical sciences, these methods have triggered many applications in this realm which in turn has fuelled the development of novel and principled approaches.

Whilst we have so far assumed that one has enough prior knowledge to encode into a model, there are many settings where one can realistically only suspect the data to have some symmetries without explicitly knowing which. An increasing body of work has been investigating this research avenue. Within a machine learning paradigm, a typical approach is to parametrise a space of inductive biases and learn the optimal inductive bias with respect to some metric of interest. This strategy has for instance been successfully employed to automatically discover invariances in the data by parametrising a class of transformations via Lie groups (Cohen and Welling, 2014; Benton et al., 2020) or by other means for learning data augmentation policies (Cubuk et al., 2019; Ratner et al., 2017). In an unsupervised setting, Skopek et al. (2020); Gu et al. (2019) inferred the latent geometry of data by parametrising an embedding space as a product of constant curvature spaces where curvatures and multiplicities are free parameters. Further work towards parametrising more expressive classes of symmetry for broad types of data is an important direction to continue expanding the practical capabilities of deep probabilistic models and more generally modern machine learning models.

Bibliography

- Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu (2016). Pixel Recurrent Neural Networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1747–1756. PMLR.
- Achille, A. and Soatto, S. (2019). Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50).
- Ahmad, S. and Scheinkman, L. (2019). How can we be so dense? The benefits of using highly sparse representations. *arXiv preprint arXiv:1903.11257*.
- Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. (2017). Deep variational information bottleneck. In *International Conference on Learning Representations*.
- Barp, A., Kennedy, A., and Girolami, M. (2019). Hamiltonian Monte Carlo on Symmetric and Homogeneous Spaces via Symplectic Reduction. *arXiv:1903.02699 [stat]*.
- Behrmann, J., Grathwohl, W., Chen, R. T. Q., Duvenaud, D., and Jacobsen, J.-H. (2019). Invertible residual networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 573–582. PMLR.
- Belkin, M. and Niyogi, P. (2003). Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396.
- Beltrami, E. (1868). *Teoria Fondamentale Degli Spazii Di Curvatura Costante: Memoria*. F. Zanetti.
- Benamou, J.-D. and Brenier, Y. (2000). A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393.
- Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1):1–127.
- Bengio, Y. and Bengio, S. (1999). Modeling High-Dimensional Discrete Data with Multi-Layer Neural Networks. *Proceedings of the 12th International Conference on Neural Information Processing Systems*, pages 400–406.
- Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.
- Benton, G., Finzi, M., Izmailov, P., and Wilson, A. G. (2020). Learning Invariances in Neural Networks. *arXiv:2010.11882 [cs, stat]*.

- Bloem-Reddy, B., Foster, A., Mathieu, E., and Teh, Y. W. (2018). Sampling and inference for beta neutral-to-the-left models of sparse networks. In *Conference on Uncertainty in Artificial Intelligence*.
- Bloem-Reddy, B., Mathieu, E., Foster, A., Rainforth, T., Teh, Y. W., Ge, H., Lomelí, M., and Ghahramani, Z. (2017). Sampling and inference for discrete random probability measures in probabilistic programs. page 10.
- Bloem-Reddy, B. and Orbanz, P. (2017). Preferential Attachment and Vertex Arrival Times. *arXiv:1710.02159 [physics, stat]*.
- Bouchacourt, D., Tomioka, R., and Nowozin, S. (2018). Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *AAAI Conference on Artificial Intelligence*.
- Bourgain, J. (1986). The metrical interpretation of superreflexivity in banach spaces. *Israel Journal of Mathematics*, 56(2):222–230.
- Chen, R. T. Q., Behrmann, J., Duvenaud, D. K., and Jacobsen, J.-H. (2019). Residual flows for invertible generative modeling. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Chen, R. T. Q., Li, X., Grosse, R., and Duvenaud, D. (2018a). Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*.
- Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018b). Neural ordinary differential equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180.
- Chevallier, E. (2017). A Family of Anisotropic Distributions on the Hyperbolic Plane. In Nielsen, F. and Barbaresco, F., editors, *Geometric Science of Information*, volume 10589, pages 717–724. Springer International Publishing, Cham.
- Chevallier, E., Barbaresco, F., and Angulo, J. (2015). Probability Density Estimation on the Hyperbolic Space Applied to Radar Processing. In Nielsen, F. and Barbaresco, F., editors, *Geometric Science of Information*, volume 9389, pages 753–761. Springer International Publishing, Cham.
- Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703.
- Cohen, S., Amos, B., and Lipman, Y. (2021). Riemannian Convex Potential Maps. *arXiv:2106.10272 [cs, stat]*.

- Cohen, T. and Welling, M. (2014). Learning the irreducible representations of commutative lie groups. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1755–1763, Beijing, China. PMLR.
- Cohen, T. and Welling, M. (2016). Group equivariant convolutional networks. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA. PMLR.
- Cohen, T. S., Geiger, M., and Weiler, M. (2019). A general theory of equivariant CNNs on homogeneous spaces. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30.
- Collins, A. and Quillian, M. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8:240–248.
- Cubuk, E. D., Zoph, B., Mané, D., Vasudevan, V., and Le, Q. V. (2019). AutoAugment: Learning augmentation strategies from data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 113–123. Computer Vision Foundation / IEEE.
- Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection*. Murray, London.
- Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. (2018). Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*.
- De Cao, N., Aziz, W., and Titov, I. (2020). Block neural autoregressive flow. In Adams, R. P. and Gogate, V., editors, *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 1263–1273. PMLR.
- de Haan, P., Cohen, T., and Welling, M. (2020). Natural Graph Networks. *arXiv:2007.08349 [cs, stat]*.
- Debbasch, F. and Chevalier, C. (2008). Diffusion Processes on Manifolds. In *Institute of Mathematical Statistics Collections*, pages 85–97. Institute of Mathematical Statistics, Beachwood, Ohio, USA.
- Dinh, L., Krueger, D., and Bengio, Y. (2015). NICE: Non-linear independent components estimation. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.
- Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using Real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

- Donoho, D. L. and Grimes, C. (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596.
- Dormand, R. J. and Prince, J. P. (1980). A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, pages 19–26.
- Du, Y. and Mordatch, I. (2019). Implicit generation and modeling with energy based models. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification (2Nd Edition)*. Wiley-Interscience, New York, NY, USA.
- Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019). Neural spline flows. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Esmaili, B., Wu, H., Jain, S., Siddharth, N., Paige, B., and van de Meent, J.-W. (2019). Hierarchical disentangled representations. *Artificial Intelligence and Statistics*.
- Falorsi, L. (2021). Continuous normalizing flows on manifolds. *arXiv:2104.14959 [cs, stat]*.
- Falorsi, L., de Haan, P., Davidson, T. R., De Cao, N., Weiler, M., Forré, P., and Cohen, T. S. (2018). Explorations in Homeomorphic Variational Auto-Encoding. *arXiv:1807.04689 [cs, stat]*.
- Fefferman, C., Mitter, S., and Narayanan, H. (2016). Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049.
- Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230.
- Finlay, C., Jacobsen, J.-H., Nurbekyan, L., and Oberman, A. M. (2020). How to train your neural ODE: The world of jacobian and kinetic regularization. In *ICML*, pages 3154–3164.
- Finzi, M., Welling, M., and Wilson, A. G. (2021). A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 3318–3328. PMLR.
- Fletcher, P., Lu, C., Pizer, S., and Joshi, S. (2004). Principal Geodesic Analysis for the Study of Nonlinear Statistics of Shape. *IEEE Transactions on Medical Imaging*, 23(8):995–1005.
- Fletcher, P. T. (2011). Geodesic Regression on Riemannian Manifolds. page 12.

- Ganea, O., Becigneul, G., and Hofmann, T. (2018). Hyperbolic neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D., and Eslami, S. M. A. (2018a). Conditional neural processes. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1704–1713. PMLR.
- Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S. M. A., and Teh, Y. W. (2018b). Neural Processes.
- Ge, H., Xu, K., and Ghahramani, Z. (2018). Turing: A language for flexible probabilistic inference. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, pages 1682–1690.
- Germain, M., Gregor, K., Murray, I., and Larochelle, H. (2015). MADE: Masked autoencoder for distribution estimation. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 881–889, Lille, France. PMLR.
- Ghahramani, Z., Jordan, M. I., and Adams, R. P. (2010). Tree-structured stick breaking for hierarchical data. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 19–27. Curran Associates, Inc.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- Goodman, N. D., Mansinghka, V. K., Roy, D., Bonawitz, K., and Tenenbaum, J. B. (2008). Church: A language for generative models.
- Gopalakrishnan, S., Marzi, Z., Madhow, U., and Pedarsani, R. (2018). Combating adversarial attacks using sparse representations.
- Gordon, A. D., Henzinger, T. A., Nori, A. V., and Rajamani, S. K. (2014). Probabilistic programming. In *Proceedings of the on Future of Software Engineering*. ACM.
- Grathwohl, W., Chen, R. T. Q., Bettencourt, J., and Duvenaud, D. (2019). Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*.
- Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

- Griffiths, T. L., Jordan, M. I., Tenenbaum, J. B., and Blei, D. M. (2004). Hierarchical topic models and the nested chinese restaurant process. In Thrun, S., Saul, L. K., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*, pages 17–24. MIT Press.
- Gu, A., Sala, F., Gunel, B., and Ré, C. (2019). Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*.
- Hamelryck, T., Kent, J. T., and Krogh, A. (2006). Sampling Realistic Protein Conformations Using Local Structural Bias. *PLoS Computational Biology*, 2(9).
- Heller, K. A. and Ghahramani, Z. (2005). Bayesian hierarchical clustering. In *International Conference on Machine Learning, ICML '05*, pages 297–304, New York, NY, USA. ACM.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). β -VAE: LEARNING BASIC VISUAL CONCEPTS WITH A CONSTRAINED VARIATIONAL FRAMEWORK. page 22.
- Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.
- Hinton, G. E., Osindero, S., and Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554.
- Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2019). Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.
- Ho, J., Chen, X., Srinivas, A., Duan, Y., and Abbeel, P. (2019). Flow++: Improving flow-based generative models with variational dequantization and architecture design. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2722–2730. PMLR.
- Ho, J., Jain, A., and Abbeel, P. (2020a). Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.
- Ho, J., Lohn, E., and Abbeel, P. (2020b). Compression with Flows via Local Bits-Back Coding. *arXiv:1905.08500 [cs, math, stat]*.
- Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. (2018). Neural Autoregressive Flows. page 10.
- Hutchinson (1990). A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450.
- Hyvärinen, A. and Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural networks*, 13(4-5):411–430.

- Ishwaran, H. and James, L. F. (2003). Generalized weighted Chinese restaurant processes for species sampling mixture models. *STATISTICA SINICA*, 13:2003.
- Jaini, P., Selby, K. A., and Yu, Y. (2019). Sum-of-squares polynomial flow. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3009–3018. PMLR.
- Jayasumana, S., Hartley, R., Salzmann, M., Li, H., and Harandi, M. (2015). Kernel Methods on Riemannian Manifolds with Gaussian RBF Kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(12):2464–2477.
- Johnson, M., Duvenaud, D. K., Wiltchko, A., Adams, R. P., and Datta, S. R. (2016). Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, pages 2946–2954.
- Jordan, M. I. (1986). Serial order: A parallel, distributed processing approach. Technical Report 8604, Institute for Cognitive Science, University of California, San Diego.
- Kalchbrenner, N., Oord, A., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., and Kavukcuoglu, K. (2017). Video Pixel Networks. In *International Conference on Machine Learning*, pages 1771–1779. PMLR.
- Karl Pearson (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.
- Karpatne, A., Ebert-Uphoff, I., Ravela, S., Babaie, H., and Kumar, V. (2019). Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 31:1544–1554.
- Keil, F. (1979). *Semantic and Conceptual Development: An Ontological Perspective*. Cognitive Science Series. Harvard University Press.
- Kelly, J., Bettencourt, J., Johnson, M. J., and Duvenaud, D. (2020). Learning differential equations that are easy to solve. In *Neural Information Processing Systems*.
- Kim, H. and Mnih, A. (2018). Disentangling by factorising. In *International Conference on Machine Learning*.
- Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. (2014). Semi-supervised learning with deep generative models. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

- Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, pages 4743–4751, Red Hook, NY, USA. Curran Associates Inc.
- Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Klimovskaia, A., Lopez-Paz, D., Bottou, L., and Nickel, M. (2019). Poincaré Maps for Analyzing Complex Hierarchies in Single-Cell Data. Preprint, Bioinformatics.
- Kruskal, J. and Wish, M. (1978). *Multidimensional Scaling*. Sage Publications.
- Kumar, A., Eslami, S. M. A., Rezende, D. J., Garnelo, M., Viola, F., Lockhart, E., and Shanahan, M. (2019). Consistent Generative Query Networks. *arXiv:1807.02033 [cs, stat]*.
- Larochelle, H. and Iain Murray (2011). The Neural Autoregressive Distribution Estimator. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pages 29–37. PMLR.
- Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Lee, J. M. (2003). *Introduction to Smooth Manifolds*. Number 218 in Graduate Texts in Mathematics. Springer, New York.
- Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.
- Liu, C. and Zhu, J. (2018). Riemannian Stein variational gradient descent for Bayesian inference. In *The 32nd AAAI Conference on Artificial Intelligence*, pages 3627–3634, New Orleans, Louisiana USA. AAAI press.
- Louppe, G., Hermans, J., and Cranmer, K. (2019). Adversarial variational optimization of non-differentiable simulators. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1438–1447. PMLR.
- Lui, Y. M. (2012). Advances in matrix manifolds for computer vision. *Image and Vision Computing*, 30(6):380–388.
- Mallasto, A. and Feragen, A. (2018). Wrapped Gaussian Process Regression on Riemannian Manifolds. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5580–5588, Salt Lake City, UT. IEEE.
- Mathieu, E., Foster, A., and Teh, Y. W. (2021). On Contrastive Representations of Stochastic Processes. *arXiv:2106.10052 [cs, stat]*.

- Mathieu, M. F., Zhao, J. J., Zhao, J., Ramesh, A., Sprechmann, P., and LeCun, Y. (2016). Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048.
- Miao, N., Mathieu, E., N, S., Teh, Y. W., and Rainforth, T. (2022). On incorporating inductive biases into VAEs. In *Submitted to the Tenth International Conference on Learning Representations*.
- Milnor and Weaver (1997). Topology from the differentiable viewpoint. Princeton University Press.
- Monge, G. (1781). *Mémoire Sur La Théorie Des Déblais et Des Remblais*. De l’Imprimerie Royale.
- Nagano, Y., Yamaguchi, S., Fujita, Y., and Koyama, M. (2019). A wrapped normal distribution on hyperbolic space for gradient-based learning. In *ICML*.
- Neal, R. M. (1992). Connectionist learning of belief networks. *Artificial Intelligence*, 56(1):71–113.
- Newman, M. (2005). Power laws, pareto distributions and zipf’s law. *Contemporary Physics*, 46(5):323–351.
- Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pages 6341–6350.
- Noé, F., Olsson, S., Köhler, J., and Wu, H. (2019). Boltzmann Generators – Sampling Equilibrium States of Many-Body Systems with Deep Learning. *arXiv:1812.01729 [cond-mat, physics:physics, stat]*.
- Noether, E. (1918). Invariante Variationsprobleme. *Nachr. d. König. Gesellsch. d. Wiss. zu Göttingen, Math-phys. Klasse, Seite 235-157*.
- Olshausen, B. and Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609.
- Oyanedel, R., Gelcich, S., Mathieu, E., and E.J., M.-G. (2021). A dynamic simulation model to support reduction in illegal trade in legal wildlife markets.
- Paganini, M., de Oliveira, L., and Nachman, B. (2018). Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multi-Layer Calorimeters. *Physical Review Letters*, 120(4):042003.
- Papamakarios, G., Pavlakou, T., and Murray, I. (2017). Masked autoregressive flow for density estimation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.

- Pennec, X. (2006). Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154.
- Pitman, J. and Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900.
- Pol, A. A., Berger, V., Germain, C., Cerminara, G., and Pierini, M. (2019). Anomaly Detection with Conditional Variational Autoencoders. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1651–1657, Boca Raton, FL, USA. IEEE.
- Pourzanjani, A., Jiang, R. M., Mitchell, B., Atzberger, P. J., and Petzold, L. (2020). General bayesian inference over the stiefel manifold via the givens transform. *Bayesian Analysis*.
- Prenger, R., Valle, R., and Catanzaro, B. (2019). Waveglow: A Flow-based Generative Network for Speech Synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 3617–3621.
- Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A. (2020). Universal Differential Equations for Scientific Machine Learning. *arXiv:2001.04385 [cs, math, q-bio, stat]*.
- Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.
- Ratner, A. J., Ehrenberg, H. R., Hussain, Z., Dunnmon, J., and Ré, C. (2017). Learning to compose domain-specific transformations for data augmentation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, pages 3239–3249, Red Hook, NY, USA. Curran Associates Inc.
- Reed, S., Sohn, K., Zhang, Y., and Lee, H. (2014). Learning to disentangle factors of variation with manifold interaction. In *International Conference on Machine Learning*, pages 1431–1439.
- Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.
- Rezende, D. J., Shakir Mohamed, and Daan Wierstra (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1278–1286. PMLR.
- Risken, H. (1989). *The Fokker-Planck Equation: Methods of Solutions and Applications, 2nd Ed.* Springer Verlag, Berlin, Heidelberg.

- Rodríguez, A. C., Kacprzak, T., Lucchi, A., Amara, A., Sgier, R., Fluri, J., Hofmann, T., and Réfrégier, A. (2018). Fast cosmic web simulations with generative adversarial networks. *Computational Astrophysics and Cosmology*, 5(1):4.
- Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837.
- Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.
- Roy, D. M., Kemp, C., Mansinghka, V. K., and Tenenbaum, J. B. (2006). Learning annotated hierarchies from relational data. In *NIPS*, pages 1185–1192. MIT Press.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, pages 318–362. MIT Press, Cambridge, MA.
- Said, S., Bombrun, L., and Berthoumieu, Y. (2014). New Riemannian Priors on the Univariate Normal Model. *Entropy*, 16(7):4015–4031.
- Said, S., Hajri, H., Bombrun, L., and Vemuri, B. C. (2018). Gaussian distributions on riemannian symmetric spaces: Statistical learning with structured covariance matrices. *IEEE Trans. Inf. Theor.*, 64(2):752–772.
- Salakhutdinov, R. R., Tenenbaum, J. B., and Torralba, A. (2011). One-shot learning with a hierarchical nonparametric bayesian model. In *ICML Unsupervised and Transfer Learning*.
- Samanta, B., De, A., Jana, G., Chattaraj, P. K., Ganguly, N., and Rodriguez, M. G. (2019). NeVAE: A Deep Generative Model for Molecular Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:1110–1117.
- Sarkar, R. (2012). Low distortion delaunay embedding of trees in hyperbolic plane. In van Kreveld, M. and Speckmann, B., editors, *Graph Drawing*, pages 355–366, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Schmidhuber, J. (1992). Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879.
- Sharir, O., Levine, Y., Wies, N., Carleo, G., and Shashua, A. (2020). Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Physical Review Letters*, 124(2):020503.
- Shimizu, R., Mukuta, Y., and Harada, T. (2021). Hyperbolic neural Networks++. In *International Conference on Learning Representations*.
- Siddharth, N., Paige, T. B., Van de Meent, J.-W., Desmaison, A., Goodman, N., Kohli, P., Wood, F., and Torr, P. (2017). Learning disentangled representations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, pages 5925–5935.

- Skopek, O., Ganea, O.-E., and Bécigneul, G. (2020). Mixed-curvature variational autoencoders. In *International Conference on Learning Representations*.
- Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907.
- Song, Y. and Kingma, D. P. (2021). How to Train Your Energy-Based Models. *arXiv:2101.03288 [cs, stat]*.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.
- Tabak, E. G. and Turner, C. V. (2013). A Family of Nonparametric Density Estimation Algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164.
- Tabak, E. G. and Vanden-Eijnden, E. (2010). Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233.
- Teh, Y. W. (2011). Dirichlet process. In *Encyclopedia of Machine Learning*, pages 280–287. Springer.
- Teh, Y. W., Daume III, H., and Roy, D. M. (2008). Bayesian agglomerative clustering with coalescents. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20*, pages 1473–1480. Curran Associates, Inc.
- Teh, Y. W., Welling, M., Osindero, S., and Hinton, G. E. (2003). Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4:1235–1260.
- Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.
- Tifrea, A., Becigneul, G., and Ganea, O.-E. (2019). Poincare glove: Hyperbolic word embeddings. In *International Conference on Learning Representations*.
- Tomczak, J. M. and Welling, M. (2017). Improving Variational Auto-Encoders using Householder Flow. *arXiv:1611.09630 [cs, stat]*.
- Townsend, J., Bird, T., and Barber, D. (2019). Practical lossless compression with latent variables using bits back coding. In *International Conference on Learning Representations*.
- Van Den Berg, R., Hasenclever, L., Tomczak, J. M., and Welling, M. (2018). Sylvester normalizing flows for variational inference. In Globerson, A., Globerson, A., and Silva, R., editors, *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018, pages 393–402. Association For Uncertainty in Artificial Intelligence (AUAI).

- van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. *arXiv:1609.03499 [cs]*.
- van den Oord, A., Kalchbrenner, N., Espeholt, L., kavukcuoglu, k., Vinyals, O., and Graves, A. (2016). Conditional image generation with PixelCNN decoders. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.
- Weiler, M., Forré, P., Verlinde, E., and Welling, M. (2021). Coordinate Independent Convolutional Networks – Isometry and Gauge Equivariant Convolutions on Riemannian Manifolds. *arXiv:2106.06020 [cs, stat]*.
- Wood, F., van de Meent, J. W., and Mansinghka, V. (2014). A new approach to probabilistic programming inference. In *AISTATS*, pages 2–46.
- Wu, D., Wang, L., and Zhang, P. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks. *Physical Review Letters*, 122(8):080602.
- Yang, H. H. and Amari, S.-i. (1997). Adaptive online learning algorithms for blind separation: Maximum entropy and minimum mutual information. *Neural computation*, 9(7):1457–1482.
- Yip, K. and Sussman, G. J. (1997). Sparse representations for fast, one-shot learning. In Kuipers, B. and Webber, B. L., editors, *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island, USA*, pages 521–527. AAAI Press / The MIT Press.
- Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Zhang, M. and Fletcher, T. (2013). Probabilistic principal geodesic analysis. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.
- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.