Likelihood Almost Free Inference Networks

Guoqing Zheng, Yiming Yang, Jaime Carbonell
Carnegie Mellon University
\{gzheng, yiming, jgc\}@cs.cmu.edu

Abstract

Variational inference for latent variable models is prevalent in various machine learning problems, typically solved by maximizing the Evidence Lower Bound (ELBO) of the true data likelihood with respect to a variational distribution. However, freely enriching the family of variational distribution is challenging since the ELBO requires variational likelihood evaluations of the latent variables. In this paper, we propose a novel framework to enrich the variational family based on an alternative lower bound, by introducing auxiliary random variables to the variational distribution only. While offering a much richer family of complex variational distributions, the resulting inference network is likelihood almost free in the sense that only the latent variables require evaluations from simple likelihoods and samples from all the auxiliary variables are sufficient for maximum likelihood inference. We show that the proposed approach is essentially optimizing a probabilistic mixture of ELBOs, thus enriching modeling capacity and enhancing robustness. It outperforms state-of-the-art methods in our experiments on several density estimation tasks.

1 Introduction

Estimating posterior distributions is the primary focus of Bayesian inference, where we are interested in how our belief over the variables in our model would change after observing a set of data. Predictions can also be benefited from Bayesian inference as every prediction will be equipped with a confidence interval representing how sure the prediction is. Compared to the maximum a posteriori (MAP) estimator of the model parameters, which is a point estimator, the posterior distribution provides richer information about model parameters and hence more justified prediction.

Among various inference algorithms for posterior estimation, variational inference (VI) and Markov Chain Monte Carlo (MCMC) are the most wisely used ones. It is well known that MCMC suffers from slow mixing time though asymptotically the chained samples will approach the true posterior. Furthermore, for latent variable models (LVMs) where each sampled data point is associated with a latent variable, the number of simulated Markov Chains increases with the number of data points, making the computation too costly. VI, on the other hand, facilitates faster inference because it optimizes an explicit objective function and its convergence can be measured and controlled. Hence, VI has been widely used in many Bayesian models, such as the mean-field approach for the Latent Dirichlet Allocation [Blei et al., 2003], etc. To enrich the family of distributions over the latent variables, neural network based variational inference methods have also been proposed, such as Variational Autoencoder (VAE) [Kingma and Welling, 2013], Importance Weighted Autoencoder (IWAE) [Burda et al., 2015] and others [Rezende and Mohamed, 2015; Mnih and Gregor, 2014; Kingma et al., 2016]. These methods outperform the traditional mean-field based inference algorithms due to their flexible distribution families and easy-to-scale algorithms, therefore becoming the state of the art for variational inference.

The aforementioned VI methods are essentially maximizing the evidence lower bound (ELBO), i.e., the lower bound of the true marginal data likelihood, defined as

$$
\log p_d(x) = \log \mathbb{E}_{z \sim q_\phi(z|x)} \frac{p(z)p(x|z)}{q(z|x)} \\
\geq \mathbb{E}_{z \sim q_\phi(z|x)} \log \frac{p(z)p(x|z)}{q(z|x)}
$$

Notice that the equality holds if and only if \(q_\phi(z|x) = p_d(z|x)\) and otherwise a gap always exists. The more flexible the variational family \(q(z|x)\) is, the more likely it can match the true posterior \(p(z|x)\). However, arbitrarily enriching the variational model family \(q\) is non-trivial, since optimizing Eq. 1 always requires evaluations of \(q(z|x)\), no matter what architecture is used to model \(q\).

In this paper we propose to optimize an alternative lower bound of the true data likelihood, by introducing auxiliary variables to the variational model only. Most importantly, likelihood evaluations are not required for the auxiliary variables and easy samplings are admitted. This essentially results in a likelihood almost free inference network, in the sense that only variational likelihoods for the actual latent variables are needed while samples for all auxiliary variables are sufficient maximum likelihood inference. We argue that the resulting inference network is essentially learning a mix of different variational posteriors, and thus enables modeling a much richer and flexible family of complex posterior distribution. It can
be shown that the new framework subsumes several recently developed neural variational methods as special cases. We conduct empirical evaluations on several density estimation tasks, which validate the effectiveness of the proposed method.

The rest of the paper is organized as follows: We briefly review two existing approaches for inference network modeling, and present our proposed likelihood almost free inference network in the following section. We then point out the connections of the proposed framework to other related ones. Empirical evaluations and analysis are carried out in Section Experiments, and we conclude this paper in the last section.

2 Preliminaries

In this section, we briefly review several existing methods that aim to enrich inference networks flexible neural network architectures.

2.1 Variational Autoencoder (VAE)

Given a generative model \( p_\theta(x, z) = p_\theta(z)p_\theta(x|z) \) defined over data \( x \) and latent variable \( z \), indexed by parameter \( \theta \), variational inference aims to approximate the intractable posterior \( p(z|x) \) with \( q_\phi(z|x) \), indexed by parameter \( \phi \), such that the ELBO is maximized

\[
\mathcal{L}_{\text{VAE}}(x) = \mathbb{E}_q \log p(x, z) − \mathbb{E}_q \log q(z|x) \leq \log p(x) \tag{2}
\]

Parameters of both generative distribution \( p \) and variational distribution \( q \) are learned by maximizing the ELBO with stochastic gradient methods. Specifically, VAE [Kingma and Welling, 2013] assumes both the conditional distribution of data given the latent codes of the generative model and the variational posterior distribution are Gaussians, whose means and diagonal covariances are parameterized by two neural networks, termed as generative network and inference network, respectively. Model learning is possible due to the re-parameterization trick [Kingma and Welling, 2013] which makes back propagation through the stochastic variables possible.

2.2 Importance Weighted Autoencoder (IWAE)

The above ELBO is a lower bound of the true data log-likelihood \( \log p(x) \), hence [Burda et al., 2015] proposed IWAE to directly estimate the true data log-likelihood with the presence of the variational model\(^1\), namely

\[
\log p(x) = \log \mathbb{E}_q p(x, z)\frac{p(x, z)}{q(z|x)} \geq \log \frac{1}{m} \sum_{i=1}^{m} p(x, z_i) \equiv \mathcal{L}_{\text{IWAE}}(x) \tag{3}
\]

where \( m \) is the number of importance weighted samples. The above bound is tighter than the ELBO used in VAE. When trained on the same network structure as VAE, with the above estimate as training objective, IWAE achieves considerable improvements over VAE on various density estimation tasks [Burda et al., 2015] and similar idea is also considered in [Mnih and Rezende, 2016].

\(^1\)We drop the dependencies of \( p \) and \( q \) on parameters \( \theta \) and \( \phi \) to prevent clutter.

\(^2\)The variational model is also referred to as the inference model, hence we use them interchangeably.

2.3 Normalizing Flows

Another idea to enrich the model capacity to cover complex distribution is normalizing flows [Rezende and Mohamed, 2015], in which lies the central idea of transforming a simple variable with known densities to construct complex densities. Given random variable \( z \in \mathbb{R}^d \) with density \( p(z) \), consider a smooth and invertible function \( f: \mathbb{R}^d \rightarrow \mathbb{R}^d \) operated on \( z \). Let \( z' = f(z) \) be the resulting random variable, hence the density of \( z' \) is

\[
p(z') = p(z) \left| \det \frac{\partial f}{\partial z} \right|^{-1} \tag{4}
\]

Normalizing flows thus considers successively transforming \( z_0 \) with a series of transformations \( \{f_1, f_2, ..., f_K\} \) to construct arbitrarily complex densities for \( z_K = f_K \circ f_{K-1} \circ ... \circ f_1(z_0) \) as

\[
\log p(z_K) = \log p(z_0) − \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k}{\partial z_{k-1}} \right| \tag{5}
\]

Hence the complexity lies in computing the determinant of the Jacobian matrix. Without further assumption about \( f \), the general complexity for that is \( O(d^3) \) where \( d \) is the dimension of \( z \). In order to accelerate this, [Rezende and Mohamed, 2015] proposed the following family of transformations that they termed as planar flow:

\[
f(z) = z + u\psi(w^T z + b) \tag{6}
\]

where \( w \in \mathbb{R}^d, u \in \mathbb{R}^d, b \in \mathbb{R} \) are parameters and \( h(\cdot) \) is a univariate non-linear function with derivative \( h'(\cdot) \). For this family of transformations, the determinant of the Jacobian matrix can be efficiently computed to facilitate model training.

\[
\det \frac{\partial f}{\partial z} = \det(I + u\psi(z)^T) = 1 + u^T \psi(z) \tag{7}
\]

where \( \psi(z) = h'(w^T z + b)w \). The computation cost of the determinant is hence reduced from \( O(d^3) \) to \( O(d) \).

Applying \( f \) to \( z \) can be viewed as feeding the input variable to a neural network with only one single hidden unit followed by a linear output layer which has the same dimension with the input layer. Obviously, because of the bottleneck caused by the single hidden unit, the capacity of the family of transformed density is hence limited.

3 The Proposed Method

In this section we propose to maximize an alternative lower bound of the data likelihood, which essentially results in an inference network which is almost likelihood free, in the sense that only the latent variables require evaluations from simple likelihoods and samples from all the auxiliary variables are sufficient for maximum likelihood inference.

3.1 Flexible Lower Bounds with Auxiliary Variables

The true data log-likelihood can be rewritten as

\[
\log p(x) = \log \mathbb{E}_{q(z|x)} \frac{p(x, z)}{q(z|x)} = \mathcal{L}_r(x) \tag{8}
\]
for any valid (conditional) variational density \( q(\tau|x) \) and \( \tau \) is an auxiliary variable (parameter) introduced into the variational model \( q \). Notice that \( \tau \) appears in the subscript of \( \mathcal{L}_\tau(x) \) to explicitly indicate that any practical estimate of \( \mathcal{L}_\tau(x) \) will depend on the choice of \( \tau \).

With the above formulation, one is tempted to maximize the benefits of auxiliary variable \( \tau \) by finding the best suitable \( \tau \) with maximizing, e.g.,

\[
\mathcal{L}_{\text{MAX}}(x) \equiv \max_{\tau} \log \mathbb{E}_{q(\tau|x)} \frac{p(x,z)}{q(z|\tau,x)}
\]  

(9)

In the context of formulating the variational distribution by neural networks, this is equivalent to adding more deterministic layers to the inference network, to accommodate the mapping from \( x \) to \( \tau \) and then the mapping from \( \tau \) to the stochastic layer on \( z \). This will definitely increase the model capacity for the variational distribution due to more parameters, however a key shortcoming is that even with the learned fixed \( \tau \), eventually it is still one single density \( q(z|\tau^*,x) \) which can hardly always capture the stochastic behaviors of the latent variable \( z \) given a data point \( x \).

For instance, when modeling binary data with classic VAE and IWAE, which are instantiated for both VAE and IWAE as Asymmetric Variational Autoencoder (ASY-VAE), which in turn is an auxiliary variable (parameter) introduced into the variational model \( q \). The resulting asymmetric structure from standard variational autoencoders and their variants, where the stochasticity of inference model \( q \) and the generative model \( p \) are both defined over \( z \), therefore we term the the model with the above bounds as Asymmetric Variational Autoencoder (ASY-VAE), which includes classic VAE and ASY-IWAE as two instantiations for VAE and IWAE, respectively.

Intuitively, ASYVAE can be thought of as optimizing a mix of various bounds \( \mathcal{L}_\tau \), thus enhances robustness for model learning which is of particular importance to Monte Carlo methods for estimating the bound and its gradients w.r.t. its parameters. Moreover, the resulting inference model enjoys higher flexibility, with the potential to capture complex structure of the posterior distribution, such as multi-modality.

For completeness, we briefly include that

**Proposition 1** Both \( \mathcal{L}_{\text{ASY-VAE}}(x) \) and \( \mathcal{L}_{\text{ASY-IWAE}}(x) \) are lower bounds of the true data log-likelihood, satisfying \( \log p(x) = \mathcal{L}_{\text{ASY-IWAE}}(x) \geq \mathcal{L}_{\text{ASY-VAE}}(x) \).

Proof is trivial from Jensen’s inequality, hence it’s omitted. Figure 1 shows a comparison of the inference models between classic VAE and the proposed ASYVAE.

**Remark 1** Though the first equality holds for any choice of distribution \( q(\tau|x) \) (whether \( \tau \) depends on \( x \) or not), for practical estimation with Monte Carlo methods, it becomes an inequality \( \log p(x) \geq \mathcal{L}_{\text{ASY-IWAE}}(x) \) and the bound tightens as the number of importance samples is increased [Burda et al., 2015]. The second inequality always holds when estimated with Monte Carlo samples.

**Remark 2** The above bounds are only concerned with one auxiliary variable \( \tau \), however \( \tau \) can also be a set of auxiliary variables. Moreover, with the same motivation, we can make the variational family of ASYVAE even more flexible by defining a series of \( k \) auxiliary variables, such that \( q(z, \tau_1, ..., \tau_k|x) = q(\tau_1|x)q(\tau_2|\tau_1) ... q(\tau_k|\tau_{k-1})q(z|\tau_1, ..., \tau_k) \) with sample generation process as

\[
\begin{align*}
\tau_1 &= f_1(x, \epsilon_1) \\
\tau_i &= f_i(\tau_{i-1}, \epsilon_k) & \text{for } i = 2, 3, ..., k
\end{align*}
\]  

(13)

and we have

**Proposition 2** The ASYVAE with \( k \) auxiliary random variables \( \{\tau_1, \tau_2, ..., \tau_k\} \) is also a lower bound to the true log-likelihood, satisfying \( \log p(x) = \mathcal{L}_{\text{ASY-IWAE-k}} \) where

\[
\mathcal{L}_{\text{ASY-IWAE-k}}(x) \equiv \mathbb{E}_{q(\tau_1|x)} \mathbb{E}_{q(\tau_2|\tau_1)} \cdots \mathbb{E}_{q(\tau_k|\tau_{k-1})} \mathbb{E}_{q(z|\tau_1, ..., \tau_k)} \left[ \log p(x, z) - \log q(z|x, \tau_1, ..., \tau_k) \right]
\]  

(14)

and

\[
\mathcal{L}_{\text{ASY-VAE-k}}(x) \equiv \mathbb{E}_{q(\tau_1|x)} \mathbb{E}_{q(\tau_2|\tau_1)} \cdots \mathbb{E}_{q(\tau_k|\tau_{k-1})} \frac{p(x,z)}{q(z|x, \tau_1, ..., \tau_k)}
\]  

(15)

Figure 1c illustrates the inference model of an ASYVAE with \( k \) auxiliary variables.

### 3.2 Learning with Monte Carlo Estimates

With no additional assumptions are made for the generative model \( p \) and variational model \( q \) other than that they are parameterized by neural network with stochastic layers, for both ASY-VAE and ASY-IWAE, we can estimate the corresponding bounds and its gradients of \( \mathcal{L}_{\text{ASY-VAE}} \) and \( \mathcal{L}_{\text{ASY-IWAE}} \) with ancestral sampling from the model.

For example, for ASY-VAE with one auxiliary variable \( \tau \), we estimate

\[
\hat{\mathcal{L}}_{\text{ASY-VAE}}(x) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left( \log p(x, z_{ij}) - \log q(z_{ij}|\tau_i, x) \right)
\]  

(16)
work essentially turns to be Normalizing Flow \cite{Rezende2015}. When \( z = f(x, \tau) \), the above bound turns to

\[
\log p_\theta(x) \geq \mathbb{E}_\tau \log \mathbb{E}_{z \sim q_\phi(z|x,\tau)} \frac{p_\theta(z)p_\theta(x|z)}{q_\phi(z|x,\tau)}
\]

(18)

\[
= \mathbb{E}_\tau \log p_\theta(z = f(x, \tau)) + \mathbb{E}_\tau \log p_\theta(x|z = f(x, \tau)) - \mathbb{E}_\tau \log q_\phi(z = f(x, \tau)|x,\tau)
\]

(19)

\[
= \mathbb{E}_\tau \log p_\theta(z = f(x, \tau)) + \mathbb{E}_\tau \log p_\theta(x|z = f(x, \tau))
\]

(20)

\[
- \left( \mathbb{E}_\tau \log q_\phi(\tau|x) - \mathbb{E}_\tau \log \det \frac{\partial f}{\partial \tau} \right)
\]

(21)

where we assume that \( f \) is invertible w.r.t \( \tau \) which is essentially the bound optimized by Normalizing Flows. When NF assumes multiple layers of transformations, it can also be thought of as defining a series of random variables based on warping variables on previous layers, such that \( z_i = f(z_{i-1}) \). However, it requires \( f \) to be bijective to enable likelihood evaluation of the intermediate variables, while VAE doesn’t place such restriction on \( f \).

4.2 Other methods with auxiliary variables

Relation to Hierarchical Variational Models (HVM) \cite{Ranganath2016} and Auxiliary Deep Generative Models (ADGM) \cite{Maaloe2016} are two closely related variational methods with auxiliary variables. HVM also considers enriching the variational model family by placing a prior over the latent variable for the variational distribution \( q(z|x) \). While ADGM takes another step to this goal, by placing a prior over the auxiliary variable on the generative model, which in some cases will keep the marginal generative distribution of the data invariant. It has been shown that HVM and ADGM are mathematically equivalent by \cite{Brümmer2016}. However, our proposed method doesn’t add any prior on the generative model and thus doesn’t change the structure of the generative model. We argue that with no priors over the introduced variables are defined, and due to the model structure that most of the auxiliary variables are not directly linked to explain the data (except for \( \tau \) when we assume multiple auxiliary variables for the variational distribution), our proposed method places the least restrictive constraints on
the variational distribution which enriches model capacity for more accurate posterior approximation.

4.3 Adversarial learning based inference models

Adversarial learning based inference models, such as Adversarial Autoencoders [Makhzani et al., 2015], Adversarial Variational Bayes [Mescheder et al., 2017], and Adversarially Learned Inference [Dumoulin et al., 2016], aim to maximize the ELBO without any variational likelihood evaluations at all. It can be shown that for the above adversarial learning based models, when the discriminator is trained to its optimum, the model is equivalent to optimizing the ELBO. However, due to the minimax game involved in the adversarial setting, practically at any moment it is not guaranteed that they are optimizing a lower bound of the true data likelihood, thus no maximum likelihood learning interpretation can be provided. However, in our proposed framework, we don’t require variational likelihood evaluations for almost all the variables in the variational model, except for the actual latent variables \( z \), while still maintaining the ML interpretation.

5 Experiments

5.1 Setups

To test our proposed AVAE for variational inference we use standard benchmark datasets MNIST\(^3\) and Omniglot\(^4\) [Lake et al., 2013]. Our method is general and can be applied to any formulation of the generative model \( p_{\theta}(x, z) \). For simplicity and fair comparison, in this paper, we focus on densities defined by stochastic neural networks, i.e., a broad family of flexible probabilistic generative models with its parameters defined by neural networks. Specifically, we consider the following two families of generative models

\[
G_1 : p_{\theta}(x, z) = p_{\theta}(z)p_{\theta}(x|z) \tag{22}
\]

\[
G_2 : p_{\theta}(x, z_1, z_2) = p_{\theta}(z_1)p_{\theta}(z_2|z_1)p_{\theta}(x|z_2) \tag{23}
\]

where \( p(z) \) and \( p(z_1) \) are the priors defined over \( z \) and \( z_1 \) for \( G_1 \) and \( G_2 \), respectively. All other conditional densities are specified with their parameters \( \theta \) defined by neural networks, therefore ending up with two stochastic neural networks. This network could have any number of layers, however in this paper, we focus on the ones which only have one and two stochastic layers, i.e., \( G_1 \) and \( G_2 \), to conduct a fair comparison with previous methods on similar network architectures, such as VAE, IWAE and Normalizing Flows.

We use the same network architectures for both \( G_1 \) and \( G_2 \) as in [Burda et al., 2015], specifically shown as follows

\( G_1 \) : A single Gaussian stochastic layer \( z \) with 50 units. In between the latent variable \( z \) and observation \( x \) there are two deterministic layers, each with 200 units;

\( G_2 \) : Two Gaussian stochastic layers \( z_1 \) and \( z_2 \) with 50 and 200 units, respectively. Two deterministic layers with

200 units connect the observation \( x \) and latent variable \( z_2 \), and two deterministic layers with 100 units are in between \( z_2 \) and \( z_1 \).

where a Gaussian stochastic layer consists of two fully connected linear layers, with one outputting the mean and the other outputting the logarithm of diagonal covariance. All other deterministic layers are fully connected with tanh nonlinearity.

For \( G_1 \), inference network with the following architecture is used

\[
\tau_1 = f_1(x||\epsilon_1) \text{ where } \epsilon_1 \sim \mathcal{N}(0, I) \tag{24}
\]

\[
\tau_i = f_i(\tau_{i-1}||\epsilon_i) \text{ where } \epsilon_i \sim \mathcal{N}(0, I)
\]

for \( i = 2, ..., k \)

\[
q(z|x, \tau_1, ..., \tau_k) = \mathcal{N}(\mu(x||\tau_1||...||\tau_k), \text{diag}(\sigma(x||\tau_1||...||\tau_k)) \tag{25}
\]

where \( || \) denotes the concatenation operator. All noise vectors \( \epsilon_s \) are set to be of 50 dimensions, and all other variables have the corresponding dimensions in the generative model. Inference network used for \( G_2 \) is the same, except for the Gaussian stochastic layer is defined for \( z_2 \). An additional Gaussian stochastic layer with \( z_2 \) as input is defined for \( z_1 \) with the dimensions of variables aligned to those in the generative model \( G_2 \). Further, Bernoulli observation models are assumed for both MNIST and Omniglot. For MNIST, we employ the static binarization strategy as in [Larochelle and Murray, 2011] while dynamic binarization is employed for Omniglot.

Our baseline models include VAE, IWAE and NF. Since our proposed method involves adding more layers to the inference network, we also include another enhanced version of VAE with more deterministic layers added to its inference network, which we term as VAE+\(^5\).\(^6\) All models are implemented in PyTorch\(^7\). Parameters of both the variational distribution and the generative distribution of all models are optimized with Adam [Kingma and Ba, 2014] for 2000 epochs, with a fixed learning rate of 0.0005; exponential decay rates for the 1st and 2nd moments at 0.9 and 0.999, respectively. Batch normalization [Ioffe and Szegedy, 2015] is also used, as it has been shown to improve learning for neural stochastic models [Sonderby et al., 2016].

5.2 Generative Density Estimation

For MNIST, models are trained and tuned on the 60,000 training and validation images, and estimated log-likelihood on the test set with 5000 importance weighted samples are reported. Table 1 presents the performance of all models, when the generative model is assumed to be from both \( G_1 \) and \( G_2 \).

Firstly, VAE+ achieves higher log-likelihood estimates than vanilla VAE due to the added more layers in the inference network, implying that a better posterior approximation is learned. Second, we observe that ASY-VAE achieves better density estimates than VAE+, which confirms our expectation that adding more auxiliary variables to the inference network

\(^3\)Data downloaded from http://www.cs.toronto.edu/~larochh/public/datasets/binarized_mnist/

\(^4\)Data downloaded from https://github.com/yburda/iwae/raw/master/datasets/OMNIGLOT/chardata.mat

\(^5\)VAE+ is a restricted version of AVAE with all the noise vectors \( \epsilon_s \) set to be constantly 0.

\(^6\)http://pytorch.org/
leads to a richer family of variational distributions. This suggests that incorporating more sources of stochasticity in the inference network is another key factor to provide a richer variational family, in addition to enlarging the variational model space by adding more deterministic layers. Similar trends can be observed on the importance weighted versions (ASY-IWAE versus IWAE). Overall, our proposed method ASY-IWAE outperforms IWAE by more than 1 nat on $G_1$ and 0.75 nat on $G_2$.

Results on OMNIGLOT are presented in Table 2 where similar trends can be observed as on MNIST. One observation different from MNIST is that, the gains from ASY-VAE and ASY-IWAE over VAE and IWAE respectively are not as large as they are on MNIST. It could be explained by the fact that Omniglot is a smaller set, roughly with a size of 40% of MNIST.

5.3 Generated Samples

After the models are trained, generative samples can be obtained by feeding $z \sim N(0, I)$ to the learned generative model $G_1$ (or $z_2 \sim N(0, I)$ to $G_2$). Since higher log-likelihood estimates are obtained on $G_2$, Figure 2 shows the random generative samples from our proposed method trained with $G_2$ on both MNIST and Omniglot, compared to real samples from the training sets. We observe the generated samples are visually consistent with the training data.

6 Conclusions

This paper presents a new framework to enrich variational family for variational inference, by introducing auxiliary random variables to the variational inference networks, based on an alternative lower bound of the data likelihood. We emphasize that the no variational likelihood evaluations are required for the auxiliary variables, hence allowing constructing complex distributions via warping a simple random noise vector as inputs to neural networks. This leads to a likelihood almost free inference network, in the sense that only variational likelihoods for the actual latent variables are needed while samples for all auxiliary variables are sufficient. It can be shown that the proposed inference network is essentially learning a richer probabilistic mixture of variational posteriors, thus achieving a much richer and flexible family of variational distributions. Empirical evaluations of the instantiated Asymmetric Variational Autoencoders (AVAE) demonstrate the effectiveness of incorporating auxiliary variables in variational inference.

It remains an interesting question of how many auxiliary variables are needed to best exploit the variational family for a specific problem. Also, since this paper only focuses on enriching the variational distribution, other techniques such as Normalizing Flows, Inverse Autoregressive Flows can be combined with the proposed framework. Hence, how to effectively aggregate the proposed method of enriching the variational family with other techniques, including adversarial learning, to achieve the best possible generative modeling is another promising direction to explore. Lastly, training deep neural network with arbitrary number of stochastic layers remains a challenging problem, a principled framework can be pursued.

References

[Blei et al., 2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3:993–1022, 2003.

[Brümmer, 2016] Niko Brümmer. Note on the equivalence of hierarchical variational models and auxiliary deep generative models. arXiv preprint arXiv:1603.02443, 2016.

[Burda et al., 2015] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. arXiv preprint arXiv:1509.00519, 2015.

[Dumoulin et al., 2016] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. arXiv preprint arXiv:1606.00704, 2016.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, pages 448–456, 2015.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.
Table 1: MNIST test set NLL with generative models $G_1$ and $G_2$ (lower is better)

| Model                        | $- \log p(x)$ on $G_1$ | $- \log p(x)$ on $G_2$ |
|------------------------------|------------------------|------------------------|
| VAE [Burda et al., 2015]    | 87.88                  | 85.65                  |
| IWAE ($IW = 50$) [Burda et al., 2015] | 86.10                  | 84.04                  |
| VAE+NF [Rezende and Mohamed, 2015] | -                     | $\leq$ 85.10           |
| VAE+ ($k = 1$)               | 87.56                  | 85.53                  |
| VAE+ ($k = 4$)               | 87.40                  | 85.23                  |
| VAE+ ($k = 8$)               | 87.28                  | 85.07                  |
| ASY-VAE ($k = 1$)            | 87.31                  | 85.23                  |
| ASY-VAE ($k = 4$)            | 87.16                  | 85.08                  |
| ASY-VAE ($k = 8$)            | 87.01                  | 84.97                  |
| ASY-IWAE ($IW = 50, k = 1$)  | 85.76                  | 83.77                  |
| ASY-IWAE ($IW = 50, k = 4$)  | 85.31                  | 83.52                  |
| ASY-IWAE ($IW = 50, k = 8$)  | 85.03                  | 83.29                  |

Table 2: Omniglot test set NLL with generative models $G_1$ and $G_2$ (lower is better)

| Model                        | $- \log p(x)$ on $G_1$ | $- \log p(x)$ on $G_2$ |
|------------------------------|------------------------|------------------------|
| VAE [Burda et al., 2015]    | 108.86                 | 107.93                 |
| IWAE ($IW = 50$) [Burda et al., 2015] | 104.87                 | 103.93                 |
| VAE+ ($k = 1$)               | 108.80                 | 107.89                 |
| VAE+ ($k = 4$)               | 108.64                 | 107.80                 |
| VAE+ ($k = 8$)               | 108.53                 | 107.67                 |
| ASY-VAE ($k = 1$)            | 108.74                 | 107.82                 |
| ASY-VAE ($k = 4$)            | 108.60                 | 107.65                 |
| ASY-VAE ($k = 8$)            | 108.41                 | 107.43                 |
| ASY-IWAE ($IW = 50, k = 1$)  | 104.83                 | 103.57                 |
| ASY-IWAE ($IW = 50, k = 4$)  | 104.80                 | 103.44                 |
| ASY-IWAE ($IW = 50, k = 8$)  | 104.63                 | 103.40                 |

[Kingma et al., 2016] Diederik P. Kingma, Tim Salimans, Rafal Józefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational autoencoders with inverse autoregressive flow. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4736–4744, 2016.

[Makhzani et al., 2015] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[Mescheder et al., 2017] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*, 2017.

[Mnih and Gregor, 2014] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1791–1799, 2014.

[Mnih and Rezende, 2016] Andriy Mnih and Danilo Jimenez Rezende. Variational inference for monte carlo objectives. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2188–2196, 2016.
[Ranganath et al., 2016] Rajesh Ranganath, Dustin Tran, and David M. Blei. Hierarchical variational models. In Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, pages 324–333, 2016.

[Rezende and Mohamed, 2015] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, pages 1530–1538, 2015.

[Sønderby et al., 2016] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pages 3738–3746, 2016.