Is the Discrete VAE's Power Stuck in its Prior?

Haydn Jones
Department of Computer Science and Engineering
New Mexico Institute of Mining and Technology
haydn.jones@student.nmt.edu

Juston Moore
Advanced Research for Cyber Systems
Los Alamos National Laboratory
jmoore01@lanl.gov

Abstract

We investigate why probabilistic neural models with discrete latent variables are effective at generating high-quality images. We hypothesize that fitting a more flexible variational posterior distribution and performing joint training of the encoder, decoder, and prior distribution should improve model fit. However, we find that modifying the training procedure for the well-known vector quantized variational autoencoder (VQ-VAE) leads to models with lower marginal likelihood for held-out data and degraded sample quality. These results indicate that current discrete VAEs use their encoder and decoder as a deterministic compression bottleneck. The distribution-matching power of these models lies solely in the prior distribution, which is typically trained after clamping the encoder and decoder.

1 Introduction

While generative adversarial networks (GANs) excel at synthesizing realistic-looking natural images [Karras et al., 2020, Park et al., 2019], probabilistic models including variational autoencoders (VAEs) [Kingma and Welling, 2014, Higgins et al., 2017] and normalizing flows [Dinh et al., 2015, Kingma and Dhariwal, 2018, Ho et al., 2019] generally lag behind as measured by qualitative assessments of sample images. However, GANs do not offer many attractive features that probabilistic models naturally provide, such as efficient encoding of inputs into a latent space representation and likelihood estimation for held-out data. Indeed, while probabilistic models must posit some probability mass over every possible image, there is no constraint in the GAN architecture ensuring that all images correspond to a representation in latent space; due to mode collapse, the typical samples from a GAN regularly under-represent the training data distribution.

The current state-of-the-art VAE, as judged by image sample quality, is the vector quantized variational autoencoder (VQ-VAE). VQ-VAEs introduce discrete latent variables and train a powerful autoregressive prior to capture high-level correlations in this discrete space [van den Oord et al., 2017]. A recent extension, dubbed VQ-VAE-2, competes with state of the art GANs as measured by Inception Score and Fréchet Inception Distance [Razavi et al., 2019b] and improves sample image diversity and resolution by pairing a hierarchical encoder/decoder with a hierarchical autoregressive prior. However, VQ-VAEs are trained in two steps. The encoder and decoder networks are first trained to minimize reconstruction error, like a traditional bottleneck autoencoder. The learned encoding from inputs to latent variables is deterministic. Next, a prior distribution is trained to model the discrete latent variables.

In this paper, we investigate the extent to which the VQ-VAE's model fit and sample quality depend on this specific training procedure. We hypothesized that jointly training the encoder/decoder with the prior would improve the model fit. Surprisingly, we find that "smoothing" the latent space through sampling joint training leads to poorer model fit and lower sample image quality. Unlike continuous VAEs, which often perform well on interpolation tasks, our results show that the distribution of latent codes in the VQ-VAE is highly structured, and most latent codes do not decode to natural images.

1st I Can't Believe It's Not Better Workshop (ICBINB@NeurIPS 2020), Vancouver, Canada.

2 Background

Variational autoencoders (VAEs) are a flexible class of probabilistic latent variable models that leverage neural networks to learn nonlinear mappings between latent variables and observed variables [Kingma and Welling, 2014, Rezende et al., 2014]. A VAE posits a joint distribution over observed variables $\boldsymbol{x}$ and latent variables $\boldsymbol{z}$: $p(\boldsymbol{x}, \boldsymbol{z}; \theta, \omega) = p(\boldsymbol{x}|\boldsymbol{z}; \theta)p(\boldsymbol{z}; \omega)$. The flexibility in VAEs comes from training a (typically deterministic) mapping $\text{dec}_\theta(\boldsymbol{z})$. The VAE likelihood $p(\boldsymbol{x}|\boldsymbol{z}; \theta)$ is then parameterized by this mapping; for continuous observations, a commonly-used modeling choice is a Gaussian distribution centered on the decoder output $\mathcal{N}(\text{dec}_\theta(\boldsymbol{z}), \sigma^2 I)$ with fixed variance σ^2.

VAEs are fit by maximizing the marginal likelihood, or *evidence*, $p(\boldsymbol{x}; \theta)$. Since this quantity is usually intractable, approximate inference is performed by positing a variational posterior distribution $q(\boldsymbol{z}|\boldsymbol{x}; \phi)$. Rather than optimizing a variational distribution for each observation $\boldsymbol{x}$, VAEs perform *amortized inference* by training an encoder $\text{enc}_\phi(\boldsymbol{x})$, which in turn parameterizes $q(\boldsymbol{z}|\boldsymbol{x}; \phi)$. The model is trained using the evidence lower bound (ELBO):

$$p(\boldsymbol{x}; \boldsymbol{z}; \theta, \omega) \geq \mathop{\mathrm{E}}_{\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x}; \phi)} \left[\log p(\boldsymbol{x}|\boldsymbol{z}; \theta) - D_{\text{KL}}(q(\boldsymbol{z}|\boldsymbol{x}; \phi)||p(\boldsymbol{z}; \omega))\right] \tag{1}$$

The outer expectation is approximated via Monte Carlo samples from the variational posterior q and the key "reparameterization trick" allow gradients to flow through the sampling operation [Kingma and Welling, 2014, Rezende et al., 2014].

2.1 Vector Quantized VAEs

Our work focuses on extending the VQ-VAE, a VAE with discrete latent variables introduced by van den Oord et al. [2017]. In this model, each latent variable $z_i \in \{1, \ldots, K\}$ indexes into a codebook $C \in \mathbb{R}^{K \times D}$ to produce a row vector C_{z_i}. The notation $C_{\boldsymbol{z}}$ will refer to a matrix of codebook elements indexed by latent variables in $\boldsymbol{z}$, where the i^{th} row in $C_{\boldsymbol{z}}$ is C_{z_i}. The full model is specified by setting a uniform categorical prior and an isotropic Gaussian likelihood:

$$p(z_i = k) = \frac{1}{K} \tag{2}$$

$$p(\boldsymbol{x}|\boldsymbol{z}; \theta) = \mathcal{N}\left(\text{dec}_\theta(C_{\boldsymbol{z}}), \sigma^2 I\right) \tag{3}$$

The authors fit this model using a maximum a posteriori (MAP) estimate for the variational posterior by quantizing the output of the encoder to the nearest codebook vector:

$$q(z_i = k \mid \boldsymbol{x}; \phi) = \begin{cases} 1 & \text{if } k = \arg\min_j ||\text{enc}_\phi(\boldsymbol{x}) - C_j||_2 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Because the prior distribution is uniform and the posterior is a categorical distribution parameterized by a one-hot probability vector, the KL-divergence term in ELBO is a constant ($\log K$). Moreover, the ELBO expectation term is over a deterministic distribution q, so the model is simply trained by maximizing the likelihood $\max_{\theta, \phi} \log p(\boldsymbol{x}|\boldsymbol{z} = \text{dec}_\theta(\boldsymbol{x}); \theta)$. The sampling and quantization operations in Eq 4 are not differentiable, so the authors apply a "straight-through" gradient estimator, which propagates the gradients with respect to the codebook element C_{z_i} through to the outputs of the decoder $\text{enc}_\phi(\boldsymbol{x})_i$. If the encoder output and codebook element are sufficiently close, this is a close approximation, but there is no bound on the bias of the gradients with this method. The VQ-VAE introduces additional terms to the loss function to move codebook entries and outputs of the encoder toward one another during training.

After training the encoder/decoder end-to-end, a flexible autoregressive prior $p(\boldsymbol{z}; \omega)$ is fit to the latent codes, obtained by the deterministic mapping in Eq 4. The original VQ-VAE paper applies a PixelCNN model for images [van der Oord et al., 2016]. In this work, we will apply a PixelCNN++ [Salimans et al., 2017]. Both the PixelCNN and PixelCNN++ are autoregressive convolutional neural networks that are trained efficiently using masked convolutions.

3 Model Variants

We make two changes to the VQ-VAE's inference procedure to address approximations made in the original paper. First, instead of fitting a MAP estimate during variational inference, we fit a

Variational Posterior	Prior	Prior Training
Quantized	PixelCNN++	Marginal
Quantized	PixelCNN++	Joint
Quantized	Uniform	N/A
Categorical	PixelCNN++	Marginal
Categorical	PixelCNN++	Joint
Categorical	Uniform	N/A

Table 1: Summary of model variants.

mean-field categorical distribution. Second, instead of fitting the encoder/decoder with a uniform prior using a straight-through gradient approximation, we train the model jointly end-to-end. The model variants we explore are shown in Table 1. In addition to the variants discussed below, we train baseline models with a Uniform categorical prior; this prior is never fine-tuned. The first model "Quantized, PixelCNN++, Marginal" is the original VQ-VAE.

3.1 Quantization vs. Categorical Posterior

The first question we ask is whether the quantization step in the VQ-VAE is responsible for the model's success, or whether an encoder can perform equivalently by directly learning to parameterize a categorical distribution. Because the variational posterior is simply a tool for amortized inference, we expect this equally-expressive alternative model to perform comparably. For simplicity, we choose a mean-field approximation $q(\boldsymbol{z} \mid \boldsymbol{x}; \phi) = \prod_{i=1}^{N} q_i(z_i; \boldsymbol{x}, \phi)$ where each q_i is a categorical distribution.

The Quantized model uses Euclidean distance between outputs of the encoder $\text{dec}_\theta(\boldsymbol{x})$ and codebook entries, similar to the original VQ-VAE. Our "Quantized, PixelCNN++, Marginal" model is equivalent to the VQ-VAE with regularization parameter $\beta = 0.25$. All other Quantized models use a categorical variational distributions parameterized by distance:

$$q(z_i = k \mid \boldsymbol{x}; \phi) = \frac{\exp\left(||\text{enc}_\phi(\boldsymbol{x})_i - C_k||_2\right)}{\sum_{k'=1}^{K} \exp\left(||\text{enc}_\phi(\boldsymbol{x})_i - C_{k'}||_2\right)} \tag{5}$$

We use the notation $\text{enc}_\phi(\boldsymbol{x})_i$ to refer to the encoder's prediction for the i^{th} latent variable.

The Categorical models directly estimate a categorical distribution without reference to the codebook:

$$q(z_i = k \mid \boldsymbol{x}; \phi) = \text{softmax}(\text{enc}_\phi(\boldsymbol{x})_i)_k \tag{6}$$

3.2 Marginal vs. Joint Training

The second question we ask is whether the joint VQ-VAE is benefiting from the VAE framework at all. Since the model is initially trained without any sampling operations, it is possible that the VQ-VAE's capability to model complex distributions, such as natural images, is entirely conferred by the subsequently-trained prior distribution.

Similar to the VQ-VAE loss function, we train the encoder/decoder in the Marginal models to maximize the expected likelihood term from ELBO: $\text{E}_{\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x};\phi)} \log p(\boldsymbol{x} \mid \boldsymbol{z}; \theta)$. This model can equivalently be viewed as a bottleneck autoencoder with sampling at the bottleneck layer, or as an amortized expectation-maximization algorithm to perform maximum likelihood estimation $\max_{\phi,\theta} \text{E}_{z \sim q(\boldsymbol{z}|\boldsymbol{x};\phi)} \log p(\boldsymbol{x} \mid \boldsymbol{z}; \theta)$ [Roy et al., 2018]. Once the encoder/decoder converge, we train a PixelCNN++ to minimize the KL divergence: $\min_\omega D_{\text{KL}}(q(\boldsymbol{z}|\boldsymbol{x}; \phi)||p(\mathbf{z}; \omega))$[1].

Our Joint models are straightforwardly trained to maximize ELBO (Eq 1). The model is jointly trained by allowing gradients to flow between the encoder, decoder, and prior. We fit the prior distribution by optimizing its hyperparameters ω (i.e. empirical Bayes). We use the Gumbel softmax trick from Jang et al. [2017] to reparameterize the categorical sampling operation. For our mean-field

[1]The prior training objective is equivalent to cross entropy loss because $H(q)$ is constant with respect to ω.

variational posterior, this sampling is computed by:

$$\boldsymbol{z}_i = \text{softmax}\left(\frac{\text{enc}_\phi(\boldsymbol{x})_i + \boldsymbol{g}}{\tau}\right) \tag{7}$$

where $\text{enc}_\phi(\boldsymbol{x})_i$ represents the K-length vector of logits for the i^{th} latent variable, and $\boldsymbol{g}$ is a K-length vector where $g_k \sim \text{Gumbel}(0, 1)$. As $\tau \to 0$, $\boldsymbol{z}_i$ approaches a one-hot vector representing a draw from the distribution $q(z_i = k; \phi) = \text{softmax}(\text{enc}_\phi(\boldsymbol{x})_i)$.

Rather than indexing into the codebook, the joint model matrix-multiplies the Gumbel-softmax samples into the codebook matrix C. We begin training with a relatively high value of τ, resulting in a weighted mixture of codebook elements being passed to the decoder. As the temperature is annealed, the samples for z_i approach one-hot vectors. In our experiments we found that annealing τ to a value of 0.1 over the course of training brought samples sufficiently close to one-hot vectors.

4 Related Work

Sønderby et al. [2017] develop the most closely-related model to ours in the literature. Rather than jointly training a flexible prior, however, this work assumes that the prior distribution is uniform. Our "Categorical, Uniform" model is similar to the GS-Soft model described in the paper. Additionally, Roy et al. [2018] develop an EM inference algorithm for the VQ-VAE model.

Kaiser et al. [2018] demonstrates that VQ-VAE model fit degrades as the length of quantization vectors grows, a phenomenon they call "index collapse." Rather than introducing sampling to avoid the problem of index collapse, the authors develop a model to quantize sub-vectors of the encoder's output using multiple codebooks. It does not appear that our Joint models suffer from index collapse (see Section 5.2).

Similar to—yet distinct from—our findings, many recent works have found that VAE models with a flexible decoder experience "posterior collapse", whereby a powerful autoregressive decoder completely ignores outputs of the encoder. The lack of a gradient from the decoder leads the variational posterior distribution q to collapse to match the prior distribution [Lucas et al., 2019, Razavi et al., 2019a]. In the models we investigate, we restrict our attention to decoders that are convolutional neural networks, so our findings cannot be explained by posterior collapse.

Several recent works have tried to improve the quality of sample images by increasing the expressiveness of VAE models. In particular, Rezende and Mohamed [2015] replaces the variational distribution in the encoder with a normalizing flow, and Gulrajani et al. [2017] uses a powerful autoregressive model as the decoder. Both approaches are complementary to our model's joint training of an expressive autoregressive prior.

5 Experiments

We present a qualitative and quantitative analysis of the model variants described in Section 3. While the ImageNet samples in the original VQ-VAE paper appear high quality, it becomes clear upon close inspection that the model is only matching general textures (for example, most of the animals are missing heads). We study facial images from the Large-scale CelebFaces Attributes (CelebA) dataset [Liu et al., 2015] because we believe the quality of samples is more immediately apparent to human evaluators [Seyama and Nagayama, 2007]. All images are downscaled to 64x64 pixels, and intensities are scaled to the range $\left[0, \frac{255}{256}\right]$. We dequantize all images in order to be able to calculate discrete bits per dimension by adding uniform noise $u \sim \mathcal{U}\left(0, \frac{1}{256}\right)$ [Kingma and Dhariwal, 2018].

In all model variants, we take one Monte Carlo sample per data point from the variational posterior distribution during training. The models use the same encoder and decoder structure, consisting of 4 residual layers with 128 feature maps in both the encoder and decoder. We set the number of codebook elements $K = 256$. We use strided convolutions to down/up sample the image spatial resolution. Our PixelCNN++ prior is slightly modified from the original implementation, using softmax output rather than a logistic mixture and 3 residual layers with 64 channels instead of the original 5 residual layers . Models are trained using the Adam optimizer for 500 epochs with a batch size of 128 and a learning rate of 3e-4. The Gumbel softmax temperature is initialized at 1.0 and is annealed exponentially to 0.1 over the course of training. The model was trained 10 times on random

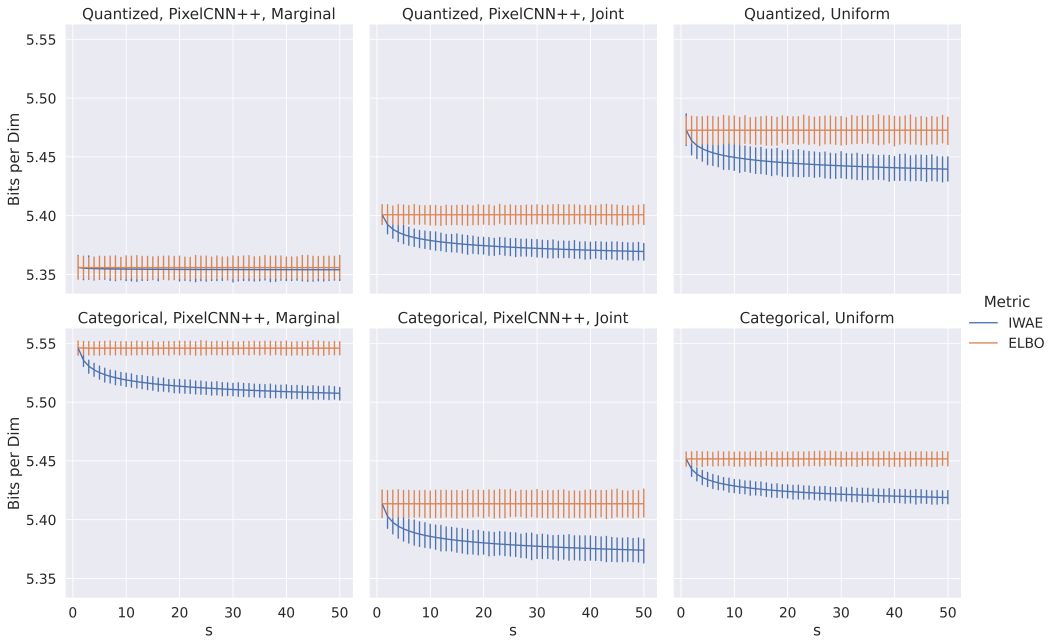

Figure 1: Model evidence for held out images from the CelebA dataset, shown as bits per dimension. Lower is better. We compute lower bounds on the model evidence across a varying number Monte Carlo samples, S, from the variational posterior distribution. Error bars indicate standard deviation across 10 random train/test splits.

train/test splits to characterize variance; 70% of the dataset was used for training, 15% for testing, and 15% for validation.

5.1 Marginal Likelihood of Held Out Data

We apply the IWAE estimator described in Burda et al. [2016] and Cremer et al. [2018] to obtain an asymptotically tight lower bound on the model evidence; note that IWAE with 1 sample is equivalent to ELBO. We compute IWAE by drawing S Monte Carlo samples $\boldsymbol{z}^{(s)} \sim q(\boldsymbol{z}|\boldsymbol{x};\phi)$ and computing:

$$IWAE_S = \log\left(\frac{1}{S}\sum_{s=1}^{S}\frac{p(\boldsymbol{x},\boldsymbol{z}^{(s)};\theta,\omega)}{q(\boldsymbol{z}^{(s)}\mid\boldsymbol{x};\phi)}\right) \tag{8}$$

We convert our results from bounds on model evidence to bits per dimension:

$$\text{Bits per Dim} = \frac{1}{N}\sum_{i=1}^{N}\frac{-1}{W\cdot H\cdot C}\log_2(p(\tilde{\boldsymbol{x}};\theta,\omega)) - \log(a) \tag{9}$$

where $\tilde{x} \sim \mathcal{U}(x, x+a)$, W, H, and C, are the width, height, and number of color channels in the image, respectively. a measures the discretization level of the data; for 8-bit images standardized to the range $\left(0, \frac{255}{256}\right)$, a will be $\frac{1}{256}$.

Figure 1 shows average bits per dimension, along with variance across 10 random train/test splits. The original VQ-VAE model ("Quantized, PixelCNN++, Marginal") outperforms all other variants with 5.35 bits per dimension. Joint training is not too far behind, with the "Categorical, PixelCNN++, Joint" model achieving 5.39 bits per dimension. The models with Uniform priors underperform, and the "Categorical, PixelCNN++, Marginal" model, which is not trained via a valid ELBO objective, performs worst.

5.2 Index Collapse

Figure 2 shows the usage patterns across all codebook elements, for each model. A large spike at 0 frequency indicates "index collapse", whereby a large set of codebook elements are rarely used; a

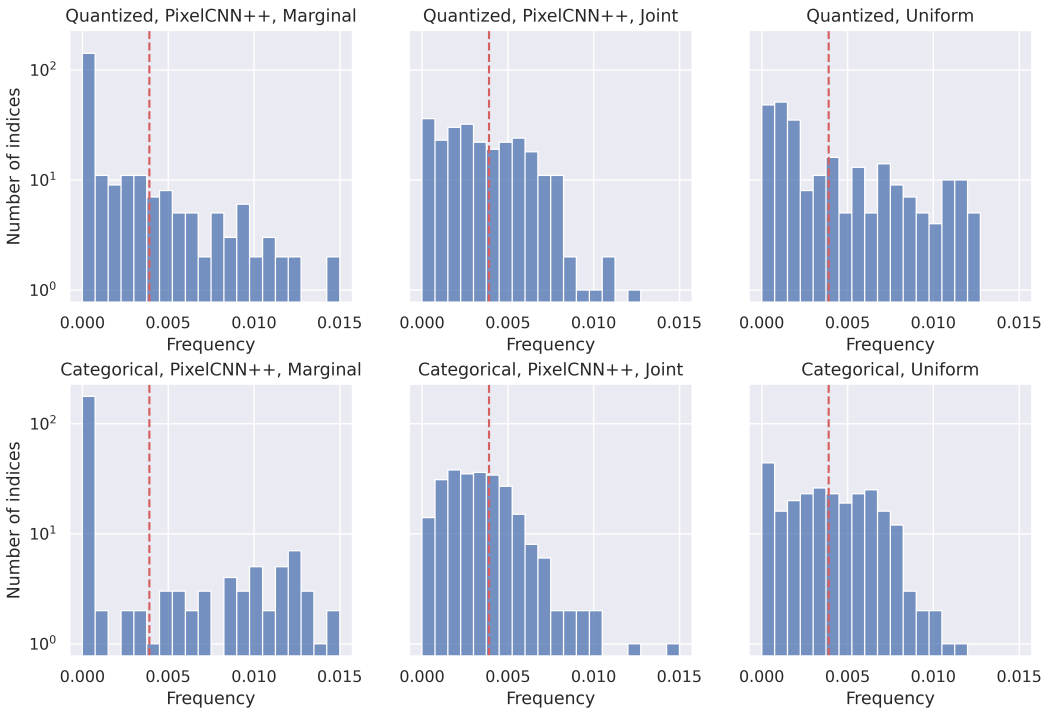

Figure 2: Histogram of codebook vector usage across models. A large number of indices with a frequency of 0 is indicative of index collapse in the model. The vertical dashed lines show the location where we would expect to see a spike if the model was experiencing posterior collapse to a uniform categorical prior.

spike at $1/256 = 0.0039$ would indicate "posterior collapse", whereby every posterior distribution q converged to a uniform prior. We see from these results that the models trained marginally do suffer from index collapse, but joint training with either a PixelCNN++ or a Uniform prior mitigates the problem. Surprisingly, the VQ-VAE has the second-largest spike at 0, but achieves the highest Bits per Dimension. No models appear to suffer from posterior collapse, as we expect since the decoder is a straightforward convolutional network.

5.3 Qualitative Evaluation

It is commonly accepted that likelihood of held-out data is not a good metric for sample quality [Theis et al., 2016]. Thus, we present samples from all model variants in Figure 3. It is clear that these samples are not comparable to the quality of state of the art GAN samples; however, the VQ-VAE and prior architecture in Razavi et al. [2019b] that generates samples competing with GANs is multi-scale and hierarchical whereas ours is not. We believe this to be the reason for the comparatively poor samples.

We see that the samples from the original VQ-VAE ("Quantized, PixelCNN++, Marginal") are moderately superior to the other models. Interestingly, the "Categorical, PixelCNN++, Marginal" model, which performed worst as measured by model evidence, produces far superior samples to the models with a Uniform prior. We note that in preliminary experiments (samples not shown), all of our model variants consistently produce high quality samples on simple images from CIFAR-10 and MNIST.

6 Conclusions

Our experiments demonstrate that modifying the inference procedure for the VQ-VAE—by introducing a more expressive posterior distribution, adding sampling, and jointly training—have a large negative impact on the model fit and small negative impact on the quality of images sampled from the

Figure 3: Samples from trained models.

model. This finding is surprising, because we would expect to achieve better model fit. After all, the original VQ-VAE's MAP posterior, i.e. a categorical distribution with all probability mass assigned to one location, could be learned by our Categorical models.

Although joint training leads to a worse model fit, it offers the possibility of learning an accurate posterior distribution. Being able to infer an accurate posterior distribution over discrete latent codes is important capability for a wide variety of inference tasks.

Our finding that learning a full variational posterior does not outperform a simple, quantized point estimate leads us to conclude that existing discrete neural models don't truly amortize posterior model inference for discrete latent variable models. Rather the encoder/decoder serve as a deterministic compression function. All of the generative modelling functionality is contained within the prior distribution. Allowing the PixelCNN++ prior to provide gradients to the encoder/decoder does not lead to improved model fit. This indicates that the PixelCNN++ is flexible enough to model a somewhat arbitrary discrete distribution produced by a pre-trained, deterministic encoder.

Acknowledgments and Disclosure of Funding

We would like to thank our anonymous reviewers for insightful suggestions and comments. Research presented in this paper was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project numbers 20200666DI and 20210043DR.

References

Y. Burda, R. B. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

C. Cremer, X. Li, and D. Duvenaud. Inference suboptimality in variational autoencoders. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1086–1094. PMLR, 2018.

L. Dinh, D. Krueger, and Y. Bengio. NICE: non-linear independent components estimation. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taïga, F. Visin, D. Vázquez, and A. C. Courville. Pixelvae: A latent variable model for natural images. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. volume 97 of *Proceedings of Machine Learning Research*, pages 2722–2730, Long Beach, California, USA, 09–15 Jun 2019.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

L. Kaiser, S. Bengio, A. Roy, A. Vaswani, N. Parmar, J. Uszkoreit, and N. Shazeer. Fast decoding in sequence models using discrete latent variables. volume 80 of *Proceedings of Machine Learning Research*, pages 2390–2399, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018.

T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems 31*, pages 10215–10224. 2018.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

J. Lucas, G. Tucker, R. B. Grosse, and M. Norouzi. Understanding posterior collapse in generative latent variable models. In *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*, 2019.

T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

A. Razavi, A. van den Oord, B. Poole, and O. Vinyals. Preventing posterior collapse with delta-vaes. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019a.

A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems 32*, pages 14866–14876. 2019b.

D. Rezende and S. Mohamed. Variational inference with normalizing flows. volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org, 2014.

A. Roy, A. Vaswani, A. Neelakantan, and N. Parmar. Theory and experiments on vector quantized autoencoders. *arXiv preprint arXiv:1805.11063*, 2018.

T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

J. Seyama and R. S. Nagayama. The uncanny valley: Effect of realism on the impression of artificial human faces. *Presence: Teleoperators and virtual environments*, 16(4):337–351, 2007.

C. Sønderby, B. Poole, and A. Mnih. Continuous relaxation training of discrete latent variable image models. In *Bayesian Deeplearning Workshop @ NIPS2017*, 2017.

L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

A. van den Oord, O. Vinyals, and k. kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30*, pages 6306–6315. 2017.

A. van der Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. volume 48 of *Proceedings of Machine Learning Research*, pages 1747–1756, New York, New York, USA, 20–22 Jun 2016.