

Learning Deep Generative Models with Annealed Importance Sampling

Xinqiang Ding¹ David J. Freedman¹

Abstract

Variational inference (VI) and Markov chain Monte Carlo (MCMC) are two main approximate approaches for learning deep generative models by maximizing marginal likelihood. In this paper, we propose using annealed importance sampling for learning deep generative models. Our proposed approach bridges VI with MCMC. It generalizes VI methods such as variational auto-encoders and importance weighted auto-encoders (IWAE) and the MCMC method proposed in (Hoffman, 2017). It also provides insights into why running multiple short MCMC chains can help learning deep generative models. Through experiments, we show that our approach yields better density models than IWAE and can effectively trade computation for model accuracy without increasing memory cost.

1. Introduction

Deep generative models with latent variables are powerful probabilistic models for modeling high dimensional data. One of the challenges for learning such models on large datasets by maximizing marginal likelihood is sampling from posterior distributions of latent variables given data, because the posterior distributions are often intractable and have complex dependency structures between latent variables. Two main approximate approaches for learning such models are variational inference (VI) (Wainwright et al., 2008; Blei et al., 2017) and Markov chain Monte Carlo (MCMC) (Neal, 1993).

In VI, a family of distributions parameterized by an inference model is used to approximate the posterior distribution of latent variables. VI approaches learn both generative and inference models simultaneously by maximizing the evidence lower bound (Wainwright et al., 2008). VI becomes especially scalable and efficient for models with continuous latent

variables when it is combined with stochastic optimization, an amortized inference model, and the reparameterization trick (Robbins & Monro, 1951; Hoffman et al., 2013; Rezende et al., 2014; Kingma & Welling, 2013). One of such efficient VI approaches is the variational auto-encoders (VAEs) (Kingma & Welling, 2013; Rezende et al., 2014). In VAEs, the variational distributions used to approximate the posterior distribution of latent variables are commonly chosen to be fully factorized, whereas the true posterior distribution is not necessarily fully factorized and might even have multiple modes. Because the optimization of generative models highly depends on the approximate posterior distribution, generative models learned using VAEs are biased toward having factorized posterior distributions and can be suboptimal. Multiple studies have proposed methods to increase the expressibility of variational distributions while maintaining efficient optimization (Rezende & Mohamed, 2015; Kingma et al., 2016; Salimans et al., 2015; Tran et al., 2016; Burda et al., 2016; Sønderby et al., 2016; Caterini et al., 2018).

MCMC approaches (Neal, 1993) aim to sample from posterior distributions by running a Markov chain with posterior distributions as its stationary distributions. Because it could take large number of steps for a Markov chain to converge, MCMC approaches were known as much slower than VI and were not as widely used as VI approaches for learning deep generative models, especially on large datasets. Compared with rapid developments of VI approaches in recent years, relatively fewer studies investigate the use of MCMC approaches for learning deep generative models (Hoffman, 2017; Han et al., 2017).

In this paper, we propose using annealed importance sampling (AIS) (Jarzynski, 1997; Neal, 2001) for learning deep generative models. Our approach bridge VI and MCMC approaches. VI approaches such as VAEs and importance weighted auto-encoders (IWAEs) can be viewed as special cases of our approach. It also generalizes the MCMC method proposed in (Hoffman, 2017). Through experiments, we show that our approach yields better density models than IWAEs and can effectively trade computation for model accuracy without increasing its memory cost.

¹Department of Neurobiology, The University of Chicago, Chicago, IL 60637 United States. Correspondence to: Xinqiang Ding <xqding@umich.edu>.

2. Background

Let us assume the generative model of interest is defined by a joint distribution of observed data variables x and continuous latent variables z : $p_\theta(x, z) = p_\theta(z)p_\theta(x|z)$, where θ represents parameters of the generative model that need to be optimized. Given training data x , we are interested in learning the generative model $p_\theta(x, z)$ by maximizing its marginal likelihood on the training data x , i.e., maximizing $\log p_\theta(x) = \log \int p_\theta(x, z) dz$. Because the marginal likelihood $\log p_\theta(x)$ is usually a high dimensional integration when z is high-dimensional and has complex dependence structure between its components, computing $\log p_\theta(x)$ directly is computationally expensive. However, we note that maximizing $\log p_\theta(x)$ with respect to θ does not necessarily require computing the value of $\log p_\theta(x)$. If we use first order gradient based optimization methods for training, what is necessarily required is to compute the gradient of $\log p_\theta(x)$ with respect to θ , i.e.,

$$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{z \sim p_\theta(z|x)} [\nabla_\theta \log p_\theta(x, z)], \quad (1)$$

and θ would follow the gradient in (1) in maximum likelihood learning. As shown in (1), computing $\nabla_\theta \log p_\theta(x)$ is equivalent to calculating the expectation of $\nabla_\theta \log p_\theta(x, z)$ with respect to the posterior distribution $p_\theta(z|x)$. Because $\nabla_\theta \log p_\theta(x, z)$ is usually analytically tractable, computing $\nabla_\theta \log p_\theta(x, z)$ for any given x and z is easy. However, the posterior distribution $p_\theta(z|x)$ is usually analytically intractable and often difficult to draw samples from for deep generative models, accurately computing $\nabla_\theta \log p_\theta(x)$ based on Eq. (1) is still computational expensive. Therefore, efficient approximation methods are required to estimate the expectation in (1).

Several VI methods, including VAEs (Kingma & Welling, 2013; Rezende et al., 2014) and IWAEs (Burda et al., 2016), proposed for learning deep generative models can be viewed/understood as devising ways of estimating the expectation in (1). As a VI approach, VAEs use an amortized inference model $q_\phi(z|x)$ to approximate the posterior distribution $p_\theta(z|x)$, where ϕ represents parameters of the inference model. Both the generative model and the inference model are learned by maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log \frac{p_\theta(x, z)}{q_\phi(z|x)} \right]. \quad (2)$$

To maximize ELBO, θ follows the gradient of $\mathcal{L}(\theta, \phi)$ with respect to θ which is

$$\nabla_\theta \mathcal{L}(\theta, \phi) = \mathbb{E}_{z \sim q_\phi(z|x)} [\nabla_\theta \log p_\theta(x, z)]. \quad (3)$$

In VAEs, the expectation in (3) is usually estimated by drawing one sample from $q_\phi(z|x)$. If we view VAEs as

an approximate method for maximum likelihood learning, comparing (3) to (1), we note that VAEs can be viewed as using the expectation in (3) to approximate the expectation in (1) using one sample from $q_\phi(z|x)$. The rationale behind the approximation is that the inference model $q_\phi(z|x)$ is optimized to approximate the posterior distribution $p_\theta(z|x)$. However, when the inference model can not approximate the posterior distribution well due to either its limited expressibility or issues in optimizing the inference model, the estimator based on (3) will have large bias with respect to the target expectation in (1).

Considering we can easily draw independent samples from $q_\phi(z|x)$ which is optimized to approximate $p_\theta(z|x)$, how can we devise a better estimator for the expectation with respect to $p_\theta(z|x)$ in (1)? Because we can easily compute $p_\theta(z|x)$ up to a scaling constant, a canonical way to better estimate (1) with samples from $q_\phi(z|x)$ is using the self-normalized importance sampling (Robert & Casella, 2013). Specifically, we can draw K independent samples, z_1, \dots, z_K , from $q_\phi(z|x)$. Then the expectation in (1) is estimated using

$$\sum_{k=1}^K \tilde{w}_k \nabla_\theta \log p_\theta(x, z_k) \quad (4)$$

where $\tilde{w}_k = w_k / \sum_{i=1}^K w_i$ and $w_k = w(x, z_k, \theta, \phi) = p_\theta(x, z_k) / q_\phi(z_k|x)$. \tilde{w}_k represents the normalized importance weight of the sample z_k . The same gradient estimator as (4) was used in IWAEs, but in IWAEs the estimator was devised as the gradient of a new ELBO function based on multiple (K) samples (Burda et al., 2016):

$$\mathcal{L}_K(\theta, \phi) = \mathbb{E}_{z_1, \dots, z_K \sim q_\phi(z|x)} \left[\log \frac{1}{K} \sum_{k=1}^K \frac{p_\theta(x, z_k)}{q_\phi(z_k|x)} \right]. \quad (5)$$

It is easy to verify that the estimator in (4) is an unbiased estimator of $\nabla_\theta \mathcal{L}_K(\theta, \phi)$. Again, if we view IWAEs as an approximate method for maximum likelihood learning, IWAEs can be viewed as using the estimator in (4), which is based on importance sampling, to approximate the expectation in (1).

Considering the idea of using multiple samples, the expectation in (3) can be also better estimated using multiple samples from $q_\phi(z|x)$, i.e., using the estimator

$$\frac{1}{K} \sum_{k=1}^K \nabla_\theta \log p_\theta(x, z_k), \quad (6)$$

where z_1, \dots, z_K are independent samples from $q_\phi(z|x)$. The difference between estimator (4) and (6) is the way that samples are weighted: samples are weighted equally in (6), whereas in (4) samples are weighted based on their importance (likelihood ratio between $p_\theta(x, z_k)$ and $q_\phi(z|x)$).

The bias of (6) for estimating (1) is due to the difference between $q_\phi(z|x)$ and $p_\theta(z|x)$. In contrast, the bias of (4) for estimating (1) is because of the self-normalizing procedure. As K goes to infinity, the estimator in (4) almost surely converges to (1), but the estimator in (6) does not. Increasing K in (6) can reduce its estimation variance, but can not reduce its bias with respect to (1). This can explain the results in (Burda et al., 2016) that increasing K in (4) can improve learned generative models much more significantly than using (6).

Another factor that affects the bias of the estimator in (4) for (1) is the difference between $q_\phi(z|x)$ and $p_\theta(z|x)$. Factors contributing to the difference include amortizing the variational parameters, limited expressiveness of variational distributions and local minimum issues in optimizing variational distributions. When $q_\phi(z|x)$ can not approximate $p_\theta(z|x)$ well because of these factors, although increasing K in estimator (4) can reduce its bias, it is very likely not the most computational efficient way. This is because information about the posterior distribution $p_\theta(z|x)$ is only used in the final weighting step after sampling and is not used in guiding the sampling steps.

3. Method

In this work, we propose using annealed importance sampling (AIS) (Jarzynski, 1997; Neal, 2001) to estimate the expectation in (1). In the following, we first briefly introduce AIS. Then we present our proposed approach for learning deep generative models with AIS. After that, we use a simple example to show the connection between IWAEs and our approach and why our approach can be computationally more efficient than IWAEs.

3.1. Annealed Importance Sampling

As a generalization of importance sampling, AIS (Jarzynski, 1997; Neal, 2001) uses samples from a sequence of distributions that bridge an initial tractable distribution and a final target distribution. Specifically, let us assume that we want to compute the expectation $\mathbb{E}_{f_T}[h]$ of a function $h(z)$ under a target probability distribution whose density is proportional to a function $f_T(z)$ with an unknown normalization constant. Let us also assume it is not feasible or computationally expensive to directly draw independent samples from $f_T(z)$. However, we can easily draw independent samples from an approximate tractable distribution whose probability density is proportional to $f_0(z)$. In AIS, a sequence of distributions with probability densities proportional to $f_1(z), \dots, f_{T-1}(z)$ is constructed to connect the tractable distribution $f_0(z)$ and the target distribution $f_T(z)$. A generally useful way to construct such

intermediate distributions is to set

$$f_t(z) = f_0(z)^{\alpha_t} f_T(z)^{\beta_t}, \quad (7)$$

where α_t and β_t are inverse temperatures for $f_0(z)$ and $f_T(z)$, respectively, and they satisfy $1 = \alpha_0 \geq \dots \geq \alpha_T = 0$ and $0 = \beta_0 \leq \dots \leq \beta_T = 1$.

To estimate the expectation $\mathbb{E}_{f_T}[h]$, we can generate K samples $\{z_1, \dots, z_K\}$ and compute their corresponding weights $\{w_1, \dots, w_K\}$ as follows. To generate the k th sample z_k and calculate its weight w_k , a sequence of samples $\{z_k^0, \dots, z_k^T\}$ is generated as follows. Initially, z_k^0 is generated by sampling from the distribution $f_0(z)$. For $1 \leq t \leq T$, z_k^t is generated using a reversible transition kernel $T_t(z|z_k^{t-1})$ that keeps f_t invariant. After T steps the sample z_k is set to $z_k = z_k^T$ and its weight w_k is calculated as:

$$w_k = \frac{f_1(z_k^0)}{f_0(z_k^0)} \frac{f_2(z_k^1)}{f_1(z_k^1)} \dots \frac{f_{T-1}(z_k^{T-2})}{f_{T-2}(z_k^{T-2})} \frac{f_T(z_k^{T-1})}{f_{T-1}(z_k^{T-1})}. \quad (8)$$

With the generated K samples z_1, \dots, z_K and their weights w_1, \dots, w_K , the expectation $\mathbb{E}_{f_T}[h]$ is estimated using

$$\mathbb{E}_{f_T}[h] \simeq \sum_{k=1}^K \tilde{w}_k h(z_k), \quad (9)$$

where $\tilde{w}_k = w_k / \sum_{i=1}^K w_i$ are normalized weights.

3.2. Learn Deep Generative Models with AIS

The main idea of our proposed method is to use AIS (9) to estimate the gradient $\nabla_\theta \log p_\theta(x)$ in (1), which is required by maximum likelihood learning. The function of interest $h(z)$ in $\mathbb{E}_{f_T}[h]$ is $\nabla_\theta \log p_\theta(x, z)$. The target distribution is the posterior distribution $p_\theta(z|x)$ which is proportional to $p_\theta(x, z)$, i.e., $f_T(x) = p_\theta(x, z)$. The initial tractable distribution $f_0(z)$ is set to the variational distribution $q_\phi(z|x)$ which is learned using VAEs. The intermediate distributions $f_t(z)$ are chosen based on (7) by setting α_t and β_t based on the constraints: $1 = \alpha_0 \geq \dots \geq \alpha_T = 0$ and $0 = \beta_0 \leq \dots \leq \beta_T = 1$. The reversible transition kernel $T_t(z|z_k^{t-1})$ that keeps $f_t(z)$ invariant is constructed using the Hamiltonian Monte Carlo (HMC) sampling method (Neal et al., 2011) in which the potential energy function $U_t(z)$ is set to $U_t(z) = -\log f_t(z)$. With the estimation of the gradient $\nabla_\theta \log p_\theta(x)$ in (1), the generative model parameters θ are optimized using stochastic gradient ascent (Robbins & Monroe, 1951; Kingma & Ba, 2014; Hoffman et al., 2013). Because the performance of AIS estimation (9) still depends on the similarity between the initial and target distributions, the initial distribution parameters ϕ is optimized by maximizing the ELBO function $\mathcal{L}(\theta, \phi)$ (2) using VAEs (Rainforth et al., 2018). In summary, the detailed procedures of our proposed method are described in Algorithm (1).

Algorithm 1 Learning Deep Generative Models with Annealed Importance Sampling

Require:

x : training data
 K : the number of annealed importance weighted samples
 $p_\theta(x, z)$: the generative model
 $q_\phi(z|x)$: the inference model
 T : the number of inverse temperatures
 $\{\alpha_t : 1 = \alpha_0 \geq \dots \geq \alpha_T = 0\}$: inverse temperatures for the inference model
 $\{\beta_t : 0 = \beta_0 \leq \dots \leq \beta_T = 1\}$: inverse temperatures for the generative model
 $\{\epsilon_t : t = 1, \dots, T\}$: the step sizes used in leapfrog integration of HMC at each inverse temperature
 L : the number of integration steps in HMC

Calculate Gradients and Optimize Parameters:

while θ, ϕ not converged **do**
 sample example(s) x from the training data;
 update the generative model parameter θ
 set $\log w_k = 0$ for $k = 1, \dots, K$;
 sample $z^0 = [z_1^0, z_2^0, \dots, z_K^0]$, where z_k^0 are i.i.d. samples from $q_\phi(z|x)$;
 $\log w_k \leftarrow (\beta_1 - \beta_0) \log p_\theta(x, z_k^0) + (\alpha_1 - \alpha_0) \log q_\phi(z_k^0)$ for $k = 1, \dots, K$;
 for $t \leftarrow 1$ **to** $T - 1$ **do**
 $z^t = \text{HMC}(z^{t-1}, \alpha_t, \beta_t, \epsilon_t, L)$, where the potential energy function is: $U_t(z) = -\log f_t(z)$ and $f_t(z) = q_\phi(z|x)^{\alpha_t} p_\theta(x, z)^{\beta_t}$;
 $\log w_k \leftarrow \log w_k + (\beta_t - \beta_{t-1}) \log p_\theta(x, z_k^{t-1}) + (\alpha_t - \alpha_{t-1}) \log q_\phi(z_k^{t-1})$ for $k = 1, \dots, K$;
 end for
 $z^T = \text{HMC}(z^{T-1}, \alpha_T, \beta_T, \epsilon_T, L)$ or set $z^T = z^{T-1}$
 set $z = [z_1, \dots, z_K] = [z_1^T, \dots, z_K^T]$ and $\widetilde{w}_k = w_k / \sum_{i=1}^K w_i$;
 estimate the gradient $\nabla_\theta \log p_\theta(x)$ with $\delta_\theta = \sum_{k=1}^K \widetilde{w}_k \nabla_\theta \log p_\theta(x, z_k)$;
 apply gradient update to θ using δ_θ ;
 update the inference model parameter ϕ
 sample $\epsilon \sim \mathcal{N}(0, \mathbf{I})$;
 set $z = \mu(\phi, x) + \sigma(\phi, x) \odot \epsilon$ and calculate $\mathcal{L}(\theta, \phi)$;
 estimate the gradient $\delta_\phi = \nabla_\phi \mathcal{L}(\theta, \phi)$ with the reparameterization trick;
 apply gradient update to ϕ using δ_ϕ ;
end while

3.3. Implicit Distributions Defined by IWAEs and Algorithm 1

Previous studies showed that IWAEs can also be interpreted as optimizing the standard ELBO (2) using a more complex variational distribution that is implicitly defined by importance sampling (IS) (Cremer et al., 2017; Mnih & Rezende,

2016). Similar interpretation can be used to understand Algorithm 1. Specifically, when using (4) or (9) to estimate the expectation in (1), IS or AIS implicitly defines a proposal distribution, $q_{\text{IS}}(z|x)$ or $q_{\text{AIS}}(z|x)$, using samples from $q_\phi(z|x)$ to approximate the posterior $p_\theta(z|x)$. We can sample from the implicitly defined proposal distributions, $q_{\text{IS}}(z|x)$ or $q_{\text{AIS}}(z|x)$, with Algorithm 2.

Algorithm 2 Sampling $q_{\text{IS}}(z|x)$ or $q_{\text{AIS}}(z|x)$

K : number of importance samples
 L : number of integration steps in HMC
 T : num of intermediate distributions
case 1: when importance sampling is used
 sample $z_1, \dots, z_K \sim q_\phi(z|x)$
 set $w_k = \frac{p_\theta(x, z_k)}{q_\phi(z_k|x)}$ and $\widetilde{w}_k = w_k / \sum_{k=1}^K w_k$
case 2: when AIS is used
 sample z_1, \dots, z_K and compute $\widetilde{w}_1, \dots, \widetilde{w}_K$ with Algorithm 1
 sample $j \sim \text{Categorical}(\widetilde{w}_1, \dots, \widetilde{w}_K)$
return z_j

One way to compare IWAEs and Algorithm (1) is to compare the computational efficiency of $q_{\text{IS}}(z|x)$ and $q_{\text{AIS}}(z|x)$ for approximating the posterior $p_\theta(z|x)$. To do that, we apply them in the following simple example. In the example, the target distribution $p(z|x)$ is a normal distribution of two correlated random variables. The proposal distribution $q(z|x)$ is set to the normal distribution of two independent random variables that minimizes the Kullback-Leibler (KL) divergence from $q(z|x)$ to $p(z|x)$, i.e., $D_{\text{KL}}(q(z|x)||p(z|x))$. If we assume the amortized inference model used in VAEs has enough flexibility, the variational distribution learned by VAE, $q_{\text{VAE}}(z|x)$ should be equal/close to $q(z|x)$. $q_{\text{IS}}(z|x)$ depends on the parameter K and the computational cost of using IS increases linearly with K . $q_{\text{AIS}}(z|x)$ depends on the parameters, L , K and T , used in AIS, and its computational cost scales linearly with $L \times K \times T$. To make a fair comparison, we compare $q_{\text{IS}}(z|x)$ and $q_{\text{AIS}}(z|x)$ under the same computational cost, i.e., K in IS is equal to $L \times K \times T$ in AIS. The inverse temperatures α_t and β_t in AIS are set to changing linearly with t from 0 to T . The results are shown in Fig. (1), where the implicit distribution probability densities are estimated using kernel methods with Gaussian Kernels. Both $q_{\text{IS}}(z|x)$ and $q_{\text{AIS}}(z|x)$ become better approximation of $p(z|x)$ when increasing K or $L \times K \times T$. Using the same amount of computational cost, $q_{\text{AIS}}(z|x)$ approximates the target distribution better than $q_{\text{IS}}(z|x)$. Therefore, using $q_{\text{AIS}}(z|x)$ is computationally more efficient than using $q_{\text{IS}}(z|x)$ for approximating $p(z|x)$. We can expect that the better performance of $q_{\text{AIS}}(z|x)$ for approximating $p(z|x)$ can help Algorithm (1) learn better generative models than IWAEs.

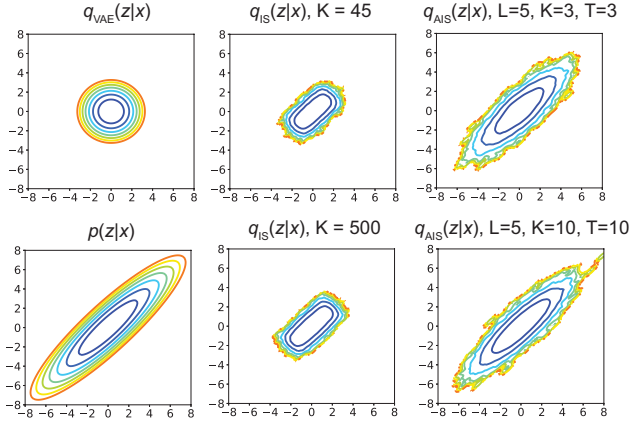


Figure 1. Comparison of implicit distributions $q_{IS}(z|x)$ and $q_{AIS}(z|x)$ defined using importance sampling and annealed importance sampling, respectively, for approximating the target distribution $p(z|x)$.

4. Related Work

Our approach (Algorithm 1) is most closely related to the IWAE approach (Burda et al., 2016) and Matthew D. Hoffman’s HMC approach (MH-HMC) (2017) for learning deep generative models with MCMC. Both methods can be viewed as special cases of our proposed approach. The IWAE approach corresponds to setting $T = 1$ and $z^1 = z^0$ in Algorithm (1). The MH-HMC approach (Hoffman, 2017) is equivalent to setting $K = 1$, $1 = \alpha_0 > \alpha_1 = \dots = \alpha_T = 0$, and $0 = \beta_0 < \beta_1 = \dots = \beta_T = 1$.

Another related approach is the variational inference approach based on sequential Monte Carlo (VSMC) (Le et al., 2018; Maddison et al., 2017; Naesseth et al., 2018). We note that our approach is different from VSMC. VSMC optimizes a lower bound defined by marginal likelihood estimation based on sequential Monte Carlo (SMC). When computing gradients of the lower bound with respect to θ and ϕ , VSMC has to run the back-propagation algorithm through all intermediate samples and the resampling operations, but multiple studies (Le et al., 2018; Maddison et al., 2017; Naesseth et al., 2018) ignored the contribution of the resampling operations to the gradients based on empirical performance. In contrast, our proposed approach is not aimed to optimize variational lower bounds, although it happens to be equivalent to optimizing a variational lower bound in the specific setting that is equivalent to IWAEs. Our approach is motivated by the fact that maximum likelihood learning requires estimating the expectation in (1). Both q_ϕ and AIS are employed to better estimate the expectation in (1). On the implementation side, when estimating the expectation in (1), we only need to consider gradients $\nabla_\theta \log p_\theta(x, z_k^T)$ for samples z_k^T from AIS at the last time

step which does not require running back-propagation and do not need to consider the contribution of the reject-accept operations in HMC.

The implicit distribution $q_{AIS}(z|x)$ defined by AIS in Algorithm (1) using samples from $q_{VAE}(z|x)$ approximates the posterior $p(z|x)$ better than the VAE variational distribution $q_{VAE}(z|x)$. From this point of view, there have been multiple studies developing VI methods with more flexible and accurate variational distributions (Rezende & Mohamed, 2015; Kingma et al., 2016; Salimans et al., 2015; Tran et al., 2016; Burda et al., 2016; Sønderby et al., 2016; Caterini et al., 2018). These approaches learn more accurate variational distributions in an explicit manner by minimizing the KL divergence, which is equivalent to maximizing ELBO, from a more flexible families of variational distributions to the target distribution. The advantage of these approaches is that generative models and inference models are learned by optimizing a single objective function (ELBO), which is also easy to estimate. On the other end, the disadvantage of these approaches is that more parameters have to be introduced in inference models and, if the reparameterization trick is used, the variational distributions are restricted to distributions that admit reparameterization. For example, in normalizing flow based approach (Rezende & Mohamed, 2015; Kingma et al., 2016; Caterini et al., 2018), more parameters are needed to increase the flexibility of variational distributions. In Hamiltonian variational inference (HVI) (Salimans et al., 2015), an extra inference model for auxiliary variables is required to reverse transformations in HMC steps. In addition, these approaches require running back-propagation through all the transformations in the flow, which makes their memory cost increase linearly with the depth of the flow or the number of HMC steps. Moreover, running back-propagation through the accept-reject operations in HMC often leads to estimations with large variance. In contrast, our approach (Algorithm 1) does not require running back-propagation through the HMC steps and therefore its memory cost is constant with respect to the number of HMC steps $L \times T$.

5. Experiment Results

5.1. Dataset and Model Setup

We conducted a series of experiments to evaluate the performance of our proposed algorithm (1) on learning deep generative models using both the Omniglot dataset (Lake et al., 2015) and the MNIST (LeCun et al., 1998) dataset. For the Omniglot dataset, we used 24,345 examples for training and 8,070 examples for testing¹. For the MNIST dataset,

¹The Omniglot data was downloaded from <https://github.com/yburda/iwae.git>

Table 1. Results of IWAE-DReG and our approach on the Omniglot dataset.

	K = 1		K=5		K=50	
	$\approx \log p(x)$	ACTIVE UNITS	$\approx \log p(x)$	ACTIVE UNITS	$\approx \log p(x)$	ACTIVE UNITS
IWAE-DReG	-109.41	26	-106.11	35	-103.91	43
OURS (T = 5)	-103.22	47	-102.47	50	-102.03	50
OURS (T = 11)	-102.45	50	-101.94	50	-101.64	50

Table 2. Results of IWAE-DReG and our approach on the MNIST dataset.

	K = 1		K=5		K=50	
	$\approx \log p(x)$	ACTIVE UNITS	$\approx \log p(x)$	ACTIVE UNITS	$\approx \log p(x)$	ACTIVE UNITS
IWAE-DReG	-86.90	16	-85.52	21	-84.38	27
OURS (T = 5)	-84.56	29	-84.25	32	-83.93	36
OURS (T = 11)	-84.14	32	-83.78	35	-83.63	40

Table 3. Results of IWAE-DReG and our approach on the Omniglot dataset with same computational cost.

	$\approx \log p(x)$	ACTIVE UNITS
IWAE-DReG (K = 55)	-103.85	44
OURS (L = 5, K = 1, T = 11)	-102.45	50
IWAE-DReG (K = 275)	-103.13	47
OURS (L = 5, K = 5, T = 11)	-101.94	50
IWAE-DReG (K = 2750)	-102.40	50
OURS (L = 5, K = 50, T = 11)	-101.64	50

Table 4. Results of MH-HMC and our approach on the Omniglot dataset with same computational cost.

	$\approx \log p(x)$
MH-HMC (L = 5, K = 5, T = 5)	-102.57
OURS (L = 5, K = 5, T = 5)	-102.47
MH-HMC (L = 5, K = 5, T = 11)	-101.32
OURS (L = 5, K = 5, T = 11)	-101.94
MH-HMC (L = 5, K = 50, T = 5)	-102.32
OURS (L = 5, K = 50, T = 5)	-102.03
MH-HMC (L = 5, K = 50, T = 11)	-101.25
OURS (L = 5, K = 50, T = 11)	-101.64

60,000 examples were used for training and 10,000 examples for testing. Images from both datasets have a dimension of 28×28 . For both training and testing, images are dynamically binarized into vectors of 0 and 1 with the probability of being 1 equal to normalized pixel values between 0 and 1.

We used same generative models and same inference models as that used in the IWAE study (Burda et al., 2016). The dimension of the latent variable z is 50. For the generative model $p_\theta(x, z) = p(z)p_\theta(x|z)$, the prior distribution $p(z)$ is a 50 dimensional standard Gaussian distribution. The conditional distribution $p_\theta(x|z)$ is a Bernoulli distribution and is parameterized by a neural network with two hidden layers, both of which have 200 hidden units with tanh activation functions. The approximate posterior distribution $q_\phi(z|x)$ is a 50 dimensional Gaussian distribution with a diagonal covariance matrix. Its mean $\mu(x)$ and variance $\sigma(x)$ are similarly parameterized by neural networks with two hidden layers, both of which have 200 hidden units with tanh activation functions.

We used the same optimization setup as that used in the IWAE study (Burda et al., 2016). The Adam optimizer (Kingma & Ba, 2014) is used with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-4}$. The optimization was run for 3^i passes over the training data with a learning rate of $0.001 \cdot 10^{-i/7}$ for $i = 0, \dots, 7$. Overall, the optimization was run for 3,280 epochs. The size of a mini-batch is set to 128 for all approaches. This is different from the IWAE study where the size of a mini-batch is set to 20.

Same deep generative models are also learned using the two closely related approaches: IWAEs and MH-HMC (2017). Because previous study (Tucker et al., 2019) showed that IWAEs with doubly reparameterized gradient estimators (IWAE-DReG) can improve the performance of IWAEs, we used IWAE-DReG in all computations involving IWAE. In the original MH-HMC approach (Hoffman, 2017), K is equal to 1. In our experiments, we implemented the MH-HMC approach with $K > 1$. We note that only when $K = 1$ can the MH-HMC approach be viewed

as a special case of our proposed approach. When $K > 1$, the MH-HMC approach is different from our proposed algorithm (1) in terms of the way of weighting samples: the MH-HMC approach assigns equal weights to the K samples whereas our proposed approach uses weights from AIS for the K samples. This difference between MH-HMC and our approach is similar to that between the estimators in (6) and (4). Each model is trained independently for three repeats. All results shown below are average of three independent repeats.

5.2. Evaluation of Learned Models

Following (Wu et al., 2017; Hoffman, 2017; Cremer et al., 2018), we evaluate learned generative models using marginal likelihoods estimated with AIS (Neal, 2001) on test datasets. To be confident that our estimated marginal likelihoods are accurate enough to be used for comparing models, we follow (Wu et al., 2017) to empirically validate our AIS estimates using Bidirectional Monte Carlo (BDMC) (Grosse et al., 2015). By running AIS in both forward and backward directions, BDMC provides both lower and upper bounds of marginal likelihoods for simulated data. The gap between the upper and the lower bound provides an estimate of the AIS estimator’s precision for simulated data. Assuming the simulated data from learned models follows similar distribution as real data, we expect that the gap computed using simulated data also provides a good estimate of the estimator’s precision on real data.

For models learned with both Omniglot and MNIST datasets, we used AIS estimates with 96 samples and 20,000 intermediate distributions $q(z|x)^{\alpha_t}p(x,z)^{\beta_t}$ with a linear schedule for inverse temperatures α_t and β_t , i.e., $\beta_t = t/20,000$ for $t = 0, \dots, 20,000$ and $\alpha_t = 1 - \beta_t$. For all the models trained on the Omniglot dataset, the gaps estimated using BDMC on 1000 simulated samples are less than 0.1 nats. On the MNIST dataset, models trained using IWAE-DReG with $K \in \{1, 5, 50\}$ also have gaps less than 0.1 nats. Models trained with our approach have gaps less than 0.1 nats for $K = 1$, 0.3 nats for $K = 5$, 0.3 nats for $K = 50$, $T = 5$ and 0.5 nats for $K = 50$, $T = 11$.

5.3. Results

Generative models are trained on both Omniglot and MNIST datasets using both IWAE-DReG and our approach (Algorithm 1) under various settings of hyper-parameters (parameter K for IWAE-DReG and parameters K and T for our approach). The number of HMC integration steps L in Algorithm (1) is set to 5. The inverse temperatures α_t and β_t in Algorithm (1) are set to change linearly between 0 and 1. The estimated marginal likelihoods on the Omniglot and MNIST testing datasets are summarized in Table 1 and 2, respectively.

For models trained with IWAE-DReG, the marginal likelihood increases on both Omniglot and MNIST testing datasets when the value of $K \in \{1, 5, 50\}$ increases. This agrees with previous studies (Burda et al., 2016; Tucker et al., 2019). For a fixed K , our approach with $T = 5$ or $T = 11$ produces better density models than IWAE-DReG. If we view IWAE-DReG as a special case of our approach with $T = 1$, then the performance of our approach improves when increasing the value of either K or T . This is because increasing K or T can make implicit distributions defined by IS and AIS better approximate the posterior distribution, which in turn improves estimators in (4) and (9) for estimating the expectation in (1).

Following the IWAE study (Burda et al., 2016), we also calculated the number of active latent units and used it to represent how much the latent space’s representation capacity was utilized in learned models. The intuition is that if a latent unit is active for encoding information in the observation, it is expected that its distribution would change with observations. Therefore, we used the following variance statistics to quantify the activity of the a latent unit h : $A_h = \text{Cov}_{x \in \text{test set}}(\mathbb{E}_{h \sim q(h|x)}[h])$, which measures how much the value of a latent unit changes when the observation in test set changes. A latent unit h is defined to be active if $A_h > 10^{-2}$. Both the statistics A_h and the cutoff value are adopted from the IWAE study (Burda et al., 2016). As shown in Table 1 and Table 2, the number of active units in models trained with our approach is much larger than that in models trained with IWAE-DReG. In addition, the number of active units in models trained with our approach increases monotonically not only with the number of samples K but also with the number of inverse temperatures T . Intuitively, when the number of inverse temperature increases, the annealing process becomes longer and smoother, making it easier for samples to explore more latent space. On the Omniglot dataset, all the latent units are active for most of the models trained with our approach.

For a fixed K , the computational cost of our approach is about $L \times T$ times that of IWAE-DReG. Therefore results in Table 1 and Table 2 only show that our approach is an effective way of trading computation for model accuracy. To show that our approach is also computationally more efficient than IWAE-DReG, we also compared learned models trained using IWAE-DReG and our approach with the same computational cost on the Omniglot dataset. The result is shown in Table 3. The computational costs of IWAE-DReG and our approach are equal if K in IWAE-DReG is equal to $L \times K \times T$ in our approach. Results in Table 3 show that, with the same computational cost, our approach leads to better density models than IWAE-DReG and models trained with our approach have more active units. Although results on the MNIST dataset also shows that models trained with our approach have higher marginal like-

likelihood than that trained with IWAE-DReG, the difference in marginal likelihood between the two methods is smaller than the gap estimated using BDMC.

We also compared our approach with the MH-HMC method for learning deep generative models with same computational cost. The MH-HMC method (Hoffman, 2017) always sets α_t and β_t as $1 = \alpha_0 > \alpha_1 = \dots = \alpha_T = 0$, and $0 = \beta_0 < \beta_1 = \dots = \beta_T = 1$. In our approach, α_t and β_t are free to choose as long as they satisfy the constraints that $1 = \alpha_0 \geq \alpha_1 \geq \dots \geq \alpha_T = 0$, and $0 = \beta_0 \leq \beta_1 \leq \dots \leq \beta_T = 1$. In this study, we set α_t and β_t to change linearly between 0 and 1. When $K = 1$, the MH-HMC approach is a special case of our approach for setting α_t and β_t . When $K > 1$, the MH-HMC method is different from our approach in the way of weighting samples. The marginal likelihoods of models learned using the MH-HMC method and our approach are shown in Table (4). Similar to models trained with our approach, models trained with the MH-HMC method also improves when increasing the value of K or T . For the cases where $T = 5$, our approach leads to better density models, whereas for the cases where $T = 11$, the MH-HMC method leads to better density models.

6. Conclusion and Discussion

In this work, we propose a new approach for learning deep generative models using AIS. Our approach connects VI methods with MCMC methods used for learning generative models. Two VI methods including VAEs and IWAEs and the MCMC method, MH-HMC, can be viewed as special cases of our approach. Experiments show that our approach can effectively trade computation for model accuracy without increasing memory cost. In addition, our approach is computationally more efficient than IWAEs for learning deep generative models and produces better density models than IWAEs under the same computational cost. Compared with normalizing flow based VI methods (Rezende & Mohamed, 2015; Kingma et al., 2016) and methods that combine VI and MCMC (Salimans et al., 2015; Caterini et al., 2018), the advantage of our approach is two folds: (1) it does not require introducing extra parameters in inference models; (2) it does not require running back-propagation through operations in HMC steps, so its memory cost does not increase with the depth of flow or the number of HMC steps used in MCMC.

Our approach does not require running the HMC steps to reach equilibrium, because the AIS estimator in (9) does not require so. In addition, because the AIS estimator in (9) improves with the number of samples K and the length of HMC steps $L \times T$, our proposed approach for learning generative models also improves with both K and $L \times T$. This explains why running multiple MCMC chains with-

out convergence can still help learn deep generative models. This insight along with the insight into the connection between VI and MCMC methods provided by our method resonate with the recent work by (Hoffman & Ma, 2019) which suggests that VI and MCMC methods are not as different as one might think and running short MCMC procedures without convergence but with multiple chains can become an attractive alternative relative to VI for learning generative models.

In current work, the initial distribution $q_\phi(z|x)$ used in AIS is learned by minimizing the KL divergence $D_{\text{KL}}(q||p)$. Minimizing $D_{\text{KL}}(q||p)$ often leads to $q_\phi(z|x)$ that underestimates the variance of $p(z|x)$. In addition, minimizing $D_{\text{KL}}(q||p)$ has the mode-seeking behavior in which $q_\phi(z|x)$ only covers one mode even when $p(z|x)$ has multiple modes. Therefore, using $q_\phi(z|x)$ as the initial distribution in AIS might not be ideal for AIS to explore more latent space and multiple modes. However, the advantage of using $q_\phi(z|x)$ as the initial distribution is that the initial samples have small variance and are from regions with high posterior density. It is not clear to us for now what alternative choice of the initial distribution would be better than $q_\phi(z|x)$. Moreover, the inverse temperatures α_t and β_t in Algorithm (1) are set to change linearly between 0 and 1. Therefore, the initial distribution $q_\phi(z|x)$ affects sampling at all intermediate time steps. Because the initial distribution $q_\phi(z|x)$ learned using VAEs usually underestimates the variance of $p(z|x)$ and has mode-seeking behavior, the current setting of α_t and β_t , allowing $q_\phi(z|x)$ to affect sampling in all intermediate steps, might not be ideal for AIS to explore more latent space and even multiple modes. In the MH-HMC approach, the inverse temperatures α_t and β_t are set as $1 = \alpha_0 > \alpha_1 = \dots = \alpha_T = 0$, and $0 = \beta_0 < \beta_1 = \dots = \beta_T = 1$ and the initial distribution can only affect sampling at the first step. This can explain the better performance of the MH-HMC approach compared to our current approach when $T = 11$ in Table (4). However, the disadvantage of the way of setting the inverse temperatures in the MH-HMC approach is that the HMC sampling could be trapped in a local minimum region when the landscape of $\log p_\theta(x, z)$ is rugged.

References

- Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.
- Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.
- Caterini, A. L., Doucet, A., and Sejdinovic, D. Hamiltonian

- variational auto-encoder. In *Advances in Neural Information Processing Systems*, pp. 8167–8177, 2018.
- Cremer, C., Morris, Q., and Duvenaud, D. Reinterpreting importance-weighted autoencoders. In *Workshop at International Conference on Learning Representations*, 2017.
- Cremer, C., Li, X., and Duvenaud, D. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pp. 1078–1086, 2018.
- Grosse, R. B., Ghahramani, Z., and Adams, R. P. Sandwiching the marginal likelihood using bidirectional monte carlo. *arXiv preprint arXiv:1511.02543*, 2015.
- Han, T., Lu, Y., Zhu, S.-C., and Wu, Y. N. Alternating back-propagation for generator network. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- Hoffman, M. D. Learning deep latent gaussian models with markov chain monte carlo. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1510–1519. JMLR. org, 2017.
- Hoffman, M. D. and Ma, Y. Langevin dynamics as non-parametric variational inference. In *2nd Symposium on Advances in Approximate Bayesian Inference*, 2019.
- Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.
- Jarzynski, C. Nonequilibrium equality for free energy differences. *Physical Review Letters*, 78(14):2690, 1997.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014.
- Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2013.
- Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pp. 4743–4751, 2016.
- Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
- Le, T. A., Igl, M., Rainforth, T., Jin, T., and Wood, F. Auto-encoding sequential monte carlo. In *International Conference on Learning Representations*, 2018.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Maddison, C. J., Lawson, J., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Teh, Y. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pp. 6573–6583, 2017.
- Mnih, A. and Rezende, D. J. Variational inference for monte carlo objectives. In *Proceedings of the 33rd International Conference on Machine Learning-Volume 48*, pp. 2188–2196, 2016.
- Naesseth, C., Linderman, S., Ranganath, R., and Blei, D. Variational sequential monte carlo. In *International Conference on Artificial Intelligence and Statistics*, pp. 968–977, 2018.
- Neal, R. M. *Probabilistic inference using Markov chain Monte Carlo methods*. Department of Computer Science, University of Toronto Toronto, ON, Canada, 1993.
- Neal, R. M. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.
- Neal, R. M. et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.
- Rainforth, T., Kosiorek, A., Le, T. A., Maddison, C., Igl, M., Wood, F., and Teh, Y. W. Tighter variational bounds are not necessarily better. In *International Conference on Machine Learning*, pp. 4277–4285, 2018.
- Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538, 2015.
- Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.
- Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.
- Robert, C. and Casella, G. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.
- Salimans, T., Kingma, D., and Welling, M. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pp. 1218–1226, 2015.
- Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. Ladder variational autoencoders. In *Advances in neural information processing systems*, pp. 3738–3746, 2016.

- Tran, D., Ranganath, R., and Blei, D. M. The variational gaussian process. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.
- Tucker, G., Lawson, D., Gu, S., and Maddison, C. J. Doubly reparameterized gradient estimators for monte carlo objectives. In *International Conference on Learning Representations*, 2019.
- Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.
- Wu, Y., Burda, Y., Salakhutdinov, R., and Grosse, R. On the quantitative analysis of decoder-based generative models. In *International Conference on Learning Representations*, 2017.