IMPROVING GAUSSIAN MIXTURE LATENT VARIABLE MODEL CONVERGENCE WITH OPTIMAL TRANSPORT

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models with both discrete and continuous latent variables are highly motivated by the structure of many real-world data sets. They present, however, subtleties in training often manifesting in the discrete latent variable not being leveraged. In this paper, we show why such models struggle to train using traditional log-likelihood maximization, and that they are amenable to training using the Optimal Transport framework of Wasserstein Autoencoders. We find our discrete latent variable to be fully leveraged by the model when trained, without any modifications to the objective function or significant fine tuning. Our model generates comparable samples to other approaches while using relatively simple neural networks, since the discrete latent variable carries much of the descriptive burden. Furthermore, the discrete latent provides significant control over generation.

1 INTRODUCTION

Unsupervised learning using generative latent variable models provides a powerful and general approach to learning the underlying, low-dimensional structure from large, unlabeled datasets. Perhaps the two most common techniques for training such models are Variational Autoencoders (VAEs) (Kingma & Welling, 2014; Rezende et al., 2014), and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). Both have advantages and disadvantages. VAEs provide a meaningful lower bound on the log likelihood that is stable under training, as well as an encoding distribution from the data into the latent. However, they generate blurry samples due to their objective being unable to handle deterministic decoders and tractability requiring simple priors (Hoffman & Johnson, 2016). On the other hand, GANs naturally enable deterministic generative models with sharply defined samples, but their training procedure is less stable (Arjovsky & Bottou, 2017).

A relatively new approach to training generative models has emerged based on minimizing the Optimal Transport (OT) distance (Villani, 2008) between the generative model distribution and that of the data. The OT approach provides a general framework for training generative models, which promises some of the best of both GANs and VAEs. Though interesting first results have been given in Arjovsky et al. (2017); Rubenstein et al. (2018); Tolstikhin et al. (2018), the OT approach to generative modelling is still nascent.

Our contributions are twofold: we seek to improve generative modelling capabilities with discrete and continuous latent variables, but importantly, we seek also to establish that training generative models with OT can be significantly more effective than the traditional VAE approach.

Discrete latent-variable models are critical to the endeavor of unsupervised learning because of the ubiquity of discreteness in the natural world, and hence in the datasets that describe it. However, they are harder to train than their continuous counterparts. This has been tackled in a number of ways (e.g., directly mitigating high-variance discrete samples (Eslami et al., 2016; Lawson et al., 2018), parametrizing discrete distributions using continuous ones (Jang et al., 2017; Maddison et al., 2017; Van den Oord et al., 2017), deliberate model design leveraging conjugacy (Johnson et al., 2016)).

However, even in the simple case where the number of mixtures is small enough that monte-carlo sampling from the discrete latent is avoidable, training can still be problematic. For example, in Dilokthanakul et al. (2016) a Gaussian-mixture latent-variable model (GM-LVM) was studied, and the authors were unable to train their model on MNIST using variational inference without substantially modifying the VAE objective. What appears to happen is that the model quickly learns to "hack" the

VAE objective function by collapsing the discrete latent variational distribution. This problem only occurs in the unsupervised setting, as Kingma et al. (2014) are able to learn the discrete latent in the semi-supervised version of the same problem once they have labeled samples for the discrete latent to latch onto. This is discussed in more detail in Section 2.1.

The OT approach to training generative models (in particular the Wasserstein distance, discussed in Section 2.2) induces a weaker topology on the space of distributions, enabling easier convergence of distributions than in the case of VAEs (Bousquet et al., 2017). Thus, one might conjecture that the OT approach would enable easier training of GM-LVMs than the VAE approach. We provide evidence that this is indeed the case, showing that GM-LVMs can be trained in the unsupervised setting on MNIST, and motivating further the value of the OT approach to generative modelling.

2 Gaussian-Mixture Wasserstein Autoencoders

We consider a hierarchical generative model p_G with two layers of latent variables, the highest one being discrete. Explicitly, if we denote the discrete latent k with density p_D (D for discrete), and the continuous latent z with density p_C (C for continuous), the generative model is given by:

$$p_G(x) = \sum_{k=1}^{K} \int_{\mathcal{Z}} dz \; p_G(x|z) \, p_C(z|k) \, p_D(k) \tag{1}$$

In this work, we consider a GM-LVM with categorical distribution $p_D = \text{Cat}(K)$ and continuous distribution $p_C(z|k) = \mathcal{N}(z; \mu_k^0, \Sigma_k^0)$. We refer to this GM-LVM as a GM-VAE when it is trained as a VAE (Kingma & Welling, 2014; Rezende et al., 2014) or GM-WAE when trained as a Wasserstein Autoencoder (Tolstikhin et al., 2018) (discussed in Section 2.2).

2.1 The difficulty of training GM-VAEs

Training GM-LVMs in the traditional VAE framework (GM-VAEs) involves maximizing the evidence lower bound (ELBO) averaged over the data. Such models are empirically hard to train (Dilokthanakul et al., 2016). This is likely due to the fact that the discrete latent variational distribution learns on a completely different scale from the generative distribution. Consequently, the discrete latent tends to instantly learn some unbalanced structure where its classes are meaningless in order to accommodate the untrained generative distribution. The generative model then learns around that structure, galvanizing the meaningless discrete distribution early in training.

More explicitly, if we choose a variational distribution $q(z,k|x) = q_C(z|k,x) \, q_D(k|x)$ to mirror the prior in Equation 1, the ELBO can be written as follows:

$$\text{ELBO} = \mathbb{E}_{q_D}\Big[\mathbb{E}_{q_C}\big[\log p_G(x|z)\big] - D_{\text{KL}}\big[q_C(z|k,x)\big|\big|p_C(z|k)\big]\Big] - D_{\text{KL}}\big[q_D(k|x)\big|\big|p_D(k)\big] \tag{2}$$

Both the first and the second term in Equation 2 depend on $q_D(k|x)$. However, the second term is much smaller than the first; it is bounded by $\log K$ for uniform p_D over K classes, whereas the first term is unbounded from above (though we will initialize the modes of q_C to match those of the priors making the continuous KL term initially small as well). As a consequence, $q_D(k|x)$ will immediately shut off the k values (i.e., $q_D(k|x) = 0 \;\forall x$) with large reconstruction losses, $\mathbb{E}_{q_C(z|k,x)}[\log p_G(x|z)]$. This is shown in the top row of Figure 1 where within the first 10 training steps the reconstruction loss has substantially decreased (Figure 1a) by simply shutting off 9 values of k in $q_D(k|x)$ (Figure 1b), resulting in a drastic increase of the discrete KL term (Figure 1a). However, this increase in the discrete KL term is negligible since the term is multiple orders of magnitude smaller than the reconstruction term in the ELBO. All of this takes place in the first few training iterations; well before the generative model has learned to use its continuous latent (see Figure 1c).

Subsequently, on a slower timescale, the generative model starts to learn to reconstruct from its continuous latent, causing $q_C(z|k,x)$ to shift away from its prior toward a more-useful distribution to the generative model. We see this in Figure 1d: the continuous KL curve grows concurrently with the downturn of the reconstruction loss term. Figure 1f shows that after this transition (taking a few thousands training steps), the reconstructions from the model start to look more like MNIST digits.

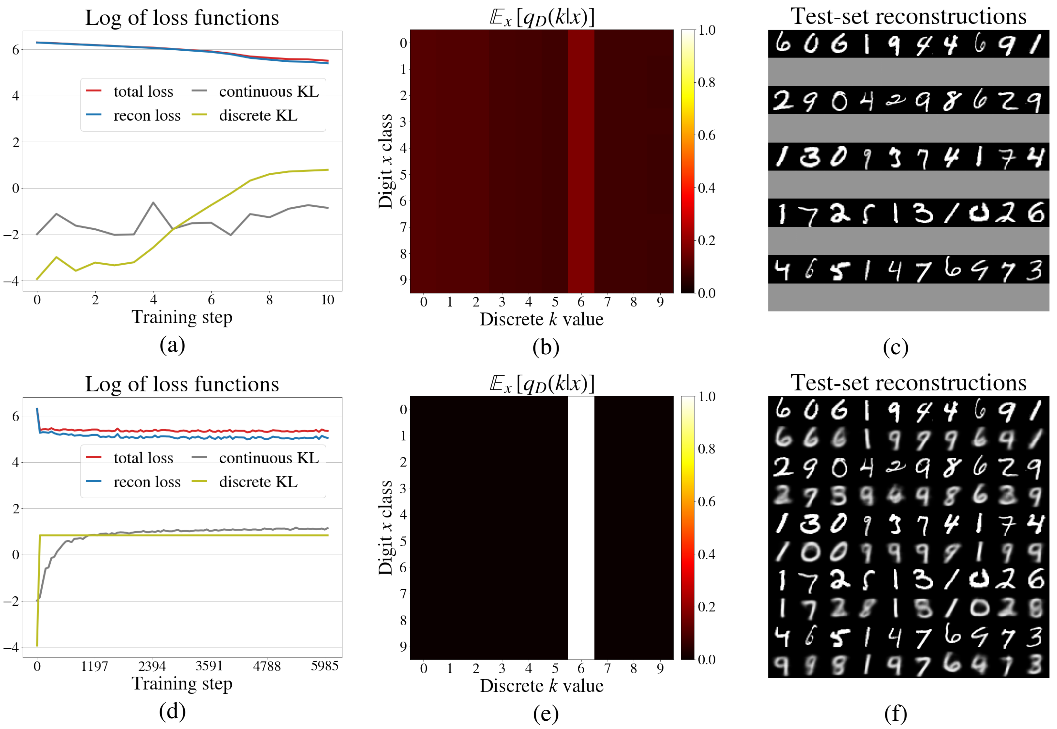


Figure 1: Top row shows a snapshot of the GM-VAE after 10 training steps. Loss curves are shown in (a), the discrete variational distribution in (b) with rows ℓ representing $\mathbb{E}_{\{x|\text{label}(x)=\ell\}} q_D(k|x)$, and reconstructions are shown in (c). Bottom row shows the same snapshot after 6000 training steps.

While the generative model learns to use the continuous latent, the discrete distribution $q_D(k|x)$ never revives the k values that it shut off. This is because the generative model would not know how to use the $z \sim q_C(z|k,x)$ values for those ks, implying a significant penalty in the reconstruction term of the ELBO. This is evidenced in Figure 1d by the discrete KL staying flat, and in Figure 1e where the columns corresponding to the shut off k values never repopulate.

We have discussed the difficulty of leveraging the structure of the latent variables in GM-VAEs using our specific implementation designed to mirror the GM-WAE of Section 2.2. Many other variants of this implementation performed similarly. Though the root cause of this difficulty has not been ascertained in generality, we expect it to be in part due to the per-data-point nature of the ELBO objective, in particular, the impact of the KL divergence term on learning the variational distribution. This point will be elaborated upon with more empirical justification in Section 3.

2.2 Optimal Transport facilitates training of GM-LVMs

The difficulty associated with training GM-VAEs may be interpreted as a problem of restricted convergence of a sequence of distributions, where the sequence is indexed by the training steps. If that were so, an objective function that induces a weaker topology (and therefore, allows sequences to converge more easily) might help GM-LVMs converge to a distribution that non-trivially uses its discrete latent variable. Hence, we are motivated to consider approaching the training of such models using the OT framework, and in particular the Wasserstein distance as our objective, as it is known to induce a weaker topology than that of maximum likelihood.

Following the OT approach of Tolstikhin et al. (2018), we would like to minimize the 2-Wasserstein distance between the underlying data distribution (from which we have samples) and our GM-LVM:

$$W_2^\dagger\left(p_{\text{data}}, p_G\right)^2 = \inf_{\substack{q(z,k|x)\in\mathcal{P}_{\mathcal{Z}\times\mathcal{K}} \\ \mathbb{E}_{p_{\text{data}}(x)}[q(z,k|x)]=p_C(z|k)p_D(k)}} \mathbb{E}_{p_{\text{data}}(x)}\mathbb{E}_{q(z,k|x)}\mathbb{E}_{p_G(y|z)}\left[||x-y||_2^2\right] \qquad (3)$$

where $\mathcal{P}_{\mathcal{Z}\times\mathcal{K}}$ is the set of all joint distributions over z and k, such that $q(z,k|x) = q_C(z|k,x)q_D(k|x)$ with q_C and q_D parametrized below. Any parametrization of $q(z,k|x)$ reduces the search space of the infimum, so $W_2^\dagger$ is in fact an upper bound on the true 2-Wasserstein distance W_2. Note that $W_2^\dagger$ is only equal to the true 2-Wasserstein distance when $p_G(y|z)$ is deterministic, providing an upper bound in the case of random generative models (Tolstikhin et al., 2018). We choose to model the "variational" distribution $q(z,k|x)$ deliberately to mirror the structure of the prior, which differs from, for example, Makhzani et al. (2016) who assume conditional independence between $z|x$ and $k|x$.

Since the constrained infimum is intractable, a relaxed version of $W_2^\dagger$ is introduced as follows:

$$\begin{aligned}\widetilde{W}_2^\dagger\big(p_{\text{data}}, p_G\big)^2 \;=\; &\inf_{q(z,k|x)\in\mathcal{P}_{\mathcal{Z}\times\mathcal{K}}} \mathbb{E}_{p_{\text{data}}(x)}\, \mathbb{E}_{q(z,k|x)}\, \mathbb{E}_{p_G(y|z)}\, \big[\, ||x-y||_2^2\, \big] \\ &+ \lambda\, \mathcal{D}\Big(\, \mathbb{E}_{p_{\text{data}}(x)}\big[q(z,k|x)\big] \,\Big|\Big|\, p_C(z|k)\, p_D(k)\,\Big)\end{aligned} \tag{4}$$

which is equivalent to the original distance when $\lambda \to \infty$. This equivalence requires only that $\mathcal{D}$ be a divergence. As in Tolstikhin et al. (2018), we use the Maximum Mean Discrepancy (MMD) with a mixture of inverse multiquadratic (IMQ) kernels with various bandwidth C^i. The MMD is a distance on the space of densities and has an unbiased U-estimator (Gretton et al., 2012a). Explicitly, if $\boldsymbol{k}$ is a reproducing positive-definite kernel and is characteristic, then the MMD associated to $\boldsymbol{k}$ is given by

$$\text{MMD}(q\,||\,p) = \mathbb{E}_{z_1,z_2\sim q}[\boldsymbol{k}(z_1,z_2)] + \mathbb{E}_{z_1,z_2\sim p}[\boldsymbol{k}(z_1,z_2)] - 2\mathbb{E}_{z_1\sim q, z_2\sim p}[\boldsymbol{k}(z_1,z_2)] \tag{5}$$

IMQ kernels have fatter tails than the classic radial basis function kernels, proving more useful early in training when the encoder has not yet learned to match the aggregated posterior with the prior. The choice of bandwidth for the kernel can be fickle, so we take a mixture of kernels with bandwidths $C^i \in \{10^j, 2\times 10^j, 5\times 10^j \,|\, j\in\{-2,\ldots,2\}\}$ reducing the sensitivity on any one choice (see Dziugaite et al. (2015); Gretton et al. (2012b); Li et al. (2015)).

Given the discrete latent in our model, we cannot directly use Equation 4 with the MMD. Instead we integrate out the discrete latent variable in Equation 3, arriving at our GM-WAE objective function:

$$\begin{aligned}\widetilde{W}_2^\dagger\big(p_{\text{data}}, p_G\big)^2 \;=\; &\inf_{\sum_k q(z,k|x)\in\mathcal{P}_{\mathcal{Z}}} \mathbb{E}_{p_{\text{data}}(x)}\, \mathbb{E}_{\sum_k q(z,k|x)}\, \mathbb{E}_{p_G(y|z)}\, \big[\, ||x-y||_2^2\, \big] \\ &+ \lambda\, \text{MMD}\Big(\, \mathbb{E}_{p_{\text{data}}(x)}\Big[\sum_k q(z,k|x)\Big] \,\Big|\Big|\, \sum_k p_C(z|k)\, p_D(k)\,\Big)\end{aligned} \tag{6}$$

This allows us to compute the MMD between two continuous distributions, where it is defined.

As mentioned in Section 1, VAEs have the disadvantage that deterministic generative models cannot be used; this is not the case for the Wasserstein distance. Thus we parametrize the generative density $p_G(x|z)$ as a deterministic distribution $x|z = g_\theta(z)$ where g_θ is a mapping from the latent to the data space specified by a deep neural network with parameters θ. This parametrization allows the minimization the objective function using stochastic gradient descent with automatic differentiation.

To enable gradient-based minimization for the infimum in Equation 6, we parametrize $q(z,k|x) = q_C(z|k,x)\, q_D(k|x)$ with neural networks. We take $q_C(z|k,x)$ to be a Gaussian with diagonal covariance for each k, mirroring the prior, and use the reparameterization trick (Kingma & Welling, 2014; Rezende et al., 2014) to compute gradients. In order to avoid back propagating through discrete variables, the expectation over the distribution $q_D(k|x)$ is computed exactly. It could be computed by sampling using standard techniques (Brooks et al. (2011); Jang et al. (2017); Maddison et al. (2017)).

As previously mentioned, the weakness of the topology induced by the Wasserstein distance on the space of distributions may enable the GM-WAE to overcome the VAE training issues presented in Section 2.1. With the objective in hand, a more precise argument can be made to support this claim.

Recall from Section 2.1 that the problem with the GM-VAE was that the objective function demands the various distributions to be optimized at the individual data-point level. For example, the $D_{\text{KL}}(q_D(k|x)||p_D(k))$ term in Equation 2 breaks off completely and becomes irrelevant due to its size. This causes the $q_D(k|x)$ distribution to shut off k values early, which becomes galvanized as the generative model learns.

However, in posing the problem in terms of the most efficient way to move one distribution p_G onto another p_{data}, via the latent distribution $q(z,k|x)$, the Wasserstein distance never demands the similarity of two distributions conditioned per data point. Indeed, the $\mathbb{E}_{p_{\text{data}}}$ in Equation 6 is inside both the infimum and the divergence $\mathcal{D}$. We expect that "aggregating" the posterior as such will allow $q(z, k|x)$ (in particular, $q_D(k|x)$) the flexibility to learn data-point specific information while still matching the prior on aggregate. Indeed, it is also found in Makhzani et al. (2016) that using an adversarial game to minimize the distance between an aggregated posterior and the prior is successful at unsupervised training on MNIST with a discrete-continuous latent-variable model.

3 Results

In this work we primarily seek to show the potential for OT techniques to enable the training of GM-LVMs. Thus, we use relatively simple neural network architectures and train on MNIST.

We use a mixture of Gaussians for the prior, with 10 mixtures to represent the 10 digits in MNIST and a non-informative uniform prior over these mixtures. Namely, for each $k \in \{0, \dots, 9\}$:

$$p_D(k) = \frac{1}{10}, \qquad p_C(z|k) = \mathcal{N}(z; \mu_k^0, \sigma_k^0 I) \tag{7}$$

where the μ_k^0 and σ_k^0 represent the mean and covariance of each mixture and are fixed before training. We found that choosing $\dim(z) = 9$ worked well. We choose the μ_k^0 to be equidistant and for each k, $\sigma_k^0 = \sigma^0$ is chosen identically in order to admit $\approx 5\%$ overlap between the 10 different modes of the prior (i.e., the distance between any pair of means $\mu_{k_1}^0$ and $\mu_{k_2}^0$ is $4\sigma^0$).

For the variational distribution, we take $q(z, k|x) = q_C(z|k, x)\, q_D(k|x)$ with

$$q_D(k|x) = \pi_k(x), \qquad q_C(z|k, x) = \mathcal{N}\Big(z; \mu_k(x), \text{diag}\big(\sigma_k(x)\big)\Big) \tag{8}$$

where each component is parametrized by a neural network. For $\pi_k(x)$ a 3-layer DCGAN-style network (Radford et al., 2015) is used with largest convolution layer composed of 64 filters. The Guassian networks $\mu_k(x), \sigma_k(x)$ are taken to be 32-unit two-hidden-layer dense networks. Finally, for the generative model, we take $p_G^\theta(x|z)$ to be deterministic with $x|z = g_\theta(z)$, using a 3-layer DCGAN-style network with smallest transpose convolution layer composed of 128 filters. All the convolutional filters have size 5×5 except for the last layer which has size 1×1.

We use batch normalisation (Ioffe & Szegedy, 2015), ReLU activation functions (Glorot et al., 2011) after each hidden layer and Adam for optimization (Kingma & Ba, 2015) with a learning rate of 0.0005. We find that $\lambda = 450$ works well, although the value of λ does not impact performance appreciably as long as it is larger than a few hundred. The (μ_k, σ_k) networks are pretrained to match the prior moments, which accelerates training and improves stability (this was also done for GM-VAE in Section 2.1).

3.1 Reconstructions and samples

Our implementation of GM-WAE is able to reconstruct MNIST digits from its latent variables well. In Figure 2a example data points from the held-out test set are shown on the odd rows, with their reconstructions on the respective rows below. The encoding of the input points is a two step process, first determining in which mode to encode the input via the discrete latent, and then drawing the continuous encoding from the corresponding mode.

Samples from the GM-WAE are shown in Figure 2b and 2c. Since the discrete prior $p_D(k)$ is uniform, we can sample evenly across the ks in order from 0 through 9, while still displaying representative samples from $p(z, k) = p_C(z|k)p_D(k)$. Again, this shows how the GM-WAE learns to leverage the structure of the prior, whereas the GM-VAE results in the collapse of the several modes of the prior.

GM-WAE has a smooth manifold structure in its latent variables. In Figure 3a the reconstructions of a linear interpolation with uniform step size in the continuous latent space is shown between pairs of data points. This compares similarly to other WAE and VAE approaches to MNIST. In Figure 3b a linear interpolation is performed between the prior mode μ_6^0, and the other nine prior modes $\mu_{k\neq 6}^0$. This not only shows the smoothness of the learned latent manifold in all directions around a single mode of the prior, but also shows that the variatonal distribution has learned to match the modes of the prior. As one would hope given the suitability of a 10-mode GM-LVM to MNIST, almost every

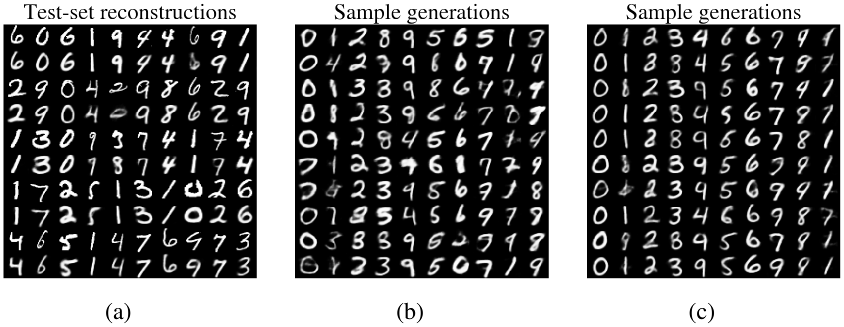


Figure 2: Shown in (a) are reconstructions of held-out data from the inferred latent variables. The first, third, etc, rows are the raw data, and the rows below show the corresponding reconstructions. Digit samples $x \sim p_G(x|z)\,p_C(z|k)$ for each discrete latent variable k are shown in (b) as well as those samples closer to each mode of the prior in (c). The samples in (c) come from z values sampled from Gaussians identical to $p_C(z|k)$, except with standard deviation scaled down by $3/5$.

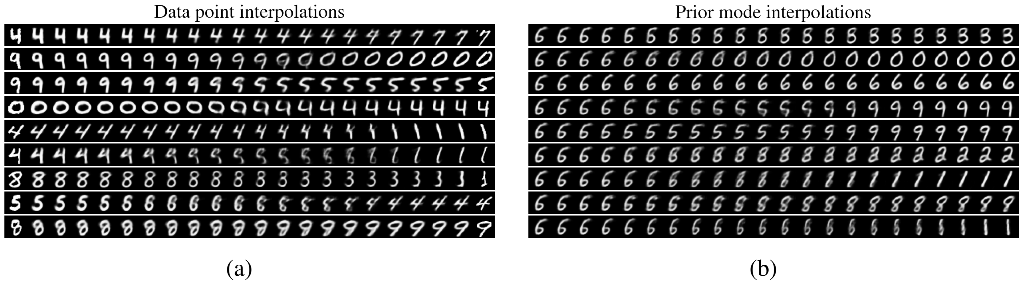


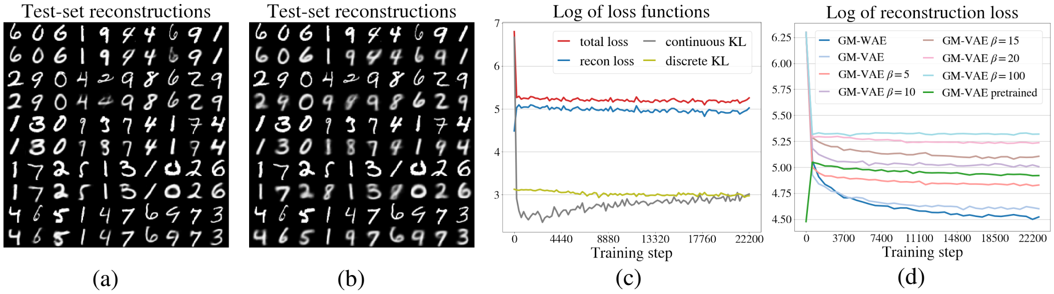
Figure 3: Reconstructions from linear interpolations in the continuous latent space between two data points (a), and between the prior mode μ^0_6, and the other nine prior modes $\mu^0_{k\neq 6}$ (b). In (a), the true data points are shown in the first and last column next to their direct reconstructions.

mode of the prior now represents a different digit. This level of control built into the prior requires not only a multi-modal prior, but also a training procedure that actually leverages the structure in both the prior and variational distribution, which seems to not be the case for VAEs (see Section 2.1).

The quality of samples from our GM-WAE is related to the ability of the encoder networks to match the prior distribution. Figure 2c and 3b demonstrate that the latent manifold learned is similar to the prior. Near the modes of the prior the samples are credible handwritten digits, with the encoder networks able to capture the structure within each mode of the data manifold (variation within each column) and clearly separate each different mode (variation between rows).

We have argued that the VAE objective itself was responsible for the collapse of certain k values in the discrete variational distribution, and that the per-data-point nature of the KL played a significant role. To test this hypothesis, and to compare directly our trained WAE with the equivalent VAE discussed in Section 2.1, we initialize the VAE with the parameters of the final trained WAE, and train it according to the VAE objective. At initialization, the VAE with trained WAE parameters produces high quality samples and reconstructions (Figure 4a). However, after a few hundred iterations, the reconstructions deteriorate significantly (Figure 4b), and are not improved with further training.

The learning curves over the period of training between Figure 4a and 4b are shown in Figure 4c, where the cause of the performance deterioration is clear: the continuous KL term in the VAE objective is multiple orders of magnitude larger than the reconstruction term, causing optimization to sacrifice reconstruction in order to reduce this KL term. Of course, the approximate posterior aggregated over the data will not be far from the prior as that distance is minimized in the WAE objective. However, this is not enough to ensure that the continuous KL term is small for every data point individually. It is thus the per-data-point nature of the KL in the VAE objective that destroys the reconstructions. Indeed, in order to minimize the per-data-point KL term in the GM-VAE objective,



(a) (b) (c) (d)

Figure 4: (a) Reconstructions for an untrained VAE initialized with same parameters as our trained WAE. (b) Those same reconstructions after a few dozen thousands training steps according to the VAE objective. (c) Learning curves from an untrained VAE initialized with same parameters as our trained WAE. (d) Reconstruction loss for different VAE variations.

$q_C(z|k,x)$ is forced toward the mean μ_k^0 for every x, causing it to lose much of its x dependence. This can be seen in Figure 4b where the reconstructions are less customized and blurrier.

To compare the performance of GM-WAE against GM-VAE more quantitatively, we directly compare the reconstruction loss from the VAE objective (the first term on the right hand side of Equation 2). Strictly speaking, this quantity is ill-defined for the GM-WAE, as the generative model is chosen to be deterministic. Instead we simply use the values returned by the GM-WAE generative model as if they were the Bernoulli mean parameters of the GM-VAE (Kingma et al., 2014). These reconstruction loss curves are shown Figure 4d. Also shown are the reconstruction losses for the GM-VAE with various rescaling factors β in front of the KL terms of Equation 2. This rescaled KL term is inspired by both Higgins et al. (2016), which studies the impact of rescaling the KL term in VAEs, as well as by the WAE objective itself where λ plays the role of a regularization coefficient. While, the GM-WAE is not trained to minimized this reconstruction loss, it actually achieves the best results. This shows that GM-WAE performs better at reconstructing MNIST digits than its VAE counterpart, as measured by the VAE's own reconstruction objective.

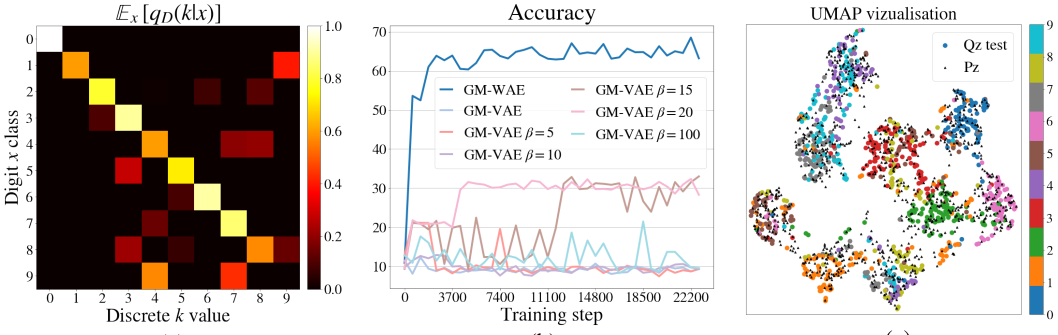
We also show in Figure 4d the reconstruction curve of a GM-VAE initialized with trained GM-WAE parameters. This echoes the previous discussion concerning the deterioration of the reconstructions in GM-VAEs due to the per-data-point KL term. In Figures 4c and 4d, the GM-VAE initialized with trained GM-WAE parameters uses a rescaling factor $\beta = 10$ for visualization purposes. The same phenomenological behavior is observed with no rescaling factor, just less visually pronounced.

Overall, our results for GM-WAE are qualitatively competitive with other approaches (Tolstikhin et al., 2018), despite a relatively low-complexity implementation. Furthermore, GM-WAE offers more control over generation and inference due to its latent-variable structure, which cannot be achieved with the GM-VAE objective.

3.2 Latent variable fidelity

We have shown that the GM-WAE is able to both reconstruct data and generate new samples meaningfully from the prior distribution. We now turn to studying the variational distributions directly, including with how much fidelity a given class of digits is paired with a given discrete latent.

Consider first the discrete distribution $q_D(k|x)$ shown in Figure 5a, where $\mathbb{E}_{\{x|\text{label}(x)=\ell\}} q_D(k|x)$ is shown in row ℓ. From the staircase structure, it is clear that this distribution learns to approximately assign each discrete latent value k to a different class of digit. However, it does not do so perfectly. This is expected as the GM-WAE seeks only to reconstruct the data from its encoding, not to encode it in any particular way. This does not mean GM-WAE is failing to use its discrete latent effectively. Indeed, when comparing Figure 2c and Figure 5a, a meaningful source of overlap between different values of k and a single digit class can be seen. For example, in Figure 5a the digit 5 is assigned partially to $k = 3$ and $k = 5$. In Figure 2c, 5s drawn with a big-round lower loop are similar to digit 3 and 5s with a small loop and long upper bar are assigned to another cluster corresponding to digit 5. A similar discussion holds for 8s and 9s.



(a)

(b)

(c)

Figure 5: Visualization of the variational distributions. (a) shows $\mathbb{E}_{\{x|\text{label}(x)=\ell\}} q_D(k|x)$ in row ℓ. (b) shows the accuracy as a function of the training steps for our method and the same VAE variations than Figure 4d. (c) shows $z|x \sim \sum_k q_C(z|k,x)q_D(k|x)$ dimensionally reduced using UMAP (McInnes & Healy, 2018). 1000 encoded test-set digits and 1000 samples from the prior are used. Encoded points are colored by their digit label.

To assess the digit-class fidelity of the discrete encoder more quantitatively, we calculate the accuracy of the digit-class assignment according to $q_D(k|x)$. To assign a digit-class label to each k value, we follow a similar protocol to that of Makhzani et al. (2016): we assign the digit-class label to the k value that maximizes the average discrete latent for that class, in decreasing order of that maximum. Figure 5b shows the resulting accuracy throughout training. Our GM-WAE achieves an accuracy on the held-out test set just shy of 70%. The corresponding accuracies for the GM-VAE variations considered in Figure 4 are also shown. The best performing GM-VAE with a scaling factor of $\beta = 20$ achieves approximately 30%. This shows again the difficulty of the GM-VAE to capture meaningful structure in the data. For reference, basic K-means clustering (MacQueen, 1967) achieves 50-60%, and Makhzani et al. (2016) achieve 90% (using 16 discrete classes, and substantially different model and training procedure).

Another way to study the latent variable structure of GM-WAE is to consider dimensionally reduced visualizations of the continuous latent z. In Figure 5c such a visualization is shown using UMAP (McInnes & Healy, 2018). Distinct clusters can indeed be seen in the prior and in the samples from $q_C(z|k,x)$. Though the clusters of $z \sim q_C(z|k,x)$ do not fully align with those from the prior $z \sim p_D(z|k)$, they maintain significant overlap. Samples from $q_C(z|k,x)$ in Figure 5c are colored according to the true digit labels, and show how GM-WAE learns to assign digits to the different clusters. In particular, the 7 / 9 cluster is clearly overlapping, as seen in Figures 5a, 2b and 2c.

We have see that the GM-WAE model is highly suited to the problem under study. It reconstructs data and provides meaningful samples, it effectively uses both discrete and continuous variational distributions, all while maintaining close proximity between the variational distribution and the prior.

4 Conclusions

We studied an unsupervised generative model with a mixture-of-Gaussians latent variable structure, well suited to data containing discrete classes of objects with continuous variation within each class. We showed that such a simple and critical class of models fails to train using the VAE framework, in the sense that it immediately learns to discard its discrete-latent structure. We further exposed the root cause of this phenomenon with empirical results. We then put to the test the abstract mathematical claim that the Wasserstein distance induces a weaker topology on the space of distributions by attempting to train the same mixture-of-Gaussians model in the WAE framework. We found the Wasserstein objective is successful at training this model to leverage its discrete-continuous latent structure fully. We provided promising results on MNIST and demonstrated the additional control available to a highly structured model with both discrete and continuous latent variables. We hope this motivates further study of the exciting but nascent field of Optimal Transport in generative modeling.

REFERENCES

M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, 2017.

O. Bousquet, S. Gelly, I. Tolstikhin, C. J. Simon-Gabriel, and B. Schölkopf. From optimal transport to generative modeling: the VEGAN cookbook. *arXiv:1705.07642*, 2017.

S. Brooks, A. Gelman, G. Jones, and M. Xiao-Li. *Handbook of Markov chain Monte Carlo*. CRC press, 2011.

N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep unsupervised clustering with Gaussian mixture variational autoencoders. *arXiv:1611.02648*, 2016.

G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Conference on Uncertainty in Artificial Intelligence*, 2015.

S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K. Kavukcuoglu, and G. E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, 2016.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2011.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems*, 2014.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. In *Journal of Machine Learning Research*, 2012a.

A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems*, 2012b.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

M. D. Hoffman and M. J. Johnson. ELBO surgery: yet another way to carve up the variational evidence lower bound. In *NIPS Workshop on Advances in Approximate Bayesian Inference*, 2016.

S. Ioffe and C. Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

M. J. Johnson, D. Duvenaud, A. B. Wiltschko, S. R. Datta, and R. P. Adams. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, 2016.

D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, 2014.

D. Lawson, G. Tucker, C.-C. Chiu, C. Raffel, K. Swersky, and N. Jaitly. Learning hard alignments with variational inference. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.

Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, 2015.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, 1967.

C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: a continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016.

L. McInnes and J. Healy. Umap: uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426*, 2018.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2015.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

P. K. Rubenstein, B. Schölkopf, and I. Tolstikhin. On the latent space of Wasserstein auto-encoders. In *Workshop track - International Conference on Learning Representations*, 2018.

I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

A. Van den Oord, O. Vinyals, and K. kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017.

C. Villani. *Optimal Transport: Old and New*. Springer Berlin Heidelberg, 2008.