VAEPP: Variational Autoencoder with a Pull-back Prior

Wenxiao Chen[1,2][0000−0001−8852−675X], Wenda Liu[1][0000−0003−4614−7545], Zhenting Cai[1][0000−0003−1965−2060], Haowen Xu[1,2][0000−0003−2841−5788], and Dan Pei[1,2][0000−0002−5113−838X]✉

[1] Department of Computer Science and Technology, Tsinghua University
[2] Beijing National Research Center for Information Science and Technology (BNRist)
{chen-wx17,liuwd17,caizt16,xhw15}@mails.tsinghua.edu.cn
peidan@tsinghua.edu.cn

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 the 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 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.

Keywords: Variational Autoencoder· Deep Generative Model · Adversarial Training.

1 Introduction

How to learn deep generative models that are able to capture complex data patterns 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) [6], flow-based models [11], PixelCNN [20], and Variational Autoencoder (VAE) [10, 21]. GANs achieve SOTA in generative models, but likelihood of GANs are poor or incalculable.

The likelihood is important for generative models. VAE uses the variational inference and re-parameterization trick to optimize the evidence lower bound of log-likelihood (ELBO). In the past, researches [12, 27] focused on enriching the variational posterior, but recently [26] showed that the standard Gaussian prior could lead to underfitting. To enrich the prior, several learnable priors have been

proposed [26, 2, 25]. Most of them focus on approximating aggregated posterior which is the integral of the variational posterior and is the optimal prior that maximizes ELBO. However, existing methods based on the aggregated posterior reach limited performance, and the practical meaning of the aggregated posterior is ambiguous. 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 [10] with gradient penalty terms, which mixes the discriminator and the gradient penalty term [7, 28] into VAE. Compared to AAE [18], VAEPP uses discriminator to adjust learnable prior while AAE uses discriminator to replace $KL(q(z)||p(z))$. Langevin dynamics, provided by [13] is used in VAEPP 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 and quality of sampling.
- 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 [10] models the joint distribution $p_\theta(x, z)$ and the marginal distribution $p_\theta(x) = \int p_\theta(x, z)\mathrm{d}z$. VAE applies variational inference to obtain the evidence lower bound objective (ELBO):

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

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 re-parameterization trick. In vanilla VAE, the prior $p_\theta(z)$ is the standard Gaussian.

Recently, [26] 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 [26]. The training objective with learnable prior $p_\lambda(z)$ is:

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

$\mathcal{I}, \mathcal{J}, \mathcal{K}$ denote 3 terms in eq. (2) 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 $q_\phi(z)$. [26, 25] obtained an approximation of $q_\phi(z)$ with their proposed prior, but reached limited performance.

2.2 GANs and Wasserstein distance

In vanilla GAN [6], 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 [1]:

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

where $Lip(D) \leq 1$ means that D is 1-Lipschitz, and μ, ν are measures. 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 for the Lipschitz constraint. WGAN-GP [7] and WGAN-div [28] improved WGAN by gradient penalty techniques, to achieve a more stable training.

3 Pull-back Prior

3.1 Intuition of Pull-back Prior

Definition 1. *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, β is a learnable scalar, $f_\lambda(z)$ denotes $p_{\mathcal{N}}(z)e^{-\beta D(G(z))}$, and $Z = \int_{\mathcal{Z}} f_\lambda(z)\mathrm{d}z$ is the partition function.

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

Fig. 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, in VAEPP the density of z which generates better images will increase, and the density of z which generates worse images will decrease.

$D(x)$ 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 theorem 2. However, it remains questions about how to obtain D and G, determine β, and calculate Z.

3.2 How to obtain D and G

In our model, $G(z) = \mathbb{E}_{p_\theta(x|z)}\, x$, *i.e.*, the mean of $p_\theta(x|z)$. In our experiments, $p_\theta(x|z)$ is chosen to be a Discretized Logistic [23] or a Bernouli. $G(z)$ 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

To maximize ELBO, we can obtain the optimal β by (λ contains β and ω, where ω denotes the parameters of D):

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

When the training coverages, $\partial\mathcal{L}/\partial\beta = 0$. The gradient $\partial\mathcal{L}/\partial\beta$ is:

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

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 few epochs in training). The 2nd term in eq. (6) is the mean of the discriminator on data generated from p_λ. $\partial\mathcal{L}/\partial\beta = 0$ 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 real data and generated data when the generator is well-trained.

Noticing that $p_{\mathcal{N}}$ is a special case of p_λ where $\beta = 0$, Pull-back Prior is a general form of the standard Gaussian. We shall compare their performance in experiments.

3.4 The upper-bound of Z

It is difficult to calculate the partition function Z exactly. Fortunately for VAEPP, 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 (note, $\hat{p}_\theta(x) \le p_\theta(x)$ indicates $\ln \hat{p}_\theta(x) \le \ln p_\theta(x)$):

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

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

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

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

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

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 [2, 25, 10] and our paper, it is a common practice to dynamically sample 0/1 binary images (which is exactly the x of our VAE and many other paper's) from real-value grayscale images (whose distribution is denoted by $p^*(e)$). Each pixel value of e is normalized into $[0, 1]$, and then 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 overestimate Z since $\frac{1}{N} q_\phi(z|x) \ll q_\phi(z)$ if directly using eq. (7). Therefore, we consider to use $p^*(e)$ instead of $p^*(x)$ to estimate a lower bound of $q_\phi(z)$ in such datasets (called Bernouli datasets in our paper). Given that $p^*(x) = \mathbb{E}_{p^*(e)} p^*(x|e)$, we shall have:

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

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:

$$\begin{aligned}
&\hat{Z} = \mathbb{E}_{p^*(e)}\, \mathbb{E}_{q_\phi(z|e)}\, \frac{f_\lambda(z)}{\frac{1}{M} q_\phi(z|e)} \geq \mathbb{E}_{p^*(e)}\, \mathbb{E}_{q_\phi(z|e)}\, \frac{f_\lambda(z)}{q_\phi(z)} = \mathbb{E}_{q_\phi(z)}\, \frac{f_\lambda(z)}{q_\phi(z)} = Z \\
&\mathbb{E}_{p^*(x)} \ln p_\theta(x) = \mathbb{E}_{p^*(e)}\, \mathbb{E}_{p^*(x|e)} \ln \mathbb{E}_{q_\phi(z|e)}\, \frac{p_\theta(x|z) p_\lambda(z)}{q_\phi(z|e)} \\
&\geq \mathbb{E}_{p^*(e)}\, \mathbb{E}_{p^*(x|e)}\, \mathbb{E}_{q_\phi(z|e)} \ln \frac{p_\theta(x|z) p_\lambda(z)}{q_\phi(z|e)} \qquad (9) \\
&= \mathbb{E}_{p^*(x)} \ln p_\theta(x) - \mathbb{E}_{p^*(e)}\, \mathbb{E}_{p^*(x|e)}\, KL(q_\phi(z|e), p_\theta(z|x))
\end{aligned}$$

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.

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. [2] 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 use $\frac{1}{N} q_\phi(z|x)$ as a lower bound of $q_\phi(z)$, to obtain $\hat{Z}$ 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 propose two training methods and a sampling method for VAEPP. The main difference between two trainings method is how to train the discriminator.

4.1 2-step training for VAEPP

The discriminator should be obtained by $W^1(p_\theta, p^*)$, suggested by WGAN [1]. 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 try to obtain the discriminator by $W^1(p^\dagger, p^*)$ instead. β is limited by a hyper-parameter. In this way, an discriminator D is trained by:

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

The other parameters of VAEPP are trained by SGVB:

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

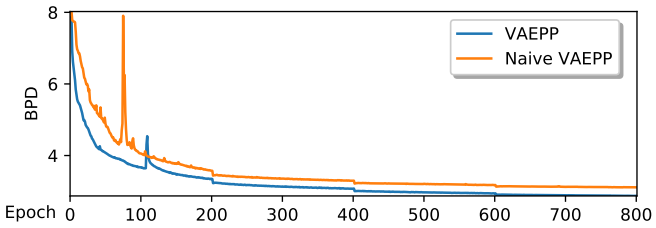

Fig. 2. Training loss of Naive VAEPP and VAEPP on CIFAR-10. Naive VAEPP is more unstable and nearly crashes at 80 epoch while VAEPP has a little acceptable gap. From global view, the training loss of VAEPP is more smooth than Naive VAEPP and is better than Naive VAEPP's over almost all training process, 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.

Above two optimizations run alternatively, as shown 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:

$$\max_{\theta,\phi,\beta} \max_{Lip(D)\leq 1} \mathcal{L}(\theta, \phi, \beta, \omega) \tag{10}$$

theorem 3 in appendix indicates that it is reasonable to obtain discriminator D during optimizing eq. (10), and the gradient penalty term should be multiplied by β. Finally, the optimizations for θ, ϕ, β and ω are combined into one, as shown 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 [24]. Moreover, [13] 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.

The sampling of VAEPP consists of 3 parts: sample initial z_0 by a GAN modeling $q_\phi(z)$; generate $z \sim p_\lambda(z)$ from initial z_0 by MALA; generate image from z with the decoder. This sampling process is similar to 2-Stage VAE [4]. The main difference between them is that VAEPP applies Langevin dynamics

Algorithm 1: 2-step training algorithm for VAEPP

Input: 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
    $\theta, \phi, \beta \leftarrow$ Adam $(\nabla_{\theta,\phi,\beta}(\frac{1}{b}\sum_i^b \mathcal{L}^{(i)} - \ln(\frac{1}{b}\sum_i^b Z^{(i)})), \{\theta, \phi, \beta\}, \tau)$;
  end
  for $i = 1, \dots, b$ **do**
    Sample $e, x \sim p^*$, $z \sim p_{\mathcal{N}}$, $\hat{e} \leftarrow G(z)$;
    get gradient penalty term $\zeta \leftarrow R(e, \hat{e})$;
    $L^{(i)} \leftarrow D(\hat{x}) - D(x) + \zeta$;
  end
  $\omega \leftarrow$ Adam $(\nabla_\omega \frac{1}{b}\sum_i^b L^{(i)}, \omega, \tau)$;
end

to sample from the explicit prior but 2-Stage VAE doesn't, since the prior of 2-Stage VAE is implicit. In experiments, sampling from the explicit prior may improve the quality of sampling in some datasets.

Accept-Reject Sampling [2] is useless for p_λ because it requires that $p_\lambda(z)/p_{\mathcal{N}}(z)$ is bounded by a constant T on the support of p_λ, such that a sample could be accepted in expected T times. But it is hard to ensure that there exists a small T in VAEPP.

5 Experiments

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 [15], Fashion-MNIST [29], and Omniglot [14], 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. 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 challenging practical work

Algorithm 2: 1-step training algorithm for VAEPP

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

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

and we leave it for future work. IvOM [19] of VAEPP reaches 0.018, 0.017 on MNIST, CIFAR-10, which shows good data coverage.

We compare 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 both) 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 $\mathcal{L}-\mathcal{I}$ [9](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 $\frac{1}{M} q_\phi(z|x)$. With 10^9 samples, the variance of $\hat{Z}$ with truncated normal and normal is 0.000967 (truncated normal) and 0.809260 (normal) respectively in MNIST. Therefore, truncated normal is chosen as the 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 [8], comparable to GANs and 2-Stage VAE (SOTA of VAE in FID), as shown in table 4. Some generated images of VAEPP are shown in fig. 3. It is important to notice that the GAN in VAEPP only plays the role that generates z_0 with high $p_\lambda(z_0)$, in latent space with small dimension, instead of image. The ability of VAEPP that generates image from z is totally depend on the decoder.

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

Table 1. Test NLL on MNIST and Bits/dim on CIFAR-10. The data are from [17, 3, 26, 2, 25]. Bits/dim means $-\log p_\theta(x|z)/3072/\ln 2$. VAEPP+Flow means VAEPP with a normalization flow on encoder. The decoder on CIFAR-10 is Discretized Logistic and the decoder on MNIST is Bernouli. 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.

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.

It is hard to reach best FID, IS [22] and log-likelihood simultaneously with one 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 [4], the variance of $p_\theta(x|z)$ is chosen as a learnable scalar γ, and the $\dim \mathcal{Z}$ is chosen as a number, slightly larger than the dimension of real data manifold. In our experiments, VAEPP reaches best FID when $\dim \mathcal{Z} = 128$.

For better understanding, the values of discriminator on training set are normalized into $\mathcal{N}(0, 1)$. To validate the eq. (6), we calculate $\mathbb{E}_{p_\lambda(z)} D(G(z))$

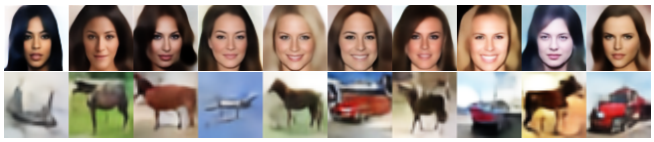

Fig. 3. Examples of generated images from VAEPP on CelebA [16] and CIFAR-10.

GP Strategy	WGAN-GP	WGAN-div-1	WGAN-div-2
Naive VAEPP	3.15	3.20	4.47
VAEPP	2.95	2.91	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 algorithm 2. WGAN-div-1 is chosen as the default gradient penalty strategy since it reaches best performance in VAEPP.

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 [4]. VAEPP uses Bernouli as decoder on MNIST and Discretized Logistic on others. GAN-VAEPP indicates that image is directly sampled from z_0, without Langevin dynamics. 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.

and $\mathbb{E}_{q_\phi(z)} D(G(z))$. They are 0.092 and 0.015 respectively on CIFAR-10, which means discriminator on generated samples and reconstructed samples are nearly same as on real data. To validate the assumption in section 7 holds in experiment, we calculate $|\mathbb{E}_{p_\theta(x|z)} D(x) - D(G(z))|$, which 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 VAEPP, by mixing the optimizations of WGAN and VAE into one. VAEPP shows impressive performance in log-likelihood and quality of sampling on common datasets. We believe that VAEPP could lead VAE models into a new stage, with clearer formula, more general framework and better performance.

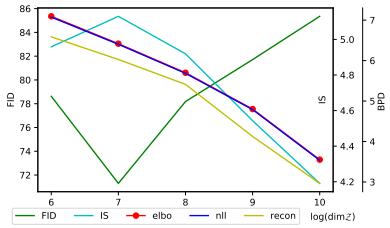

Fig. 4. Comparison of VAEPP with a learnable scalar γ (variance of $p_\theta(x|z)$), as the dimension of latent space varies on CIFAR-10, with metrics BPD, FID and IS. FID and BPD are better when it is smaller and IS is better when it is larger. When dim $\mathcal{Z}$ is greater than 128, the quality 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 [4]. It shows an interesting phenomenon that trends of FID and IS, are not same as BPD, maybe greatly different.

7 Derivation of Pull-back Prior

For any given θ, ϕ, search the optimal prior that minimizes the $W^1(p_\theta, p^*)$:

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

We use an assumption $\mathbb{E}_{p_\theta(x|z)} D(x) = D(G(z))$ 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:

$$\min_{\lambda}\{\mathbb{E}_{p_\lambda(z)} D(G(z)) - \mathbb{E}_{p^*(x)} D(x)\} \quad \textbf{s.t. } KL(p_\lambda, p_\mathcal{N}) = \alpha, \int_\mathcal{Z} p_\lambda(z)\mathrm{d}z = 1$$

Theorem 2. *The optimal solution for this simplified optimization is the Pull-back Prior.*

Proof. It could be solved by Lagrange multiplier method introduced by calculus of variation [5]. The Lagrange function with Lagrange multiplier η, γ is:

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

By Euler-Lagrange equation, the optimal p_λ satisfies $\frac{\delta F}{\delta p_\lambda} = 0$. Therefore, we obtain

$$\frac{\delta F}{\delta p_\lambda} = D(G(z)) + \eta + \gamma \log \frac{p_\lambda(z)}{p_\mathcal{N}(z)} + \gamma p_\lambda(z) * \frac{1}{p_\lambda(z)} = 0$$

Rewritten it into $\ln p_\lambda(z) = -\frac{1}{\gamma} D(G(z)) + \ln p_\mathcal{N}(z) - (\frac{\eta}{\gamma} + 1)$, which is the Pull-back Prior with $\beta = \frac{1}{\gamma}, \ln Z = (1 + \frac{\eta}{\gamma})$. β is determined by α. In simplified optimization, α is static and need to be searched, *i.e.*, β need to be searched, as section 3.3 does.

Theorem 3. *The optimization process of* $\max_{Lip(D)\leq 1} \mathcal{L}(\theta, \phi, \beta, \omega)$ *is equivalent to the* $\max_{Lip(D)\leq 1} \mathcal{K}$*, which is a lower-bound of* $\beta W^1(p^\dagger, p^*)$.

Proof. $\mathcal{L} = \mathcal{I} + \mathcal{J} + \mathcal{K}$, where $\mathcal{I} + \mathcal{J}$ is independent to D, then $\mathcal{I} + \mathcal{J}$ is constant.

$$\mathcal{K} = -\mathbb{E}_{q_\phi(z)}\, \beta * D(G(z)) - \ln Z \leq \beta\, \mathbb{E}_{p_\mathcal{N}(z)}\, D(G(z)) - E_{q_\phi(z)} D(G(z))$$

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

$$\max_{Lip(D)\leq 1} \mathcal{K} \leq \beta \max_{Lip(D)\leq 1} \{\mathbb{E}_{p^\dagger(x)}\, D(x) - E_{p_r(x)} D(x)\} = \beta W^1(p^\dagger, p_r)$$

where $p_r(x) = \mathbb{E}_{q_\phi(z)}\, p_\theta(x|z)$ and $p_r \approx p^*$ is observed in experiments.

Acknowledgments

This work has been supported by National Key R&D Program of China 2019YFB1802504 and the Beijing National Research Center for Information Science and Technology (BNRist) key projects.

References

1. Arjovsky, M., Chintala, S., et al.: Wasserstein generative adversarial networks. In: International conference on machine learning. pp. 214–223 (2017)
2. Bauer, M., Mnih, A.: Resampled priors for variational autoencoders. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 66–75 (2019)
3. Chen, X., Mishra, N., et al.: Pixelsnail: An improved autoregressive generative model. In: International Conference on Machine Learning. pp. 863–871 (2018)
4. Dai, B., Wipf, P.D.: Diagnosing and enhancing vae models. ICLR (2019)
5. Gelfand, I.M., Silverman, R.A., et al.: Calculus of variations. Courier Corporation (2000)
6. Goodfellow, I., Pouget-Abadie, J., et al.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
7. Gulrajani, I., Ahmed, F., et al.: Improved training of wasserstein gans. In: NIPS (2017)
8. Heusel, M., Ramsauer, H., et al.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems. pp. 6626–6637 (2017)
9. Hoffman, M.D., Johnson, M.J.: Elbo surgery: yet another way to carve up the variational evidence lower bound. In: Workshop in Advances in Approximate Bayesian Inference, NIPS. vol. 1 (2016)
10. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
11. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: Advances in Neural Information Processing Systems. pp. 10215–10224 (2018)
12. Kingma, D.P., Salimans, T., et al.: Improved variational inference with inverse autoregressive flow. In: Advances in neural information processing systems. pp. 4743–4751 (2016)

13. Kumar, R., Goyal, A., et al.: Maximum entropy generators for energy-based models. arXiv preprint arXiv:1901.08508 (2019)
14. Lake, B.M., Salakhutdinov, R., et al.: Human-level concept learning through probabilistic program induction. Science **350**(6266), 1332–1338 (2015)
15. Larochelle, H., Murray, I.: The neural autoregressive distribution estimator. AISTATS pp. 29–37 (2011)
16. Liu, Z., Luo, P., et al.: Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3730–3738 (2015)
17. Maaløe, L., Fraccaro, M., et al.: Biva: A very deep hierarchy of latent variables for generative modeling. NeurIPS (2019)
18. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, J.I.: Adversarial autoencoders. CoRR (2015)
19. Metz, L., Poole, B., et al.: Unrolled generative adversarial networks. ICLR (2017)
20. Van den Oord, A., Kalchbrenner, N., et al.: Conditional image generation with pixelcnn decoders. In: Advances in neural information processing systems. pp. 4790–4798 (2016)
21. Rezende, D.J., Mohamed, S., et al.: Stochastic backpropagation and approximate inference in deep generative models. In: ICML (2014)
22. Salimans, T., Goodfellow, I., et al.: Improved techniques for training gans. In: Advances in neural information processing systems. pp. 2234–2242 (2016)
23. Salimans, T., Karpathy, A., Chen, X., Kingma, P.D.: Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. ICLR (2017)
24. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: Advances in Neural Information Processing Systems. pp. 11895–11907 (2019)
25. Takahashi, H., Iwata, T., et al.: Variational autoencoder with implicit optimal priors. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5066–5073 (2019)
26. Tomczak, J., Welling, M.: Vae with a vampprior. In: International Conference on Artificial Intelligence and Statistics. pp. 1214–1223 (2018)
27. Tomczak, J.M., Welling, M.: Improving variational auto-encoders using householder flow. arXiv preprint arXiv:1611.09630 (2016)
28. Wu, J., Huang, Z., et al.: Wasserstein divergence for gans. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 653–668 (2018)
29. Xiao, H., Rasul, K., et al.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017)