Supplementary Materials

In this document, we give additional information of the following topics related to our paper:

- the mathematical derivation for the re-arrangement of the cross-entropy term in the VAE ELBO loss (Section A),
- an algorithmic demonstration for our block coordinate ascent algorithm (Section B),
- the derivation for the reconstruction likelihood in VAE ELBO loss when a Laplace likelihood is used (Section C),
- and details of the datasets and pre-processing, experiment set-up, model architecture, ablation study and more results (Section D, E and F).

A. Mathematical Derivations

Here we give details of the mathematical derivation for the re-arrangement of the cross-entropy term in the VAE ELBO loss, which is mentioned in Section 2 and 3.1 of the main texts.

A.1. Re-arrangement of cross-entropy wrt prior

The cross-entropy term $\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{z})]$ appears as ③ in Eq (2) in the main texts. It is commonly used as a regulariser to the sample posteriors $q_\phi(\mathbf{z}|\mathbf{x})$. With the following re-arrangement, we show that this term is in fact an alignment requirement between the prior distribution $p(\mathbf{z})$ and the inferred aggregate posterior $q_\phi(\mathbf{z})$, i.e. the population distribution of all the samples in the learnt latent space. The re-arrangement is done as follows:

$$\begin{aligned}
\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{z})] &= \int_\mathbf{z}\int_\mathbf{x} \log p(\mathbf{z}) q_\phi(\mathbf{z}|\mathbf{x}) p_\mathcal{D}(\mathbf{x})\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{z} &(1)\\
&= \int_\mathbf{z} \log p(\mathbf{z}) \Big(\int_\mathbf{x} q_\phi(\mathbf{z}|\mathbf{x}) p_\mathcal{D}(\mathbf{x})\,\mathrm{d}\mathbf{x}\Big)\mathrm{d}\mathbf{z} &(2)\\
&= \int_\mathbf{z} \log p(\mathbf{z})\, \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}[q_\phi(\mathbf{z}|\mathbf{x})]\,\mathrm{d}\mathbf{z} &(3)\\
&= \int_\mathbf{z} \log p(\mathbf{z})\, q_\phi(\mathbf{z})\mathrm{d}\mathbf{z} &(4)\\
&= \mathbb{E}_{q_\phi(\mathbf{z})}[\log p(\mathbf{z})], &(5)
\end{aligned}$$

where

$$\text{aggregate posterior}\ \ q_\phi(\mathbf{z}) = \int q_\phi(\mathbf{z}|\mathbf{x}) p_\mathcal{D}(\mathbf{x})\mathrm{d}\mathbf{x} = \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}[q_\phi(\mathbf{z}|\mathbf{x})] \approx \frac{1}{N}\sum_{i=1}^{N} q_\phi(\mathbf{z}|\boldsymbol{x}_i). \tag{6}$$

When a cross-entropy is maximised between two distributions, the two distributions will try to have more overlaps and the optimal solution will be reached when the two distributions become the same. Therefore, this cross-entropy term in the ELBO objective simply implies that the optimal prior should be matching the aggregate posterior.

A.2. Re-arrangement of the Cross-entropy between Hyper Prior and Aggregate Hyper Posterior

A similar re-arrangement for the cross-entropy term in the prior VAE ELBO (Eq (4) in the main texts) can be also achieved in the following equations. This result indicates that the optimal hyper prior $p(\mathbf{t})$ should be chosen to best fit the aggregate hyper posterior $q_{\beta,\phi}(\mathbf{t})$, which is defined as an average encoding of all possible data encodings projected to the t-space. This fitting is crucial and failure to do so will result in poor modelling performance as shown in Section 4.1 of the main texts for the hierarchical model.

$$\begin{aligned}
\mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\mathbb{E}_{q_\beta(\mathbf{t}|\mathbf{z})}[\log p(\mathbf{t})] &= \mathbb{E}_{q_{\beta,\phi}(\mathbf{t})}[\log p(\mathbf{t})] && (7)\\
&= \int_{\mathbf{t}}\int_{\mathbf{z}}\int_{\mathbf{x}} \log p(\mathbf{t}) q_\beta(\mathbf{t}|\mathbf{z}) q_\phi(\mathbf{z}|\mathbf{x}) p_{\mathcal{D}}(\mathbf{x})\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{t} && (8)\\
&= \int_{\mathbf{t}}\int_{\mathbf{z}} \log p(\mathbf{t}) q_\beta(\mathbf{t}|\mathbf{z}) \Big(\int_{\mathbf{x}} q_\phi(\mathbf{z}|\mathbf{x}) p_{\mathcal{D}}(\mathbf{x})\,\mathrm{d}\mathbf{x} \Big) \mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{t} && (9)\\
&= \int_{\mathbf{t}} \log p(\mathbf{t}) \Big(\int_{\mathbf{z}} q_\beta(\mathbf{t}|\mathbf{z}) q_\phi(\mathbf{z}) \mathrm{d}\mathbf{z} \Big)\,\mathrm{d}\mathbf{t} && (10)\\
&= \int_{\mathbf{t}} \log p(\mathbf{t}) q_{\phi,\beta}(\mathbf{t})\,\mathrm{d}\mathbf{t} && (11)\\
&= \mathbb{E}_{q_{\phi,\beta}(\mathbf{t})}[\log p(\mathbf{t})], && (12)
\end{aligned}$$

where

$$\text{aggregate hyper posterior}\quad q_{\phi,\beta}(\mathbf{t}) = \int q_\beta(\mathbf{t}|\mathbf{z}) q_\phi(\mathbf{z}) \mathrm{d}\mathbf{z} = \mathbb{E}_{q_\phi(\mathbf{z})}[q_\phi(\mathbf{t}|\mathbf{z})] = \mathbb{E}_{q_\phi(\mathbf{z})}[q_\beta(\mathbf{t}|\mathbf{z})]. \tag{13}$$

B. Block Coordinate Ascent Algorithm

Here we give an algorithmic demonstration for our block coordinate ascent algorithm, which is introduced in Section 3.3 of the main texts as our optimisation strategy for all the model parameters in our LaDDer model (i.e. the data VAE parameters θ, ϕ, the prior VAE parameters α, β and the GMM hyper prior ξ).

Algorithm 1 Block coordinate ascent to optimise model parameters $\theta, \phi, \alpha, \beta, \xi$

```
$\phi$, $\theta$ $\alpha$ $\beta$ $\leftarrow$ Initialize parameters
$\xi \leftarrow$ Initialize as unit Gaussian
**repeat** until convergence of $\mathcal{L}''(\mathbf{x}; \theta, \phi, \alpha, \beta, \xi)$
    $\theta, \phi \leftarrow$ Update by optimising $\mathcal{L}''(\mathbf{x}; \theta, \phi \,|\, \alpha, \beta, \xi)$ using AdamOptimiser
    $\alpha, \beta \leftarrow$ Update by optimising $\mathcal{L}'(\mathbf{z}; \alpha, \beta \,|\, \xi, \phi)$ using AdamOptimiser
    $\xi \leftarrow$ At the end of an epoch, update by optimising $\mathcal{L}(\mathbf{t}; \xi)$ using the updates in coordinate ascent (Blei et al., 2006)
**return** $\theta, \phi, \alpha, \beta, \xi$
```

C. Laplace Distribution to Model the Decoder Likelihood

Throughout our experiments, we use a Laplace distribution to model the reconstruction likelihood $p_\theta(\boldsymbol{x}_i|\mathbf{z})$ (i.e. ① in Eq (2) from the main texts). Here we give the mathematical details of this distribution and it will become clear that this choice of the decoder distribution results in a L1 reconstruction error loss in the overall learning objective, which is more suited for modelling image data. The derivation follows the idea proposed in (Lin et al., 2019).

The Laplace likelihood is defined as

$$p_\theta(\boldsymbol{x}_i|\mathbf{z}) = \prod_{k=1}^{d} \frac{1}{2\sigma} \exp - \frac{|\boldsymbol{x}_i^k - \mu_\theta^k(\mathbf{z})|}{\sigma}, \tag{14}$$

where σ is a hyper parameter which we also optimise under the ELBO objective. Substituting the above equation into the negative ELBO loss in Eq (2) of the main texts, we have

$$-\mathcal{L}(\mathbf{x}; \theta, \phi, \sigma) = \frac{1}{\sigma} \mathbb{E}_{\mathbf{z}} \Bigg[\underbrace{\sum_{k=1}^{d} |\mu_\theta^k(\mathbf{z}) - \boldsymbol{x}_i^k|}_{①} \Bigg] + D_{\mathrm{KL}}\big[\, q_\phi(\mathbf{z}|\boldsymbol{x}_i) \,\|\, p(\mathbf{z}) \big] + d \log 2\sigma, \tag{15}$$

where $\mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}[\cdot]$ is omitted. Now it is clear that the reconstruction likelihood ① simplifies to an element-wise absolute error (or L1 norm) also labelled as ① here.

D. Datasets and Pre-processing

We ran experiments on MNIST, Fashion MNIST and CelebA datasets. For all datasets, we take the images as real-valued data. Here we give details of these datasets and how we pre-process them for our training the VAE models in Section 4 of the main texts.

D.1. MNIST and Fashion MNIST

MNIST and Fashion MNIST datasets both consist of a training and a test set with 60k and 10k images of hand-written digits/clothes, where each image is a 28×28 grey-scale image. We rescale the original images so that the pixel values are within the range $[0, 1]$.

D.2. CelebA

CelebA dataset consists of 202599 RGB images of celebrity faces. We first resize all the images to 128 $\times$ 128 and rescale the pixel values to be within $[0, 1]$. We then randomly select 20k images (about 10%) to be the test set.

E. Model Architectures and Training Parameters

E.1. MNIST and Fashion MNIST

We use CNN architectures for both MNIST and Fashion MNIST encoder and decoder network (with additional depth to space operations for the decoder), as detailed below. We use the exactly same network for both datasets.

Encoder network:

$$\begin{aligned}
\boldsymbol{x} \in \mathcal{R}^{28\times 28\times 1} \rightarrow \text{Symmetric padding} \rightarrow \mathcal{R}^{32\times 32\times 1} &\rightarrow \text{Conv}^{3\times 3}_{16}\ (\text{stride}=2) \rightarrow \text{ReLU}\\
&\rightarrow \text{Conv}^{3\times 3}_{64}\ (\text{stride}=2) \rightarrow \text{ReLU}\\
&\rightarrow \text{Conv}^{3\times 3}_{256}\ (\text{stride}=2) \rightarrow \text{ReLU}\\
&\rightarrow \text{FC}_{64} \rightarrow \text{FC}_{d_z} \Rightarrow\ \mu \in \mathcal{R}^{d_z}\\
&\qquad\quad \searrow\\
&\qquad\qquad \text{FC}_{d_z} \rightarrow \text{ReLU} \Rightarrow\ \sigma \in \mathcal{R}^{d_z}
\end{aligned}$$

Decoder network:

$$\begin{aligned}
\boldsymbol{z} = \mu + \epsilon \odot \sigma &\rightarrow \text{FC}_{4096} \rightarrow \text{ReLU} \rightarrow \text{Reshape} \rightarrow \mathcal{R}^{1\times 1\times 4096}\\
&\rightarrow \text{depth to space}_4 \rightarrow \mathcal{R}^{4\times 4\times 256} \rightarrow \text{Conv}^{3\times 3}_{256}\ (\text{stride}=1) \rightarrow \text{ReLU}\\
&\rightarrow \text{depth to space}_2 \rightarrow \mathcal{R}^{8\times 8\times 64} \rightarrow \text{Conv}^{3\times 3}_{64}\ (\text{stride}=1) \rightarrow \text{ReLU}\\
&\rightarrow \text{depth to space}_2 \rightarrow \mathcal{R}^{16\times 16\times 16} \rightarrow \text{Conv}^{3\times 3}_{16}\ (\text{stride}=1) \rightarrow \text{ReLU}\\
&\rightarrow \text{depth to space}_2 \rightarrow \mathcal{R}^{32\times 32\times 4} \rightarrow \text{Conv}^{5\times 5}_{1}\ (\text{stride}=1, \text{padding=valid})\\
&\rightarrow \text{Sigmoid} \rightarrow \text{Reshape} \Rightarrow\ \hat{\boldsymbol{x}} \in \mathcal{R}^{28\times 28\times 1}
\end{aligned}$$

Prior VAE network:

$$\begin{aligned}
&\boldsymbol{z} \rightarrow (\text{FC}_{512} \rightarrow \text{Leaky ReLU}) \times 3 \rightarrow \text{FC}_{d_t} \rightarrow\ \mu_t \in \mathcal{R}^{d_t}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \searrow\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{FC}_{d_t} \rightarrow \text{ReLU} \Rightarrow\ \sigma_t \in \mathcal{R}^{d_t}\\
&\boldsymbol{t} = \mu_t + \epsilon \odot \sigma_t \rightarrow (\text{FC}_{512} \rightarrow \text{Leaky ReLU}) \times 3\\
&\quad \rightarrow \text{FC}_{d_z} \rightarrow \hat{z} \in \mathcal{R}^{d_z}
\end{aligned}$$

d_z is the dimension of z-space, d_t is the latent dimension of our prior VAE, μ is code mean and σ is decoder variance parameter introduced in Section C, $\epsilon \sim \mathcal{N}(0, \mathcal{I})$ and unless otherwise specified the padding in the convolution operation is 'same'. We use the following hyper-parameters to train the network:

Batch size	d_z	d_t	N epoch	Optimizer	Learning rate	Padding
256	16	2	50	Adam	5×10^{-4}	SAME

E.2. CelebA

For CelebA dataset, we implement a CNN encoder and a Style-GAN generator, as detailed below. Encoder network:

$$\begin{aligned}
\boldsymbol{x} \in \mathcal{R}^{128\times128\times3} &\rightarrow \text{Conv}_{32}^{3\times3}\ (\text{stride}=2) \rightarrow \text{Leaky ReLU}\\
&\rightarrow \text{Conv}_{64}^{3\times3}\ (\text{stride}=2) \rightarrow \text{Leaky ReLU}\\
&\rightarrow \text{Conv}_{128}^{3\times3}\ (\text{stride}=2) \rightarrow \text{Leaky ReLU}\\
&\rightarrow \text{Conv}_{256}^{3\times3}\ (\text{stride}=2) \rightarrow \text{Leaky ReLU}\\
&\rightarrow \text{Conv}_{512}^{3\times3}\ (\text{stride}=2) \rightarrow \text{Leaky ReLU}\\
&\rightarrow \text{Conv}_{1024}^{3\times3}\ (\text{stride}=2, \text{padding=valid}) \rightarrow \text{Leaky ReLU}\\
&\rightarrow \text{FC}_{512} \rightarrow \text{FC}_{d_z} \Rightarrow\ \mu \in \mathcal{R}^{d_z}\\
&\qquad\qquad \searrow\\
&\qquad\qquad\quad \text{FC}_{d_z} \Rightarrow\ \sigma \in \mathcal{R}^{d_z}
\end{aligned}$$

Decoder network which is inspired by the Style-GAN architecture is given below. The Style Module refers to a fully connected a layer that first doubles the channel dimensions and then merges the channel dimension back to the original dimension by summing the first half channels with the original channel value weighted second half channels. This operation is modified from the original Style-GAN generator to fit with a VAE model.

$$\begin{aligned}
\boldsymbol{z} = \mu + \epsilon \odot \sigma &\rightarrow (\text{FC}_{512} \rightarrow \text{Leaky ReLU}) \times 8 \rightarrow \text{Reshape} \rightarrow \mathcal{R}^{1\times1\times512}\\
&\rightarrow \text{Resize}_{128}^{2\times2} \rightarrow \text{CNN}_{512}^{3\times3} \rightarrow \text{Instance Norm} \rightarrow \text{Style Module} \rightarrow \mathcal{R}^{2\times2\times512}\\
&\rightarrow \text{Resize}_{128}^{4\times4} \rightarrow \text{CNN}_{512}^{3\times3} \rightarrow \text{Instance Norm} \rightarrow \text{Style Module} \rightarrow \mathcal{R}^{4\times4\times512}\\
&\rightarrow \text{Resize}_{128}^{8\times8} \rightarrow \text{CNN}_{512}^{3\times3} \rightarrow \text{Instance Norm} \rightarrow \text{Style Module} \rightarrow \mathcal{R}^{8\times8\times512}\\
&\rightarrow \text{Resize}_{128}^{16\times16} \rightarrow \text{CNN}_{256}^{3\times3} \rightarrow \text{Instance Norm} \rightarrow \text{Style Module} \rightarrow \mathcal{R}^{16\times16\times256}\\
&\rightarrow \text{Resize}_{64}^{32\times32} \rightarrow \text{CNN}_{256}^{3\times3} \rightarrow \text{Instance Norm} \rightarrow \text{Style Module} \rightarrow \mathcal{R}^{32\times32\times256}\\
&\rightarrow \text{Resize}_{64}^{64\times64} \rightarrow \text{CNN}_{128}^{3\times3} \rightarrow \text{Instance Norm} \rightarrow \text{Style Module} \rightarrow \mathcal{R}^{64\times64\times128}\\
&\rightarrow \text{Resize}_{32}^{128\times128} \rightarrow \text{CNN}_{128}^{3\times3} \rightarrow \text{Instance Norm} \rightarrow \text{Style Module} \rightarrow \mathcal{R}^{128\times128\times128}\\
&\rightarrow \text{CNN}_{128}^{3\times3} \rightarrow \text{Leaky ReLU} \rightarrow \text{CNN}_{3}^{3\times3} \rightarrow \mathcal{R}^{128\times128\times3}
\end{aligned}$$

Prior VAE network:

$$\begin{aligned}
\boldsymbol{z} &\rightarrow (\text{FC}_{512} \rightarrow \text{Leaky ReLU}) \times 5 \rightarrow \text{FC}_{d_t} \rightarrow\ \mu_t \in \mathcal{R}^{d_t}\\
&\qquad\qquad\qquad\qquad\qquad \searrow\\
&\qquad\qquad\qquad\qquad\qquad\quad \text{FC}_{d_t} \rightarrow \text{ReLU} \Rightarrow\ \sigma_t \in \mathcal{R}^{d_t}\\
\boldsymbol{t} &= \mu_t + \epsilon \odot \sigma_t \rightarrow (\text{FC}_{512} \rightarrow \text{Leaky ReLU}) \times 5\\
&\rightarrow \text{FC}_{d_z} \rightarrow \hat{z} \in \mathcal{R}^{d_z}
\end{aligned}$$

We use the following hyper-parameters to train the network:

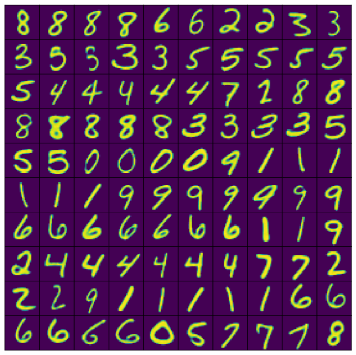
Batch size	d_z	d_t	N epoch	Optimizer	Learning rate	Padding
256	256	32	75	Adam	5×10^{-4}	SAME

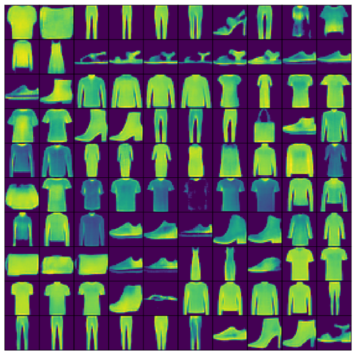
F. More Experimental Results

Now we show more results that supplement the results from the main texts. We give more generated samples from each VAE model, FID scores on the MNIST and fashion-MNIST datasets, an ablation study on the impacts of the prior VAE's latent dimension on the overall modelling performance and finally more examples of latent space interpolation using the derived latent distribution.

F.1. Generated Samples from Inferred Prior Distribution

Here we generate samples from the inferred prior distribution of each VAE model. The sample quality indicates how well the prior distribution matches the learnt latent data distribution. As can be seen, our method and GMM prior produce the best quality samples. However, in the higher dimensional latent space which is often needed for more complex datasets such as CelebA, our method outperforms the GMM prior. It is also very clear that the standard VAE with a normal prior fails to represent the inferred data distribution at all. This calls for caution in using the unit Gaussian prior if the inferred latent distribution is actually needed to represent the realistic data samples in later tasks.


(a) MNIST.


(b) Fashion MNIST.

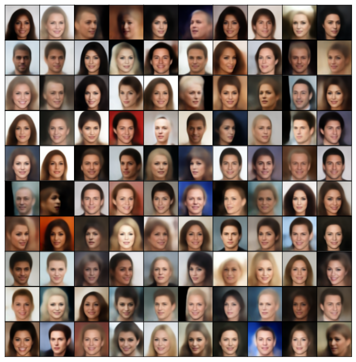
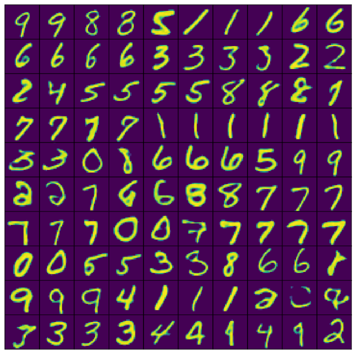
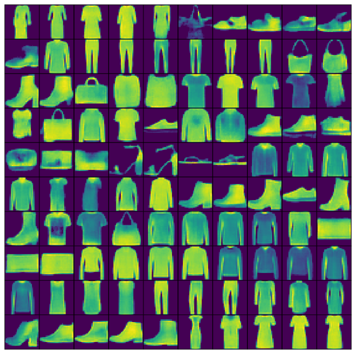

(c) CelebA.

Figure 1. 100 images generated from the inferred prior distribution in our LaDDer model for each of the 3 datasets.


(a) MNIST.


(b) Fashion MNIST.

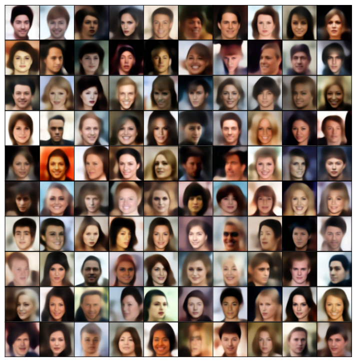
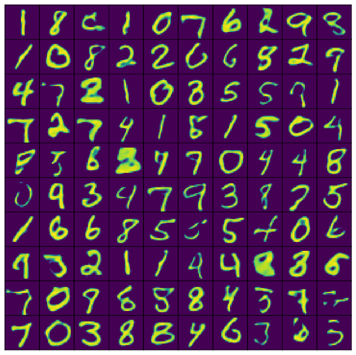
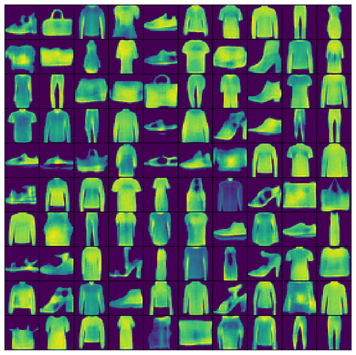

(c) CelebA.

Figure 2. 100 images generated from the inferred prior distribution in a GMM-VAE model for each of the 3 datasets.


(a) MNIST.


(b) Fashion MNIST.

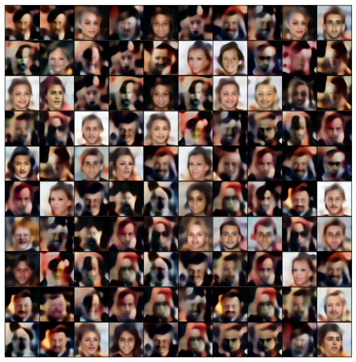
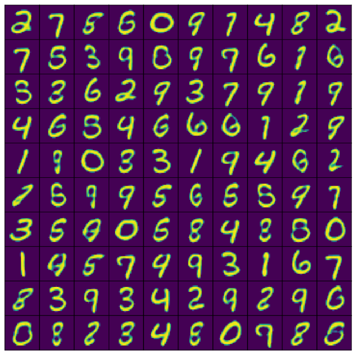
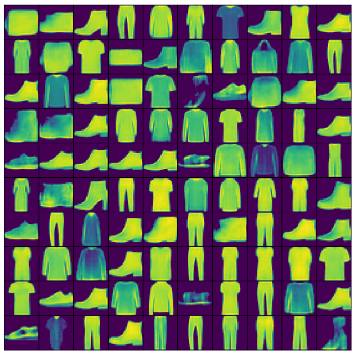

(c) CelebA.

Figure 3. 100 images generated from the inferred prior distribution in VampPrior model for each of the 3 datasets.


(a) MNIST.


(b) Fashion MNIST.

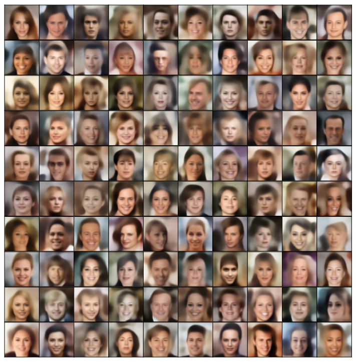
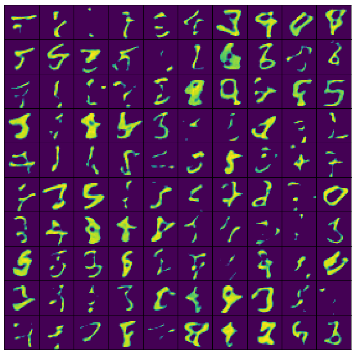
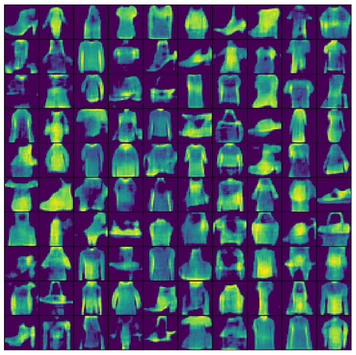

(c) CelebA.

Figure 4. 100 images generated from the inferred prior distribution in a VAE with a hierarchical prior for each of the 3 datasets.


(a) MNIST.


(b) Fashion MNIST.

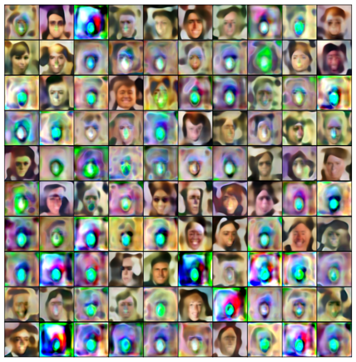

(c) CelebA.

Figure 5. 100 images generated from the inferred prior distribution in standard VAE (normal prior) for each of the 3 datasets.

F.2. FID Scores on MNIST and fashion-MNIST

In Section 4.1 from the main texts, we demonstrate that our generative prior can accurately model the inferred latent data distribution by evaluating the FID scores of both generated and reconstructed samples. We pay special attention to the difference in the visual quality between the generation and reconstruction. This is because the reconstruction samples quality of a VAE indicates the best sample quality the VAE can generate (if the prior 100% matches the inferred data manifold point-by-point). We give the FID score for CelebA dataset in the main texts. Here we also give the FID scores for MNIST and fashion-MNIST datasets, which tell a very similar story that our LaDDer model can generate very good quality samples and the gap between generation and reconstruction is small.

Table 1. FID scores (lower is better) of generated and reconstructed samples from 5 VAE models with different priors. Our generative prior and GMM hyper prior achieves best sample quality in both generation and reconstruction and the minimal gap between the two.

Dataset	Prior model	Normal	VampPrior	GMM	Hierarchical	Ours	Real image
MNIST digit	Generation	401.4 ± 4.4	47.2 ± 3.6	2.3 ± 0.5	52.4 ± 4.7	6.4 ± 1.3	2.3 ± 0.4
	Reconstruction	2.1 ± 0.8	1.4 ± 0.4	2.5 ± 0.8	5.4 ± 1.4	2.4 ± 0.4	
	Difference	399.3 ± 5.2	45.8 ± 4.0	-0.2 ± 1.3	47.0 ± 3.3	4.0 ± 1.7	
MNIST fashion	Generation	266.1 ± 2.4	23.6 ± 1.2	6.0 ± 1.0	243.8 ± 2.9	5.2 ± 0.9	0.4 ± 0.1
	Reconstruction	3.2 ± 0.4	3.0 ± 0.2	2.9 ± 0.4	1.8 ± 0.2	1.0 ± 0.4	
	Difference	262.9 ± 2.8	20.6 ± 1.4	3.1 ± 1.4	242.0 ± 3.1	4.2 ± 1.3	

An interesting effect shown here is that the VAE model with a GMM prior also achieves very good performance. We reckon this is because the MNIST datasets are relatively simple and hence a GMM fitting in a relatively low-dimensional z-space can match the overall data distribution fairly well. However, when the dataset becomes more complex, such as the CelebA dataset, our generative prior is more flexible and outperforms the GMM prior, as shown in the main texts Section 4.1.

F.3. Ablation Study: the Impact of Hyper Latent Dimension on Performance

Here we carry out an ablation study on the impact of the latent dimension of our prior VAE model on the overall modelling performance, specifically on the generation quality. Over all three datasets, the generation quality improves as we allow the prior VAE to adopt a higher latent dimension. However, the benefit of increasing the latent dimension diminishes as the dimension reaches certain level. This indicates that the dimension reduction achieved by the prior VAE module is still valid and does not limit the model's performance as long as its latent dimension is not set to be overly small.

Table 2. FID scores (lower is better) of generated samples from our generative prior with different latent dimension d_t. It is clear with higher latent dimension, the generation improves. However, the benefits diminish as the dimension increases.

Dataset	Ours					Real image
	2d	4d	8d	16d	32d	
MNIST digit	58.7 ± 1.8	35.1 ± 8.1	6.4 ± 1.3	6.9 ± 0.8	N/A	2.3 ± 0.4
MNIST fashion	25.3 ± 1.6	9.9 ± 1.2	5.5 ± 0.7	5.2 ± 0.9	N/A	0.4 ± 0.1
CelebA	175.7 ± 0.5	148.4 ± 0.6	138.4 ± 0.5	133.5 ± 0.6	132.7 ± 0.5	5.9 ± 0.5

F.4. Latent Space Interpolation Through Our Shortest Likelihood Path

In the main texts, we demonstrate how to use the estimated latent distribution to facilitate further tasks through a latent space interpolation example. We show that the derived latent distribution can inform the interpolation to only step on high likelihood regions in the latent space. As a result, the inferred samples remain realistic throughout the path and render smooth transitions into the target image. Here, we show more examples on the interpolation task and we also show our shortest likely path traversal on two other VAE models to indicate it can be generally applied to other methods. Across the interpolation results, the interpolation informed with the learnt latent distribution generates better samples. The exception is the VAE with a normal prior and this is because the normal prior poorly models the true data distribution.

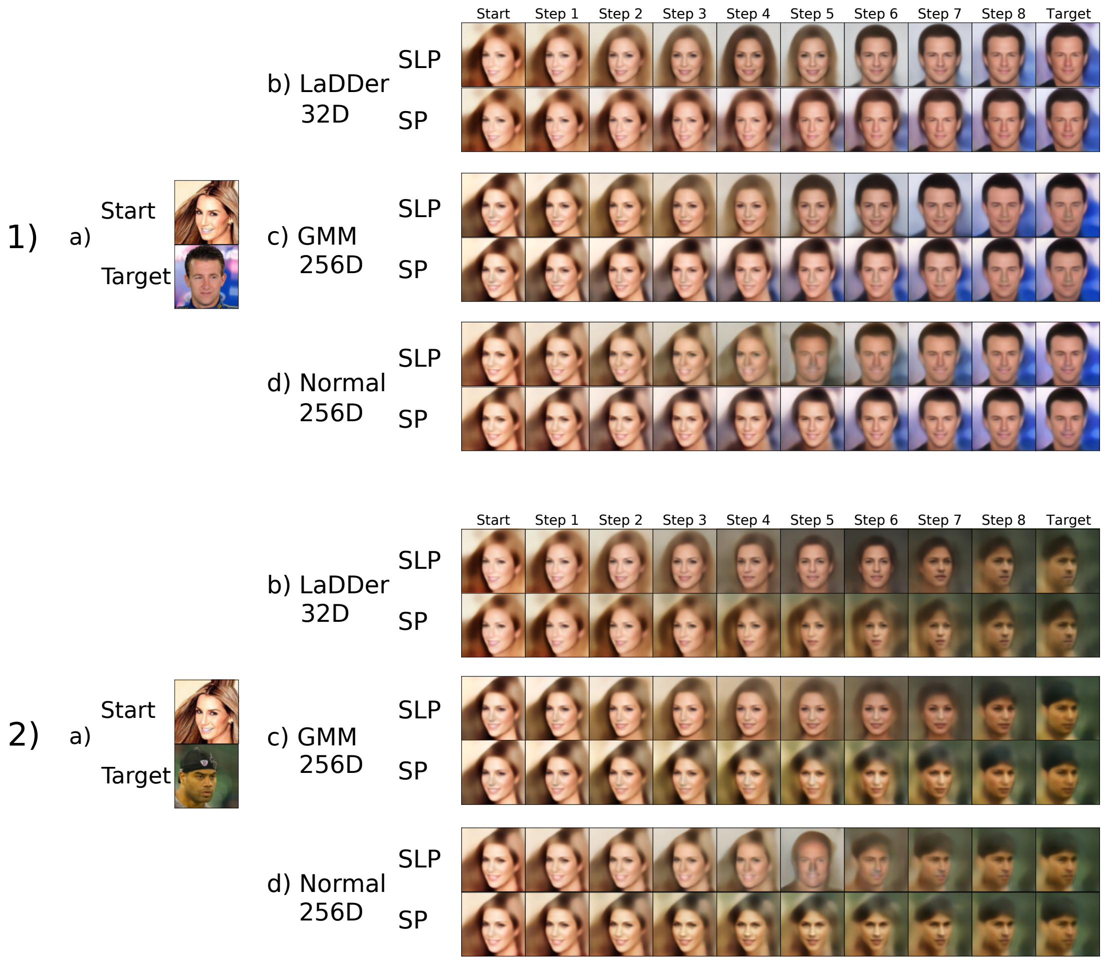


Figure 6. Latent space interpolation between a pair of CelebA images. We show the interpolated images for 3 VAE models: (b) our proposed LaDDer, (c) a GMM prior and (d) a normal prior. For each method, top row gives the interpolation of our shortest likely path (SLP) method and bottom row gives the interpolation of the shortest path (SP) method.

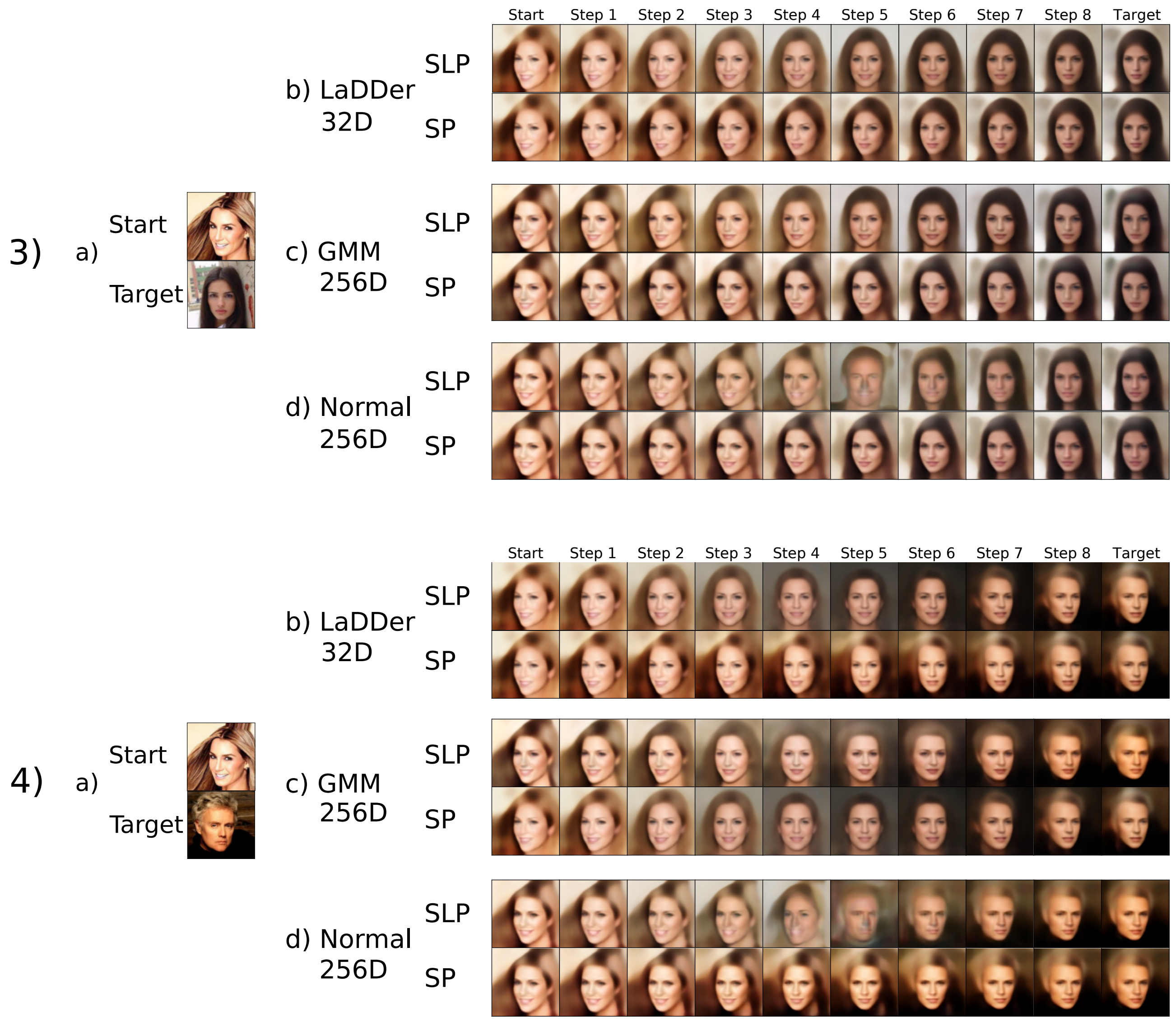


Figure 7. Latent space interpolation between a pair of CelebA images. We show the interpolated images for 3 VAE models: (b) our proposed LaDDer, (c) a GMM prior and (d) a normal prior. For each method, top row gives the interpolation of our shortest likely path (SLP) method and bottom row gives the interpolation of the shortest path (SP) method.

References

Blei, D. M., Jordan, M. I., et al. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

Lin, S., Roberts, S. J., Trigoni, N., and Clark, R. Balancing reconstruction quality and regularisation in ELBO for vaes. *arXiv*, 2019.