VAEPP: Variational Autoencoder with a Pull-back Prior

Abstract

Many approaches to training generative models by distinct training objectives have been proposed in the past. Variational Autoencoder (VAE) is an outstanding model of them based on log-likelihood. In this paper, we propose a novel learnable prior, Pull-back Prior, for VAEs by adjusting the density of the prior through a discriminator that can assess the quality of data. It involves the discriminator from theory of GANs to enrich the prior in VAEs. Based on it, we propose a more general framework, VAE with a Pull-back Prior (VAEPP), which uses the existing techniques of VAEs and WGANs, to improve the log-likelihood, quality of sampling and stability of training. In MNIST and CIFAR-10, the log-likelihood of VAEPP outperforms models without autoregressive components and is comparable to autoregressive models. In MNIST, Fashion-MNIST, CIFAR-10 and CelebA, the FID of VAEPP is comparable to GANs and SOTA of VAEs.

1 Introduction

How to learn deep generative models that are able to capture complex data pattern in high dimension space, *e.g.*, image datasets, is one of the major challenges in machine learning. Many approaches to training generative models by distinct training objectives have been proposed in the past, *e.g.*, Generative Adversarial Network (GAN) [Goodfellow *et al.*, 2014], flow-based models [Dinh *et al.*, 2016; Kingma and Dhariwal, 2018], PixelCNN [Van den Oord *et al.*, 2016], and Variational Autoencoder (VAE) [Kingma and Welling, 2014; Rezende *et al.*, 2014].

VAE uses the variational inference and re-parameterization trick to optimize the evidence lower bound of log-likelihood (ELBO). In the past, many researches [Kingma *et al.*, 2016; Tomczak and Welling, 2016] focused on enriching the variational posterior, but recently [Tomczak and Welling, 2018] showed that the standard Gaussian prior could lead to underfitting. To enrich the prior, several learnable priors have been proposed [Tomczak and Welling, 2018; Bauer and Mnih, 2019; Takahashi *et al.*, 2019]. Most of them focus on approximating aggregated posterior which is the integral of the variational posterior and is shown as the optimal prior that minimizes ELBO. However, existing methods based on the aggregated posterior reach limited performance, and the practical meaning of the aggregated posterior is ambiguous. Recently, we notice that a discriminator can assess the quality of data and **we argue that it is advisable to adjust the learnable prior by the discriminator, where the discriminator has clear practical meaning.**

We propose a novel learnable prior, Pull-back Prior, based on the discriminator. Firstly, a discriminator $D(x)$ is trained for assessing the quality of images. Then, we define a pull-back discriminator on latent space, by $D(G(z))$, where $G(z)$ is the generator. Finally, we adjust the density of the prior according to the pull-back discriminator.

We propose a training algorithm for VAE with Pull-back Prior (VAEPP), based on SGVB [Kingma and Welling, 2014] with gradient penalty terms, which mixes the discriminator and the gradient penalty term into VAE. In this way, We extend VAE to a more general framework, VAEPP. In VAEPP, we use the gradient penalty terms of WGAN-GP [Gulrajani *et al.*, 2017] and WGAN-div [Wu *et al.*, 2018] to achieve Lipschitz constraint. We also use Langevin dynamics, provided by [Kumar *et al.*, 2019] to improve the quality of sampling.

The main contributions of this paper are in the following:

- We propose a novel learnable prior, Pull-back Prior, which is adjusted by a discriminator that can assess the quality of data.
- We propose VAEPP framework to use existing techniques of VAE, *e.g.*, flow posterior, WGAN, *e.g.*, gradient penalty strategy, and Langevin dynamics to improve the log-likelihood, quality of sampling and stability of training.
- In MNIST and CIFAR-10, the log-likelihood of VAEPP outperforms models without autoregressive components and is comparable to autoregressive models. In MNIST, Fashion-MNIST, CIFAR-10 and CelebA, the FID of VAEPP is comparable to GANs and SOTA of VAEs.

2 Background

2.1 VAEs and learnable priors

Many generative models aim to minimize the KL-divergence between the empirical distribution $p^*(x)$ and the model distribution $p_\theta(x)$, which leads to maximization likelihood estimation. The vanilla VAE [Kingma and Welling, 2014] models

the joint distribution $p_\theta(x, z)$ and the marginal distribution by $p_\theta(x) = \int p_\theta(x, z)\mathrm{d}z$. VAE applies variational inference to obtain the evidence lower bound objective (ELBO):

$$\ln p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\ln p_\theta(x|z) + \ln p_\theta(z) - \ln q_\phi(z|x)] \triangleq \mathcal{L}(x; \theta, \phi) \quad (1)$$

where $q_\phi(z|x)$ is the variational encoder and $p_\theta(x|z)$ is the generative decoder. The training objective of VAE is $\mathbb{E}_{p^*(x)}[\mathcal{L}(x; \theta, \phi)]$ and it is optimized by SGVB with the reparameterization trick. In vanilla VAE, the prior $p_\theta(z)$ is chosen as the standard Gaussian distribution.

Recently, [Tomczak and Welling, 2018] showed that the simplistic prior could lead to underfitting. Since then many learnable priors are proposed to enrich the prior. Most of them focused on the aggregated posterior $q_\phi(z)$, which was shown to be the optimal prior that maximizes ELBO according to [Tomczak and Welling, 2018] where $p_\lambda(z)$ denotes the learnable prior:

$$\mathcal{L}(\theta, \phi, \lambda) = \mathbb{E}_{p^*(x)}\mathbb{E}_{q_\phi(z|x)}[\ln p_\theta(x|z)] + \mathbb{E}_{p^*(x)}[\mathbb{H}[q_\phi(z|x)]] + \mathbb{E}_{q_\phi(z)}\ln p_\lambda(z) = \mathcal{I} + \mathcal{J} + \mathcal{K} \quad (2)$$

We use $\mathcal{I}, \mathcal{J}, \mathcal{K}$ to denote 3 terms respectively for short thereafter. Notice that $p_\lambda(z)$ only appears in the last term $\mathcal{K}$ and the optimal solution of $p_\lambda(z)$ is $q_\phi(z)$. [Tomczak and Welling, 2018; Takahashi *et al.*, 2019] obtained an approximation of $q_\phi(z)$ with their proposed prior, but reached limited performance.

2.2 GANs and Wasserstein distance

In vanilla GAN [Goodfellow *et al.*, 2014], a generator is trained to generate samples for deceiving the discriminator, and a discriminator is trained to distinguish generated samples and real samples. However, vanilla GAN is unstable during the training process. To tackle this problem, Wasserstein distance is introduced by WGAN [Arjovsky *et al.*, 2017]. Wasserstein distance $W^1(\mu, \nu)$ between measures μ, ν is:

$$W^1(\mu, \nu) = \sup_{Lip(D)\leq 1}\{\mathbb{E}_{\mu(x)}D(x) - \mathbb{E}_{\nu(x)}D(x)\} \quad (3)$$

where $Lip(D) \leq 1$ means that D is 1-Lipschitz. WGAN is optimized by minimizing $W^1(p^*, p_\theta)$ which can be seen as a min-max optimization.

WGAN makes progress toward stable training but sometimes fails to converge since it uses weight clipping to achieve the Lipschitz constraint. WGAN-GP [Gulrajani *et al.*, 2017] and WGAN-div [Wu *et al.*, 2018] improved WGAN by gradient penalty techniques, to achieve a more stable training.

3 Pull-back Prior

3.1 Intuition of Pull-back Prior

The formula of Pull-back Prior is given by:

$$p_\lambda(z) = \frac{1}{Z}p_\mathcal{N}(z) \cdot e^{-\beta D(G(z))} \quad (4)$$

where $p_\mathcal{N}$ is a simple prior, D is a discriminator, G is a generator, $f_\lambda(z)$ denotes $p_\mathcal{N}(z)e^{-\beta D(G(z))}$, Z is the partition function $Z = \int_\mathcal{Z} f_\lambda(z)\mathrm{d}z$ and β is a learnable scalar.

Figure 1: The discriminators on above images (generated by linear interpolation of two sample from $q_\phi(z)$), are better at both sides and worse at the middle, which validates the intuition that a discriminator can assess the quality of images. Moreover, the density of z which generates better images will increase, and the density of z which generates worse images will decrease.

A design proposition of Pull-back Prior is that we increase $p_\lambda(z)$ where z generates better data and decrease $p_\lambda(z)$ where z generates worse data. In Pull-back Prior, D is a discriminator to assess the quality of x, where smaller $D(x)$ indicates x being more similar to real data, as shown in fig. 1. Such discriminator $D(x)$ is defined on x, and the pull-back discriminator on z is defined by $D(G(z))$, where $D(G(z))$ represents the ability of z that can generate data with high quality. To increase $p_\lambda(z)$ at the better z and decrease $p_\lambda(z)$ at the worse z, we modify $p_\mathcal{N}(z)$ by $\beta D(G(z))$, and then normalize it by Z. Finally, we obtain the basic formula of Pull-back Prior.

The theoretical derivation for Pull-back Prior is provided in appendix A. However, it remains questions about how to obtain D and G, determine β, and calculate Z.

3.2 How to obtain D and G

We choose $G(z) = \mathbb{E}_{p_\theta(x|z)}x$ in our model, *i.e.*, the mean of the $p_\theta(x|z)$. Notice that $p_\theta(x|z)$ is chosen to be a Discretized Logistic [Salimans *et al.*, 2017] or a Bernouli in our experiments. $G(z)$ is generated by a neural network and it is set as the mean of $p_\theta(x|z)$.

D plays an important role in Pull-back Prior. We shall propose two ways to obtain D in section 4.1 and section 4.2, and compare them later in our experiments.

3.3 How to determine β

Theoretically, β in eq. (4) represents how far p_λ is from $p_\mathcal{N}$, as proved in appendix A. To maximize ELBO, we can obtain the optimal β by:

$$\beta = \arg\min_\beta \mathcal{L}(\theta, \phi, \lambda) = \arg\min_\beta \mathcal{L}(\theta, \phi, \beta, \omega) \quad (5)$$

The gradient $\partial\mathcal{L}/\partial\beta$ is:

$$\begin{aligned}\frac{\partial \ln Z}{\partial \beta} &= \frac{1}{Z}\int_\mathcal{Z} p_\mathcal{N}(z)e^{-\beta D(G(z))} \cdot (-D(G(z)))\mathrm{d}z \\ &= \mathbb{E}_{p_\lambda(z)}[-D(G(z))] \\ \frac{\partial \mathcal{L}}{\partial \beta} &= \mathbb{E}_{q_\phi(z)}[-D(G(z))] - \frac{\partial \ln Z}{\partial \beta} \\ &= -\mathbb{E}_{q_\phi(z)}[D(G(z))] + \mathbb{E}_{p_\lambda(z)}[D(G(z))] \end{aligned} \quad (6)$$

The 1st term in eq. (6) is the mean of the discriminator on reconstructed data (reconstructed data are nearly same as real data in VAE, after only few epochs in training). The 2nd term in eq. (6) is the mean of the discriminator on data generated from p_λ. Hence, $\partial\mathcal{L}/\partial\beta = 0$ means that the discriminator can't distinguish reconstructed data and generated data when the training converges. It coincides with the philosophy of GANs that the discriminator can't distinguish the real data and generated data when the generator is well-trained.

Noticing that $p_{\mathcal{N}}$ is a special case of p_λ where $\beta = 0$, we shall compare them in experiments.

3.4 The upper-bound of Z

It is difficult to calculate the exact partition function Z. Fortunately for VAE, it is acceptable to obtain an upper-bound of Z, denoted by $\hat{Z}$. Using the upper-bound $\hat{Z}$ in training and evaluation, we can obtain lower-bounds of log-likelihood and ELBO (notice $\hat{p}_\theta(x) \le p_\theta(x)$ means $\ln \hat{p}_\theta(x) \le \ln p_\theta(x)$):

$$\hat{p}_\theta(x) = \int \frac{p_\theta(x|z) f_\lambda(z)}{\hat{Z}} \mathrm{d}z \le \int \frac{p_\theta(x|z) f_\lambda(z)}{Z} \mathrm{d}z = p_\theta(x)$$

$$\hat{\mathcal{K}} = \mathbb{E}_{q_\phi(z)} \ln \frac{1}{\hat{Z}} f_\lambda(z) \le \mathbb{E}_{q_\phi(z)} \ln \frac{1}{Z} f_\lambda(z) = \mathcal{K}$$

$$\hat{\mathcal{L}} = \mathcal{I} + \mathcal{J} + \hat{\mathcal{K}} \le \mathcal{I} + \mathcal{J} + \mathcal{K} = \mathcal{L}$$

The upper-bound $\hat{Z}$ is derived as follows:

$$\hat{Z} = \mathbb{E}_{p^*(x)} \mathbb{E}_{q_\phi(z|x)} \frac{f_\lambda(z)}{\frac{1}{N} q_\phi(z|x)} \ge \mathbb{E}_{q_\phi(z)} \frac{f_\lambda(z)}{q_\phi(z)} = Z \quad (7)$$

The fact that $\hat{Z}$ is an upper-bound of Z comes from:

$$\frac{q_\phi(z|x)}{N} \le \frac{1}{N} \sum_{i=1}^{N} q_\phi(z|x^{(i)}) \approx \mathbb{E}_{p^*(x)} q_\phi(z|x) = q_\phi(z)$$

In previous VAE literatures and our paper, it is a common practice to dynamically sample 0/1 binary images (which is exactly the x of our VAE and may other paper's) from real-value grayscale images (whose distribution is denoted by $p^*(e)$). Each pixel value of e is normalized to $[0, 1]$, and is used as the probability of the corresponding pixel of x being 1 (denoted by $p^*(x|e)$). In such situation, even when the size M of original grayscale image dataset is moderate, the size N of the sampled images dataset is exponentially large. Hence, we shall severely underestimate $q_\phi(z)$ if directly using eq. (7). Because of this, we consider to use $p^*(e)$ instead of $p^*(x)$ to estimate $q_\phi(z)$ in such datasets (called Bernouli datasets in our paper). Given that $p^*(x) = \mathbb{E}_{p^*(e)} p^*(x|e)$, we shall have:

$$q_\phi(z) = \mathbb{E}_{p^*(e)} \mathbb{E}_{p^*(x|e)} q_\phi(z|x) = \mathbb{E}_{p^*(e)} q_\phi(z|e) \quad (8)$$

where $q_\phi(z|e)$ denotes $\mathbb{E}_{p^*(x|e)} q_\phi(z|x)$. eq. (8) suggests that we may train a variational encoder $q_\phi(z|e)$ instead of $q_\phi(z|x)$ along with a generative decoder $p_\theta(x|z)$, while the log-likelihood estimator is still correct:

$$\mathbb{E}_{p^*(x)} \log p_\theta(x) = \mathbb{E}_{p^*(e)} \mathbb{E}_{p^*(x|e)} \log p_\theta(x)$$
$$= \mathbb{E}_{p^*(e)} \mathbb{E}_{p^*(x|e)} \log \mathbb{E}_{q_\phi(z|e)} \frac{p_\theta(x,z)}{q_\phi(z|e)}$$

Based on this idea, we then derive $\hat{Z}$ and ELBO as:

$$\hat{Z} = \mathbb{E}_{p^*(e)} \mathbb{E}_{q_\phi(z|e)} \frac{f_\lambda(z)}{\frac{1}{M} q_\phi(z|e)} \ge \mathbb{E}_{q_\phi(z)} \frac{f_\lambda(z)}{q_\phi(z)} = Z$$

$$\mathbb{E}_{p^*(x)} \ln p_\theta(x) = \mathbb{E}_{p^*(e)} \mathbb{E}_{p^*(x|e)} \ln \mathbb{E}_{q_\phi(z|e)} \frac{p_\theta(x|z) p_\lambda(z)}{q_\phi(z|e)}$$

$$\ge \mathbb{E}_{p^*(e)} \mathbb{E}_{p^*(x|e)} \mathbb{E}_{q_\phi(z|e)} \ln \frac{p_\theta(x|z) p_\lambda(z)}{q_\phi(z|e)} \quad (9)$$

$$= \mathbb{E}_{p^*(x)} \ln p^*(x) - \mathbb{E}_{p^*(e)} \mathbb{E}_{p^*(x|e)} KL(q_\phi(z|e), p_\theta(z|x))$$

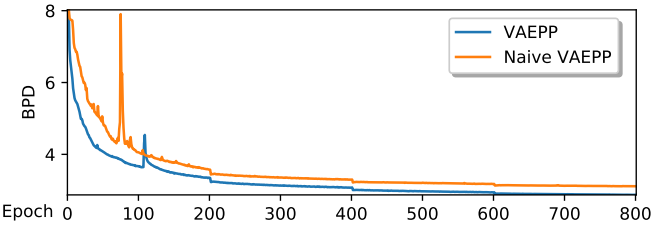

Figure 2: Training loss of Naive VAEPP and VAEPP on CIFAR-10. Naive VAEPP is more unstable and nearly crashes at 80 epoch while VAEPP has a little acceptable gap. From global view, the training loss of VAEPP is more smooth than Naive VAEPP and is better than Naive VAEPP's over almost all training process, which validates the motivation in section 4.2. There are little gaps at per 200 epoch because learning rate is reduced to half at every 200 epoch.

eq. (9) is similar to the original ELBO, and the conclusions in this paper hold for eq. (9) by repeating derivations for eq. (9). $\mathcal{L}(\theta, \phi, \lambda)$ denotes eq. (9) in Bernouli datasets. For other datasets, the distribution $p^*(e)$ and $p^*(x)$ is same, and we shall use $p^*(x|e) = \delta(x - e)$, where δ is Dirac delta function.

Review the estimation of Z. By the theory of importance sampling, p_λ is the optimal choice for the proposal distribution in the estimation of Z. However, it is intractable to sample from p_λ. [Bauer and Mnih, 2019] uses $p_{\mathcal{N}}$ as the proposal distribution to estimate Z but when $KL(p_{\mathcal{N}}, p_\lambda)$ is high, the variance of this estimation will be large.

In our experiments, $KL(q_\phi, p_\lambda)$ is much smaller than $KL(p_{\mathcal{N}}, p_\lambda)$. Therefore, we choose $q_\phi(z)$ as the proposal distribution and replace $q_\phi(z)$ by $\hat{q}_\phi(z|x)$, to obtain the lower-bound of $\hat{Z}$ in eq. (7). The variance of $\hat{Z}$ is acceptable in experiments. In training, $p_{\mathcal{N}}(z)$ could be used together with $q_\phi(z)$, as the proposal distributions, since $KL(p_{\mathcal{N}}, p_\lambda)$ is small in the beginning of training.

4 Training and Sampling

In this section, we will propose two training methods and a sampling method for VAEPP. The main difference between these two trainings method is how to train the discriminator.

4.1 2-step training for VAEPP

The discriminator should be obtained by $W^1(p_\theta, p^*)$, as WGAN [Arjovsky *et al.*, 2017] did. However in VAEPP, p_θ is intractable for sampling, since $p_\theta(x) = \mathbb{E}_{p_\lambda(z)} p_\theta(x|z)$ and $p_\lambda(z)$ is intractable for sampling.

When β is small enough, $p_\lambda(z)$ is near to $p_{\mathcal{N}}(z)$ which is feasible for sampling. Then, $p_\theta(x)$ is near to $p^\dagger(x)$, where $p^\dagger(x) = \mathbb{E}_{p_{\mathcal{N}}(z)} p_\theta(x|z)$ and $p^\dagger(x)$ is feasible for sampling. Therefore, we suspect that $W^1(p_\theta, p^*)$ could be replace by $W^1(p^\dagger, p^*)$, which is feasible since $p^\dagger(x), p^*(x)$ are feasible for sampling. To ensure β is small, β is limited by a hyper-parameter. In this way, an discriminator D is trained by:

$$W^1(p^\dagger, p^*) = \sup_{Lip(D) \le 1} \mathbb{E}_{p^\dagger(x)} D(x) - \mathbb{E}_{p^*(x)} D(x)$$

The other parameters of VAEPP is trained by SGVB:

$$\max_{\theta, \phi, \beta} \mathcal{L}(\theta, \phi, \beta, \omega)$$

Algorithm 1 2-step training algorithm for VAEPP

Require: The gradient penalty algorithm R, the batch size b, the number of critic iterations per generator iteration n_c, the parameters for Adam Optimizers, τ.

while $\theta, \phi, \beta, \omega$ have not converged **do**
  for $k = 1, \dots n_c$ **do**
    for $i = 1, \dots, b$ **do**
      Sample $e, x \sim p^*$, $z \sim q_\phi(z|e)$, $\epsilon \sim p_{\mathcal{N}}$
      $Z^{(i)} \leftarrow \frac{1}{2}(e^{-\beta D(G(\epsilon))} + \frac{f_\lambda(z)}{\frac{1}{M} q_\phi(z|e)})$
      $\mathcal{L}^{(i)} \leftarrow \ln p_\theta(x|z) + \ln f_\lambda(z) - \ln q_\phi(z|e)$
    end for
    $\mathcal{L} \leftarrow \frac{1}{b} \sum_i^b \mathcal{L}^{(i)} - \ln(\frac{1}{b} \sum_i^b Z^{(i)})$
    $\theta, \phi, \beta \leftarrow \text{Adam}\ (\nabla_{\theta,\phi,\beta} \mathcal{L}, \{\theta, \phi, \beta\}, \tau)$
  end for
  for $i = 1, \dots, b$ **do**
    Sample $e, x \sim p^*$, latent variable $z \sim p_{\mathcal{N}}$
    $\hat{e} = G(z)$, get gradient penalty term $\zeta \leftarrow R(e, \hat{e})$
    $L^{(i)} \leftarrow D(\hat{x}) - D(x) + \zeta$
  end for
  $\omega \leftarrow \text{Adam}\ (\nabla_\omega \frac{1}{b} \sum_i^b L^{(i)}, \omega, \tau)$
end while

Above two training process run alternatively, which is called the 2-step training algorithm for VAEPP, in algorithm 1. The model trained by 2-step training algorithm is called Naive VAEPP.

4.2 1-step training for VAEPP

However, the training process of algorithm 1 is unstable and inefficient, as shown in fig. 2. We suspect that the two independent optimizations instead of one whole optimization, may lower the log-likelihood and stability. Therefore, we try to combine the training for $\theta, \phi, \beta, \omega$ into a whole optimization. Our solution is to use SGVB with the gradient penalty term to train VAEPP:

$$\max_{\theta,\phi,\beta} \max_{Lip(D) \le 1} \mathcal{L}(\theta, \phi, \beta, \omega)$$

The optimization $\max_{Lip(D)\le 1} \mathcal{L}(\theta, \phi, \beta, \omega)$ is equivalent to optimization $\max_{Lip(D)\le 1} \{-\mathbb{E}_{q_\phi(z)}\, \beta * D(G(z)) - \ln Z\}$. The latter is a lower-bound of $\beta W^1(p^\dagger, p^*)$:

$$\begin{aligned}
&\max_{Lip(D)\le 1} \{-\mathbb{E}_{q_\phi(z)}\, \beta * D(G(z)) - \ln Z\} \\
&\le \beta \max_{Lip(D)\le 1} \{\mathbb{E}_{p_{\mathcal{N}}(z)}\, D(G(z)) - E_{q_\phi(z)} D(G(z))\} \\
&= \beta \max_{Lip(D)\le 1} \{\mathbb{E}_{p^\dagger(x)}\, D(x) - E_{p_r(x)} D(x)\} \\
&= \beta W^1(p^\dagger, p_r) \approx \beta W^1(p^\dagger, p^*) \qquad (10)
\end{aligned}$$

where $p_r(x) = \mathbb{E}_{q_\phi(z)}\, p_\theta(x|z)$ and the inequality of $\ln Z$ is:

$$\ln Z = \ln \mathbb{E}_{p_{\mathcal{N}}(z)}\, e^{-\beta * D(G(z))} \ge \mathbb{E}_{p_{\mathcal{N}}(z)}[-\beta * D(G(z))]$$

The last approximation sign in eq. (10) is from our expectation that $p_r \to p^*$, which is also observed in experiments, after few epochs in training. eq. (10) also uses the assumption $\mathbb{E}_{p_\theta(x|z)}\, D(x) = D(\mathbb{E}_{p_\theta(x|z)}\, x) = D(G(z))$ introduced in appendix A to simplify equation at the 2nd column.

Algorithm 2 1-step training algorithm for VAEPP

Require: The gradient penalty algorithm R, the batch size b, the parameters for Adam Optimizers, τ.

while $\theta, \phi, \beta, \omega$ have not converged **do**
  for $i = 1, \dots, b$ **do**
    Sample $e, x \sim p^*$, $z \sim q_\phi(z|e)$, $\epsilon \sim p_{\mathcal{N}}$
    $\hat{e} = G(\epsilon)$, get gradient penalty term $\zeta \leftarrow R(e, \hat{e})$
    $Z^{(i)} \leftarrow \frac{1}{2}(e^{-\beta D(G(\epsilon))} + \frac{f_\lambda(z)}{\frac{1}{M} q_\phi(z|e)})$
    $\mathcal{L}^{(i)} \leftarrow \ln p_\theta(x|z) + \ln f_\lambda(z) - \ln q_\phi(z|e) + \beta\zeta$
  end for
  $\mathcal{L} \leftarrow \frac{1}{b} \sum_i^b \mathcal{L}^{(i)} - \ln(\frac{1}{b} \sum_i^b Z^{(i)})$
  $\theta, \phi, \beta, \omega \leftarrow \text{Adam}\ (\nabla_{\theta,\phi,\beta} \mathcal{L}, \{\theta, \phi, \beta, \omega\}, \tau)$
end while

eq. (10) indicates that it is reasonable to obtain discriminator D during optimizing section 4.2. eq. (10) also indicates that the gradient penalty term should be multiplied by β. Finally, the optimizations for θ, ϕ, β and ω are combined into one, which is called the 1-step training algorithm for VAEPP, in algorithm 2. The model trained by 1-step training algorithm is called VAEPP.

4.3 Sampling from VAEPP

We apply Langevin dynamics to sample z from $p_\lambda(z)$. It could generate natural and sharp images and only requires that $\nabla_z \log p_\lambda(z)$ is computable and $p_\lambda(z_0)$ is high enough where z_0 is the initial point of Langevin dynamics [Song and Ermon, 2019]. Moreover, [Kumar *et al.*, 2019] has implemented a Metropolis-Adjusted Langevin Algorithm (MALA) for sampling, where the formula of density also contains a discriminator term. But how to obtain the initial z_0 whose density is high enough is still a problem.

Following the philosophy of VAEPP, *i.e.*, using the technique of GANs to assist VAEs, it is natural to use a GAN to model the distribution $q_\phi(z)$, and use samples of the GAN as the initial z_0 for MALA.

The sampling of VAEPP consists of 3 parts:

1. generate initial z_0 by a GAN modeling $q_\phi(z)$
2. generate $z \sim p_\lambda(z)$ from initial z_0 by MALA
3. generate image from z with a decoder

This sampling process is similar to 2-Stage VAE [Dai and Wipf, 2019]. The main difference between them is that VAEPP applies Langevin dynamics to sample from the explicit prior but 2-Stage VAE doesn't, since the prior of 2-Stage VAE is implicit. In experiments, we found that sampling from the explicit learnable prior might improve the quality of sampling in some datasets.

Accept/Reject Sampling (ARS) [Bauer and Mnih, 2019] is useless for p_λ because ARS requires that $p_\lambda(z)/p_{\mathcal{N}}(z)$ is bounded by a constant M for any z, such that a sample could be sampled in M times. But it is hard to ensure that there exists an enough small M in VAEPP.

5 Experiments

VAEPP is evaluated in common datasets including MNIST, Static-MNIST[Larochelle and Murray, 2011], Fashion-

Model	MNIST	CIFAR
With autoregressive		
PixelCNN	81.30	3.14
DRAW	80.97	3.58
IAFVAE	79.88	3.11
PixelVAE++	78.00	2.90
PixelRNN	79.20	3.00
VLAE	79.03	2.95
PixelSNAIL		2.85
PixelHVAE with VampPrior	78.45	
Without autoregressive		
Implicit Optimal Priors	83.21	
Discrete VAE	81.01	
LARS	80.30	
VampPrior	79.75	
BIVA	78.59	3.08
Naive VAEPP	76.49	3.15
VAEPP	76.37	2.91
VAEPP+Flow	76.23	2.84

Table 1: Test NLL on MNIST and Bits/dim on CIFAR-10. Bits/dim means $-\log p_\theta(x|z)/(3072 * \ln(2))$. The data are from [Maaløe *et al.*, 2019], [Chen *et al.*, 2018], [Tomczak and Welling, 2018], [Bauer and Mnih, 2019] and [Takahashi *et al.*, 2019]. VAEPP+Flow means VAEPP with a normalization flow on encoder. The posterior on CIFAR-10 is Discretized Logistic. Additional, we compare VAE based on $q_\phi(z|x)$ and $q_\phi(z|e)$ on MNIST, whose NLL are 81.10 and 83.30 respectively. Moreover, evaluation using importance sampling based on $q_\phi(z|e)$ has enough small standard deviation (0.01) with 10^8 samples altogether. It validates that $q_\phi(z|e)$ is stable for evaluation and doesn't improve the performance. VAEPP reaches SOTA without autoregressive component, and is comparable to models with autoregressive component.

MNIST [Xiao *et al.*, 2017], Omniglot [Lake *et al.*, 2015], and CIFAR-10 [Krizhevsky *et al.*, 2009] with log-likelihood. The quality of sampling of VAEPP is evaluated in MNIST, Fashion-MNIST, CIFAR-10 and CelebA [Liu *et al.*, 2015], with FID [Heusel *et al.*, 2017].

5.1 Log-likelihood Evaluatoin

We compare our algorithms with other models based on log-likelihood, on MNIST and CIFAR-10 as shown in table 1, and on Static-MNIST, Fashion-MNIST, and Omniglot, as shown in table 2. Because the improvement of auto-regressive components is significant, we separate models by whether they use an auto-regressive component. VAEPP outperforms most of the models without autoregressive component and is comparable to the models with autoregressive component. The reason of why VAEPP doesn't use an auto-regressive component is that VAEPP is time-consuming in training, evaluation and sampling due to the huge structure (need additional discriminator) and Langevin dynamics. It is not easy to apply an auto-regressive component on VAEPP since auto-regressive component is also time-consuming. Therefore, how to apply an autoregressive component on VAEPP is a valuable and challenging practical work and we leave it for future work.

Model	Static MNIST	Fashion	Omniglot
Naive VAEPP	78.06	214.63	90.72
VAEPP	77.73	213.24	89.60
VAEPP+Flow	77.66	213.19	89.24

Table 2: Test NLL on Static MNIST, Fashion-MNIST and Omniglot.

GP Strategy	Naive VAEPP	VAEPP
WGAN-GP	3.15	2.95
WGAN-div-1	3.20	2.91
WGAN-div-2	4.47	2.99

Table 3: Comparison between Naive VAEPP and VAEPP when gradient penalty strategy varies on CIFAR-10 with dim $\mathcal{Z} = 1024$. For any gradient penalty strategy in the table, VAEPP outperforms Naive VAEPP, which validates the our intuition of design of algorithm 2. WGAN-div-1 is chosen as our default gradient penalty strategy since it reaches best performance in VAEPP.

We evaluate and compare the performance of Naive VAEPP trained by algorithm 1 and VAEPP trained by algorithm 2 on CIFAR-10, as the gradient penalty algorithm is chosen from 3 strategies: WGAN-GP, WGAN-div-1 (sampling the linear interpolation of real data and generated data) and WGAN-div-2 (sampling real data and generated data), as shown in table 3.

To validate that it is better to use $q_\phi(z)$ to evaluate Z than $p_\mathcal{N}(z)$ in section 3.4, we calculate the $KL(q_\phi(z)||p_\lambda(z))$ and $KL(p_\mathcal{N}(z)||p_\lambda(z))$ on CIFAR-10 and MNIST. The former is smaller than $\mathcal{L} - \mathcal{I}$ [Hoffman and Johnson, 2016](180.3 on CIFAR-10 and 12.497 on MNIST), and the latter can be evaluated directly (1011.30 on CIFAR-10 and 57.45 on MNIST). Consequently, $q_\phi(z)$ is much closer to $p_\lambda(z)$ than $p_\mathcal{N}(z)$.

To ensure the variance of estimation $\hat{Z}$ is small enough, the $q_\phi(z|e)$ is chosen as truncated normal distribution (drop the sample whose magnitude is more than 2 standard deviation from the mean) instead of normal distribution, which may reduce the gap between $q_\phi(z)$ and $\hat{q}_\phi(z)$. With 10^9 samples, the variance of $\hat{Z}$ with truncated normal and normal are 0.000967 (truncated normal) and 0.809260 (normal) respectively in MNIST. Therefore, truncated normal is chosen as default setting.

5.2 Quality of Sampling

As a common sense, the quality of sampling of VAEs is worse than GANs, and it is indeed a reason that we involve the techniques of GAN to improve VAE model: We use the discriminator to adjust learnable prior and a GAN to sample the initial z_0 for Langevin dynamics. These techniques will help VAEPP improve the quality of samples. The samples of VAEPP gets good FID, comparable to GANs and 2-Stage VAE (which is the SOTA of VAE in FID), as shown in table 4. Some generated images are shown in fig. 3.

It is hard to reach best FID, IS [Salimans *et al.*, 2016] and log-likelihood simultaneously with the same setting. We observe the fact that when $\dim \mathcal{Z}$ (the dimension of latent space) increases, the trends of FID and IS are greatly different to log-likelihood's, as shown in fig. 4. As diagnosis in [Dai and

Figure 3: Examples of generated images.

Model	MNIST	Fashion	CIFAR	CelebA
Best GAN	~ 10	~ 32	~ 70	~ 49
VAE+Flow	54.8	62.1	81.2	65.7
WAE-MMD	115.0	101.7	80.9	62.9
2-StageVAE	12.6	29.3	72.9	44.4
GAN-VAEPP	12.7	26.4	74.1	53.4
VAEPP	12.0	26.4	71.0	53.4

Table 4: FID comparison of GANs and VAEs. Best GAN indicates the best FID on each dataset across all GAN models when trained using settings suggested by original authors. VAEPP uses Bernouli as posterior on MNIST and Discretized Logistic on others. GAN-VAEPP indicates that image is directly sampled from z_0, generated by GAN without Langevin dynamics. The data of Best GAN and other VAEs is from [Dai and Wipf, 2019]. In experiments, we found that the FID of VAEPP is usually better than GAN-VAEPP, which means that the explicit prior and Langevin dynamics might be useful for improving the quality of sampling in some datasets.

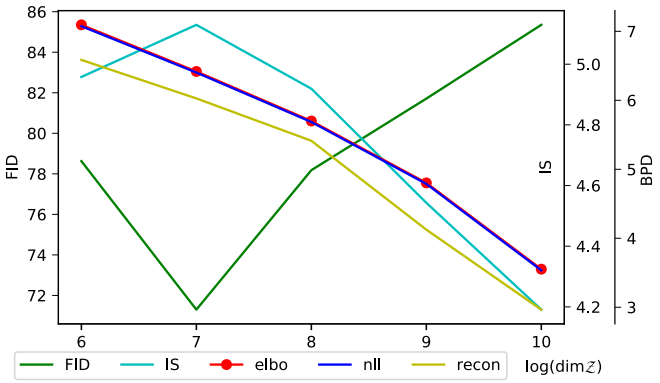

Figure 4: Comparison of VAEPP with a learnable scalar γ (variance of $p_\theta(x|z)$), as the dimension of latent space varies on CIFAR-10, with metrics BPD, FID and IS. FID and BPD is better when it is smaller and IS is better when it is larger. When $\dim \mathcal{Z}$ is greater than 128, the quality of sampling becomes worse and BPD becomes better as $\dim \mathcal{Z}$ increases. It validates the proposition that $\dim \mathcal{Z}$ should be chosen as a minimal number of active latent dimensions in [Dai and Wipf, 2019]. The reconstruction term is optimized more as $\dim \mathcal{Z}$ increases, because larger latent space could keep more information. Meanwhile, the $KL(q_\phi(z)||p_\lambda(z))$ (Bounded by the difference between ELBO and reconstruction term) increases not much. It also shows an interesting phenomenon that the trends of FID and IS, are not always same as BPD, maybe greatly different.

Wipf, 2019], the variance of $p_\theta(x|z)$ is chosen as a learnable scalar γ, and the $\dim \mathcal{Z}$ is chosen as a number, a little larger than the dimension of real data manifold. $\dim \mathcal{Z} = 128$, by our experimental results.

In this section, for better understanding, the values of discriminator in training dataset are normalized into $\mathcal{N}(0, 1)$.

To validate the eq. (6), we calculate the $\mathbb{E}_{p_\lambda(z)}[D(G(z))]$ (discriminator on generated samples) and $\mathbb{E}_{q_\phi(z)}[D(G(z))]$ (discriminator on reconstructed samples). They are 0.092 and 0.015 respectively on CIFAR-10, which means discriminator on generated samples and reconstructed samples are nearly same as the discriminator on real data.

To validate the assumption introduced in appendix A indeed holds in practical experiment, $|\mathbb{E}_{p_\theta(x|z)} D(x) - D(G(z))|$ is calculated and it is an acceptable value (0.019) on CIFAR-10.

6 Conclusion

We propose a novel learnable prior, Pull-back Prior, for VAE, by adjusting the prior through a discriminator assessing the quality of data, with a solid derivation and an intuitive explanation. We propose an efficient and stable training method for VAE with Pull-back Prior, by mixing the optimizations of WGAN and VAE into one. VAEPP is evaluated on common datasets, and shows impressive performance in log-likelihood and quality of sampling. We believe that this paper could lead VAE models into a new stage, with a clear formula, a general framework and powerful performance.

A Derivation of Pull-back Prior

For any given θ, ϕ, search the optimal prior that minimizes the Wasserstein distance between p_θ and p^*:

$$\min_\lambda \sup_{Lip(D)\le 1} \{\mathbb{E}_{p_\lambda(z)} \mathbb{E}_{p_\theta(x|z)} D(x) - \mathbb{E}_{p^*(x)} D(x)\} \quad (11)$$

We apply an assumption $\mathbb{E}_{p_\theta(x|z)} D(x) = D(G(z))$ (it indeed defines a discriminator on e by $D(e) = \mathbb{E}_{p*(x|e)} D(x)$ in Bernouli dataset) and an approximation D to simplify it. The D in eq. (11) could be replaced by an approximation D in $W^1(p^\dagger, p^*)$, if p_λ is near $p_\mathcal{N}$, as section 4.1 and section 4.2 does. The simplified optimization is:

$$\min_\lambda \{\mathbb{E}_{p_\lambda(z)} D(G(z)) - \mathbb{E}_{p^*(x)} D(x)\} \quad (12)$$

$$\textbf{s.t. } KL(p_\lambda, p_\mathcal{N}) = \alpha, \qquad \int_\mathcal{Z} p_\lambda(z)\mathrm{d}z = 1$$

It could be solved by Lagrange multiplier method introduced by calculus of variation [Gelfand *et al.*, 2000]. The Lagrange function with Lagrange multiplier η, γ is:

$$F(p_\lambda, \eta, \gamma) = \mathbb{E}_{p_\lambda(z)} D(G(z)) - \mathbb{E}_{p^*(x)} D(x) + \eta(\int_\mathcal{Z} p_\lambda(z)\mathrm{d}z - 1) + \gamma(KL(p_\lambda, p_\mathcal{N}) - \alpha) \quad (13)$$

We solve eq. (13) by Euler-Lagrange equation:

$$\ln p_\lambda(z) = \frac{1}{\gamma} D(G(z)) + \ln p_\mathcal{N}(z) + (\frac{\eta}{\gamma} - 1) \quad (14)$$

where γ is determined by α and η is determined from condition $\int_\mathcal{Z} p_\lambda(z)\mathrm{d}z = 1$. Consequently, β is determined by α, representing how far p_λ is from $p_\mathcal{N}$. In eq. (12), α is static and should be searched as an appropriate value, *i.e.*, β should be searched, as section 3.3 does.

References

[Arjovsky *et al.*, 2017] Martin Arjovsky, Soumith Chintala, et al. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223, 2017.

[Bauer and Mnih, 2019] Matthias Bauer and Andriy Mnih. Resampled priors for variational autoencoders. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 66–75, 2019.

[Chen *et al.*, 2018] Xi Chen, Nikhil Mishra, et al. Pixelsnail: An improved autoregressive generative model. In *International Conference on Machine Learning*, pages 863–871, 2018.

[Dai and Wipf, 2019] Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.

[Dinh *et al.*, 2016] Laurent Dinh, Jascha Sohl-Dickstein, et al. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[Gelfand *et al.*, 2000] Izrail Moiseevitch Gelfand, Richard A Silverman, et al. *Calculus of variations*. Courier Corporation, 2000.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, et al. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, et al. Improved training of wasserstein gans. In *NIPS*, 2017.

[Heusel *et al.*, 2017] Martin Heusel, Hubert Ramsauer, et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[Hoffman and Johnson, 2016] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016.

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

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[Kingma *et al.*, 2016] Durk P Kingma, Tim Salimans, et al. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[Kumar *et al.*, 2019] Rithesh Kumar, Anirudh Goyal, et al. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019.

[Lake *et al.*, 2015] Brenden M Lake, Ruslan Salakhutdinov, et al. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[Larochelle and Murray, 2011] Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 29–37, 2011.

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, et al. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[Maaløe *et al.*, 2019] Lars Maaløe, Marco Fraccaro, et al. Biva: A very deep hierarchy of latent variables for generative modeling. *arXiv preprint arXiv:1902.02102*, 2019.

[Rezende *et al.*, 2014] Danilo Jimenez Rezende, Shakir Mohamed, et al. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[Salimans *et al.*, 2016] Tim Salimans, Ian Goodfellow, et al. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[Salimans *et al.*, 2017] Tim Salimans, Andrej Karpathy, et al. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[Song and Ermon, 2019] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.

[Takahashi *et al.*, 2019] Hiroshi Takahashi, Tomoharu Iwata, et al. Variational autoencoder with implicit optimal priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5066–5073, 2019.

[Tomczak and Welling, 2016] Jakub M Tomczak and Max Welling. Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

[Tomczak and Welling, 2018] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223, 2018.

[Van den Oord *et al.*, 2016] Aaron Van den Oord, Nal Kalchbrenner, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

[Wu *et al.*, 2018] Jiqing Wu, Zhiwu Huang, et al. Wasserstein divergence for gans. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 653–668, 2018.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, et al. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.