NCP-VAE: VARIATIONAL AUTOENCODERS WITH NOISE CONTRASTIVE PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Variational autoencoders (VAEs) are one of the powerful likelihood-based generative models with applications in various domains. However, they struggle to generate high-quality images, especially when samples are obtained from the prior without any tempering. One explanation for VAEs' poor generative quality is the prior hole problem: the prior distribution fails to match the aggregate approximate posterior. Due to this mismatch, there exist areas in the latent space with high density under the prior that do not correspond to any encoded image. Samples from those areas are decoded to corrupted images. To tackle this issue, we propose an energy-based prior defined by the product of a base prior distribution and a reweighting factor, designed to bring the base closer to the aggregate posterior. We train the reweighting factor by noise contrastive estimation, and we generalize it to hierarchical VAEs with many latent variable groups. Our experiments confirm that the proposed noise contrastive priors improve the generative performance of state-of-the-art VAEs by a large margin on the MNIST, CIFAR-10, CelebA 64, and CelebA HQ 256 datasets.

1 INTRODUCTION

Variational autoencoders (VAEs) [Kingma & Welling, 2014; Rezende et al., 2014] are one of the powerful likelihood-based generative models that have applications in image generation [Brock et al., 2018; Karras et al., 2019; Razavi et al., 2019], music synthesis [Dhariwal et al., 2020], speech generation [Oord et al., 2016], image captioning [Aneja et al., 2019; Deshpande et al., 2019; Aneja et al., 2018], semi-supervised learning [Kingma et al., 2014; Parhi et al., 2020], and representation learning [Van Den Oord et al., 2017; Fortuin et al., 2018]. Although there has been tremendous progress in improving the expressivity of the approximate posterior, several studies have observed that VAE priors fail to match the aggregate (approximate) posterior [Rosca et al., 2018; Hoffman & Johnson, 2016]. This phenomenon is sometimes described as holes in the prior, referring to regions in the latent space that are not decoded to data-like samples. Such regions often have a high density under the prior but have a low density under the aggregate approximate posterior.

The prior hole problem is commonly tackled by increasing the flexibility of the prior via hierarchical priors [Klushyn et al., 2019], autoregressive models [Gulrajani et al., 2016], a mixture of approximate posteriors [Tomczak & Welling, 2018], normalizing flows [Xu et al., 2019; Chen et al., 2016], resampled priors [Bauer & Mnih, 2019], and energy-based models [Pang et al., 2020; Valldat et al., 2018b; Pang et al., 2020]. Among them, energy-based models (EBMs) [Du & Mordatch, 2019; Pang et al., 2020] have shown promising results in learning expressive priors. However, they require running iterative MCMC steps during training which is computationally expensive, especially when the energy function is represented by a
neural network. Moreover, they scale poorly to hierarchical models where an EBM is defined on each group of latent variables.

Our key insight in this paper is that a trainable prior is brought as close as possible to the aggregate posterior as a result of training a VAE. The mismatch between the prior and the aggregate posterior can be reduced by simply reweighting the prior to re-adjust its likelihood in the area of mismatch with the aggregate posterior. To represent this reweighting mechanism, we formulate the prior using an EBM that is defined by the product of a reweighting factor and a base trainable prior as shown in Fig. 1. We represent the reweighting factor using neural networks and the base prior using Normal distributions.

Instead of expensive MCMC sampling, we use noise contrastive estimation (NCE) (Gutmann & Hyvärinen, 2010) for training the EBM prior. We show that NCE naturally trains the reweighting factor in our prior by learning a binary classifier to distinguish samples from a target distribution (i.e., samples from the approximate posterior) vs. samples from a noise distribution (i.e., the base trainable prior). However, since NCE’s success depends on how close the noise distribution is to the target distribution, we first train the VAE with the base prior to bring it close to the aggregate posterior. And then, we train the EBM prior using NCE.

In this paper, we make the following contributions: i) We propose an EBM prior termed noise contrastive prior (NCP) which is trained by contrast sampling from the aggregate posterior to samples from a base prior. NCPs are learned as a post-training mechanism to replace the original prior with a more flexible prior, which can improve the generative performance of VAEs with any structure. ii) We also show how NCPs are trained on hierarchical VAEs with many latent variable groups. We show that training hierarchical NCPs scales easily to many groups, as they are trained for each latent variable group in parallel. iii) Finally, we demonstrate that NCPs improve the generative quality of VAEs by a large margin across datasets.

2 BACKGROUND

We first review VAES, their extension to hierarchical VAEs, and the prior hole problem.

Variational Autoencoders: VAES learn a generative distribution \( p(x, z) = p(z)p(x|z) \) where \( p(z) \) is a prior distribution over the latent variable \( z \) and \( p(x|z) \) is a likelihood function that generates the data \( x \) given \( z \). VAEs are trained by maximizing a variational lower bound on the log-likelihood \( \log p(x) \):

\[
\log p(x) \geq \mathbb{E}_{z \sim q(z|x)}[\log p(x|z)] - \text{KL}(q(z|x)||p(z)) := \mathcal{L}_{\text{VAE}}(x),
\]

where \( q(z|x) \) is an approximate posterior and KL is the Kullback–Leibler divergence. The final training objective is formulated by \( \mathbb{E}_{p_d(x)}[\mathcal{L}_{\text{VAE}}(x)] \) where \( p_d(x) \) is the data distribution (Kingma & Welling, 2014).

Hierarchical VAEs (HVAEs): To increase the expressivity of both prior and approximate posterior, earlier work adapted a hierarchical latent variable structure (Vahdat & Kautz, 2020; Kingma et al., 2016; Sønderby et al., 2016; Gregor et al., 2016). In HVAEs, the latent variable \( z \) is divided into \( K \) separate groups, \( z = \{z_1, \ldots, z_K\} \). The approximate posterior and the prior distributions are then defined by:

\[
q(z|x) = \prod_{k=1}^{K} q(z_k|x_{<k}, x) \quad \text{and} \quad p(z) = \prod_{k=1}^{K} p(z_k|z_{<k}).
\]

Using these, the training objective becomes:

\[
\mathcal{L}_{\text{HVAE}}(x) := \mathbb{E}_{q(z|x)}[\log p(x|z)] - \sum_{k=1}^{K} \mathbb{E}_{q(z_{<k}|x)} \left[ \text{KL}(q(z_k|x_{<k}, x)||p(z_k|z_{<k})) \right],
\]

where \( q(z_{<k}|x) = \prod_{s=1}^{k-1} q(z_s|x_{<s}, x) \) is the approximate posterior up to the \( (k-1) \)th group.

The Prior Hole Problem: Let \( q(z) \triangleq \mathbb{E}_{p_d(x)}[q(z|x)] \) denote the aggregate (approximate) posterior. In Appendix B.1, we show that maximizing \( \mathbb{E}_{p_d(x)}[\mathcal{L}_{\text{VAE}}(x)] \) with respect to the prior parameters corresponds to bringing the prior as close as possible to the aggregate posterior by minimizing \( \text{KL}(q(z)||p(z)) \) w.r.t. \( p(z) \). Formally, the prior hole problem refers to the phenomenon that \( p(z) \) fails to match \( q(z) \).

\[\text{For } k = 1, \text{ the expectation inside the summation is simplified to } \text{KL}(q(z_{1}|x)||p(z_{1})).\]
3 Noise Contrastive Priors (NCPs)

One of the main causes of the prior hole problem is the limited expressivity of the prior that prevents it from matching the aggregate posterior. Recently, energy-based models have shown promising results in representing complex distributions. Motivated by their success, we introduce the noise contrastive prior (NCP) \( p_{\text{NCP}}(z) = \frac{1}{Z} r(z)p(z) \), where \( p(z) \) is a base prior distribution, e.g., a Normal, \( r(z) \) is a reweighting factor, and \( Z = \int r(z)p(z)dz \) is the normalization constant. The function \( r : \mathbb{R}^n \rightarrow \mathbb{R}^+ \) maps the latent variable \( z \in \mathbb{R}^n \) to a positive scalar, and can be implemented using neural nets.

The reweighting factor \( r(z) \) can be trained using MCMC sampling as discussed in Appendix A. However, MCMC requires expensive sampling iterations that scale poorly to hierarchical VAEs. To address this, we describe a noise contrastive estimation based approach to train \( p_{\text{NCP}}(z) \) without MCMC sampling.

3.1 Learning the Reweighting Factor with Noise Contrastive Estimation

Recall that training VAEs closes the gap between the prior and the aggregate posterior by minimizing \( \text{KL}(q(z)||p(z)) \) with respect to prior. Assuming the base prior \( p(z) \) to be fixed, \( \text{KL}(q(z)||p_{\text{NCP}}(z)) \) is zero when \( r(z) = q(z)/p(z) \). However, since we do not have the density function for \( q(z) \), we cannot compute the ratio explicitly. Instead, in this paper, we propose to estimate \( r(z) \) using noise contrastive estimation (Gutmann & Hyvärinen 2010), also known as the likelihood ratio trick that has been popularized in machine learning by predictive coding (Oord et al. 2018) and generative adversarial networks (GANs) (Goodfellow et al. 2014). Since, we can generate samples from both \( p(z) \) and \( q(z) \), we train a binary classifier to distinguish samples from \( q(z) \) and samples from the base prior \( p(z) \) by minimizing the binary cross-entropy loss:

\[
\min_D - \mathbb{E}_{z \sim q(z)} [\log D(z)] - \mathbb{E}_{z \sim p(z)} [\log (1 - D(z))]. \tag{3}
\]

Here, \( D : \mathbb{R}^n \rightarrow (0, 1) \) is a binary classifier that generates the classification prediction probabilities. Eq. (3) is minimized when \( D(z) = \frac{q(z)}{q(z) + p(z)} \). Denoting the classifier at optimality by \( D^*(z) \), we estimate the reweighting factor \( r(z) = \frac{q(z)}{p(z)} \approx \frac{D^*(z)}{1 - D^*(z)} \). The appealing advantage of this estimator is that it is obtained by simply training a binary classifier rather than using expensive MCMC sampling.

3.2 Two-stage Training for Noise Contrastive Priors

To properly learn the reweighting factor, NCE training requires the base prior distribution to be close to the target distribution. Intuitively, if \( p(z) \) is very close to \( q(z) \) (i.e., \( p(z) \approx q(z) \)), the optimal classifier will have a large loss value in Eq. (3), and we will have \( r(z) \approx 1 \). If \( p(z) \) is instead far from \( q(z) \), the binary classifier will easily learn to distinguish samples from the two distributions and it will not learn the likelihood ratios correctly. If \( p(z) \) is roughly close to \( q(z) \), then the binary classifier can learn the ratios.

To ensure that the base prior distribution is close to the target aggregate posterior distribution, we propose a two-stage training algorithm. In the first stage, we train the VAE with only the base prior \( p(z) \). From Appendix B.1, we know that at the end of training, \( p(z) \) is as close as possible to \( q(z) \). In the second stage, we freeze the VAE model including the approximate posterior \( q(z|x) \), the base prior \( p(z) \), and the likelihood \( p(x|z) \), and we only train the reweighting factor \( r(z) \) using Eq. (3). The second stage can be thought of as replacing the base distribution \( p(z) \) with a more expressive distribution of the form \( p_{\text{NCP}}(z) \propto r(z)p(z) \). Hence, NCP matches the prior to the aggregate posterior \( q(z) \) by using the reweighting factor. Note that our proposed method is generic as it only assumes that we can draw samples from \( q(z) \) and \( p(z) \), which applies to any VAE. Our training is illustrated in Fig. 2.

\[\text{We generate samples from the aggregate posterior } q(z) = \mathbb{E}_{p_d(x)}[q(z|x)] \text{ via ancestral sampling: draw data from the training set } (x \sim p_d(x)) \text{ and then sample from } z \sim q(z|x).\]
Figure 2: NCP-VAE is trained in two stages. In the first stage, we train a VAE using the original VAE objective. In the second stage, we train the reweighting factor $r(z)$ using noise contrastive estimation (NCE). NCE trains a classifier to distinguish samples from the prior and samples from the aggregate posterior. Our noise contrastive prior (NCP) is then constructed by the product of the base prior and the reweighting factor, formed via the classifier. At test time, we sample from NCP using SIR or LD. These samples are then passed to the decoder to generate output samples.

3.3 Test Time Sampling

To sample from a VAE with an NCP, we first generate samples from the NCP before passing them to the decoder to generate output samples (shown in Fig. 2). We propose two methods for sampling from NCPs.

Sampling-Importance-Resampling (SIR): We first generate $M$ samples from the base prior distribution $\{z^{(m)} \}_{m=1}^{M} \sim p(z)$. We then resample one of the $M$ proposed samples using importance weights proportional to $w^{(m)} = p_{\text{NCP}}(z^{(m)})/p(z^{(m)}) = r(z^{(m)})$. The benefit of this technique: both proposal generation and the evaluation of $r$ on the samples are done in parallel.

Langevin Dynamics (LD): Since our NCP is an EBM, we can use LD for sampling. Denoting the energy function by $E(z) = -\log r(z) - \log p(z)$, we initialize a sample $z_0$ by drawing from $p(z)$ and update the sample iteratively using: $z_{t+1} = z_t - 0.5 \lambda \nabla \mathbb{E}_x E(z) + \sqrt{\lambda} \epsilon_t$ where $\epsilon_t \sim N(0, 1)$ and $\lambda$ is the step size. LD is run for a finite number of iterations, and in contrast to SIR, it can be slow given its sequential form.

3.4 Generalization to Hierarchical VAEs

The state-of-the-art VAE (Vahdat & Kautz 2020) uses a hierarchical $q(z|x)$ and $p(z)$. Here $p(z)$ is chosen to be a Gaussian distribution. Appendix B.2 shows that training a HVAE encourages the prior to minimize $\mathbb{E}_{q(z_{<k})} [\text{KL}(q(z_k|z_{<k}) || p(z_k|z_{<k}))]$ for each conditional, where $q(z_{<k}) \triangleq \mathbb{E}_{p_{\text{ud}}(x)} [q(z_{<K} | x)]$ is the aggregate posterior up to the $(k-1)^{th}$ group, and $q(z_k|z_{<k}) \triangleq \mathbb{E}_{p_{\text{ud}}(x)} [q(z_k | z_{<k}, x)]$ is the aggregate conditional for the $k^{th}$ group. Given this observation, we extend NCPs to hierarchical models to match each conditional in the prior with $q(z_k | z_{<k})$. Formally, we define hierarchical NCPs by $p_{\text{NCP}}(z) = \frac{1}{Z} \prod_{k=1}^{K} r(z_k | z_{<k}) p(z_k | z_{<k})$ where each factor is an EBM. $p_{\text{NCP}}(z)$ resembles energy machines with an autoregressive structure among groups (Nash & Durkan 2019).

In the first stage, we train the HVAE with prior $\prod_{k=1}^{K} p(z_k | z_{<k})$. For the second stage, we use $K$ binary classifiers, each for a hierarchical group. Following Appendix C, we train each classifier via:

$$
\min_{D_k} \mathbb{E}_{p_{\text{ud}}(x)q(z_{<k} | x)} \left[ -\mathbb{E}_{q(z_k | z_{<k}, x)} [\log D_k(z_k, c(z_{<k}))] - \mathbb{E}_{p(z_k | z_{<k})} [\log (1 - D_k(z_k, c(z_{<k})))]) \right], \quad (4)
$$
where the outer expectation samples from groups up to the $(k - 1)^{th}$ group, and the inner expectations sample from approximate posterior and base prior for the $k^{th}$ group, conditioned on the same $z_{<k}$. The discriminator $D_k$ classifies samples $z_k$ while conditioning its prediction on $z_{<k}$ using a shared context feature $c(z_{<k})$.

The NCE training in Eq. (4) is minimized when \( D_k(z_k, c(z_{<k})) = \frac{q(z_k|z_{<k})}{q(z_k|z_{<k}) + p(z_k|z_{<k})} \). Denoting the classifier at optimality by \( D_k^*(z, c(z_{<k})) \), we obtain the reweighting factor \( r(z_k|z_{<k}) \approx \frac{D_k^*(z_k, c(z_{<k}))}{1 - D_k^*(z_k, c(z_{<k}))} \) in the second stage. Given our hierarchical NCP, we use ancestral sampling to sample from the prior. For sampling from each group, we can use SIR or LD as discussed before.

The context feature $c(z_{<k})$ extracts a representation from $z_{<k}$. Instead of learning a new representation at stage two, we simply use the representation that is extracted from $z_{<k}$ in the hierarchical prior, trained in the first stage. Note that the binary classifiers in the second stage are trained in parallel for all groups.

\section{4 Related Work}

In this section, we review prior works related to the proposed method.

**Energy-based Models (EBMs):** Early work on EBMs for generative learning goes back to 1980s (Ackley et al., 1985; Hinton et al., 1986). Prior to the modern deep learning era, most attempts for building generative models using EBMs were centered around Boltzmann machines (Hinton, 2002; Hinton et al., 2006) and their “deep” extensions (Salakhutdinov & Hinton, 2009; Larochelle & Bengio, 2008). Although the energy function in these models is restricted to simple bilinear functions, they have been proven effective for representing the prior in discrete VAEs (Rolfe, 2016; Vahdat et al., 2018a; 2020). Recently, EBMs with neural energy functions have gained popularity for representing complex data distribution (Du & Mordatch, 2019). Pang et al. (2020) have shown that neural EBMs can represent expressive prior distributions. However, in this case, the prior is trained using MCMC sampling, and it has been limited to a single group of latent variables. To eliminate MCMC sampling, NCE (Gutmann & Hyvärinen, 2010) has recently been used for training a normalizing flow on data distributions (Gao et al., 2020). Moreover, Han et al. (2019; 2020) have used divergence triangulation to sidesteps MCMC sampling. In contrast, we use NCE to train an EBM prior where a noise distribution is easily available through a pre-trained VAE.

**Adversarial Training:** Similar to NCE, generative adversarial networks (GANs) (Goodfellow et al., 2014) also rely on a discriminator to learn the likelihood ratio between noise and real images. However, GANs use the discriminator to update the generator, whereas in NCE, the noise generator is fixed. In spirit similar are recent works (Azadi et al., 2018; Turner et al., 2019; Che et al., 2020) that link GANs, defined in the pixels space, to EBMs. We apply the likelihood ratio trick to the latent space of VAEs. The main difference: the base prior and approximate posterior are trained with the VAE objective rather than the adversarial loss. Adversarial loss has been used for training implicit encoders in VAEs (Makhzani et al., 2015; Mescheder et al., 2017; Engel et al., 2018). But, they have not been linked to energy-based priors as we do explicitly.

**Prior Hole Problem:** Among prior works on this problem, VampPrior (Tomczak & Welling, 2018) uses a mixture of encoders to represent the prior. However, this requires storing training data or pseudo-data to generate samples at test time. Takahashi et al. (2019) use the likelihood ratio estimator to train a simple prior distribution. However at test time, the aggregate posterior is used for sampling in the latent space. Bauer & Mnih (2019) propose a reweighting factor similar to ours, but it is trained via importance sampling. Recently, Lawson et al. (2019) introduced energy-inspired models (EIMs) that define distributions induced by the sampling processes used by Bauer & Mnih (2019) as well as our SIR sampling. Although, EIMs have the advantage of end-to-end training and can be used as either prior or decoder in VAEs, they require multiple samples during training (up to 1K). This can make application of EIMs to deep hierarchical models such as NVAEs very challenging as these models are often very memory intensive and are trained with a few training samples per GPU.
Two-stage VAEs: VQ-VAE (Van Den Oord et al., 2017; Razavi et al., 2019) first trains an autoencoder and then fits an autoregressive PixelCNN (Van Den Oord et al., 2016) prior to the latent variables which is slow to sample from. Two-stage VAE (2s-VAE) (Dai & Wipf, 2018) trains a VAE on the data, and then, trains another VAE in the latent space. Regularized autoencoders (RAE) (Ghosh et al., 2020) train an autoencoder, and subsequently a Gaussian mixture model on latent codes. In contrast, we train the model with the original VAE objective in the first stage, and we improve the expressivity of the prior using an EBM.

5 Experiments

Our implementation of NCP-VAE builds upon NVAE (Vahdat & Kautz, 2020), the state-of-the-art hierarchical VAE. We examine NCP-VAE on four datasets including dynamically binarized MNIST (LeCun, 1998), CIFAR-10 (Krizhevsky et al., 2009), CelebA-64 (Liu et al., 2015) and CelebA-HQ-256 (Karras et al., 2017). For CIFAR-10 and CelebA-64, the model has 30 groups, and for CelebA-HQ-256 it has 20 groups. We sample from NCP-VAE using SIR with 5K proposal samples at the test (image generation) time. On these datasets, we measure the sample quality using the Fréchet Inception Distance (FID) score (Heusel et al., 2017) with 50,000 samples, as computing the log-likelihood requires estimating the intractable normalization constant. To report log-likelihood results, we train an NVAE model with a small latent space on MNIST with 10 groups of $4 \times 4$ latent variables. We estimate log normalization constant in NCPs using 1000 importance weighted samples. We intentionally limit the latent space to ensure that we can estimate the normalization constant correctly (standard deviation of $\log Z$ estimation $\leq 0.23$). Thus, on this dataset, we report negative log-likelihood (NLL). Implementation details are provided in Appendix E.

5.1 Quantitative Results

The quantitative results are reported in Tab. 1, Tab. 2, Tab. 3, and Tab. 4. On all four datasets, our model improves upon state-of-the-art NVAE, and it reduces the gap with GANs by a large margin. On CelebA 64, we improve NVAE from an FID of 13.48 to 5.25, comparable to GANs. On CIFAR-10, NCP-VAE
improves the NV AE FID of 51.71 to 24.08. On MNIST, although our latent space is much smaller, our model outperforms previous VAEs. NV AE has reported 78.01 nats on this dataset with a larger latent space.

**Reconstruction:** In the last row of Tab. 1, Tab. 2, and Tab. 3, we report the FID score for the reconstructed images of the NV AE baseline. Note how reconstruction FID is much lower than our FID, indicating that our model is far from memorizing the training data. In Appendix G, we also provide nearest neighbours from training data for generated samples. In Appendix F, we present our NCP applied to vanilla VAEs.

5.2 **Qualitative Results**

We visualize samples generated by NCP-VAE in Fig. 3 without any manual intervention. We adopt the common practice of reducing the temperature of the base prior \( p(z) \) by scaling down the standard-deviation of the conditional Normal distributions [Kingma & Dhariwal, 2018](#) [Brock et al., 2018](#) Vahdat & Kautz (2020) also observe that re-adjusting the batch-normalization (BN), given a temperature applied to the prior, improves the generative quality. Similarly, we achieve diverse, high-quality images by re-adjusting the BN statistics as described by Vahdat & Kautz (2020). Additional qualitative results are shown in Appendix H.

5.3 **Additional Studies**

We perform additional experiments to study i) how hierarchical NCPs perform as the number of latent groups increases, ii) the impact of SIR and LD hyperparameters, and iii) what the classification loss in NCE training conveys about \( p(z) \) and \( q(z) \). All experiments in this section are performed on CelebA-64 data.

---

3Lowering of the temperature is only used to obtain qualitative samples. It’s not used when computing any of the quantitative results in Sec. 5.1.
### Table 6: Effect of SIR sample size and LD iterations. Time-$N$ is the time used to generate a batch of $N$ images.

<table>
<thead>
<tr>
<th># SIR proposal samples</th>
<th>FID↓</th>
<th>Time-1 (sec)</th>
<th>Time-10 (sec)</th>
<th>Memory (GB)</th>
<th># LD iterations</th>
<th>FID↓</th>
<th>Time-1 (sec)</th>
<th>Time-10 (sec)</th>
<th>Memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>11.75</td>
<td>0.34</td>
<td>0.42</td>
<td>1.96</td>
<td>5</td>
<td>14.44</td>
<td>3.08</td>
<td>3.07</td>
<td>1.94</td>
</tr>
<tr>
<td>50</td>
<td>8.58</td>
<td>0.40</td>
<td>1.21</td>
<td>4.30</td>
<td>50</td>
<td>12.76</td>
<td>27.85</td>
<td>28.55</td>
<td>1.94</td>
</tr>
<tr>
<td>500</td>
<td>6.76</td>
<td>1.25</td>
<td>9.43</td>
<td>20.53</td>
<td>500</td>
<td>8.12</td>
<td>276.13</td>
<td>260.35</td>
<td>1.94</td>
</tr>
<tr>
<td>5000</td>
<td><strong>5.25</strong></td>
<td><strong>10.11</strong></td>
<td><strong>95.67</strong></td>
<td><strong>23.43</strong></td>
<td>1000</td>
<td><strong>6.98</strong></td>
<td><strong>552</strong></td>
<td><strong>561.44</strong></td>
<td><strong>1.94</strong></td>
</tr>
</tbody>
</table>

![Figure 4: (a) Classification loss for binary classifiers on latent variable groups. A larger final loss upon training indicates that \( q(z) \) and \( p(z) \) are more similar. (b) The effective sample size vs. the final loss value at the end of training. Higher effective sample size implies similarity of two distributions.](image)

**Number of latent variable groups:** Tab. 5 shows the generative performance of hierarchical NCP with different amounts of latent variable groups. As we increase the number of groups, the FID score of both NVAE and our model improves. This demonstrates the efficacy of our NCPs, even with expressive hierarchical priors and in the presence of many groups.

**SIR and LD parameters:** The computational complexity of SIR is similar to LD if we set the number of proposal samples in SIR equal to the number LD iterations. Tab. 6 reports the impact of these parameters. We observe that increasing both the number of proposal samples in SIR and the LD iterations leads to a noticeable improvement in FID score. For SIR, the proposal generation and the evaluation of \( r(z) \) are parallelizable. Hence, as shown in Tab. 6, image generation is faster with SIR than with LD (LD iterations are sequential). However, GPU memory usage scales with the number of SIR proposals, but not with the number of LD iterations. Interestingly, SIR, albeit simple, performs better than LD when using about the same compute.

**Classification loss in NCE:** We can draw a direct connection between the classification loss in Eq. 3 and the similarity of \( p(z) \) and \( q(z) \). Denoting the classification loss in Eq. 3 at optimality by \( \ell^* \), Goodfellow et al. (2014) show that \( \text{JSD}(p(z)||q(z)) = \log 2 - 0.5 \times \ell^* \) where JSD denotes the Jensen–Shannon divergence between two distributions. Fig. 4(a) plots the classification loss (Eq. 3) for each classifier for a 15-group NCP trained on the CelebA-64 dataset. Assume that the classifier loss at the end of training is a good approximation of \( \ell^* \). We observe that 8 out of 15 groups have \( \ell^* \geq 0.4 \), indicating a good overlap between \( p(z) \) and \( q(z) \) for those groups. To further assess the impact of the distribution match on SIR sampling, in Fig. 4(b), we visualize the effective sample size (ESS) in SIR vs. \( \ell^* \) for the same group. We observe a strong correlation between \( \ell^* \) and the effective sample size. SIR is more reliable on the same 8 groups that are more similar.

Table 5: \# groups & generative performance in FID↓

<table>
<thead>
<tr>
<th># groups</th>
<th>NVAE</th>
<th>NCP-VAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>33.18</td>
<td>18.68</td>
</tr>
<tr>
<td>15</td>
<td>14.96</td>
<td>5.96</td>
</tr>
<tr>
<td>30</td>
<td>13.48</td>
<td><strong>5.25</strong></td>
</tr>
</tbody>
</table>

---

4ESS measures reliability of SIR via \( 1/\sum_m (\hat{w}^{(m)})^2 \), where \( \hat{w}^{(m)} = r(z^{(m)})/\sum_m r(z^{(m)}) \) (Owen, 2013).
5.4 Additional Comparison to Prior Art

In our previous experiments, we rely on NVAE as the VAE backbone. Although this shows that our approach can be applied to large scale models successfully, this may make comparison against the prior art unfair as they are often applied to smaller models. To provide a fair comparison, in this section, we apply NCP to several commonly used small VAE models.

Comparison against RAE, 2s-VAE, and WAE: In Tab. 7 we show the generative performance of our approach applied to the VAE architecture in RAE (Ghosh et al., 2020). Note that this VAE architecture has only one latent variable group. The same base architecture was used in the implementation of 2s-VAE (Dai & Wipf, 2018) and WAE (Tolstikhin et al., 2018). We borrow the training setup from Ghosh et al. (2020) on the CelebA-64 dataset, and compare to the baselines reported in this work. We apply our NCP-VAE on top of vanilla VAE with a Gaussian prior as well as a 10-component Gaussian mixture model (GMM) prior that was proposed in RAEs. Our NCP-VAE improves the performance on the base VAE, improving the FID score to 41.28 from 48.12. Additionally, when NCP is applied to the VAE with GMM prior (the RAE model), it improves its performance from 40.95 to the FID score of 39.00.

Comparison against LARS and SNIS: In Tab. 8 we compare NCP-VAE to LARS (Bauer & Mnih, 2019) and SNIS (Lawson et al., 2019) priors on MNIST. We implement our NCP-VAE using the VAE and energy-function architectures that were used by Lawson et al. (2019) as well as their approach for obtaining a lower bound on the log likelihood. NCP-VAE obtains the negative log-likelihood (NLL) of 82.82, comparable to Lawson et al. (2019), while outperforming LARS (Bauer & Mnih, 2019).

Table 7: Generative performance on CelebA-64 with the RAE (Ghosh et al., 2020) architecture

<table>
<thead>
<tr>
<th>Model w/ Gaussian prior</th>
<th>FID</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE</td>
<td>48.12</td>
</tr>
<tr>
<td>2s-VAE (Dai &amp; Wipf, 2018)</td>
<td>49.70</td>
</tr>
<tr>
<td>WAE (Tolstikhin et al., 2018)</td>
<td>42.73</td>
</tr>
<tr>
<td>RAE (Ghosh et al., 2020)</td>
<td>40.95</td>
</tr>
<tr>
<td>NCP w/ Gaussian prior as base</td>
<td>41.28</td>
</tr>
<tr>
<td>NCP w/ GMM prior as base</td>
<td><strong>39.00</strong></td>
</tr>
</tbody>
</table>

Table 8: Likelihood results on MNIST on a single latent group model with architecture from LARS (Bauer & Mnih, 2019) & SNIS (Lawson et al., 2019) (results in nats)

<table>
<thead>
<tr>
<th>Model</th>
<th>NLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE w/ Gaussian prior</td>
<td>84.82</td>
</tr>
<tr>
<td>VAE w/ LARS prior (Bauer &amp; Mnih, 2019)</td>
<td>83.03</td>
</tr>
<tr>
<td>VAE w/ SNIS prior (Lawson et al., 2019)</td>
<td><strong>82.52</strong></td>
</tr>
<tr>
<td>NCP-VAE</td>
<td>82.82</td>
</tr>
</tbody>
</table>

6 Conclusions

The prior hole problem is one of the main reasons for VAEs’ poor generative quality. In this paper, we tackled this problem by introducing the noise contrastive prior (NCP), defined by the product of a reweighting factor and a base prior. We showed how the reweighting factor is trained by contrasting samples from the aggregate posterior with samples from the base prior. Our proposal is simple and can be applied to any VAE to increase its prior’s expressivity. We also showed how NCP training scales to large hierarchical VAEs, as it can be done in parallel simultaneously for all the groups. Finally, we demonstrated that NCPs improve the generative performance of state-of-the-art NVAEs by a large margin, closing the gap to GANs.

References


A TRAINING ENERGY-BASED PRIORS USING MCMC

In this section, we show how a VAE with energy-based model in its prior can be trained. Assuming that the prior is in the form \( p_{EBM}(z) = \frac{1}{Z}r(z)p(z) \), the variational bound is of the form:

\[
E_{p_{EBM}(z)}[L_{VAE}] = E_{p_{EBM}(z)}\left[ E_{q(z|x)}[\log p(x|z)] - KL(q(z|x)||p_{EBM}(z)) \right]
= E_{p_{EBM}(z)}\left[ E_{q(z|x)}[\log p(x|z) - \log q(z|x) + \log r(z) + \log p(z)] \right] - \log Z,
\]

where the expectation term, similar to VAEs, can be trained using the reparameterization trick. The only problematic term is the log-normalization constant \( \log Z \), which captures the gradient with respect to the parameters of the prior \( p_{EBM}(z) \). Denoting these parameters by \( \theta \), the gradient of \( \log Z \) is obtained by:

\[
\frac{\partial}{\partial \theta} \log Z = \frac{1}{Z} \int \frac{\partial (r(z)p(z))}{\partial \theta} dz = \frac{1}{Z} \int \frac{r(z)p(z)}{Z} \frac{\partial \log (r(z)p(z))}{\partial \theta} dz = E_{p_{EBM}(z)}\left[ \frac{\partial \log (r(z)p(z))}{\partial \theta} \right],
\]

where the expectation can be estimated using MCMC sampling from the EBM prior.

B MAXIMIZING THE VARIATIONAL BOUND FROM THE PRIOR’S PERSPECTIVE

In this section, we discuss how maximizing the variational bound in VAEs from the prior’s perspective corresponds to minimizing a KL divergence from the aggregate posterior to the prior. Note that this relation has been explored by Hoffman & Johnson (2016); Rezende & Viola (2018); Tomczak & Welling (2018) and we include it here for completeness.

B.1 VAE WITH A SINGLE GROUP OF LATENT VARIABLES

Denote the aggregate (approximate) posterior by \( q(z) \triangleq E_{p_{EBM}(z)}[q(z|x)] \). Here, we show that maximizing the \( E_{p_{EBM}(z)}[L_{VAE}(x)] \) with respect to the prior parameters corresponds to learning the prior by minimizing \( KL(q(z)||p(z)) \). To see this, note that the prior \( p(z) \) only participates in the KL term in \( L_{VAE} \) (Eq. 1). We hence have:

\[
\arg \max_{p(z)} E_{p_{EBM}(z)}[L_{VAE}(x)] = \arg \min_{p(z)} E_{p_{EBM}(z)}[KL(q(z|x)||p(z))]
= \arg \min_{p(z)} -E_{p_{EBM}(z)}[H(q(z|x))] - E_{q(z)}[\log p(z)]
= \arg \min_{p(z)} -H(q(z)) - E_{q(z)}[\log p(z)]
= \arg \min_{p(z)} KL(q(z)||p(z)),
\]

where \( H(\cdot) \) denotes the entropy. Above, we replaced the expected entropy \( E_{p_{EBM}(z)}[H(q(z|x))] \) with \( H(q(z)) \) as the minimization is with respect to the parameters of the prior \( p(z) \).

B.2 HIERARCHICAL VAEs

Denote hierarchical approximate posterior and prior distributions by: \( q(z|x) = \prod_{k=1}^{K} q(z_k|z_{<k}, x) \) and \( p(z) = \prod_{k=1}^{K} p(z_k|z_{<k}) \). The hierarchical VAE objective becomes:

\[
L_{HVAE}(x) = E_{q(z|x)}[\log p(x|z)] - \sum_{k=1}^{K} E_{q(z_{<k}|x)}[KL(q(z_k|z_{<k}, x)||p(z_k|z_{<k}))],
\]

(6)
where \( q(z_{<k}|x) = \prod_{i=1}^{k-1} q(z_i|z_{<i}, x) \) is the approximate posterior up to the \((k-1)^{th}\) group. Denote the aggregate posterior up to the \((K-1)^{th}\) group by \( q(z_{<K}) \triangleq \mathbb{E}_{p_d(x)}[q(z_{<K}|x)] \) and the aggregate conditional for the \(k^{th}\) group given the previous groups \( q(z_k|z_{<k}) \triangleq \mathbb{E}_{p_d(x)}[q(z_k|z_{<k}, x)] \).

Here, we show that maximizing \( \mathbb{E}_{p_d(x)}[L_{HVAE}(x)] \) with respect to the prior corresponds to learning the prior by minimizing \( \mathbb{E}_{q(z_{<k})}[KL(q(z_k|z_{<k})||p(z_k|z_{<k}))] \) for each conditional:

\[
\arg \max_{p(z_k|z_{<k})} \mathbb{E}_{p_d(x)}[L_{HVAE}(x)] = \arg \min_{p(z_k|z_{<k})} \mathbb{E}_{p_d(x)}[KL(q(z_k|z_{<k}, x)||p(z_k|z_{<k}))]
\]

\[
= \arg \min_{p(z_k|z_{<k})} -\mathbb{E}_{p_d(x)q(z_{<k}|x)q(z_k|z_{<k}, x)}[\log p(z_k|z_{<k})]
\]

\[
= \arg \min_{p(z_k|z_{<k})} -\mathbb{E}_{q(z_{<k})}[\log p(z_k|z_{<k})]
\]

\[
= \arg \min_{p(z_k|z_{<k})} \mathbb{E}_{q(z_{<k})}[\log p(z_k|z_{<k})]
\]

\[
= \arg \min_{p(z_k|z_{<k})} H_k(q(z_k|z_{<k})) - \mathbb{E}_{q(z_{<k})}[\log p(z_k|z_{<k})]
\]

\[
= \arg \min_{p(z_k|z_{<k})} KL(q(z_k|z_{<k})||p(z_k|z_{<k}))
\]

\[
= \arg \min_{p(z_k|z_{<k})} \mathbb{E}_{q(z_{<k})}[KL(q(z_k|z_{<k})||p(z_k|z_{<k}))].
\]

(7)

### C Conditional NCE for Hierarchical VAEs

In this section, we describe how we derive the NCE training objective for hierarchical VAEs given in Eq. 4. Our goal is to learn the likelihood ratio between the aggregate conditional \( q(z_k|z_{<k}) \) and the prior \( p(z_k|z_{<k}) \). We can define the NCE objective to train the discriminator \( D_k(z_k, z_{<k}) \) that classifies \( z_k \) given samples from the previous groups \( z_{<k} \) using:

\[
\min_{D_k} \mathbb{E}_{q(z_{<k})}[-\log D_k(z_k, z_{<k}) - \mathbb{E}_{p(z_k|z_{<k})}[\log(1 - D_k(z_k, z_{<k})] \quad \forall z_{<k}.
\]

(8)

Since \( z_{<k} \) is in a high dimensional space, we cannot apply the minimization \( \forall z_{<k} \). Instead, we sample from \( z_{<k} \) using the aggregate approximate posterior \( q(z_{<k}) \) as done for the KL in a hierarchical model (Eq. 7):

\[
\min_{D_k} \mathbb{E}_{q(z_{<k})}[-\log D_k(z_k, z_{<k}) - \mathbb{E}_{p(z_k|z_{<k})}[\log(1 - D_k(z_k, z_{<k})] \quad \forall z_{<k}.
\]

(9)

Since \( q(z_{<k})q(z_k|z_{<k}) = q(z_k, z_{<k}) \), we have:

\[
\min_{D_k} \mathbb{E}_{p_d(x)q(z_{<k}|x)}[-\log D_k(z_k, z_{<k}) - \mathbb{E}_{p(z_k|z_{<k})}[\log(1 - D_k(z_k, z_{<k})] \quad \forall z_{<k}.
\]

(10)

Finally, instead of passing all the samples from the previous latent variables groups to \( D \), we can pass the context feature \( c(z_{<k}) \) that extracts a representation from all the previous groups:

\[
\min_{D_k} \mathbb{E}_{p_d(x)q(z_{<k}|x)}[-\log D_k(z_k, c(z_{<k})) - \mathbb{E}_{p(z_k|z_{<k})}[\log(1 - D_k(z_k, c(z_{<k}))] \quad \forall z_{<k}.
\]

(11)

### D NVAE Based Model and Context Feature

**Context Feature:** The base model NVAE [Vahdat & Kautz 2020] is hierarchical. To encode the information from the lower levels of the hierarchy to the higher levels, during training of the binary classifiers, we concatenate the context feature \( c(z_{<k}) \) to the samples from both \( p(z) \) and \( q(z) \). The context feature for each group is the output of the residual cell of the top-down model and encodes a representation from \( z_{<k} \).
Figure 5: Residual blocks used in the binary classifier. We use $s$, $p$ and $C$ to refer to the stride parameter, the padding parameter and the number of channels in the feature map, respectively.

**Image Decoder $p(x|z)$:** The base NVAE (Vahdat & Kautz 2020) uses a mixture of discretized logistic distributions for all the datasets but MNIST, for which it uses a Bernoulli distribution. In our model, we observe that replacing this with a Normal distribution for the RGB image datasets leads to significant improvements in the base model performance. This is also reflected in the gains of our approach.

**E Implementation Details**

The binary classifier is composed of two types of residual blocks as in Fig. 5. The residual blocks use batch-normalization (Ioffe & Szegedy 2015), the Swish activation function (Ramachandran et al. 2017), and the Squeeze-and-Excitation (SE) block (Hu et al. 2018). SE performs a squeeze operation (e.g., mean) to obtain a single value for each channel. An excitation operation (non-linear transformation) is applied to these values to get per-channel weights. The Residual-Block-B differs from Residual-Block-A in that it doubles the number of channels ($C \rightarrow 2C$), while down-sampling the other spatial dimensions. It therefore also includes a factorized reduction with $1 \times 1$ convolutions along the skip-connection. The complete architecture of the classifier is:

**F Performance On Single Group VAE**

To demonstrate the efficacy of our approach on any off-the-shelf VAE, we apply our NCE based approach to the VAE in (Ghosh et al. 2020). Note we use the vanilla VAE with Normal prior provided by the authors. The FID for CelebA 64 improves from 48.12 to 41.28. Note that the FID for reconstruction is reported as 39.12. Single group VAEs are known to perform poorly on the task of image generation, which is reflected in the high FID value of reconstruction.

**G Nearest Neighbors from the Training Dataset**

To highlight that hierarchical NCP generates unseen samples at test time rather than memorizing the training dataset, Figures 6-7 visualize samples from the model along with a few training images that are most similar.
Conv 3x3 (s1, p1) + ReLU
↓
Residual-Block-A
↓
Residual-Block-A
↓
Residual-Block-A
↓
Residual-Block-B
↓
Residual-Block-A
↓
Residual-Block-A
↓
Residual-Block-A
↓
Residual-Block-A
↓
Residual-Block-B
↓
2D average pooling
↓
Linear + Sigmoid

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Adam (Kingma &amp; Ba, 2014)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Rate</td>
<td>Initialize at 1e-3, CosineAnnealing (Loshchilov &amp; Hutter, 2016) to 1e-7</td>
</tr>
<tr>
<td>Batch size</td>
<td>512 (MNIST, CIFAR-10), 256 (CelebA-64), 128 (CelebA HQ 256)</td>
</tr>
</tbody>
</table>

**Table 9:** Hyper-parameters for training the binary classifiers.

To get the similarity score for a pair of images, we downsample to $64 \times 64$, center crop to $40 \times 40$ and compute the Euclidean distance. The KD-tree algorithm is used to fetch the nearest neighbors. We note that the generated samples are quite distinct from the training images.
Figure 6: Query images (left) and their nearest neighbors from the CelebA-HQ-256 training dataset.
Figure 7: Query images (left) and their nearest neighbors from the CelebA-HQ-256 training dataset.
Figure 8: Additional samples from CelebA-64 at $t = 0.7$. 
Figure 9: Additional samples from CelebA-HQ-256 at $t = 0.7$. 
Figure 10: Selected good quality samples from CelebA-HQ-256.
I EXPERIMENT ON SYNTHETIC DATA

In Fig. 11 we demonstrate the efficacy of our approach on the 25-Gaussians dataset, that is generated by a mixture of 25 two-dimensional Gaussian distributions that are arranged on a grid. The encoder and decoder of the VAE have 4 fully connected layers with 256 hidden units, with 20 dimensional latent variables. The discriminator has 4 fully connected layers with 256 hidden units. Note that the samples decoded from prior $p(z)$ (Fig. 11(b)) without the NCP approach generates many points from the low density regions in the data distribution. These are removed by using our NCP approach (Fig. 11(c)).

![Figure 11: Qualitative results on mixture of 25-Gaussians.](image)

We use 50k samples from the true distribution to estimate the log-likelihood. Our NCP approach obtains an average log-likelihood of $-0.954$ nats compared to the log-likelihood obtained by sampling from the Gaussian prior, $-2.753$ nats. We use 20k Monte Carlo samples to estimate the log partition function for the calculation of log-likelihood.
**J  ADDITIONAL QUALITATIVE EXAMPLES**

In Fig. [12] we show additional examples of images generated by NVAE [Vahdat & Kautz, 2020] and our NCP-VAE. We use temperature=0.7 for both. Visually corrupt images are highlighted with a red square.

**Figure 12:** Additional samples from CelebA-64 at $t = 0.7$. 