LaDDer: Latent Data Distribution Modelling with a Generative Prior

Shuyu Lin
https://shuyulin.co.uk/

Computer Science Department,
University of Oxford,
Oxford, UK

Ronald Clark
http://www.ronnieclark.co.uk/

Imperial College London
South Kensington,
London, UK

Abstract

In this paper, we show that the performance of a learnt generative model is closely related to the model's ability to accurately represent the inferred **latent data distribution**, i.e. its topology and structural properties. We propose LaDDer to achieve accurate modelling of the latent data distribution in a variational autoencoder framework and to facilitate better representation learning. The central idea of LaDDer is a meta-embedding concept, which uses multiple VAE models to learn an embedding of the embeddings, forming a ladder of encodings. We use a non-parametric mixture as the hyper prior for the innermost VAE and learn all the parameters in a unified variational framework. From extensive experiments, we show that our LaDDer model is able to accurately estimate complex latent distribution and results in improvement in the representation quality. We also propose a novel latent space interpolation method that utilises the derived data distribution. The code and demos are available at https://github.com/lin-shuyu/ladder-latent-data-distribution-modelling.

1 Introduction

Variational autoencoders (VAEs) [22, 35] are probabilistic latent variable models that aim to learn rich representations from large amounts of data in an unsupervised manner. A trained VAE consists of a generative decoder that generates a data sample from a latent code and a variational encoder that maps a data sample to an approximate posterior distribution over latent variables. Amongst different types of generative models including GANs [12], flow models [8, 9] and autoregressive models [39], VAEs have been favoured for their training stability, strong theoretical grounding and ability to learn well-structured latent representations. In general, the quality of a learned generative model is dependent on two factors: 1) the quality of the inferred data distribution, which defines the topology and structural properties of the **latent space**; 2) the ability to generate good quality samples in the **data space**.

Thanks to all the preferable properties, VAEs have been widely applied in many computer vision applications, including image synthesis [11, 17, 41], human motion modelling [14, 36] and 3D reconstruction [1, 4, 6, 10, 29]. However, VAEs still struggle when modelling complex data such as images. Specifically, compared to other models, the generated images

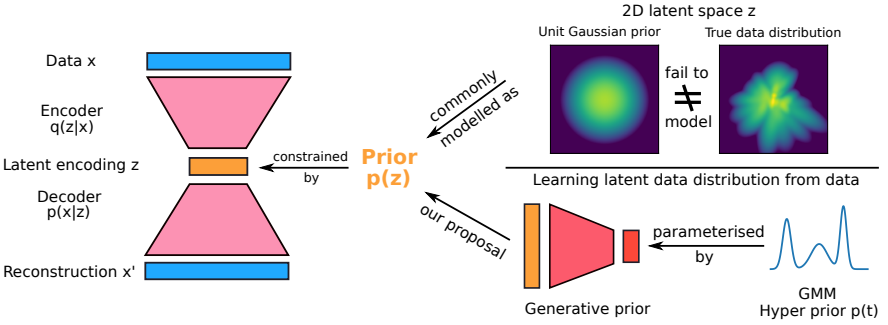

Figure 1: **LaDDer's key concept**. LaDDer adopts a generative prior, which consists of a mixture hyper prior and a series of VAEs each acting on its predecessor's latent encodings.

of VAEs can be somewhat blurry, lack of fine details and have limited diversity. An entire research field is dedicated to improving VAE learning quality. In this paper, we focus on a key component of VAE learning objective that involves restricting the learned latent space representation to a chosen prior probability. As we cannot observe the true latent distribution from which the data were generated, it is impossible to choose a perfect prior distribution a priori. Therefore, many works, which adopt a simple and inflexible prior in the form of a unit Gaussian distribution, often suffer from an over-regularised latent representation, as the model tries to get the encoder to shoehorn the data to fit the simple prior, sacrificing the quality of generated images. This suggests that using a more flexible prior distribution, or treating the prior as a parameterised model, can reduce or eliminate the over-regularisation issue and lead to improvement in the representation quality.

Following these insights, we propose LaDDer, a method that allows us to accurately model the prior distribution in a VAE framework. The central idea of our method is a meta-embedding concept which, in short, derives a latent embedding of a latent embedding. Specifically as shown in Fig. 1, our approach consists of multiple VAE models each acting on its predecessor's latent representation and forming a ladder of encodings. We use a non-parametric mixture as the hyper prior for the innermost VAE. The hyper prior together with the intermediate encoders forms a **generative prior** for the outermost VAE. We learn the parameters of all the VAE networks, along with the non-parametric mixture, in a unified variational inference framework. From extensive experiments, we show that our LaDDer model is able to accurately estimate complex **latent data distribution** and results in improvement in the representation quality. We also propose a **novel latent space interpolation** method and show how to best utilise the derived latent distribution in further tasks.

2 Background

Here we introduce VAE models and explain why the prior distribution is crucial in producing good learning outcomes. Given a dataset of N observations $\mathcal{D}_N = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$, VAEs assume all data samples $\boldsymbol{x}_i$ are generated from a low-dimensional latent space $\mathbf{z}$ under a latent variable model $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, where $p_\theta(\mathbf{x}|\mathbf{z})$ denotes the generative model (decoder) parameterised by θ. VAEs learn the model parameters θ by maximising the marginal log likelihood for all data points in $\mathcal{D}_N$, i.e. $\arg\max_\theta \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}[\log p_\theta(\mathbf{x})]$, where $p_\mathcal{D}(\mathbf{x})$ is the empirical data distribution and $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z}$. However, directly evaluating the marginal log likelihood is often not feasible, as integration over the network $p_\theta(\mathbf{x}|\mathbf{z})$ is not

trivial. To obtain an analytical learning objective, VAEs [22, 35] use variational inference and derive an evidence lower bound (ELBO) $\mathcal{L}(\mathbf{x};\theta,\phi)$ to the marginal log likelihood, i.e. $\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}[\log p_\theta(\mathbf{x})] \geq \mathcal{L}(\mathbf{x};\theta,\phi)$, as shown in Eq (1):

$$\mathcal{L}(\mathbf{x};\theta,\phi) \triangleq \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\big[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}[q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})]\big] \tag{1}$$

$$\triangleq \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\big[\underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{① reconstruction likelihood}} - \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log q_\phi(\mathbf{z}|\mathbf{x})]}_{\text{② posterior entropy}} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{z})]}_{\text{③ cross-entropy wrt prior}}\big]. \tag{2}$$

By breaking down the KL divergence term in Eq (1) using the definition of KL, we obtain an ELBO expression in Eq (2). Here we see that the ELBO contains three terms: ① a reconstruction likelihood that encourages good reconstruction through the auto-encoding process, ② a negative posterior entropy that favours $q_\phi(\mathbf{z}|\mathbf{x})$ with large variances and finally ③ a cross-entropy between the posterior $q_\phi(\mathbf{z}|\mathbf{x})$ and the prior $p(\mathbf{z})$ that regularises the posteriors to comply with the target prior distribution. We can re-arrange ③ into a cross-entropy between the aggregate posterior $q_\phi(\mathbf{z})$ and the prior $p(\mathbf{z})$, as shown below:

$$\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{z})] = \mathbb{E}_{q_\phi(\mathbf{z})}[\log p(\mathbf{z})], \text{ where } q_\phi(\mathbf{z}) = \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}[q_\phi(\mathbf{z}|\mathbf{x})]. \tag{3}$$

This re-arrangement reveals that term ③ in the ELBO loss encourages the prior $p(\mathbf{z})$ and the inferred latent data distribution $q_\phi(\mathbf{z})$ to match each other. If an overly limiting prior is used, then the learnt data distribution will diverge from the true data distribution. Details of the re-arrangement are given in Supplementary Materials (SM) A1.

2.1 Related work

Many approaches have been proposed to improve VAE's modelling ability for rich, high-dimensional data. For example, [23, 32, 34, 37] propose to use more flexible sample posteriors $q_\phi(\mathbf{z}|\boldsymbol{x})$ to increase the encoder's expressive power and improve the model's performance in all three loss terms of Eq (2). [5, 16, 19] focus on deriving alternative learning objectives in order to produce representations with preferred qualities, such as disentanglement.

We take a different approach from these lines of research. In this work, we focus on modelling the prior distribution $p(\mathbf{z})$ accurately, which then results in improved learning performance and facilitates further tasks, such as latent space interpolation. Several works have considered using more flexible distributions than a unit Gaussian as priors, such as a stick-breaking prior [30], a Chinese Restaurant Process prior [13] or a Gaussian mixture (GM) prior [7, 18, 27, 38]. Those methods often have limited performance for complex datasets where high-dimensional latent space is needed to facilitate the generative modelling. A closely related work is [24], where a generative model is used to parameterise the prior. Our method differs from [24], as we realise that the introduction of a generative prior alone does not guarantee accurate modelling of the latent data distribution. Thus, we introduce a GM hyper prior to complete the modelling task. In addition, we optimise all the model parameters in a coherent lower bound objective, whereas [24] has to set up a constrained optimisation to replace the ELBO objective in order to learn the model parameters.

Another line of research in VAEs is devoted to improving the generation quality, leading to the impressive image quality shown in VQ-VAE2 [33]. A major difference between VQ-VAE2 and our work lies in the different approaches taken to model the prior distribution. VQ-VAE2 learns an auto-regressive prior (using a pixelCNN model [26]) in a post-hoc, 2nd stage (after the autoencoder is trained). Our model trains both autoencoder and prior modules

jointly under the same objective. Further, VQ-VAE2 employs a 2D latent representation, where a single feature vector corresponds to a local patch of the generated image. It is extremely difficult to manipulate such a 2D representation to generate globally consistent changes. In contrast, we adopt a global code to represent the whole image. We can easily traverse across the derived data manifold and generate smoothly changing data samples.

3 Our Method

As shown in Fig. 1, we propose a generative prior, which includes an additional VAE model to project the original data VAE's encodings to an even lower dimensional space and a hyper prior for this prior VAE. We parameterise the hyper prior as a Gaussian mixture model (GMM). This design allows us to accurately estimate the latent data distribution, as the generative prior is sufficiently flexible to fit any arbitrarily complex distribution. At the same time, the optimisation of our LaDDer model can be easily integrated into the VAE ELBO objective, which we will demonstrate in this section. In the end, we also demonstrate how to use the latent data distribution derived from our LaDDer learning to facilitate a latent space interpolation task.

3.1 The VAE Unit in Our Generative Prior

We first introduce the VAE unit in our generative prior. Similar to the VAE for data samples introduced in Section 2, the prior VAE is also formulated by a latent variable model $p_\alpha(\mathbf{z},\mathbf{t}) = p_\alpha(\mathbf{z}|\mathbf{t})p(\mathbf{t})$, which governs the generation of latent encodings $\mathbf{z}_i$ through 1) a hyper prior $p(\mathbf{t})$ that resides in an even lower dimensional space and 2) an encoding decoder $p_\alpha(\mathbf{z}|\mathbf{t})$ which is parameterised by a neural network with parameters α. To optimise the prior VAE, we can introduce a variational encoder $q_\beta(\mathbf{t}|\mathbf{z})$ parameterised by β and learn both α and β by maximising an ELBO objective $\mathcal{L}(\mathbf{z};\alpha,\beta)$ similar to Eq (2) for this prior model, i.e. $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\alpha(\mathbf{z})] \geq \mathcal{L}(\mathbf{z};\alpha,\beta)$, where:

$$\mathcal{L}(\mathbf{z};\alpha,\beta) \triangleq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[\mathbb{E}_{q_\beta(\mathbf{t}|\mathbf{z})}[\log p_\alpha(\mathbf{z}|\mathbf{t})] - \mathbb{E}_{q_\beta(\mathbf{t}|\mathbf{z})}[\log q_\beta(\mathbf{t}|\mathbf{z})] + \mathbb{E}_{q_\beta(\mathbf{t}|\mathbf{z})}[\log p(\mathbf{t})]\big]. \quad (4)$$

Notice that $\mathcal{L}(\mathbf{z};\alpha,\beta)$ is a lower bound to the likelihood $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\alpha(\mathbf{z})]$, which is equivalent to the cross-entropy term ③ in Eq (2). This connection allows us to integrate the learning objective of this prior VAE, i.e. $\mathcal{L}(\mathbf{z};\alpha,\beta)$, into the ELBO for the original data VAE, i.e. $\mathcal{L}(\mathbf{x};\theta,\phi)$, and obtain a new lower bound $\mathcal{L}'(\mathbf{x};\theta,\phi,\alpha,\beta)$ to the data ELBO $\mathcal{L}(\mathbf{x};\theta,\phi)$, i.e. $\mathcal{L}(\mathbf{x};\theta,\phi) \geq \mathcal{L}'(\mathbf{x};\theta,\phi,\alpha,\beta)$, where:

$$\mathcal{L}'(\mathbf{x};\theta,\phi,\alpha,\beta) = \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\big[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log q_\phi(\mathbf{z}|\mathbf{x})] + \mathcal{L}(\mathbf{z};\alpha,\beta)\big]. \quad (5)$$

3.2 Variational Gaussian Mixture Model for the Hyper Prior

The prior VAE unit defined in Section 3.1 introduces a hyper prior $p(\mathbf{t})$. From Eq (3), we know that the optimal hyper prior should be matched to the aggregate hyper posterior, i.e. $p(\mathbf{t}) \approx q_{\beta,\phi}(\mathbf{t})$. To facilitate the matching, we parameterise $p(\mathbf{t})$ with a Gaussian mixture model (GMM) of M components (M $\ll$ N): $p(\mathbf{t}) = \sum_{m=1}^{M} w_m \mathcal{N}(\mathbf{t};\mu_m,\Sigma_m)$, where w_m is the weight for each Gaussian mixture ($w_m > 0$ and $\sum_m w_m = 1$) and μ_m and Σ_m are the mean and covariance matrix for the m-th Gaussian mixture. To fit $p(\mathbf{t})$ to $q_{\beta,\phi}(\mathbf{t})$, we resort to variational inference techniques introduced in [3]. Here we give a high-level sketch of the

algorithm and we refer interested readers to [3] for further mathematical details. Firstly, we define the generative model of the GMM by introducing the following distributions to the GMM parameters $(\boldsymbol{w}, \mu, \Sigma)$ and all N prior encoding samples $\boldsymbol{t}_i$ (the encodings of $\boldsymbol{z}_i$): $w_m \sim \mathcal{B}eta(1, \alpha_0)$, $\mu_m \sim \mathcal{N}(0, \mathcal{I})$, $\Sigma_m \sim \mathcal{W}(d_t, \mathcal{I})$, $k_i \sim \text{Cat}(\boldsymbol{w})$, $\boldsymbol{t}_i \sim \mathcal{N}(\mu_{k_i}, \Sigma_{k_i})$, where k_i indicates the choice of mixture components for i-th sample $\boldsymbol{t}_i$. Secondly, we introduce the following variational distributions under the mean-field assumption for all the model variables $W = (\boldsymbol{w}, \mu, \Sigma, k_i)$: $w_m \sim \mathcal{B}eta(\gamma_{m,1}, \gamma_{m,2})$, $\mu_m \sim \mathcal{N}(\boldsymbol{\nu}_m, \mathcal{I})$, $\Sigma_m \sim \mathcal{W}(a_m, \mathcal{B}_m)$, $k_i \sim$ Discrete$(\boldsymbol{r}_i)$, where $\xi = (\gamma_{m,1}, \gamma_{m,2}, \boldsymbol{\nu}_m, a_m, \mathcal{B}_m, \boldsymbol{r}_i)$ for $m = 1, \cdots, M$ and $i = 1, \cdots, N$ denotes variational variables and needs to be optimised. Thirdly, a variational bound on the log likelihood of $p(\mathbf{t})$ is introduced as the learning objective to optimise ξ, as shown below:

$$\log p(\mathbf{t}|\alpha_0, M) \geq \mathcal{L}(\mathbf{t}; \xi) = \mathbb{E}_{q_\xi(W)}[\log p(W, \mathbf{t}|\alpha_0, M)] - \mathbb{E}_{q_\xi(W)}[\log q_\xi(W)], \tag{6}$$

where $p(W, \mathbf{t}|\alpha_0, M)$ denotes the product of all the generative distributions introduced in step 1 and $q_\xi(W)$ denotes the product of all the variational distributions introduced in step 2.

Now substituting $\mathcal{L}(\mathbf{t}; \xi)$ in Eq (6) into Eq (4) to replace $\log p(\mathbf{t})$, we obtain a lower bound $\mathcal{L}'(\mathbf{z}; \alpha, \beta, \xi)$ to the prior ELBO $\mathcal{L}(\mathbf{z}; \alpha, \beta)$, i.e. $\mathcal{L}(\mathbf{z}; \alpha, \beta) \geq \mathcal{L}'(\mathbf{z}; \alpha, \beta, \xi)$, as

$$\mathcal{L}'(\mathbf{z}; \alpha, \beta, \xi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\mathbb{E}_{q_\beta(\mathbf{t}|\mathbf{z})}[\log p_\alpha(\mathbf{z}|\mathbf{t})] - \mathbb{E}_{q_\beta(\mathbf{t}|\mathbf{z})}[\log q_\beta(\mathbf{t}|\mathbf{z})] + \mathbb{E}_{q_\beta(\mathbf{t}|\mathbf{z})}[\mathcal{L}(\mathbf{t}; \xi)]\right]. \tag{7}$$

Now substituting $\mathcal{L}'(\mathbf{z}; \alpha, \beta, \xi)$ into the lower bound $\mathcal{L}'(\mathbf{x}; \theta, \phi, \alpha, \beta)$ in Eq (5) to replace $\mathcal{L}(\mathbf{z}; \alpha, \beta)$, we obtain a final lower bound $\mathcal{L}''(\mathbf{x}; \Theta)$ to the data likelihood $\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}[\log p_\theta(\mathbf{x})]$, i.e. $\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}[\log p_\theta(\mathbf{x})] \geq \mathcal{L}''(\mathbf{x}; \Theta)$, where $\Theta = [\theta, \phi, \alpha, \beta, \xi]$ denotes all the parameters in the data VAE, prior VAE and GMM hyper prior, and

$$\mathcal{L}''(\mathbf{x}; \Theta) = \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log q_\phi(\mathbf{z}|\mathbf{x})] + \mathcal{L}'(\mathbf{z}; \alpha, \beta, \xi)\right]. \tag{8}$$

3.3 Block Coordinate Ascent to Optimise Model Parameters

With the new lower bound in Eq (8), we are ready to introduce our block coordinate ascent algorithm which optimises all the model parameters $\Theta = [\theta, \phi, \alpha, \beta, \xi]$. Notice that the GMM hyper prior's parameter ξ are only involved in the term $\mathcal{L}(\mathbf{t}; \xi)$ from Eq (6). We can update ξ by maximising $\mathcal{L}(\mathbf{t}; \xi)$ alone. With the newly updated ξ, we can then update the prior VAE's parameter α, β, which only present in the prior VAE's ELBO $\mathcal{L}'(\mathbf{z}; \alpha, \beta|\xi)$ from Eq (7). Finally, with ξ, α, β updated, we can now update the data VAE's parameters θ, ϕ by maximising $\mathcal{L}''(\mathbf{x}; \theta, \phi|\alpha, \beta, \xi)$ from Eq (8) with α, β, ξ being fixed. An algorithmic illustration for this optimisation procedure is given in SM B. Notice that we initialise the GMM hyper prior as an uninformative unit Gaussian to begin the optimisation.

3.4 Shortest Likely Path to Traverse the Data Manifold

In Section 3.1-3.3, we introduce a generative prior and an optimisation scheme to facilitate accurate modelling of the latent data distribution. Now we demonstrate how to use the derived latent distribution in a latent space traversal task, where a path along the learnt data manifold needs to be inferred to interpolate between two data samples. Such interpolation has been used commonly in previous works [2, 19, 21, 31] to illustrate the smoothness of the learnt data manifold. Traditionally, the traversal is done by a linear interpolation between the encodings of a pair of query images, which we refer as shortest path (SP) interpolation. This method ignores the nonlinear topology of the data manifold in the latent space and often results in poor interpolated images, which are generated faraway from the data manifold.

To achieve better interpolation results, we propose to formulate the traversal task as an optimisation problem which aims to find an optimal path that is shortest while remaining close to the data manifold. We refer to our interpolation scheme as shortest likelihood path (SLP) interpolation and design an optimisation objective as follows:

$$\mathcal{O}_{\text{SLP}} = \underbrace{\text{L}_{\text{path}}(\boldsymbol{s})}_{\text{path length}} + \underbrace{\text{std}_{\text{step length}}(\boldsymbol{s})}_{\text{equal step length}} - \underbrace{\log p(\mathbf{t}=\boldsymbol{s})}_{\text{path likelihood}}, \tag{9}$$

where $\boldsymbol{s} = [s_1, \cdots, s_J]^T$ denotes a list of J steps along the traversal path and each s_j denotes a single encoding that can be projected back to the data space as an interpolated image. As can be seen, our $\mathcal{O}_{\text{SLP}}$ objective contains three terms: 1) $\text{L}_{\text{path}}(\boldsymbol{s})$ minimises the current path length, 2) $\text{std}_{\text{step length}}(\boldsymbol{s})$ requires steps to be evenly distributed along the path and 3) $\log p(\mathbf{t}=\boldsymbol{s})$ ensures all the interpolated encodings to be generated from the inferred data distribution $p(\mathbf{t})$. To find an optimal path $\boldsymbol{s}$, we minimise $\mathcal{O}_{\text{SLP}}$ wrt $\boldsymbol{s}$ by a standard optimisation scheme, such as AdamOptimiser [20] that is used in our experiments. Notice that our SLP interpolation can be generally applied to any representation learning algorithm, as long as the likelihood of the latent data distribution $p(\mathbf{z})$ can be easily evaluated.

4 Experiments and Results

We carry out extensive experiments on MNIST [25], Fashion MNIST [40] and CelebA [28] datasets to evaluate our LaDDer model. For all datasets, images are treated as real-valued data and we use a Laplace distribution to model the decoder $p_\theta(\mathbf{x}|\mathbf{z})$. The corresponding reconstruction likelihood (term ① in Eq (2)) is derived in SM C, following [27]. We compare our method to 4 other approaches, including the original VAE with a normal prior [22, 35], VAE with a GMM prior [7, 18, 27], VAE with a hierarchical prior [24] where a generative prior with a normal hyper prior is used, and VampPrior [38] where the prior is modelled as an average encoding of a set of inferred pseudo-inputs. More results, details of data preprocessing and model architectures are given in SM D-F.

4.1 Our Method Better Estimates the Latent Data Distribution

We first show that our LaDDer model can achieve a better modelling of the latent data distribution, i.e. more accurate estimation of the aggregate posterior $q_\phi(\mathbf{z})$. Fig. 2 visualises the aggregate posterior $q_\phi(\mathbf{z})$ and the optimised priors from 5 different approaches, where VAE models are trained for 2D latent space on MNIST dataset. The true data distribution shown in Fig. 2a contains complex low density regions, which correspond to natural boundaries between different object classes. An inflexible prior model, such as a unit Gaussian (Fig. 2b), over-represents such low density regions. The hierarchical prior (Fig. 2c) also fails to model $q_\phi(\mathbf{z})$ accurately, because the prior VAE is overly regularised by its inflexible hyper prior. The other three methods (Fig. 2d-f), where the prior $p(\mathbf{z})$ is parameterised with mixture models, obtain better fitting. The difference across these three methods lies in the number of mixture components. Both the GMM prior and the VampPrior have a finite number of mixture components, whereas our LaDDer prior contains an infinite number of mixture components, as our prior is estimated by integrating over all t-values: $p_\alpha(\mathbf{z}) = \int p_\alpha(\mathbf{z}|\mathbf{t})p(\mathbf{t})\,\mathrm{d}\mathbf{t}$. As a result, our method is able to smoothly fit any arbitrarily complex $q_\phi(\mathbf{z})$ with no artificial boundaries as the ones introduced in VampPrior.

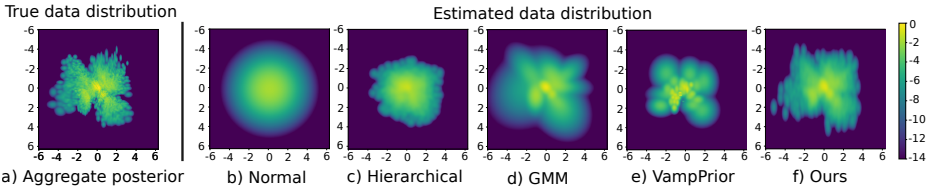

Figure 2: **Visualising the modelling of MNIST data distribution in a 2D latent space**. We first train a VAE model with the unit Gaussian prior. Then we fix the autoencoder model and fit different prior models to the derived encodings. The true data distribution (aggregate posterior $q_\phi(\mathbf{z})$) is shown in (a), where we encode 10k training images and visualise their posteriors. In b-f, we visualise the pdfs of the 5 different prior methods. Our generative prior produces the best fit to the true data distribution.

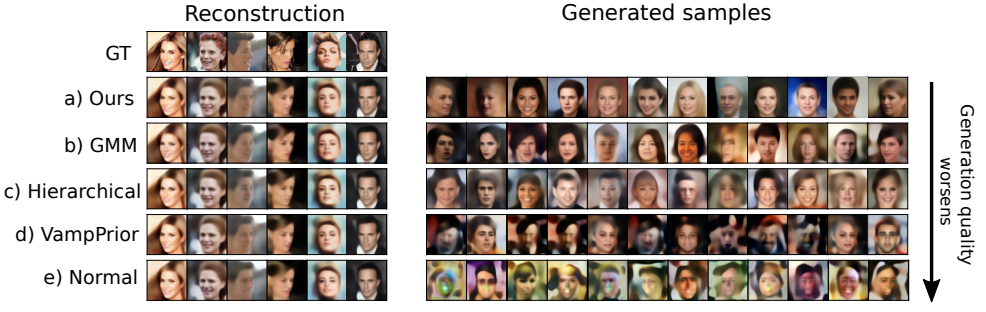

Figure 3: **Comparison of reconstruction and generation quality across different VAE models**. All five models trained with celebA dataset can produce similar quality reconstruction, but the generation quality varies significantly samples. The difference between reconstruction and generation quality indicates the importance of a prior model in modelling the data distribution and generating good quality samples. Our generative prior can model the data distribution $q_\phi(\mathbf{z})$ well, hence generating best quality samples with minimal gap to the quality of reconstructed samples. More examples are in Supplementary Materials F1.

Having a prior that can better fit $q_\phi(\mathbf{z})$ also leads to improved generation quality. Fig. 3 gives examples of both reconstructed and generated samples from the 5 VAE models trained under the CelebA dataset. All 5 methods can produce good reconstruction, but the generation quality varies significantly. This indicates that the bad generation quality commonly reported in VAE models does not merely result from the autoencoder architecture, but is also a result of the prior $p(\mathbf{z})$ failing to represent the topology of the learned latent data distribution. Generating from the prior that does not represent the true data distribution unsurprisingly produce unrealistic samples. Our proposed generative prior addresses this issue and produces generation quality that is almost as good as the model's reconstruction quality. The improved generation quality is supported by the FID score [15], which is a quantitative measure to evaluate visual quality of image samples, shown in Table 1.

We also evaluate the cross-entropy $E_{q_\phi(\mathbf{z})}[\log p(\mathbf{z})]$ (③ in Eq (2)) between the aggregate posterior $q_\phi(\mathbf{z})$ and the prior $p(\mathbf{z})$ to measure the level of matching between the two distributions. Higher $E_{q_\phi(\mathbf{z})}[\log p(\mathbf{z})]$ indicates a better fit of the prior $p(\mathbf{z})$ to the learnt data distribution $q_\phi(\mathbf{z})$. As shown in Table 2, GMM priors can achieve very good fitting for relatively simple datasets, such as MNIST. However, when the dataset becomes more complex

Table 1: **FID scores** (lower is better) of generated and reconstructed samples. Our LaDDer model achieves the best sample quality and the minimal gap between the two.

	Normal	VampPrior	GMM	Hierarchical	Ours
Generation	250.3 ± 1.5	182.9 ± 1.2	140.4 ± 0.9	143.8 ± 1.0	**132.7** ± 0.8
Reconstruction	100.3 ± 0.6	95.5 ± 0.7	102.3 ± 0.5	99.7 ± 0.6	**95.3** ± 0.5
Difference	150.0 ± 2.1	87.4 ± 1.9	38.1 ± 1.4	44.1 ± 1.6	**37.4** ± 1.3

Table 2: **Cross-entropy** $E_{q_\phi(\mathbf{z})}[\log p(\mathbf{z})]$ (higher is bettter) between the aggregate posterior and the prior. Our LaDDer model achieves the best fit to $q_\phi(\mathbf{z})$ in more complex datasets.

Dataset	Normal	VampPrior	GMM	Hierarchical	Ours
MNIST	-19.0 ± 0.1	-16.2 ± 0.1	**-8.4** ± 0.9	-17.5 ± 0.1	-12.3 ± 0.1
fashion-MNIST	-52.5 ± 0.6	-60.8 ± 1.0	-45.2 ± 0.8	-63.8 ± 0.9	**-32.8** ± 0.7
CelebA	-351.4 ± 4.3	-380.7 ± 3.4	-290.6 ± 54.8	-358.0 ± 7.3	**-71.9** ± 2.1

and the required latent dimension increases, such as fashion-MNIST and CelebA datasets, our generative prior, which is more flexible to fit complex distributions, obtains the highest cross-entropy and achieve the best fit.

4.2 Overall Generative Modelling Performance Improves

A more flexible prior $p(\mathbf{z})$ also leads to better generative modelling performance, in terms of better reconstruction quality and higher ELBO objective. This effect is clearly shown in Table 3, where our method produces the highest ELBO value and the lowest reconstruction error across all datasets. All models are trained under the same conditions and the evaluation is repeated for 5 times to estimate the variance of the results.

Table 3: **ELBO** (higher is better) and per pixel **reconstruction error** (lower is better). Our method achieves highest ELBO and lowest reconstruction error across all datasets.

Metric	Dataset	Normal	VampPrior	GMM	Hierarchical	Ours
ELBO	MNIST	1528.6 ± 21.3	1476.3 ± 19.4	1558.2 ± 24.4	1534.9 ± 19.8	**1562.5** ± 25.1
	fashion	1078.4 ± 23.4	1069.6 ± 22.1	1133.3 ± 23.9	1152.9 ± 24.9	**1174.0** ± 29.7
	CelebA	55564.8 ± 1751.3	56463.4 ± 1683.9	54162.5 ± 1716.5	56176.2 ± 1763.9	**58596.1** ± 1815.3
Per pixel recons. error (×0.01)	MNIST	2.3 ± 0.06	2.5 ± 0.06	2.3 ± 0.05	2.4 ± 0.06	**2.2** ± 0.06
	fashion	4.3 ± 0.12	4.1 ± 0.11	4.0 ± 0.11	3.7 ± 0.11	**3.6** ± 0.11
	CelebA	5.8 ± 0.21	5.9 ± 0.25	5.9 ± 0.20	5.7 ± 0.17	**5.5** ± 0.18

Furthermore, our LaDDer model produces an embedding scheme that better preserves the semantics of the data. To see this, we visualise 30k latent encodings of the MNIST dataset in Fig. 4, where a VAE of 2D latent space is trained with a GMM prior, VampPrior and our generative prior respectively. All encodings are coloured by their class labels. Our method clearly gives better clustering results, where encodings of different classes are better separated. Furthermore, the mixture components estimated in our method are better aligned with class labels, whereas the mixtures from VampPrior and GMM prior have significant overlaps and do not have a consistent correspondence to specific class labels.

4.3 Our Shortest Likely Path Traversal Gives Better Interpolation

In Section 3.4, we formulate the latent space interpolation between a pair of images as a shortest likely path (SLP) optimisation task, which favours the paths that go through high density regions of the inferred latent data distribution $p(\mathbf{z})$ or $p(\mathbf{t})$. Here we demonstrate our SLP method informed by the learnt latent distribution outperforms the conventional linear shortest path (SP) interpolation.

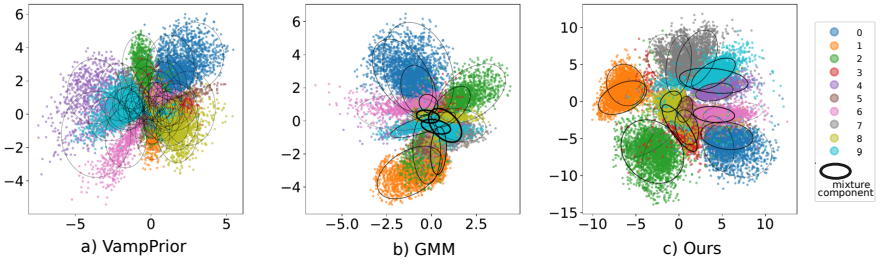

Figure 4: **The mixtures in our hyper prior naturally aligns with MNIST class labels**. We visualise encodings coloured by the digit labels for (a) VampPrior, (b) GMM prior and (c) our generative prior. Our method produces better clustering and the mixture components inferred in our method are naturally aligned with different classes.

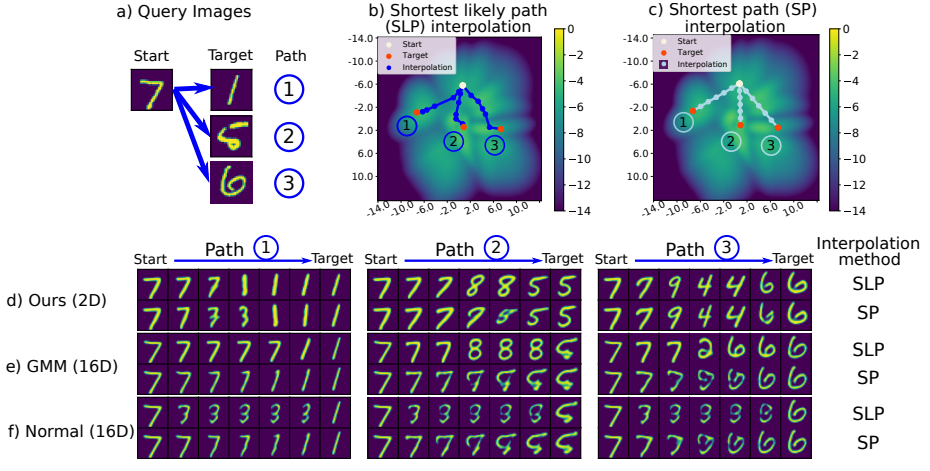

Figure 5: **Latent space interpolation for MNIST**. (a) The 3 pairs of MNIST images for the interpolation task. (b-c) The traversed paths produced by shortest likely path (SLP) and shortest path (SP) are visualised in our very low-dimensional t-space. (d-f) The interpolated images for both interpolation methods using the data distribution estimated by our method, a GMM prior and a normal prior.

Fig. 5 illustrates the interpolation between 3 pairs of MNIST images through the data manifolds estimated by our method, a GMM prior and a normal prior. Notice that our LaDDer model can produce the same quality samples from a much lower dimension latent space compared to the VAE models with the GMM and the normal prior (2D vs 16D). The super low latent dimension allows us to visualise the learnt data distribution $p(\mathbf{t})$ and the traversed path in Fig. 5b-c. Our SLP interpolation results in paths that only step on regions with high likelihood of $p(\mathbf{t})$, whereas SP interpolation ignores the topology of $p(\mathbf{t})$ and often lands on the low density regions which do not correspond to realistic data samples, hence producing unrealistic samples (bottom rows in Fig. 5d-f). Notice that neither SLP nor SP gives a good interpolation for the normal prior model. This is because the normal prior poorly represents the latent data distribution and even if an encoding has a high likelihood wrt the normal prior, it does not correspond to a realistic data sample. This reinforces the importance of obtaining an accurate modelling of the inferred data distribution in being able to utilise it for later tasks.

Fig. 6 illustrates the interpolation between 2 pairs of CelebA images over the data mani-

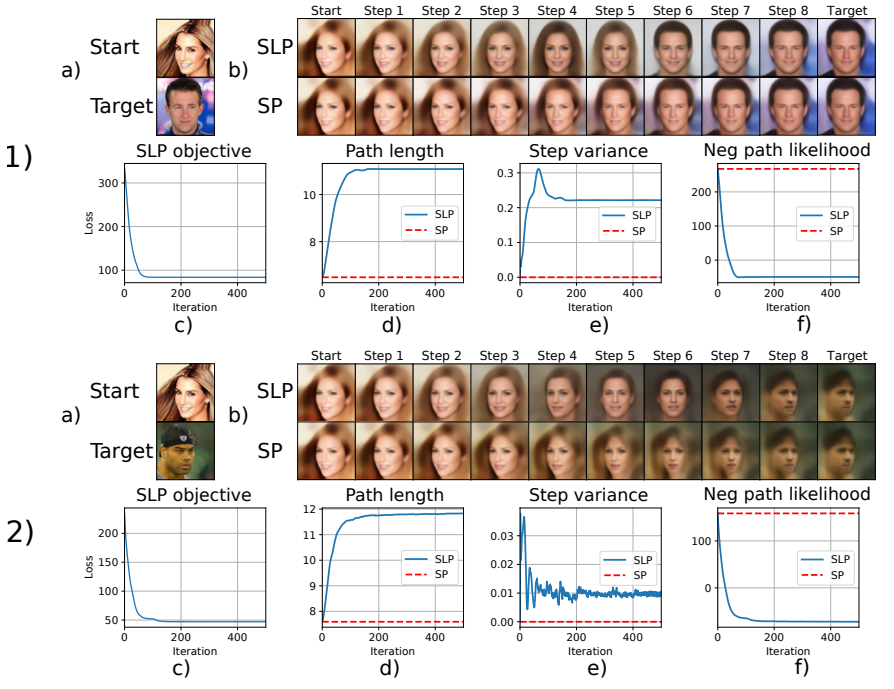

Figure 6: **Latent space interpolation for CelebA**. (a) The pair of images to be interpolated. (b) The sequence of interpolated images along the data manifold estimated by our LaDDer model, using two different interpolation methods. (c-f) Optimisation of different objectives in the our SLP objective. More examples are given in Supplementary Materials F4.

fold produced by our method, where the SLP optimisation takes place in the 32D latent space of the prior VAE. Here, we plot the different objectives in our SLP optimisation (blue line) for each example in (c-f). The overall objective ($\mathcal{O}_{\text{SLP}}$ in Eq (9)) smoothly converges in all examples. It is clear that our SLP solution trade-offs path length and allows different step lengths for obtaining high likelihood over the traversed path. As a result, our SLP method only produces realistic images which smoothly transform into the target face, whereas the SP method produces unrealistic faces along the traversed path.

5 Conclusion and Future Work

In this paper, we recognise the importance of adopting a sufficiently flexible prior in a VAE model to facilitate accurate modelling of the inferred latent data distribution. We propose LaDDer, which consists of multiple VAE models each acting on its predecessor's latent representation and a non-parametric mixture as the hyper prior for the innermost VAE. From extensive experiments, we show that our method is able to accurately model latent data distribution of complex data. We also demonstrate how to use the derived latent distribution to facilitate further tasks, such as producing better interpolation along the derived data manifold. We believe that LaDDer can be helpful in estimating the data distribution for many challenging datasets and the derived data distribution can be useful for a wide range of applications. We will continue to explore along these directions.

Acknowledgements

We thank Stephen Roberts and Niki Trigoni for engaging discussions and supporting the research behind this project. Shuyu Lin is supported by the EPSRC Centre for Doctoral Training, EP/L015897/1, and the China Scholarship Council. Ronald Clark is supported by an Imperial College Research Fellowship. We thank the reviewers for helpful comments.

References

[1] Tristan Aumentado-Armstrong, Stavros Tsogkas, Allan Jepson, and Sven Dickinson. Geometric Disentanglement for Generative Latent Shape Models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 8181–8190, 2019.

[2] David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and Improving Interpolation in Autoencoders via an Adversarial Regularizer. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[3] David M Blei, Michael I Jordan, et al. Variational inference for Dirichlet process mixtures. *Bayesian analysis*, 2006.

[4] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. CodeSLAM - learning a compact, optimisable representation for dense visual SLAM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2560–2568, 2018.

[5] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance Weighted Autoencoders, 2015.

[6] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 2020.

[7] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[8] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear Independent Components Estimation. *Conference on Robot Learning (CoRL)*, 2014.

[9] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[10] Anastasia Dubrovina, Fei Xia, Panos Achlioptas, Mira Shalah, and Leonidas J. Guibas. Composite Shape Modeling via Latent Space Factorization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[11] Patrick Esser, Johannes Haux, and Bjorn Ommer. Unsupervised Robust Disentangling of Latent Characteristics for Image Synthesis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2699–2709, 2019.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Neural Information Processing Systems (NeurIPS)*, pages 2672–2680.

[13] Prasoon Goyal, Zhiting Hu, Xiaodan Liang, Chenyu Wang, Eric P. Xing, and Carnegie Mellon. Nonparametric Variational Auto-Encoders for Hierarchical Representation Learning. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[14] Ikhsanul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. A recurrent Variational Autoencoder for Human Motion Synthesis. In *British Machine Vision Conference (BMVC)*, 2017.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems (NeurIPS)*, pages 6626–6637, 2017.

[16] Irina Higgins, Loïc 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 (ICLR)*, 2017.

[17] Yedid Hoshen, Ke Li, and Jitendra Malik. Non-Adversarial Image Synthesis With Generative Latent Nearest Neighbors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] Matthew J. Johnson, David Duvenaud, Alexander B. Wiltschko, Ryan P. Adams, and Sandeep R. Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Neural Information Processing Systems (NeurIPS)*, pages 2946–2954, 2016.

[19] Hyunjik Kim and Andriy Mnih. Disentangling by Factorising. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2649–2658. PMLR, 2018.

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for Stochastic Optimization. In *International Conference on Learning Representations, ICLR*, 2015.

[21] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Neural Information Processing Systems (NeurIPS)*, 2018.

[22] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv*, 2013.

[23] Diederik P. Kingma, Tim Salimans, and Max Welling. Improving Variational Inference with Inverse Autoregressive Flow. *arXiv*, 2016.

[24] Alexej Klushyn, Nutan Chen, Richard Kurle, Botond Cseke, and Patrick van der Smagt. Learning Hierarchical Priors in Vaes. In *Neural Information Processing Systems (NeurIPS)*, pages 2866–2875, 2019.

[25] Y. LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998. URL https://ci.nii.ac.jp/naid/10027939599/en/.

[26] Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett. Conditional image generation with pixelcnn decoders. In *Neural Information Processing Systems (NeurIPS)*, 2016.

[27] Shuyu Lin, Stephen J. Roberts, Niki Trigoni, and Ronald Clark. Balancing Reconstruction Quality and Regularisation in ElbO for Vaes. *arXiv*, 2019.

[28] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[29] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D reconstruction in Function Space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[30] Eric T. Nalisnick and Padhraic Smyth. Stick-Breaking Variational Autoencoders. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[31] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[32] Rajesh Ranganath, Dustin Tran, and David M. Blei. Hierarchical Variational Models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 324–333, 2016.

[33] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2. In *Neural Information Processing Systems (NeurIPS)*, pages 14837–14847, 2019.

[34] Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1530–1538, 2015.

[35] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1278–1286, 2014.

[36] Henglin Shi, Xin Liu, Xiaopeng Hong, and Guoying Zhao. Bidirectional Long Short-Term Memory Variational Autoencoder. In *British Machine Vision Conference (BMVC)*, page 165, 2018.

[37] Jakub M. Tomczak and Max Welling. Improving Variational Auto-Encoders using Householder Flow. *arXiv*, 2016.

[38] Jakub M. Tomczak and Max Welling. VAE with a VampPrior. In *Artificial Intelligence*, pages 1214–1223, 2018.

[39] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural Autoregressive Distribution Estimation. *J. Mach. Learn. Res.*, 2016.

[40] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[41] Linlin Yang and Angela Yao. Disentangling Latent Hands for Image Synthesis and Pose Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.