Symmetric Variational Autoencoder and Connections to Adversarial Learning

Liqun Chen[1] **Shuyang Dai**[1] **Yunchen Pu**[1] **Erjin Zhou**[4] **Chunyuan Li**[1]

Qinliang Su[2] **Changyou Chen**[3] **Lawrence Carin**[1]

[1]Duke University, [2]Sun Yat-Sen University, China, [3]University at Buffalo, SUNY, [4]Megvii Research
liqun.chen@duke.edu

Abstract

A new form of the variational autoencoder (VAE) is proposed, based on the *symmetric* Kullback-Leibler divergence. It is demonstrated that learning of the resulting symmetric VAE (sVAE) has close connections to previously developed adversarial-learning methods. This relationship helps unify the previously distinct techniques of VAE and adversarially learning, and provides insights that allow us to ameliorate shortcomings with some previously developed adversarial methods. In addition to an analysis that motivates and explains the sVAE, an extensive set of experiments validate the utility of the approach.

1 Introduction

Generative models [Pu et al., 2015, 2016b] that are *descriptive* of data have been widely employed in statistics and machine learning. Factor models (FMs) represent one commonly used generative model [Tipping and Bishop, 1999], and mixtures of FMs have been employed to account for more-general data distributions [Ghahramani and Hinton, 1997]. These models typically have latent variables (e.g., factor scores) that are inferred given observed data; the latent variables are often used for a down-stream goal, such as classification [Carvalho et al., 2008]. After training, such models are useful for inference tasks given subsequent observed data. However, when one draws from such models, by drawing latent variables from the prior and pushing them through the model to synthesize data, the synthetic data typically do not appear to be realistic. This suggests that while these models may be useful for analyzing observed data in terms of inferred latent variables, they are

also capable of describing a large set of data that do not appear to be real.

The generative adversarial network (GAN) [Goodfellow et al., 2014] represents a significant recent advance toward development of generative models that are capable of synthesizing realistic data. Such models also employ latent variables, drawn from a simple distribution analogous to the aforementioned prior, and these random variables are fed through a (deep) neural network. The neural network acts as a functional transformation of the original random variables, yielding a model capable of representing sophisticated distributions. Adversarial learning discourages the network from yielding synthetic data that are unrealistic, from the perspective of a learned neural-network-based classifier. However, GANs are notoriously difficult to train, and multiple generalizations and techniques have been developed to improve learning performance [Salimans et al., 2016], for example Wasserstein GAN (WGAN) [Arjovsky and Bottou, 2017, Arjovsky et al., 2017] and energy-based GAN (EB-GAN) [Zhao et al., 2017].

While the original GAN and variants were capable of synthesizing highly realistic data (e.g., images), the models lacked the ability to infer the latent variables given observed data. This limitation has been mitigated recently by methods like adversarial learned inference (ALI) [Dumoulin et al., 2017], and related approaches. However, ALI appears to be inadequate from the standpoint of inference, in that, given observed data and associated inferred latent variables, the subsequently synthesized data often do not look particularly close to the original data.

The variational autoencoder (VAE) [Kingma and Welling, 2014] is a class of generative models that precedes GAN. VAE learning is based on optimizing a variational lower bound, connected to inferring an approximate posterior distribution on latent variables; such learning is typically not performed in an adversarial manner. VAEs have been demonstrated to be effective models for inferring latent variables, in that the reconstructed data do typically look like the original data, albeit in a blurry manner [Dumoulin et al., 2017, Pu et al., 2016a, 2017a]. The form of the VAE

has been generalized recently, in terms of the adversarial variational Bayesian (AVB) framework [Mescheder et al., 2016]. This model yields general forms of encoders and decoders, but it is based on the original variational Bayesian (VB) formulation. The original VB framework yields a lower bound on the log likelihood of the observed data, and therefore model learning is connected to maximum-likelihood (ML) approaches. From the perspective of designing generative models, it has been recognized recently that ML-based learning has limitations [Arjovsky and Bottou, 2017]: such learning tends to yield models that match observed data, but also have a high probability of generating unrealistic synthetic data.

The original VAE employs the Kullback-Leibler divergence to constitute the variational lower bound. As is well known, the KL distance metric is asymmetric. We demonstrate that this asymmetry encourages design of decoders (generators) that often yield unrealistic synthetic data when the latent variables are drawn from the prior. From a different but related perspective, the encoder infers latent variables (across all training data) that only encompass a subset of the prior. As demonstrated below, these limitations of the encoder and decoder within conventional VAE learning are intertwined.

We consequently propose a new *symmetric* VAE (sVAE), based on a symmetric form of the KL divergence and associated variational bound. The proposed sVAE is learned using an approach related to that employed in the AVB [Mescheder et al., 2016], but in a new manner connected to the symmetric variational bound. Analysis of the sVAE demonstrates that it has close connections to ALI [Dumoulin et al., 2017], WGAN [Arjovsky et al., 2017] and to the original GAN [Goodfellow et al., 2014] framework; in fact, ALI is recovered exactly, as a special case of the proposed sVAE. This provides a new and explicit linkage between the VAE (after it is made symmetric) and a wide class of adversarially trained generative models. Additionally, with this insight, we are able to ameliorate much of the aforementioned limitations of ALI, from the perspective of data reconstruction. In addition to analyzing properties of the sVAE, we demonstrate excellent performance on an extensive set of experiments.

2 Review of Variational Autoencoder

2.1 Background

Assume observed data samples $x \sim q(x)$, where $q(x)$ is the true and unknown distribution we wish to approximate. Consider $p_\theta(x|z)$, a model with parameters θ and latent code z. With prior $p(z)$ on the codes, the modeled generative process is $x \sim p_\theta(x|z)$, with $z \sim p(z)$. We may marginalize out the latent codes, and hence the model is $x \sim p_\theta(x) = \int dz p_\theta(x|z)p(z)$. To learn θ,

we typically seek to maximize the expected log likelihood: $\hat{\theta} = \text{argmax}_\theta \, \mathbb{E}_{q(x)} \log p_\theta(x)$, where one typically invokes the approximation $\mathbb{E}_{q(x)} \log p_\theta(x) \approx \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(x_n)$ assuming N iid observed samples $\{x_n\}_{n=1,N}$.

It is typically intractable to evaluate $p_\theta(x)$ directly, as $\int dz p_\theta(x|z)p(z)$ generally doesn't have a closed form. Consequently, a typical approach is to consider a model $q_\phi(z|x)$ for the posterior of the latent code z given observed x, characterized by parameters ϕ. Distribution $q_\phi(z|x)$ is often termed an *encoder*, and $p_\theta(x|z)$ is a *decoder* [Kingma and Welling, 2014]; both are here stochastic, *vis-à-vis* their deterministic counterparts associated with a traditional autoencoder [Vincent et al., 2010]. Consider the variational expression

$$\mathcal{L}_x(\theta, \phi) = \mathbb{E}_{q(x)} \mathbb{E}_{q_\phi(z|x)} \log \left[\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} \right] \quad (1)$$

In practice the expectation wrt $x \sim q(x)$ is evaluated via sampling, assuming N observed samples $\{x_n\}_{n=1,N}$. One typically must also utilize sampling from $q_\phi(z|x)$ to evaluate the corresponding expectation in (1). Learning is effected as $(\hat{\theta}, \hat{\phi}) = \text{argmax}_{\theta, \phi} \, \mathcal{L}_x(\theta, \phi)$, and a model so learned is termed a variational autoencoder (VAE) [Kingma and Welling, 2014].

It is well known that $\mathcal{L}_x(\theta, \phi) = \mathbb{E}_{q(x)}[\log p_\theta(x) - \text{KL}(q_\phi(z|x) \| p_\theta(z|x))] \leq \mathbb{E}_{q(x)}[\log p_\theta(x)]$. Alternatively, the variational expression may be represented as

$$\mathcal{L}_x(\theta, \phi) = -\text{KL}(q_\phi(x, z) \| p_\theta(x, z)) + C_x \quad (2)$$

where $q_\phi(x, z) = q(x)q_\phi(z|x)$, $p_\theta(x, z) = p(z)p_\theta(x|z)$ and $C_x = \mathbb{E}_{q(x)} \log q(x)$. One may readily show that

$$\text{KL}(q_\phi(x, z) \| p_\theta(x, z))$$
$$= \mathbb{E}_{q(x)} \text{KL}(q_\phi(z|x) \| p_\theta(z|x)) + \text{KL}(q(x) \| p_\theta(x)) \quad (3)$$
$$= \mathbb{E}_{q_\phi(z)} \text{KL}(q_\phi(x|z) \| p_\theta(x|z)) + \text{KL}(q_\phi(z) \| p(z)) \quad (4)$$

where $q_\phi(z) = \int q(x)q_\phi(z|x)dx$. To maximize $\mathcal{L}_x(\theta, \phi)$, we seek minimization of $\text{KL}(q_\phi(x, z) \| p_\theta(x, z))$. Hence, from (3) the goal is to align $p_\theta(x)$ with $q(x)$, while from (4) the goal is to align $q_\phi(z)$ with $p(z)$. The other terms seek to match the respective conditional distributions. All of these conditions are implied by minimizing $\text{KL}(q_\phi(x, z) \| p_\theta(x, z))$. However, the KL divergence is *asymmetric*, which yields limitations wrt the learned model.

2.2 Limitations of the VAE

The support $\mathcal{S}_{p(z)}^\epsilon$ of a distribution $p(z)$ is defined as the member of the set $\{\tilde{\mathcal{S}}_{p(z)}^\epsilon : \int_{\tilde{\mathcal{S}}_{p(z)}^\epsilon} p(z)dz = 1 - \epsilon\}$ with minimum size $\|\tilde{\mathcal{S}}_{p(z)}^\epsilon\| \triangleq \int_{\tilde{\mathcal{S}}_{p(z)}^\epsilon} dz$. We are typically interested in $\epsilon \to 0^+$. For notational convenience we replace $\mathcal{S}_{p(z)}^\epsilon$ with $\mathcal{S}_{p(z)}$, with the understanding ϵ is small. We also

Liqun Chen[1], **Shuyang Dai**[1], **Yunchen Pu**[1], **Erjin Zhou**[4], **Chunyuan Li**[1]

define $\mathcal{S}_{p(\boldsymbol{z})_-}$ as the largest set for which $\int_{\mathcal{S}_{p(\boldsymbol{z})_-}} p(\boldsymbol{z})d\boldsymbol{z} = \epsilon$, and hence $\int_{\mathcal{S}_{p(\boldsymbol{z})}} p(\boldsymbol{z})d\boldsymbol{z} + \int_{\mathcal{S}_{p(\boldsymbol{z})_-}} p(\boldsymbol{z})d\boldsymbol{z} = 1$. For simplicity of exposition, we assume $\mathcal{S}_{p(\boldsymbol{z})}$ and $\mathcal{S}_{p(\boldsymbol{z})_-}$ are unique; the meaning of the subsequent analysis is unaffected by this assumption.

Consider $-\text{KL}(q(\boldsymbol{x})\|p_{\boldsymbol{\theta}}(\boldsymbol{x})) = \mathbb{E}_{q(\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - C_x$, which from (2) and (3) we seek to make large when learning $\boldsymbol{\theta}$. The following discussion borrows insights from [Arjovsky et al., 2017], although that analysis was different, in that it was not placed within the context of the VAE. Since $\int_{\mathcal{S}_{q(\boldsymbol{x})_-}} q(\boldsymbol{x}) \log p_{\boldsymbol{\theta}}(\boldsymbol{x})d\boldsymbol{x} \approx 0$, $\mathbb{E}_{q(\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \approx \int_{\mathcal{S}_{q(\boldsymbol{x})}} q(\boldsymbol{x}) \log p_{\boldsymbol{\theta}}(\boldsymbol{x})d\boldsymbol{x}$, and $\mathcal{S}_{q(\boldsymbol{x})} = (\mathcal{S}_{q(\boldsymbol{x})} \cap \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}) \cup (\mathcal{S}_{q(\boldsymbol{x})} \cap \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})_-})$. If $\mathcal{S}_{q(\boldsymbol{x})} \cap \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})_-} \neq \emptyset$, there is a strong (negative) penalty introduced by $\int_{\mathcal{S}_{q(\boldsymbol{x})} \cap \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})_-}} q(\boldsymbol{x}) \log p_{\boldsymbol{\theta}}(\boldsymbol{x})d\boldsymbol{x}$, and therefore maximization of $\mathbb{E}_{q(\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ encourages $\mathcal{S}_{q(\boldsymbol{x})} \cap \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})_-} = \emptyset$. By contrast, there is not a substantial penalty to $\mathcal{S}_{q(\boldsymbol{x})_-} \cap \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})} \neq \emptyset$.

Summarizing these conditions, the goal of maximizing $-\text{KL}(q(\boldsymbol{x})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}))$ encourages $\mathcal{S}_{q(\boldsymbol{x})} \subset \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}$. This implies that $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ can synthesize all $\boldsymbol{x}$ that may be drawn from $q(\boldsymbol{x})$, but additionally there is (often) high probability that $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ will synthesize $\boldsymbol{x}$ that will *not* be drawn from $q(\boldsymbol{x})$.

Similarly, $-\text{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z})\|p(\boldsymbol{z})) = h(q_{\boldsymbol{\phi}}(\boldsymbol{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})} \log p(\boldsymbol{z})$ encourages $\mathcal{S}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})} \subset \mathcal{S}_{p(\boldsymbol{z})}$, and the commensurate goal of increasing differential entropy $h(q_{\boldsymbol{\phi}}(\boldsymbol{z})) = -\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})} \log q_{\boldsymbol{\phi}}(\boldsymbol{z})$ encourages that $\mathcal{S}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})} \cap \mathcal{S}_{p(\boldsymbol{z})}$ be as large as possible.

Hence, the goal of large $-\text{KL}(q(\boldsymbol{x})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}))$ and $-\text{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z})\|p(\boldsymbol{z}))$ are saying the same thing, from different perspectives: (*i*) seeking large $-\text{KL}(q(\boldsymbol{x})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}))$ implies that there is a high probability that $\boldsymbol{x}$ drawn from $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ will be different from those drawn from $q(\boldsymbol{x})$, and (*ii*) large $-\text{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z})\|p(\boldsymbol{z}))$ implies that $\boldsymbol{z}$ drawn from $p(\boldsymbol{z})$ are likely to be different from those drawn from $q_{\boldsymbol{\phi}}(\boldsymbol{z})$, with $\boldsymbol{z} \in \{\mathcal{S}_{p(\boldsymbol{z})} \cap \mathcal{S}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})_-}\}$ responsible for the $\boldsymbol{x}$ that are inconsistent with $q(\boldsymbol{x})$. These properties are summarized in Fig. 1.

Considering the remaining terms in (3) and (4), and using similar logic on $-\mathbb{E}_{q(\boldsymbol{x})}\text{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})) = h(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})) + \mathbb{E}_{q(\boldsymbol{x})}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, the model encourages $\mathcal{S}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \subset \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}$. From $-\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\text{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})) = h(q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$, the model also encourages $\mathcal{S}_{q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})} \subset \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})}$. The differential entropies $h(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}))$ and $h(q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z}))$ encourage that $\mathcal{S}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \cap \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}$ and $\mathcal{S}_{q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})} \cap \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})}$ be as large as possible. Since $\mathcal{S}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \subset \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}$, it is anticipated that $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ will under-estimate the variance of $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$, as is common with the variational approximation to the
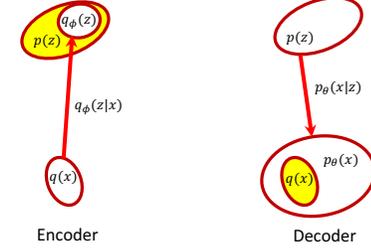

Figure 1: Characteristics of the encoder and decoder of the conventional VAE $\mathcal{L}_x$, for which the support of the distributions satisfy $\mathcal{S}_{q(\boldsymbol{x})} \subset \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}$ and $\mathcal{S}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})} \subset \mathcal{S}_{p(\boldsymbol{z})}$, implying that the generative model $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ has a high probability of generating unrealistic draws.
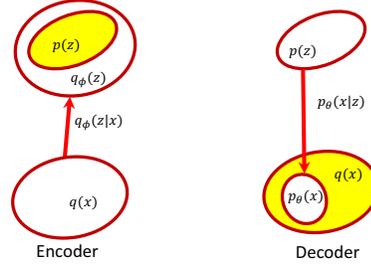

Figure 2: Characteristics of the new VAE expression, $\mathcal{L}_z$.

posterior [Blei et al., 2017].

3 Refined VAE: Imposition of Symmetry

3.1 Symmetric KL divergence

Consider the new variational expression

$$\begin{aligned}\mathcal{L}_z(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \mathbb{E}_{p(\boldsymbol{z})}\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})} \log\Big[\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})}\Big] \quad (5)\\ &= -\text{KL}(p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})\|q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})) + C_z \quad (6)\end{aligned}$$

where $C_z = -h(p(\boldsymbol{z}))$. Using logic analogous to that applied to $\mathcal{L}_x$, maximization of $\mathcal{L}_z$ encourages distribution supports reflected in Fig. 2.

Defining $\mathcal{L}_{xz}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathcal{L}_x(\boldsymbol{\theta}, \boldsymbol{\phi}) + \mathcal{L}_z(\boldsymbol{\theta}, \boldsymbol{\phi})$, we have

$$\mathcal{L}_{xz}(\boldsymbol{\theta}, \boldsymbol{\phi}) = -\text{KL}_s(q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})) + K \quad (7)$$

where $K = C_x + C_z$, and the *symmetric* KL divergence is $\text{KL}_s(q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})) \triangleq \text{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})) + \text{KL}(p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})\|q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}))$. Maximization of $\mathcal{L}_{xz}(\boldsymbol{\theta}, \boldsymbol{\phi})$ seeks minimizing $\text{KL}_s(q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}))$, which *simultaneously* imposes the conditions summarized in Figs. 1 and 2.

One may show that

$$\begin{aligned}\text{KL}_s(q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})) &= \mathbb{E}_{p(\boldsymbol{z})}\text{KL}(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\|q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z}))\\ &+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\text{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})) + \text{KL}_s(p(\boldsymbol{z})\|q_{\boldsymbol{\phi}}(\boldsymbol{z})) \quad (8)\\ &= \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}\text{KL}(p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\|q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}))\\ &+ \mathbb{E}_{q(\boldsymbol{x})}\text{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})) + \text{KL}_s(p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})) \quad (9)\end{aligned}$$

Considering the representation in (9), the goal of small $\mathrm{KL}_s(p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x}))$ encourages $\mathcal{S}_{q(\boldsymbol{x})} \subset \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}$ *and* $\mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})} \subset \mathcal{S}_{q(\boldsymbol{x})}$, and hence that $\mathcal{S}_{q(\boldsymbol{x})} = \mathcal{S}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}$. Further, since $-\mathrm{KL}_s(p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})) = \mathbb{E}_{q(\boldsymbol{x})}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) + \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}\log q(\boldsymbol{x}) + h(p_{\boldsymbol{\theta}}(\boldsymbol{x})) - C_x$, maximization of $-\mathrm{KL}_s(p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x}))$ seeks to minimize the cross-entropy between $q(\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, encouraging a complete matching of the distributions $q(\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, not just shared support. From (8), a match is simultaneously encouraged between $p(\boldsymbol{z})$ and $q_{\boldsymbol{\phi}}(\boldsymbol{z})$. Further, the respective conditional distributions are also encouraged to match.

3.2 Adversarial solution

Assuming fixed $(\boldsymbol{\theta}, \boldsymbol{\phi})$, and using logic analogous to Proposition 1 in [Mescheder et al., 2016], we consider

$$
\begin{aligned}
g(\boldsymbol{\psi}) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})}\log(1 - \sigma(f_{\boldsymbol{\psi}}(\boldsymbol{x},\boldsymbol{z})) \\
&+ \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}\log\sigma(f_{\boldsymbol{\psi}}(\boldsymbol{x},\boldsymbol{z}))
\end{aligned} \tag{10}
$$

where $\sigma(\zeta) = 1/(1 + \exp(-\zeta))$. The scalar function $f_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z})$ is represented by a deep neural network with parameters $\boldsymbol{\psi}$, and network inputs $(\boldsymbol{x}, \boldsymbol{z})$. For fixed $(\boldsymbol{\theta}, \boldsymbol{\phi})$, the parameters $\boldsymbol{\psi}^*$ that maximize $g(\boldsymbol{\psi})$ yield

$$
f_{\boldsymbol{\psi}^*}(\boldsymbol{x}, \boldsymbol{z}) = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) \tag{11}
$$

and hence

$$
\begin{aligned}
\mathcal{L}_x(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})}f_{\boldsymbol{\psi}^*}(\boldsymbol{x}, \boldsymbol{z}) + C_x \tag{12} \\
\mathcal{L}_z(\boldsymbol{\theta}, \boldsymbol{\phi}) &= -\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}f_{\boldsymbol{\psi}^*}(\boldsymbol{x}, \boldsymbol{z}) + C_z \tag{13}
\end{aligned}
$$

Hence, to optimize $\mathcal{L}_{xz}(\boldsymbol{\theta}, \boldsymbol{\phi})$ we consider the cost function

$$
\begin{aligned}
\ell(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\psi}^*) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})}f_{\boldsymbol{\psi}^*}(\boldsymbol{x}, \boldsymbol{z}) \\
&\quad - \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}f_{\boldsymbol{\psi}^*}(\boldsymbol{x}, \boldsymbol{z})
\end{aligned} \tag{14}
$$

Assuming (11) holds, we have

$$
\ell(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\psi}^*) = -\mathrm{KL}_s(q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})) \leq 0 \tag{15}
$$

and the goal is to achieve $\ell(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\psi}^*) = 0$ through joint optimization of $(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\psi}^*)$. Model learning consists of alternating between (10) and (14), maximizing (10) wrt $\boldsymbol{\psi}$ with $(\boldsymbol{\theta}, \boldsymbol{\phi})$ fixed, and maximizing (14) wrt $(\boldsymbol{\theta}, \boldsymbol{\phi})$ with $\boldsymbol{\psi}$ fixed.

The expectations in (10) and (14) are approximated by averaging over samples, and therefore to implement this solution we need only be able to sample from $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ and $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, and we do not require explicit forms for these distributions. For example, a draw from $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ may be constituted as $\boldsymbol{z} = h_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{\epsilon})$, where $h_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{\epsilon})$ is implemented as a neural network with parameters $\boldsymbol{\phi}$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I})$.

3.3 Interpretation in terms of LRT statistic

In (10) a classifier is designed to distinguish between samples $(\boldsymbol{x}, \boldsymbol{z})$ drawn from $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ and from $q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) = q(\boldsymbol{x})q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. Implicit in that expression is that there is equal probability that either of these distributions are selected for drawing $(\boldsymbol{x}, \boldsymbol{z})$, i.e., that $(\boldsymbol{x}, \boldsymbol{z}) \sim [p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) + q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})]/2$. Under this assumption, given observed $(\boldsymbol{x}, \boldsymbol{z})$, the probability of it being drawn from $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$ is $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})/(p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) + q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}))$, and the probability of it being drawn from $q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})$ is $q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})/(p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) + q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}))$ [Goodfellow et al., 2014]. Since the denominator $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) + q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})$ is shared by these distributions, and assuming function $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})/q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})$ is known, an observed $(\boldsymbol{x}, \boldsymbol{z})$ is inferred as being drawn from the underlying distributions as

$$
\text{if } p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})/q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) > 1, \ (\boldsymbol{x}, \boldsymbol{z}) \to p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) \tag{16}
$$
$$
\text{if } p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})/q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) < 1, \ (\boldsymbol{x}, \boldsymbol{z}) \to q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) \tag{17}
$$

This is the well-known likelihood ratio test (LRT) [Trees, 2001], and is reflected by (11). We have therefore derived a learning procedure based on the log-LRT, as reflected in (14). The solution is "adversarial," in the sense that when optimizing $(\boldsymbol{\theta}, \boldsymbol{\phi})$ the objective in (14) seeks to "fool" the LRT test statistic, while for fixed $(\boldsymbol{\theta}, \boldsymbol{\phi})$ maximization of (10) wrt $\boldsymbol{\psi}$ corresponds to updating the LRT. This adversarial solution comes as a natural consequence of symmetrizing the traditional VAE learning procedure.

4 Connections to Prior Work

4.1 Adversarially Learned Inference

The adversarially learned inference (ALI) [Dumoulin et al., 2017] framework seeks to learn both an encoder and decoder, like the approach proposed above, and is based on optimizing

$$
\begin{aligned}
(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}) &= \operatorname{argmin}_{\boldsymbol{\theta}, \boldsymbol{\phi}} \max_{\boldsymbol{\psi}}\{\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}\log\sigma(f_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z})) \\
&+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})}\log(1 - \sigma(f_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z})))\}
\end{aligned} \tag{18}
$$

This has similarities to the proposed approach, in that the term $\max_{\boldsymbol{\psi}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}\log\sigma(f_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})}\log(1 - \sigma(f_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z})))$ is identical to our maximization of (10) wrt $\boldsymbol{\psi}$. However, in the proposed approach, rather than directly then optimizing wrt $(\boldsymbol{\theta}, \boldsymbol{\phi})$, as in (18), in (14) the result from this term is used to define $f_{\boldsymbol{\psi}^*}(\boldsymbol{x}, \boldsymbol{z})$, which is then employed in (14) to subsequently optimize over $(\boldsymbol{\theta}, \boldsymbol{\phi})$.

Note that $\log\sigma(\cdot)$ is a monotonically increasing function, and therefore we may replace (14) as

$$
\begin{aligned}
\ell'(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\psi}^*) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})}\log\sigma(f_{\boldsymbol{\psi}^*}(\boldsymbol{x}, \boldsymbol{z})) \\
&+ \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}\log\sigma(-f_{\boldsymbol{\psi}^*}(\boldsymbol{x}, \boldsymbol{z}))
\end{aligned} \tag{19}
$$

and note $\sigma(-f_{\boldsymbol{\psi}^*}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta}, \boldsymbol{\phi})) = 1 - \sigma(f_{\boldsymbol{\psi}^*}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta}, \boldsymbol{\phi}))$. Maximizing (19) wrt $(\boldsymbol{\theta}, \boldsymbol{\phi})$ with fixed $\boldsymbol{\psi}^*$ corresponds to

Liqun Chen[1], Shuyang Dai[1], Yunchen Pu[1], Erjin Zhou[4], Chunyuan Li[1]

the minimization wrt $(\boldsymbol{\theta}, \boldsymbol{\phi})$ reflected in (18). Hence, the proposed approach is *exactly* ALI, if in (14) we replace $\pm f_{\boldsymbol{\psi}^*}$ with $\log \sigma(\pm f_{\boldsymbol{\psi}^*})$.

4.2 Original GAN

The proposed approach assumed both a decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ and an encoder $p_{\gamma}(\boldsymbol{z}|\boldsymbol{x})$, and we considered the symmetric $\text{KL}_s(q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) \| p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}))$. We now simplify the model for the case in which we only have a decoder, and the synthesized data are drawn $\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ with $\boldsymbol{z} \sim p(\boldsymbol{z})$, and we wish to learn $\boldsymbol{\theta}$ such that data synthesized in this manner match observed data $\boldsymbol{x} \sim q(\boldsymbol{x})$. Consider the symmetric

$$\begin{aligned} \text{KL}_s(q(\boldsymbol{x}) \| p_{\boldsymbol{\theta}}(\boldsymbol{x})) &= \mathbb{E}_{p(\boldsymbol{z})} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})} f_{\boldsymbol{\psi}^*}(\boldsymbol{x}) \\ &\quad - \mathbb{E}_{q(\boldsymbol{x})} f_{\boldsymbol{\psi}^*}(\boldsymbol{x}) \end{aligned} \quad (20)$$

where for fixed $\boldsymbol{\theta}$

$$f_{\boldsymbol{\psi}^*}(\boldsymbol{x}) = \log(p_{\boldsymbol{\theta}}(\boldsymbol{x})/q(\boldsymbol{x})) \quad (21)$$

We consider a simplified form of (10), specifically

$$\begin{aligned} g(\boldsymbol{\psi}) &= \mathbb{E}_{p(\boldsymbol{z})} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})} \log \sigma(f_{\boldsymbol{\psi}}(\boldsymbol{x})) \\ &\quad + \mathbb{E}_{q(\boldsymbol{x})} \log(1 - \sigma(f_{\boldsymbol{\psi}}(\boldsymbol{x})) \end{aligned} \quad (22)$$

which we seek to maximize wrt $\boldsymbol{\psi}$ with fixed $\boldsymbol{\theta}$, with optimal solution as in (21). We optimize $\boldsymbol{\theta}$ seeking to maximize $-\text{KL}_s(q(\boldsymbol{x}) \| p_{\boldsymbol{\theta}}(\boldsymbol{x}))$, as $\text{argmax}_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \boldsymbol{\psi}^*)$ where

$$\ell(\boldsymbol{\theta}; \boldsymbol{\psi}^*) = \mathbb{E}_{q(\boldsymbol{x})} f_{\boldsymbol{\psi}^*}(\boldsymbol{x}) - \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})} f_{\boldsymbol{\psi}^*}(\boldsymbol{x}) \quad (23)$$

with $\mathbb{E}_{q(\boldsymbol{x})} f_{\boldsymbol{\psi}^*}(\boldsymbol{x})$ independent of the update parameter $\boldsymbol{\theta}$. We observe that in seeking to maximize $\ell(\boldsymbol{\theta}; \boldsymbol{\psi}^*)$, parameters $\boldsymbol{\theta}$ are updated as to "fool" the log-LRT $\log[q(\boldsymbol{x})/p_{\boldsymbol{\theta}}(\boldsymbol{x})]$. Learning consists of iteratively updating $\boldsymbol{\psi}$ by maximizing $g(\boldsymbol{\psi})$ and updating $\boldsymbol{\theta}$ by maximizing $\ell(\boldsymbol{\theta}; \boldsymbol{\psi}^*)$.

Recall that $\log \sigma(\cdot)$ is a monotonically increasing function, and therefore we may replace (23) as

$$\ell'(\boldsymbol{\theta}; \boldsymbol{\psi}^*) = \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})} \log \sigma(-f_{\boldsymbol{\psi}^*}(\boldsymbol{x})) \quad (24)$$

Using the same logic as discussed above in the context of ALI, maximizing $\ell'(\boldsymbol{\theta}; \boldsymbol{\psi}^*)$ wrt $\boldsymbol{\theta}$ may be replaced by minimization, by transforming $\sigma(\mu) \to \sigma(-\mu)$. With this simple modification, minimizing the modified (24) wrt $\boldsymbol{\theta}$ and maximizing (22) wrt $\boldsymbol{\psi}$, we *exactly* recover the original GAN [Goodfellow et al., 2014], for the special (but common) case of a sigmoidal discriminator.

4.3 Wasserstein GAN

The Wasserstein GAN (WGAN) [Arjovsky et al., 2017] setup is represented as

$$\boldsymbol{\theta} = \text{argmin}_{\boldsymbol{\theta}} \max_{\boldsymbol{\psi}} \{ \mathbb{E}_{q(\boldsymbol{x})} f_{\boldsymbol{\psi}}(\boldsymbol{x}) - \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})} f_{\boldsymbol{\psi}}(\boldsymbol{x}) \} \quad (25)$$

where $f_{\boldsymbol{\psi}}(\boldsymbol{x})$ must be a 1-Lipschitz function. Typically $f_{\boldsymbol{\psi}}(\boldsymbol{x})$ is represented by a neural network with parameters $\boldsymbol{\psi}$, with parameter clipping or ℓ_2 regularization on the weights (to constrain the amplitude of $f_{\boldsymbol{\psi}}(\boldsymbol{x})$). Note that WGAN is closely related to (23), but in WGAN $f_{\boldsymbol{\psi}}(\boldsymbol{x})$ doesn't make an explicit connection to the underlying likelihood ratio, as in (21).

It is believed that the current paper is the first to consider symmetric variational learning, introducing $\mathcal{L}_z$, from which we have made explicit connections to previously developed adversarial-learning methods. Previous efforts have been made to match $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ to $p(\boldsymbol{z})$, which is a consequence of the proposed symmetric VAE (sVAE). For example, [Makhzani et al., 2016] introduced a modification to the original VAE formulation, but it loses connection to the variational lower bound [Mescheder et al., 2016].

4.4 Amelioration of vanishing gradients

As discussed in [Arjovsky et al., 2017], a key distinction between the WGAN framework in (25) and the original GAN [Goodfellow et al., 2014] is that the latter uses a binary discriminator to distinguish real and synthesized data; the $f_{\boldsymbol{\psi}}(\boldsymbol{x})$ in WGAN is a 1-Lipschitz function, rather than an explicit discriminator. A challenge with GAN is that as the discriminator gets better at distinguishing real and synthetic data, the gradients wrt the discriminator parameters vanish, and learning is undermined. The WGAN was designed to ameliorate this problem [Arjovsky et al., 2017].

From the discussion in Section 4.1, we note that the key distinction between the proposed sVAE and ALI is that the latter uses a binary discriminator to distinguish $(\boldsymbol{x}, \boldsymbol{z})$ manifested via the generator from $(\boldsymbol{x}, \boldsymbol{z})$ manifested via the encoder. By contrast, the sVAE uses a log-LRT, rather than a binary classifier, with it inferred in an adversarial manner. ALI is therefore undermined by vanishing gradients as the binary discriminator gets better, with this avoided by sVAE. The sVAE brings the same intuition associated with WGAN (addressing vanishing gradients) to a generalized VAE framework, with a generator *and* a decoder; WGAN only considers a generator. Further, as discussed in Section 4.3, unlike WGAN, which requires gradient clipping or other forms of regularization to approximate 1-Lipschitz functions, in the proposed sVAE the $f_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z})$ arises naturally from the symmetrized VAE and we do not require imposition of Lipschitz conditions. As discussed in Section 6, this simplification has yielded robustness in implementation.

5 Model Augmentation

A significant limitation of the original ALI setup is an inability to accurately reconstruct observed data via the process $\boldsymbol{x} \rightarrow \boldsymbol{z} \rightarrow \hat{\boldsymbol{x}}$ [Dumoulin et al., 2017]. With the

proposed sVAE, which is intimately connected to ALI, we may readily address this shortcoming. The variational expressions discussed above may be written as $\mathcal{L}_x = \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) - \mathbb{E}_{q(\boldsymbol{x})}\text{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}))$ and $\mathcal{L}_z = \mathbb{E}_{p_\theta(\boldsymbol{x},\boldsymbol{z})} \log p_\phi(\boldsymbol{z}|\boldsymbol{x}) - \mathbb{E}_{p(\boldsymbol{z})}\text{KL}(p_\theta(\boldsymbol{x}|\boldsymbol{z})\|q(\boldsymbol{x}))$. In both of these expressions, the first term to the right of the equality enforces model fit, and the second term penalizes the posterior distribution *for individual data samples* for being dissimilar from the prior (i.e., penalizes $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ from being dissimilar from $p(\boldsymbol{z})$, and likewise wrt $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ and $q(\boldsymbol{x})$). The proposed sVAE encourages the *cumulative* distributions $q_\phi(\boldsymbol{z})$ and $p_\theta(\boldsymbol{x})$ to match $p(\boldsymbol{z})$ and $q(\boldsymbol{x})$, respectively. By simultaneously encouraging more peaked $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, we anticipate better "cycle consistency" [Zhu et al., 2017] and hence more accurate reconstructions.

To encourage $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ that are more peaked in the space of $\boldsymbol{z}$ for individual $\boldsymbol{x}$, and also to consider more peaked $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, we may augment the variational expressions as

$$\begin{aligned}
\mathcal{L}'_x &= (\lambda+1)\mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) \\
&\quad -\mathbb{E}_{q(\boldsymbol{x})}\text{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})) \quad (26) \\
\mathcal{L}'_z &= (\lambda+1)\mathbb{E}_{p_\theta(\boldsymbol{x},\boldsymbol{z})} \log p_\phi(\boldsymbol{z}|\boldsymbol{x}) \\
&\quad -\mathbb{E}_{p(\boldsymbol{z})}\text{KL}(p_\theta(\boldsymbol{x}|\boldsymbol{z})\|q(\boldsymbol{x})) \quad (27)
\end{aligned}$$

where $\lambda \geq 0$. For $\lambda = 0$ the original variational expressions are retained, and for $\lambda > 0$, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ are allowed to diverge more from $p(\boldsymbol{z})$ and $q(\boldsymbol{x})$, respectively, while placing more emphasis on the data-fit terms. Defining $\mathcal{L}'_{xz} = \mathcal{L}'_x + \mathcal{L}'_z$, we have

$$\begin{aligned}
\mathcal{L}'_{xz} = \mathcal{L}_{xz} \quad &+ \quad \lambda[\mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) \\
&+ \mathbb{E}_{p_\theta(\boldsymbol{x},\boldsymbol{z})} \log p_\phi(\boldsymbol{z}|\boldsymbol{x})] \quad (28)
\end{aligned}$$

Model learning is the same as discussed in Sec. 3.2, with the modification

$$\begin{aligned}
\ell'(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{\psi}^*) = \quad &\mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})}[f_{\psi^*}(\boldsymbol{x},\boldsymbol{z}) + \lambda \log p_\theta(\boldsymbol{x}|\boldsymbol{z})] \\
- \quad &\mathbb{E}_{p_\theta(\boldsymbol{x},\boldsymbol{z})}[f_{\psi^*}(\boldsymbol{x},\boldsymbol{z}) - \lambda \log p_\phi(\boldsymbol{z}|\boldsymbol{x})] \quad (29)
\end{aligned}$$

A disadvantage of this approach is that it requires explicit forms for $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ and $p_\phi(\boldsymbol{z}|\boldsymbol{x})$, while the setup in Sec. 3.2 only requires the ability to sample from these distributions.

We can now make a connection to additional related work, particularly [Pu et al., 2017b], which considered a similar setup to (26) and (27), for the special case of $\lambda = 1$. While [Pu et al., 2017b] had a similar idea of using a symmetrized VAE, they didn't make the theoretical justification presented in Section 3. Further, and more importantly, the way in which learning was performed in [Pu et al., 2017b] is distinct from that applied here, in that [Pu et al., 2017b] required an additional adversarial learning step, increasing implementation complexity. Consequently, [Pu et al., 2017b] did not use adversarial learning to approximate the log-LRT, and therefore it cannot make the explicit connection to ALI and WGAN that were made in Sections 4.1 and 4.3, respectively.
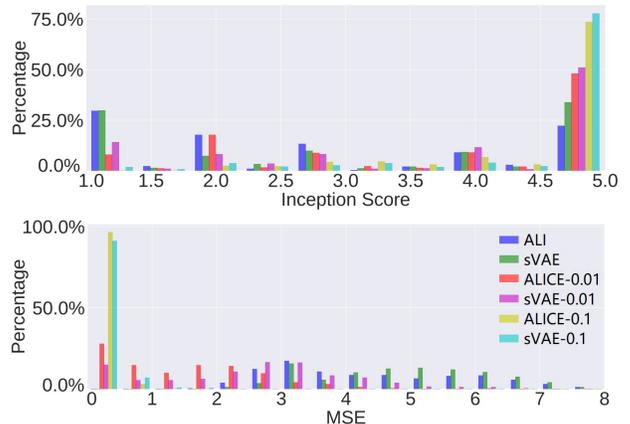

Figure 3: sVAE results on toy dataset. Top: Inception Score for ALI and sVAE with $\lambda = 0, 0.01, 0.1$. Bottom: Mean Squared Error (MSE).

6 Experiments

In addition to evaluating our model on a toy dataset, we consider MNIST, CelebA and CIFAR-10 for both reconstruction and generation tasks. As done for the model ALI with Cross Entropy regularization (ALICE) [Li et al., 2017], we also add the augmentation term ($\lambda > 0$ as discussed in Sec. 5) to sVAE as a regularizer, and denote the new model as sVAE-r. More specifically, we show the results based on the two models: i) sVAE: the model is developed in Sec. 3 to optimize $g(\boldsymbol{\psi})$ in (10) and $\ell(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{\psi}^*)$ in (14). ii) sVAE-r: the model is sVAE with regularization term to optimize $g(\boldsymbol{\psi})$ in (10) and $\ell'(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{\psi}^*)$ in (29). The quantitative evaluation is based on the mean square error (MSE) of reconstructions, log-likelihood calculated via the annealed importance sampling (AIS) [Wu et al., 2016], and inception score (IS) [Salimans et al., 2016].

All parameters are initialized with Xavier [Glorot and Bengio, 2010] and optimized using Adam [Kingma and Ba, 2015] with learning rate of 0.0001. No dataset-specific tuning or regularization, other than dropout [Srivastava et al., 2014], is performed. The architectures for the encoder, decoder and discriminator are detailed in the Appendix. All experimental results were performed on a single NVIDIA TITAN X GPU.

6.1 Toy Data

In order to show the robustness and stability of our model, we test sVAE and sVAE-r on a toy dataset designed in the same manner as the one in ALICE [Li et al., 2017]. In this dataset, the true distribution of data $\boldsymbol{x}$ is a two-dimensional Gaussian mixture model with five components. The latent code $\boldsymbol{z}$ is a standard Gaussian distribution $\mathcal{N}(0, 1)$. To perform the test, we consider using different values of λ for both sVAE-r and ALICE. For each λ, 576 experiments with different choices of architecture and hyper-parameters are

Liqun Chen[1], **Shuyang Dai**[1], **Yunchen Pu**[1], **Erjin Zhou**[4], **Chunyuan Li**[1]
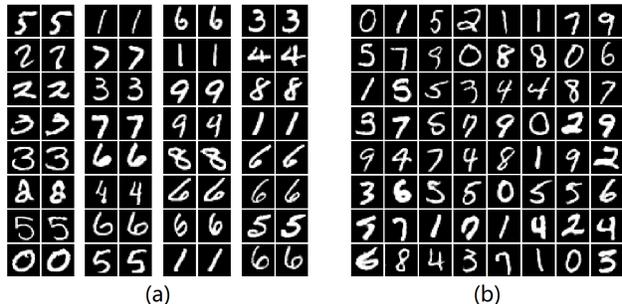
Figure 4: sVAE results on MNIST. (a) is reconstructed images by sVAE-r: in each block, column one is ground-truth and column two is reconstructed images. (b) are generated sample images by sVAE-r. Note that λ is set to 0.1 for sVAE-r.

conducted. In all experiments, we use mean square error (MSE) and inception score (IS) to evaluate the performance of the two models. Figure 3 shows the histogram results for each model. As we can see, both ALICE and sVAE-r are able to reconstruct images when $\lambda = 0.1$, while sVAE-r provides better overall inception score.

6.2 MNIST

The results of image generation and reconstruction for sVAE, as applied to the MNIST dataset, are shown in Figure 4. By adding the regularization term, sVAE overcomes the limitation of image reconstruction in ALI. The log-likelihood of sVAE shown in Table 1 is calculated using the annealed importance sampling method on the binarized MNIST dataset, as proposed in [Wu et al., 2016]. Note that in order to compare the model performance on binarized data, the output of the decoder is considered as a Bernoulli distribution instead of the Gaussian approach from the original paper. Our model achieves -79.26 nats, outperforming normalizing flow (-85.1 nats) while also being competitive to the state-of-the-art result (-79.2 nats). In addition, sVAE is able to provide compelling generated images, outperforming GAN [Goodfellow et al., 2014] and WGAN-GP [Ishaan Gulrajani, 2017] based on the inception scores.

Table 1: Quantitative Results on MNIST. † is calculated using AIS. ‡ is reported in [Hu et al., 2017].

Model	$\log p(x) \geq$	IS
NF (k=80) [Rezende et al., 2015]	-85.1	-
PixelRNN [Oord et al., 2016]	-79.2	-
AVB [Mescheder et al., 2016]	-79.5	-
ASVAE [Pu et al., 2017b]	-81.14	-
GAN [Goodfellow et al., 2014]	-114.25 †	8.34 ‡
WGAN-GP [Ishaan Gulrajani, 2017]	-79.92 †	8.45 ‡
DCGAN [Radford et al., 2016]	-79.47 †	8.93
sVAE (ours)	-80.42 †	8.81
sVAE-r (ours)	-79.26 †	9.12

Figure 5: CelebA generation results. Left block: sVAE-r generation. Right block: ALICE generation. $\lambda = 0, 0.1, 1$ and 10 from left to right in each block.

Figure 6: CelebA reconstruction results. Left column: The ground truth. Middle block: sVAE-r reconstruction. Right block: ALICE reconstruction. $\lambda = 0, 0.1, 1$ and 10 from left to right in each block.

6.3 CelebA

We evaluate sVAE on the CelebA dataset and compare the results with ALI. In experiments we note that for high-dimensional data like the CelebA, ALICE [Li et al., 2017] shows a trade-off between reconstruction and generation, while sVAE-r does not have this issue. If the regularization term is not included in ALI, the reconstructed images do not match the original images. On the other hand, when the regularization term is added, ALI is capable of reconstructing images but the generated images are flawed. In comparison, sVAE-r does well in both generation and reconstruction with different values of λ. The results for both sVAE and ALI are shown in Figure 5 and 6.
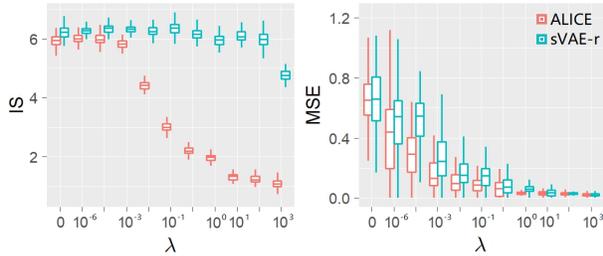
Figure 7: sVAE-r and ALICE CIFAR quantitative evaluation with different values of λ. Left: IS for generation; Right: MSE for reconstruction. The result is the average of multiple tests.

Generally speaking, adding the augmentation term as shown in (28) should encourage more peaked $q_\phi(z|x)$ and $p_\theta(x|z)$. Nevertheless, ALICE fails in the inference process and performs more like an autoencoder. This is due to the fact that the discriminator becomes too sensitive to the regularization term. On the other hand, by using the symmetric KL (14) as the cost function, we are able to alleviate this issue, which makes sVAE-r a more stable model than ALICE. This is because sVAE updates the generator using the discriminator output, before the sigmoid, a non-linear transformation on the discriminator output scale.

6.4 CIFAR-10

The trade-off of ALICE [Li et al., 2017] mentioned in Sec. 6.3 is also manifested in the results for the CIFAR-10 dataset. In Figure 7, we show quantitative results in terms of inception score and mean squared error of sVAE-r and ALICE with different values of λ. As can be seen, both models are able to reconstruct images when λ increases. However, when λ is larger than 10^{-3}, we observe a decrease in the inception score of ALICE, in which the model fails to generate images.

Table 2: Unsupervised Inception Score on CIFAR-10

| Model	IS
ALI [Dumoulin et al., 2017]	$5.34 \pm .05$
DCGAN [Radford et al., 2016]	$6.16 \pm .07$
ASVAE [Pu et al., 2017b]	$6.89 \pm .05$
WGAN-GP	$6.56 \pm .05$
WGAN-GP ResNet [Ishaan Gulrajani, 2017]	$7.86 \pm .07$
sVAE (ours)	$6.76 \pm .046$
sVAE-r (ours)	$6.96 \pm .066$

The CIFAR-10 dataset is also used to evaluate the generation ability of our model. The quantitative results, i.e., the inception scores, are listed in Table 2. Our model shows improved performance on image generation compared to ALI and DCGAN. Note that sVAE also gets comparable result as WGAN-GP [Ishaan Gulrajani, 2017] achieves. This can be interpreted using the similarity between (23) and (25) as summarized in the Sec. 4. The generated images are shown in Figure 8. More results are in the Appendix.
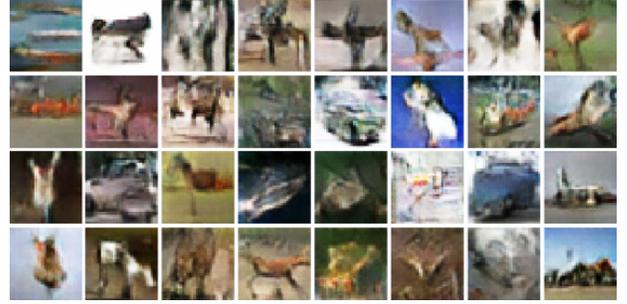
(a) sVAE CIFAR unsupervised generation.

(b) sVAE-r (with $\lambda = 1$) CIFAR unsupervised generation.
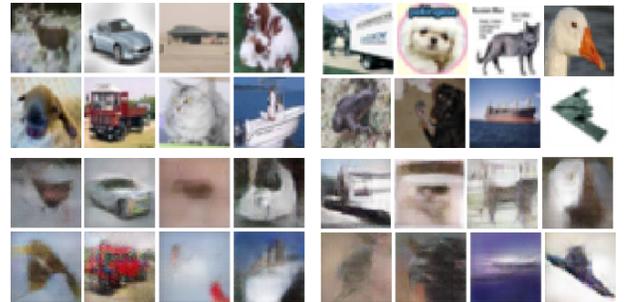
(c) sVAE-r (with $\lambda = 1$) CIFAR unsupervised reconstruction. First two rows are original images, and the last two rows are the reconstructions.

Figure 8: sVAE CIFAR results on image generation and reconstruction.

7 Conclusions

We present the symmetric variational autoencoder (sVAE), a novel framework which can match the joint distribution of data and latent code using the *symmetric* Kullback-Leibler divergence. The experiment results show the advantages of sVAE, in which it not only overcomes the missing mode problem [Hu et al., 2017], but also is very stable to train. With excellent performance in image generation and reconstruction, we will apply sVAE on semi-supervised learning tasks and conditional generation tasks in future work. Moreover, because the latent code z can be treated as data from a different domain, i.e., images [Zhu et al., 2017, Kim et al., 2017] or text [Gan et al., 2017], we can also apply sVAE to domain transfer tasks.

Liqun Chen[1], **Shuyang Dai**[1], **Yunchen Pu**[1], **Erjin Zhou**[4], **Chunyuan Li**[1]

References

M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. In *ICLR*, 2017.

D. Blei, A. Kucukelbir, and J. McAuliffe. Variational inference: A review for statisticians. In *arXiv:1601.00670v5*, 2017.

C. Carvalho, J. Chang, J. Lucas, J. Nevins, Q. Wang, and M. West. High-dimensional sparse factor modeling: Applications in gene expression genomics. *JASA*, 2008.

V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, O. Mastropietro, and A. Courville. Adversarially learned inference. In *ICLR*, 2017.

Z. Gan, L. Chen, W. Wang, Y. Pu, Y. Zhang, H. Liu, C. Li, and L. Carin. Triangle generative adversarial networks. *arXiv preprint arXiv:1709.06548*, 2017.

Z. Ghahramani and G. Hinton. The em algorithm for mixtures of factor analyzers. Technical report, University of Toronto, 1997.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

Z. Hu, Z. Yang, R. Salakhutdinov, and E. P. Xing. On unifying deep generative models. In *arXiv*, 2017.

M. A. V. D. A. C. Ishaan Gulrajani, Faruk Ahmed. Improved training of wasserstein gans. In *arXiv*, 2017.

T. Kim, M. Cha, H. Kim, J. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *arXiv*, 2017.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin. Towards understanding adversarial learning for joint distribution matching. *arXiv preprint arXiv:1709.01215*, 2017.

A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv:1511.05644v2*, 2016.

L. Mescheder, S. Nowozin, and A. Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *arXiv*, 2016.

A. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural network. In *ICML*, 2016.

Y. Pu, X. Yuan, and L. Carin. Generative deep deconvolutional learning. In *ICLR workshop*, 2015.

Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin. Variational autoencoder for deep learning of images, labels and captions. In *NIPS*, 2016a.

Y. Pu, X. Yuan, A. Stevens, C. Li, and L. Carin. A deep generative deconvolutional image model. *Artificial Intelligence and Statistics (AISTATS)*, 2016b.

Y. Pu, Z. Gan, R. Henao, C. L, S. Han, and L. Carin. VAE learning via stein variational gradient descent. In *NIPS*, 2017a.

Y. Pu, W.Wang, R. Henao, L. Chen, Z. Gan, C. Li, and L. Carin. Adversarial symmetric variational autoencoder. *NIPS*, 2017b.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

D. Rezende, S. Mohamed, and et al. Variational inference with normalizing flows. In *ICML*, 2015.

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2016.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014.

M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

H. V. Trees. Detection, estimation, and modulation theory. In *Wiley*, 2001.

P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 2010.

Y. Wu, Y. Burda, R. Salakhutdinov, and R. Grosse. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016.

R. Zhang, C. Li, C. Chen, and L. Carin. Learning structural weight uncertainty for sequential decision-making. *AISTATS*, 2018.

Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen, and L. Carin. Adversarial feature matching for text generation. *arXiv preprint arXiv:1706.03850*, 2017a.

Y. Zhang, D. Shen, G. Wang, Z. Gan, R. Henao, and L. Carin. Deconvolutional paragraph representation learning. In *Advances in Neural Information Processing Systems*, pages 4172–4182, 2017b.

J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. In *ICLR*, 2017.

J. Zhu, T. Park, P. Isola, and A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *arXiv:1703.10593*, 2017.