
On Unifying Deep Generative Models

Zhiting Hu^{1,2} Zichao Yang¹ Ruslan Salakhutdinov¹ Eric P. Xing^{1,2}
Carnegie Mellon University¹, Petuum Inc.²

Abstract

Deep generative models have achieved impressive success in recent years. Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), as powerful frameworks for deep generative model learning, have largely been considered as two distinct paradigms and received extensive independent study respectively. This paper establishes formal connections between deep generative modeling approaches through a new formulation of GANs and VAEs. We show that GANs and VAEs are essentially minimizing KL divergences with opposite directions and reversed latent/visible treatments, extending the two learning phases of classic wake-sleep algorithm, respectively. The unified view provides a powerful tool to analyze a diverse set of existing model variants, and enables to exchange ideas across research lines in a principled way. For example, we transfer the importance weighting method in VAE literatures for improved GAN learning, and enhance VAEs with an adversarial mechanism. Quantitative experiments show generality and effectiveness of the imported extensions.

1 Introduction

Recent years have seen remarkable advances in deep generative modeling. Generative Adversarial Networks (GANs) [13] and Variational Autoencoders (VAEs) [24] are two of the emerging approaches for learning neural generative models, and have achieved impressive success in a myriad of applications, such as image and text generation [39, 21, 28], disentangled representation learning [7, 26], and semi-supervised learning [40, 25]. On the other hand, the classic wake-sleep algorithm dates back to work of Hinton et al. [17] for training deep generative models such as Helmholtz machines [10].

Past research has largely viewed these approaches as distinct paradigms for generative model learning. For instance, GANs aim to achieve an equilibrium between a generator and a discriminator, while VAEs are devoted to maximizing a lower bound of the data likelihood. A rich array of theoretical analyses and model extensions have been developed independently for GANs [1, 2, 40, 34] and VAEs [4, 9, 21, 14], respectively, with few work combining the two objectives in a single model for improved inference and generation [27, 29, 42, 43]. Despite the significant progress specific to each method, it is unclear how these apparently divergent approaches connect to each other in a principled way.

In this paper, we bridge the gap between GANs and VAEs by establishing a new formulation that connects the two in a unified view, and links to the wake-sleep algorithm, showing that GANs and VAEs are in effect minimizing KL divergences in opposite directions, extending the sleep and wake phases respectively for generative model learning. More specifically, we develop a Bayesian re-formulation of GANs by introducing real/fake random variable y as visible, and treating generations x as latent to infer *a posteriori*, leading to an objective that resembles variational inference as in VAEs. As a counterpart, VAEs reverse the visible/latent treatments and KL direction, with perfect inference on real/fake variable y which results in a degenerated adversary.

The proposed interpretation enables deeper understanding of the landscape of generative modeling by providing a useful tool to analyze a broad class of most recent GAN and VAE based algorithms. For instance, we can easily extend the formulation to subsume InfoGAN [7] that additionally infers

	ADA	GANs	VAEs
\mathbf{x}	features	data/generations	data/generations
y	domain indicator	real/fake indicator	(degenerated) real/fake indicator
\mathbf{z}	data examples	code vector	code vector
$p_\theta(\mathbf{x} y)$	feature distr.	generation distr., Eq.2	$p_\theta(\mathbf{x} \mathbf{z}, y)$, generation distr., Eq.9
$q_\phi(y \mathbf{x})$	discriminator	discriminator	$q^*(y \mathbf{x})$, degenerated discriminator
$p_\eta(\mathbf{z} \mathbf{x}, y)$	—	infer net (InfoGAN)	infer net
$p_{\theta_0}(\mathbf{x}) = \mathbb{E}_{p(y)}[p_{\theta_0}(\mathbf{x} y)]$	—	prior of \mathbf{x}	prior of \mathbf{x}

Table 1: Correspondence between different approaches in the proposed formulation.

hidden representations of examples, VAE/GAN [27, 5] that combines the two models for improved generation and reduced mode missing, and adversarial domain adaptation (ADA) [12, 37, 38, 8] that is traditionally framed in the discriminative setting.

The close parallels between GANs and VAEs further ease mutual exchange of ideas that were proposed separately for improving either class of models. We give two examples in the paper. 1) Drawn inspiration from importance weighted VAE (IWAE) [4] we straightforwardly derive importance weighted GAN (IWGAN) that maximizes a tighter lower bound on the marginal likelihood compared to the vanilla GAN. 2) Motivated by the GAN adversarial game we activate the originally degenerated discriminator in VAEs, resulting in a full-fledged model that adaptively leverages both real and fake examples. Our quantitative empirical results show that the techniques imported from the other class are generally applicable to the base model and its variants, yielding consistently better performance.

2 Related Work

There is a surge of research interest in deep generative models. Remarkable progress has been made in each class of algorithms. Arjovsky and Bottou [1] perform theoretical analysis of the instability of GAN training. Arora et al. [2] discuss the generalization of GANs and show that the assumption of ideal discriminator in analysis can be problematic. Mohamed and Lakshminarayanan [32] connect GANs to density ratio estimation techniques. The f -GAN [34] generalizes GANs to optimize a diverse set of f -divergences. Our work differs in that we propose Bayesian reinterpretation of the standard GANs based on which we reveal rich connections between a broad class of generative models. Sønderby et al. [42] adapt GANs to minimize a reversed KL divergence. Our work is distinct as we are not inventing new model instances but aim to study the most popular deep generative models and establish new formulations that are generally applicable to base models and diverse sets of variants. In the line of VAEs [24], Johnson et al. [23] leverage VAEs for structured graphical model inference. Burda et al. [4] develop importance weighted VAEs to optimize a tighter lower bound on data likelihood. Norouzi et al. [33] relate maximum likelihood framework as in VAEs to reward based objectives. The two lines of research have largely been independent and specific to either class of algorithms. We establish deep connections between the two classes, which helps to transfer ideas across the vivid research areas.

A handful of previous work combines GANs and VAEs through simple structural similarity between them. Larsen et al. [27] integrate GAN objective in VAEs by tying the generators in both model. Several work [43, 30, 22] leverages adversarial mechanism to learn implicit inference distributions in VAEs, while Zhai et al. [44] exploit GAN principles to learn energy based models. Our proposed interpretation of GANs and VAEs reveals new insights into the close relations between them, and can inspire more integrations of two approaches in a principled way. Chen et al. [7] and Hu et al. [21] develop GAN and VAE extensions, respectively, and relate them to the wake-sleep algorithm. This paper instead bases on the general models and provides results that easily extend to these variants.

3 Bridging the Gap

In GANs, the generative model is trained by passing generated samples to a discriminator and minimizing the resulting error evaluated by the discriminator. Intuitively, the reliance on fake samples for learning resembles the sleep phase in the wake-sleep algorithm. In contrast, VAEs train the

generative model by reconstructing observed real examples, sharing similarity to the wake phase. This section formally explores these connections.

For ease of presentation and establishing notations in the paper, we start with a new interpretation of adversarial domain adaptation (ADA) within our proposed formulation. We then show GANs are a special case of ADA with a degenerated source domain, and reveal close relations to VAEs and wake-sleep algorithm through KL divergence interpretation of the objectives. Table 1 lists the correspondence of each components in these approaches.

3.1 Adversarial Domain Adaptation (ADA)

ADA aims to transfer prediction knowledge learned from a source domain with labeled data to a target domain without labels, by learning domain-invariant features [12, 37, 38, 8]. That is, it learns a feature extractor whose output cannot be distinguished by a discriminator between the source and target domains.

We frame our new interpretation of ADA, and review conventional formulations in the supplementary materials. To make clear notational correspondence to other models in the sequel, let \mathbf{z} be a data example either in the source or target domain, and $y \in \{0, 1\}$ be the domain indicator with $y = 0$ indicating the target domain and $y = 1$ the source domain. The data distributions are then denoted as $p(\mathbf{z}|y)$. Let $p(y)$ be the prior distribution (e.g., uniform). The feature extractor maps \mathbf{z} to representations $\mathbf{x} = G_\theta(\mathbf{z})$ with parameters θ . The data distributions over \mathbf{z} and deterministic transformation G_θ together form an *implicit* distribution over \mathbf{x} , denoted as $p_\theta(\mathbf{x}|y)$, which is intractable to evaluate likelihood but easy to sample from:

To enforce domain invariance of feature \mathbf{x} , a discriminator is trained to adversarially distinguish between the two domains, which defines a conditional distribution $q_\phi(y|\mathbf{x})$ with parameters ϕ , and the feature extractor is optimized to fool the discriminator. Let $q_\phi^r(y|\mathbf{x}) = q_\phi(1 - y|\mathbf{x})$ be the reversed distribution over domains. The objectives of ADA are therefore given as:

$$\begin{aligned} \max_\phi \mathcal{L}_\phi &= \mathbb{E}_{p_\theta(\mathbf{x}|y)p(y)} [\log q_\phi(y|\mathbf{x})] \\ \max_\theta \mathcal{L}_\theta &= \mathbb{E}_{p_\theta(\mathbf{x}|y)p(y)} [\log q_\phi^r(y|\mathbf{x})], \end{aligned} \quad (1)$$

where we omit the additional loss of θ to fit to the data label pairs of source domain (see supplements for more details). In conventional view, the first equation minimizes the discriminator binary cross entropy, while the second trains the feature extractor to maximize the cross entropy. Alternatively, we can interpret the objectives as optimizing the reconstruction of the domain variable y conditioned on feature \mathbf{x} . We explore this perspective more in the next section. Note that the only (but critical) difference between the objective of θ from ϕ is the replacement of $q(y|\mathbf{x})$ with $q^r(y|\mathbf{x})$. This is where the adversarial mechanism comes about.

3.2 Generative Adversarial Networks (GANs)

GANs [13] can be seen as a special case of ADA. Taking image generation for example, intuitively, we want to transfer the properties of the source domain (real images) to the target domain (generated images), making them indistinguishable to the discriminator.

Formally, \mathbf{x} now denotes a real example or a generated sample, \mathbf{z} is the respective latent code. For the generated sample domain ($y = 0$), the implicit distribution $p_\theta(\mathbf{x}|y = 0)$ is defined by the prior of \mathbf{z} and the generator $G_\theta(\mathbf{z})$, which is also denoted as $p_g(\mathbf{x})$ in the literature [13]. For the real example domain ($y = 1$), the code space and generator are degenerated, and we are directly presented with a fixed $p(\mathbf{x}|y = 1)$, which is just the real data distribution $p_{data}(\mathbf{x})$. Note that $p_{data}(\mathbf{x})$ is also an implicit distribution allowing efficient empirical sampling. In summary, the distribution over \mathbf{x} is constructed as

$$p_\theta(\mathbf{x}|y) = \begin{cases} p_g(\mathbf{x}) & y = 0 \\ p_{data}(\mathbf{x}) & y = 1. \end{cases} \quad (2)$$

Here, free parameters θ are only associated with $p_g(\mathbf{x})$ of the generated sample domain, while $p_{data}(\mathbf{x})$ is constant. As in ADA, discriminator D_ϕ is simultaneously trained to infer the probability that \mathbf{x} comes from the real data domain. That is, $q_\phi(y = 1|\mathbf{x}) = D_\phi(\mathbf{x})$.

With the established correspondence between GANs and ADA, we can see that the objectives of GANs are exactly expressed as in Eq.(1). To make this clearer, we recover the classical form by unfolding over y and plugging in conventional notations. For instance, the objective of the generative parameters θ is translated into

$$\begin{aligned}\max_{\theta} \mathcal{L}_{\theta} &= \mathbb{E}_{p_{\theta}(\mathbf{x}|y=0)p(y=0)} [\log q_{\phi}^r(y=0|\mathbf{x})] + \mathbb{E}_{p_{\theta}(\mathbf{x}|y=1)p(y=1)} [\log q_{\phi}^r(y=1|\mathbf{x})] \\ &= \frac{1}{2} \mathbb{E}_{\mathbf{x}=G_{\theta}(\mathbf{z}), \mathbf{z} \sim p(\mathbf{z}|y=0)} [\log D_{\phi}(\mathbf{x})] + \frac{1}{2} \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [\log(1 - D_{\phi}(\mathbf{x}))] \\ &= \frac{1}{2} \mathbb{E}_{\mathbf{x}=G_{\theta}(\mathbf{z}), \mathbf{z} \sim p(\mathbf{z}|y=0)} [\log D_{\phi}(\mathbf{x})] + const,\end{aligned}\quad (3)$$

where the prior $p(y)$ is uniform as is widely set, leading to the constant scaling factor $1/2$. Note that here the generator is trained using the unsaturated objective [13] which is commonly used in practice.

We now take a closer look at the form of Eq.(1) which is essentially reconstructing the real/fake indicator y (or its reverse) conditioned on \mathbf{x} . Further, for each optimization step of $p_{\theta}(\mathbf{x}|y)$ at point (θ_0, ϕ_0) in the parameter space, we have

Lemma 1. *Let $p(y)$ be the uniform distribution. Let $p_{\theta_0}(\mathbf{x}) = \mathbb{E}_{p(y)}[p_{\theta_0}(\mathbf{x}|y)]$, and $q^r(\mathbf{x}|y) \propto q_{\phi_0}^r(y|\mathbf{x})p_{\theta_0}(\mathbf{x})$. Therefore, the updates of θ at θ_0 have*

$$\begin{aligned}\mathbb{E}_{p(y)} [\nabla_{\theta} \mathbb{E}_{p_{\theta}(\mathbf{x}|y)} [\log q_{\phi_0}^r(y|\mathbf{x})] |_{\theta=\theta_0}] &= \\ - \mathbb{E}_{p(y)} [\nabla_{\theta} KL(p_{\theta}(\mathbf{x}|y) \| q^r(\mathbf{x}|y)) - JSD(p_{\theta}(\mathbf{x}|y=0) \| p_{\theta}(\mathbf{x}|y=1)) |_{\theta=\theta_0}],\end{aligned}\quad (4)$$

where $KL(\cdot \| \cdot)$ and $JSD(\cdot \| \cdot)$ are the KL and Jensen-Shannon Divergences, respectively.

We provide the proof in the supplement materials. Eq.(4) offers several insights into the generator learning in GANs.

- If we treat y as visible and \mathbf{x} as latent (as in ADA), it is straightforward to see the connection to variational inference where $p_{\theta}(\mathbf{x}|y)$ plays the role of inference network, p_{θ_0} the prior, and $q^r(\mathbf{x}|y)$ the posterior. Optimizing the generator G_{θ} in GANs is equivalent to minimizing the KL divergence, minus a JSD between the generator distribution $p_g(\mathbf{x})$ and data distribution $p_{data}(\mathbf{x})$.
- By definition, $p_{\theta_0}(\mathbf{x}) = (p_g(\mathbf{x}) + p_{data}(\mathbf{x}))/2$ is an average over $p_g(\mathbf{x})$ and $p_{data}(\mathbf{x})$, and the ‘‘posterior’’ $q^r(\mathbf{x}|y)$ smooths $p_{\theta_0}(\mathbf{x})$ by combining the discriminator $q_{\phi_0}^r(y|\mathbf{x})$. Thus, minimizing the KL divergence between $p_{\theta}(\mathbf{x}|y)$ and $q^r(\mathbf{x}|y)$ in effect drives $p_g(\mathbf{x})$ (i.e., $p(\mathbf{x}|y=0)$) to a mixed distribution of $p_g(\mathbf{x})$ and $p_{data}(\mathbf{x})$. Since $p_{data}(\mathbf{x})$ is fixed, $p_g(\mathbf{x})$ gets close to $p_{data}(\mathbf{x})$.
- The negative JSD term is due to the extra prior regularization in the KL divergence. As JSD is symmetric, the missing mode phenomena widely observed in GAN generator [31, 5] is explained by the asymmetry of the KL divergence which tends to concentrate $p_{\theta}(\mathbf{x}|y)$ to large modes of $q^r(\mathbf{x}|y)$ and ignore smaller ones.

Arjovsky and Bottou [1] derive a similar result of minimizing the KL divergence between $p_g(\mathbf{x})$ and $p_{data}(\mathbf{x})$. Our result does not rely on assumptions of (near) optimal discriminator, thus is more close to the practice [2]. Indeed, when the discriminator distribution $q_{\phi_0}(y|\mathbf{x})$ gives uniform guesses, the gradients of the KL and JSD terms in Eq.(4) cancel out, disabling the learning of generator. Moreover, the Bayesian interpretation of our result enables us to discover connections to VAEs, as we discuss in the next section.

InfoGAN Chen et al. [7] developed InfoGAN for disentangled representation learning which additionally recovers (part of) the latent code \mathbf{z} given example \mathbf{x} . This can be straightforwardly formulated in our framework by introducing an extra conditional $q_{\eta}(\mathbf{z}|\mathbf{x}, y)$ parameterized by η . As discussed above, GANs assume a degenerated code space for real examples, thus $q_{\eta}(\mathbf{z}|\mathbf{x}, y=1)$ is fixed without free parameters to learn, and η is only associated to $y=0$. The InfoGAN is then recovered by combining $q_{\eta}(\mathbf{z}|\mathbf{x}, y)$ with $q(y|\mathbf{x})$ in Eq.(1) to perform full reconstruction of both \mathbf{z} and y :

$$\begin{aligned}\max_{\phi} \mathcal{L}_{\phi} &= \mathbb{E}_{p_{\theta}(\mathbf{x}|y)p(y)} [\log q_{\eta}(\mathbf{z}|\mathbf{x}, y)q_{\phi}(y|\mathbf{x})] \\ \max_{\theta, \eta} \mathcal{L}_{\theta, \eta} &= \mathbb{E}_{p_{\theta}(\mathbf{x}|y)p(y)} [\log q_{\eta}(\mathbf{z}|\mathbf{x}, y)q_{\phi}^r(y|\mathbf{x})],\end{aligned}\quad (5)$$

where the ground-truth \mathbf{z} to reconstruct is sampled from the prior $p(\mathbf{z}|y)$ and encapsulated in the implicit distribution $p_{\theta}(\mathbf{x}|y)$. Let $q^r(\mathbf{x}|\mathbf{z}, y) \propto q_{\eta_0}(\mathbf{z}|\mathbf{x}, y)q_{\phi_0}^r(y|\mathbf{x})p_{\theta_0}(\mathbf{x})$, the result in the form of

Eq.(4) still holds by replacing $q_{\phi_0}^r(y|\mathbf{x})$ with $q_{\eta_0}(z|\mathbf{x}, y)q_{\phi_0}^r(y|\mathbf{x})$, and $q^r(\mathbf{x}|y)$ with $q^r(\mathbf{x}|\mathbf{z}, y)$:

$$\begin{aligned} \mathbb{E}_{p(y)} [\nabla_{\theta} \mathbb{E}_{p_{\theta}(\mathbf{x}|y)} [\log q_{\eta_0}(z|\mathbf{x}, y)q_{\phi_0}^r(y|\mathbf{x})] |_{\theta=\theta_0}] = \\ - \mathbb{E}_{p(y)} [\nabla_{\theta} \text{KL}(p_{\theta}(\mathbf{x}|y) \| q^r(\mathbf{x}|\mathbf{z}, y)) - \text{JSD}(p_{\theta}(\mathbf{x}|y=0) \| p_{\theta}(\mathbf{x}|y=1)) |_{\theta=\theta_0}], \end{aligned} \quad (6)$$

As a side result, the idea of interpreting \mathbf{x} as latent variables immediately discovers relations between InfoGAN with Adversarial Autoencoder (AAE) [29] and Predictability Minimization [41]. That is, InfoGAN is precisely an AAE which treats \mathbf{x} as latents and \mathbf{z} as visibles.

3.3 Variational Autoencoders (VAEs)

We next explore the second class of deep generative model learning algorithms. The resemblance of GAN generator learning to variational inference as shown in Eq.(4) suggests strong relations between VAEs [24] and GANs. We build correspondence between the two approaches, and show that VAEs are basically minimizing a KL divergence with an opposite direction, with a degenerated adversarial discriminator.

The conventional definition of VAEs is written as:

$$\max_{\theta, \eta} \mathcal{L}_{\theta, \eta}^{\text{vae}} = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\mathbb{E}_{\tilde{q}_{\eta}(z|\mathbf{x})} [\log \tilde{p}_{\theta}(\mathbf{x}|\mathbf{z})] - \text{KL}(\tilde{q}_{\eta}(z|\mathbf{x}) \| \tilde{p}(z))], \quad (7)$$

where $\tilde{p}_{\theta}(\mathbf{x}|\mathbf{z})$ is the generator, $\tilde{q}_{\eta}(z|\mathbf{x})$ the inference network, and $\tilde{p}(z)$ the prior over z . The parameters to learn are intentionally denoted with the notations of corresponding modules in GANs. At first glance, VAEs appear to differ from GANs greatly as they use only real examples and lack adversarial mechanism. However, our interpretation shows VAEs indeed include a degenerated adversarial discriminator that blocks out generated samples from contributing to training.

Specifically, we again introduce the real/fake variable y , and assume a perfect discriminator $q_*(y|\mathbf{x})$ which always predicts $y = 1$ with probability 1 given real examples, and $y = 0$ given generated samples. Again, for notational simplicity, let $q_*^r(y|\mathbf{x}) = q_*(1 - y|\mathbf{x})$ be the reversed distribution.

Lemma 2. Let $p_{\theta}(z, y|\mathbf{x}) \propto p_{\theta}(\mathbf{x}|\mathbf{z}, y)p(z|y)p(y)$. Therefore,

$$\begin{aligned} \mathcal{L}_{\theta, \eta}^{\text{vae}} &= 2 \cdot \mathbb{E}_{p_{\theta_0}(\mathbf{x})} [\mathbb{E}_{q_{\eta}(z|\mathbf{x}, y)q_*^r(y|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z}, y)] - \text{KL}(q_{\eta}(z|\mathbf{x}, y)q_*^r(y|\mathbf{x}) \| p(z|y)p(y))] \\ &= 2 \cdot \mathbb{E}_{p_{\theta_0}(\mathbf{x})} [-\text{KL}(q_{\eta}(z|\mathbf{x}, y)q_*^r(y|\mathbf{x}) \| p_{\theta}(z, y|\mathbf{x}))]. \end{aligned} \quad (8)$$

Here most of the components have exact correspondences (and the same definitions) in GANs and InfoGAN (Table 1), except that the generation distribution $p_{\theta}(\mathbf{x}|\mathbf{z}, y)$ differs slightly from its counterpart $p_{\theta}(\mathbf{x}|y)$ in Eq.(2) to additionally account for the uncertainty of generating \mathbf{x} given z :

$$p_{\theta}(\mathbf{x}|\mathbf{z}, y) = \begin{cases} p_{\theta}(\mathbf{x}|\mathbf{z}) & y = 0 \\ p_{\text{data}}(\mathbf{x}) & y = 1. \end{cases} \quad (9)$$

The resulting KL divergence closely relates to that in GANs (Eq.4) and InfoGAN (Eq.6), with the generative module $p_{\theta}(\mathbf{x}|\mathbf{z}, y)$ and inference networks $q_{\eta}(z|\mathbf{x}, y)q^r(y|\mathbf{x})$ placed in the opposite directions, and with inverted hidden/visible treatments of (z, y) and \mathbf{x} . In section 6, we give a general discussion that the difference between GANs and VAEs in hidden/visible treatments is relatively minor.

The proof is provided in the supplementary materials. Intuitively, recall that for the real example domain with $y = 1$, both $q_{\eta}(z|\mathbf{x}, y = 1)$ and $p_{\theta}(\mathbf{x}|\mathbf{z}, y = 1)$ are constant distributions. Therefore, with fake sample \mathbf{x} generated from $p_{\theta_0}(\mathbf{x})$, the reversed perfect discriminator $q_*^r(y|\mathbf{x})$ always gives prediction $y = 1$, making the reconstruction loss on fake samples degenerated to a constant. Hence only real examples, where q_*^r predicts $y = 0$ with probability 1, are effective for learning, which is identical to Eq.(7). We extend VAEs to also leverage fake samples in section 4.

VAE/GAN Joint Models Previous work has explored combination of VAEs and GANs for improved generation. This can be naturally motivated by the asymmetric behaviors of the KL divergences that the two algorithms aim to optimize respectively. Specifically, the VAE/GAN model [27] that improves the sharpness of VAE generated images can be alternatively motivated by remedying the mode covering behavior of the KL in VAEs. That is, the KL tends to drive the generative model

to cover all modes of the data distribution as well as regions with small values of p_{data} , resulting in implausible samples. Incorporation of GAN objectives alleviates the issue as the inverted KL enforces the generator to focus on meaningful data modes. From the other perspective, augmenting GANs with VAE objectives helps addressing the mode missing problem, which justifies the intuition of [5].

3.4 Wake Sleep Algorithm (WS)

We next discuss the connections of GANs and VAEs to the classic wake-sleep algorithm [17] which was proposed for learning deep generative models such as Helmholtz machines [10]. WS consists of wake phase and sleep phase, which optimize the generative network and inference network, respectively. We follow the above notations, and introduce new notations \mathbf{h} to denote general latents and λ for general parameters. The wake-sleep algorithm is thus written as:

$$\begin{aligned} \text{Wake : } & \max_{\theta} \mathbb{E}_{q_{\lambda}(\mathbf{h}|\mathbf{x})p_{data}(\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{h})] \\ \text{Sleep : } & \max_{\lambda} \mathbb{E}_{p_{\theta}(\mathbf{x}|\mathbf{h})p(\mathbf{h})} [\log q_{\lambda}(\mathbf{h}|\mathbf{x})] \end{aligned} \quad (10)$$

The relations between VAEs and WS are clear in previous discussions [3, 24]. Indeed, WS was originally proposed to minimize the variational lower bound as in VAEs (Eq.7) with sleep phase approximation [17]. Alternatively, VAEs can be seen as extending the wake phase. Specifically, if we instantiate \mathbf{h} with \mathbf{z} and λ with η , the wake phase objective recovers VAEs (Eq.7) in terms of generator optimization (i.e., optimizing θ). Therefore, we can see VAEs as generalizing the wake phase by also optimizing the inference network q_{η} , with additional prior regularization on latents \mathbf{z} .

On the other hand, our interpretation of GANs reveals close resemblance to the sleep phase. To make this clearer, we instantiate \mathbf{h} with y and λ with ϕ , resulting in a sleep phase objective identical to that of optimizing the discriminator q_{ϕ} in Eq.(1), which is to reconstruct y given sample \mathbf{x} . We thus can view GANs as generalizing the sleep phase by also optimizing the generative network p_{θ} to reconstruct reversed y . InfoGAN (Eq.5) further extends the correspondence to reconstruction of latents \mathbf{z} .

4 Applications

We have established close correspondence between GANs and VAEs through the proposed formulations, which not only provides deeper understanding of the existing approaches, but also facilitates to draw inspirations cross the two classes of algorithms to develop enhanced variants. In this section, we give example extensions to GANs and VAEs, respectively, by directly importing ideas from the other approaches.

4.1 Importance Weighted GANs (IWGAN)

Burda et al. [4] proposed importance weighted autoencoders (IWAE) that maximizes a tighter lower bound on the marginal likelihood. In our framework it is straightforward to develop importance weighted GANs by copying the derivations of IWAE side by side with little adaptations. Here we outline the development and give the details in the supplementary materials.

The variational inference interpretation of the objective in Eq.(4) suggests GANs can be approximately viewed as maximizing a lower bound of the marginal likelihood on y (putting aside the negative JSD term):

$$\begin{aligned} \log q(y) &= \log \int p_{\theta}(\mathbf{x}|y) \frac{q_{\phi_0}^r(y|\mathbf{x})p_{\theta_0}(\mathbf{x})}{p_{\theta}(\mathbf{x}|y)} d\mathbf{x} \\ &\geq \int p_{\theta}(\mathbf{x}|y) \log \frac{q_{\phi_0}^r(y|\mathbf{x})p_{\theta_0}(\mathbf{x})}{p_{\theta}(\mathbf{x}|y)} d\mathbf{x} = -\text{KL}(p_{\theta}(\mathbf{x}|y)||q^r(\mathbf{x}|y)) + \text{const}. \end{aligned} \quad (11)$$

Following [4], we derive a tighter lower bound through a k -sample importance weighting estimate of the log-likelihood:

$$\log q(y) \geq \mathbb{E} \left[\log \frac{1}{k} \sum_{i=1}^k \frac{q_{\phi_0}^r(y|\mathbf{x}_i)p_{\theta_0}(\mathbf{x}_i)}{p_{\theta}(\mathbf{x}_i|y)} \right] = \mathbb{E} \left[\log \frac{1}{k} \sum_{i=1}^k w_i \right] := \mathcal{L}_k(y) \quad (12)$$

where $w_i = \frac{q_{\phi_0}^r(y|\mathbf{x}_i)p_{\theta_0}(\mathbf{x}_i)}{p_{\theta}(\mathbf{x}_i|y)}$ is the unnormalized importance weight. The lower bound of Eq.(11) is recovered with $k = 1$. Taking derivative of $\mathcal{L}_k(y)$ and applying the reparameterization trick on samples \mathbf{x}_i , we have:

$$\nabla_{\theta} \mathcal{L}_k(y) = \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_k \sim p(\mathbf{z}|y)} \left[\sum_{i=1}^k \widetilde{w}_i \nabla_{\theta} \log w(y, \mathbf{x}(\mathbf{z}_i, \boldsymbol{\theta}), \boldsymbol{\theta}) \right]. \quad (13)$$

Here $\widetilde{w}_i = w_i / \sum_{i=1}^k w_i$, in which w_i can be approximated by assuming optimal discriminator distribution:

$$w_i \approx \frac{q_{\phi_0}^r(y|\mathbf{x}_i)}{q_{\phi_0}(y|\mathbf{x}_i)}. \quad (14)$$

The derivative of w_i at $\boldsymbol{\theta}_0$ is computed as:

$$\nabla_{\theta} \log w(y, \mathbf{x}(\mathbf{z}_i, \boldsymbol{\theta}), \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} = \nabla_{\theta} \left(\log q_{\phi_0}^r(y|\mathbf{x}(\mathbf{z}_i, \boldsymbol{\theta})) + \log \frac{p_{\theta_0}(\mathbf{x}_i)}{p_{\theta}(\mathbf{x}_i|y)} \right)|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}. \quad (15)$$

In analog to the standard GANs which omit priors by subtracting the JSD term (Eq.4), we also omit the second term in the derivative relevant to the prior $p_{\theta_0}(\mathbf{x})$. The resulting update rule for the generator is thus of the following form:

$$\nabla_{\theta} \mathcal{L}_k(y) = \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_k \sim p(\mathbf{z}|y)} \left[\sum_{i=1}^k \widetilde{w}_i \nabla_{\theta} \log q_{\phi_0}^r(y|\mathbf{x}(\mathbf{z}_i, \boldsymbol{\theta})) \right]. \quad (16)$$

As in GANs, only $y = 0$ (i.e., generated samples) is effective for learning the parameters $\boldsymbol{\theta}$. Intuitively, the algorithm assigns higher weights to those samples that are more realistic and fool the discriminator better, which is consistent to IWAE that emphasizes more on code states providing better reconstructions. Hjelm et al. [18], Che et al. [6] developed a similar sample weighting scheme when maximizing generator likelihood. In practice, the k samples in Eq.(16) correspond to a minibatch of samples in standard GAN update. Thus the only computational cost added by the importance weighting method is evaluating the weight for each sample, which is generally negligible. The discriminator is trained in the same way as in the standard GANs.

4.2 Adversary Activated VAEs (AAVAE)

In our formulation, VAEs include a degenerated adversarial discriminator which blocks out generated samples from contributing to model learning. We enable adaptive incorporation of fake samples by activating the adversarial mechanism. Again, derivations are straightforward by making literal analog to GANs.

We replace the perfect discriminator $q_*(y|\mathbf{x})$ in vanilla VAEs with the discriminator network $q_{\phi}(y|\mathbf{x})$ parameterized with ϕ as in GANs, resulting in an adapted objective of Eq.(8):

$$\max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \mathcal{L}_{\boldsymbol{\theta}, \boldsymbol{\eta}}^{\text{aavae}} = \mathbb{E}_{p_{\theta_0}(\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\eta}}(\mathbf{z}|\mathbf{x}, y) q_{\phi}^r(y|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z}, y)] - \text{KL}(q_{\boldsymbol{\eta}}(\mathbf{z}|\mathbf{x}, y) q_{\phi}^r(y|\mathbf{x}) \| p(\mathbf{z}|y) p(y)) \right]. \quad (17)$$

The form of Eq.(17) is precisely symmetric to the objective of InfoGAN in Eq.(5) with the additional KL prior regularization. Before analyzing the effect of adding the learnable discriminator, we first look at how the discriminator is learned. In analog to GANs as in Eqs.(1) and (5), the objective of optimizing ϕ is obtained by simply replacing the inverted distribution $q_{\phi}^r(y|\mathbf{x})$ with $q_{\phi}(y|\mathbf{x})$:

$$\max_{\phi} \mathcal{L}_{\phi}^{\text{aavae}} = \mathbb{E}_{p_{\theta_0}(\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\eta}}(\mathbf{z}|\mathbf{x}, y) q_{\phi}(y|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z}, y)] - \text{KL}(q_{\boldsymbol{\eta}}(\mathbf{z}|\mathbf{x}, y) q_{\phi}(y|\mathbf{x}) \| p(\mathbf{z}|y) p(y)) \right]. \quad (18)$$

Intuitively, the discriminator is trained to distinguish between real and fake instances by predicting appropriate y that selects the components of $q_{\boldsymbol{\eta}}(\mathbf{z}|\mathbf{x}, y)$ and $p_{\theta}(\mathbf{x}|\mathbf{z}, y)$ to best reconstruct \mathbf{x} . The difficulty of Eq.(18) is that $p_{\theta}(\mathbf{x}|\mathbf{z}, y = 1) = p_{\text{data}}(\mathbf{x})$ is an implicit distribution which is intractable for likelihood evaluation. We thus use the alternative objective as in GANs to train a binary classifier:

$$\max_{\phi} \mathcal{L}_{\phi}^{\text{aavae}} = \mathbb{E}_{p_{\theta}(\mathbf{x}|\mathbf{z}, y) p(\mathbf{z}|y) p(y)} [\log q_{\phi}(y|\mathbf{x})]. \quad (19)$$

The activated discriminator enables an effective data selection mechanism. First, AAVAE uses not only real examples, but also generated samples for training. Each sample is weighted by the inverted

	MNIST	SVHN		MNIST	SVHN		1%	10%
GAN	8.34 \pm .03	5.18 \pm .03	CGAN	0.985 \pm .002	0.797 \pm .005	SVAE	0.9412	0.9768
IWGAN	8.45\pm.04	5.34\pm.03	IWCGAN	0.987\pm.002	0.798\pm.006	AASVAE	0.9425	0.9797

Table 2: **Left:** Inception scores of vanilla GANs and the importance weighted extension. **Middle:** Classification accuracy of the generations by class-conditional GANs and the IW extension. **Right:** Classification accuracy of semi-supervised VAEs and the adversary activated extension on the MNIST test set, with varying size of real labeled training examples.

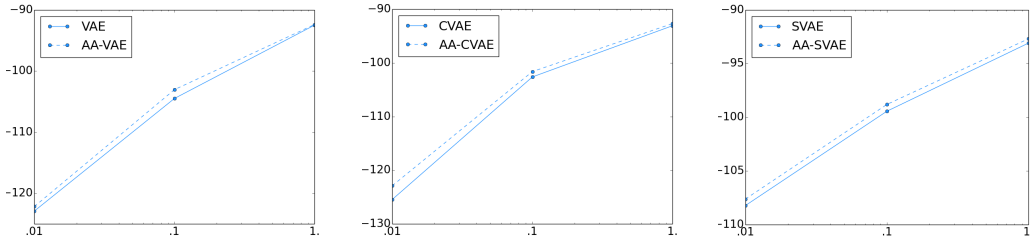


Figure 1: Lower bound values on the MNIST test set. X-axis represents the ratio of training data used for learning (0.01, 0.1, and 1.). Y-axis represents the value of lower bound. Solid lines represent the base models; dashed lines represent the adversary activated models. **Left:** VAE vs. AA-VAE. **Middle:** CVAE vs. AA-CVAE. **Right:** SVAE vs. AA-SVAE, where remaining training data are used as unsupervised data.

discriminator $q_\phi^r(y|\mathbf{x})$, so that only those samples that resemble real data and successfully fool the discriminator will be incorporated for training. This is consistent with the importance weighting strategy in IWGAN. Second, real examples are also weighted by $q_\phi^r(y|\mathbf{x})$. An example receiving large weight indicates it is easily recognized by the discriminator, which further indicates the example is hard to be simulated from the generator. That is, AAVAE emphasizes more on harder examples.

5 Experiments

We perform extensive quantitative experiments to evaluate the importance weighting method for GANs and the adversary activating method for VAEs. To show the generality of the imported ideas, we apply the extensions to vanilla models as well as several popular variants, and obtain greatly improved results.

5.1 Importance Weighted GANs

We extend both vanilla GANs and class-conditional GANs (CGAN) with the importance weighting method. The base GAN model is implemented with the DCGAN architecture and hyperparameter setting [39]. We do not tune the hyperparameters for the importance weighted extensions. We use MNIST and SVHN for evaluation. For vanilla GANs and its IW extension, we measure inception scores [40] on the generated samples. We train deep residual networks [15] provided in the tensorflow library as evaluation networks, which achieve inception scores of 9.09 and 6.55 on the test sets of MNIST and SVHN, respectively. For conditional GANs we evaluate the accuracy of conditional generation [21]. That is, we generate samples given class labels, and then use the pre-trained classifier to predict class labels of the generated samples. The accuracy is calculated as the percentage of the predictions that match the conditional labels. The evaluation networks achieve accuracy of 0.990 and 0.902 on the test sets of MNIST and SVHN, respectively.

Table 2, left panel, shows the inception scores of GANs and IW-GAN, and the middle panel gives the classification accuracy of the conditional GANs and its importance weighted extension IW-CGAN. We report the averaged results \pm one standard deviation over 5 runs. We see that the importance weighting strategy gives consistent improvements over the base models.

5.2 Adversary Activated VAEs

We apply the adversary activating method on vanilla VAEs, class-conditional VAEs (CVAE), and semi-supervised VAEs (SVAE) [25]. We evaluate on the MNIST data. The generative networks have the same architecture as the generators in GANs in the above experiments, with sigmoid activation functions on the last layer to compute the means of Bernoulli distributions over pixels. The inference networks, discriminators, and the classifier in SVAE share the same architecture as the discriminators in the GAN experiments.

We evaluate the lower bound value on the test set, with varying number of real training examples. For each minibatch of real examples we generate equal number of fake samples for training. In the experiments we found it is generally helpful to smooth the discriminator distributions by setting the temperature of the output sigmoid function larger than 1. This basically encourages the use of fake data for learning. We select the best temperature from $\{1, 1.5, 3, 5\}$ through cross-validation. We do not tune other hyperparameters for the adversary activated extensions. Figure 1 shows the results of activating the adversary mechanism on the VAE models. We see that the adversary activated models consistently outperform their respective base models. Generally, larger improvement can be obtained with smaller set of real training data. Table 2, right panel, further shows the classification accuracy of semi-supervised VAE and its adversary activated variants with different size of labeled training data. We can observe improved performance of the AA-SVAE model. The full results of standard deviations are reported in the supplementary materials.

6 Discussions

Our new interpretations of GANs and VAEs have revealed strong connections between them, and linked the emerging new approaches to the classic wake-sleep algorithm. The generality of the proposed formulation helps with deeper understanding of the broad landscape of deep generative modeling, and encourages mutual exchange of improvement ideas across research lines. It is interesting to further generalize the framework to connect to other learning paradigms such as reinforcement learning as previous work has started exploration [11, 36]. GANs simultaneously learn a metric (defined by the discriminator) to guide the generator learning, which resembles the iterative teacher-student distillation framework [19, 20] where a teacher network is simultaneously learned from structured knowledge (e.g., logic rules) and provides knowledge-informed learning signals for student networks of interest. It is exciting to build formal connections between these approaches and enable incorporation of structured knowledge in deep generative modeling.

Traditional modeling approaches usually distinguish between latent and visible variables clearly and treat them in very different ways. One of the key thoughts in our formulation is that it is not necessary to make clear boundary between latents and visibles (and between inference and generation), but instead, treating them as a symmetric pair helps with modeling and understanding. For instance, we treat the generation space x in GANs as latents, which immediately reveals the connection between GANs and adversarial domain adaptation, and provides an inference interpretation of the generation. A second example is the classic wake-sleep algorithm, where the wake phase reconstructs visibles conditioned on latents, while the sleep phase reconstructs latents conditioned on visibles (i.e., generated samples). Hence, visibles and latents are treated in a completely symmetric manner. Generally, we have prior distributions over latent space, and empirical data distributions over visible space. Both are pre-defined and can be easily sampled from. There are two major differences. (1) Empirical data distributions are usually implicit, i.e., easy for sampling from but intractable for evaluating likelihood. In contrast, priors are usually defined as explicit distributions, amiable for likelihood evaluation; (2) The complexity of the two distributions are different, because visible space is usually complex while latent space tends to be simple. However, the adversarial approach in GANs and other techniques of density ratio estimation [32] have provided useful tools to bridge the gap in (1). For instance, adversarial autoencoder (AAE) leverages the adversarial approach to allow implicit prior distributions over latent space, and a few most recent work [30, 43, 22] uses implicit variational distributions to perform inference. Indeed, the reparameterization trick in VAEs closely resembles construction of implicit distributions (as also seen in the derivations of IWGANs in Eq.13). Adversarial approach is exploited to replace intractable minimization of the prior KL divergence. The difference of space complexity in (2) guides us to choose appropriate tools (e.g., adversarial approach or reconstruction optimization, etc) to minimize the distance between distributions to learn and their targets. However, the tools chosen do not affect the underlying modeling mechanism. For instance,

VAEs and adversarial autoencoder both regularize the model by minimizing the distance between the posterior and certain prior, though VAEs choose KL divergence loss while AAE selects adversarial loss.

We can further extend the symmetric treatment of visible/latent x/z pair to data/label x/c pair, leading to a unified view of the generative and discriminative paradigms for unsupervised and semi-supervised learning. Specifically, conditional generative models create (data, label) pairs by generating data x given label c . These pairs can be used for classifier training [21, 35]. In parallel, discriminative approaches such as knowledge distillation [19, 16] create (data, label) pairs by generating label c conditioned on data x . With the symmetric view of x and c spaces, and neural network based black-box mappings across spaces, we can see the two approaches are essentially the same.

References

- [1] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017.
- [2] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and equilibrium in generative adversarial nets (GANs). *arXiv preprint arXiv:1703.00573*, 2017.
- [3] J. Bornschein and Y. Bengio. Reweighted wake-sleep. *arXiv preprint arXiv:1406.2751*, 2014.
- [4] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.
- [5] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. *ICLR*, 2017.
- [6] T. Che, Y. Li, R. Zhang, R. D. Hjelm, W. Li, Y. Song, and Y. Bengio. Maximum-likelihood augmented discrete generative adversarial networks. *arXiv preprint arXiv:1702.07983*, 2017.
- [7] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, pages 2172–2180, 2016.
- [8] X. Chen, Y. Sun, B. Athiwaratkun, C. Cardie, and K. Weinberger. Adversarial deep averaging networks for cross-lingual sentiment classification. *arXiv preprint arXiv:1606.01614*, 2016.
- [9] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. *ICLR*, 2017.
- [10] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.
- [11] C. Finn, P. Christiano, P. Abbeel, and S. Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*, 2016.
- [12] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.
- [13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.
- [14] P. Goyal, Z. Hu, X. Liang, C. Wang, and E. Xing. Nonparametric variational auto-encoders for hierarchical representation learning. *arXiv preprint arXiv:1703.07027*, 2017.
- [15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [16] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

- [17] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158, 1995.
- [18] R. D. Hjelm, A. P. Jacob, T. Che, K. Cho, and Y. Bengio. Boundary-seeking generative adversarial networks. *arXiv preprint arXiv:1702.08431*, 2017.
- [19] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing. Harnessing deep neural networks with logic rules. In *ACL*, 2016.
- [20] Z. Hu, Z. Yang, R. Salakhutdinov, and E. P. Xing. Deep neural networks with massive learned knowledge. In *EMNLP*, 2016.
- [21] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing. Controllable text generation. In *ICML*, 2017.
- [22] F. Huszár. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.
- [23] M. J. Johnson, D. Duvenaud, A. B. Wiltschko, S. R. Datta, and R. P. Adams. Composing graphical models with neural networks for structured representations and fast inference. In *Neural Information Processing Systems*, 2016.
- [24] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [25] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *NIPS*, pages 3581–3589, 2014.
- [26] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, pages 2539–2547, 2015.
- [27] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.
- [28] X. Liang, Z. Hu, H. Zhang, C. Gan, and E. P. Xing. Recurrent topic-transition GAN for visual paragraph generation. *arXiv preprint arXiv:1703.07022*, 2017.
- [29] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.
- [30] L. Mescheder, S. Nowozin, and A. Geiger. Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*, 2017.
- [31] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *ICLR*, 2017.
- [32] S. Mohamed and B. Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.
- [33] M. Norouzi, S. Bengio, N. Jaitly, M. Schuster, Y. Wu, D. Schuurmans, et al. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems*, pages 1723–1731, 2016.
- [34] S. Nowozin, B. Cseke, and R. Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pages 271–279, 2016.
- [35] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier GANs. *ICML*, 2017.
- [36] D. Pfau and O. Vinyals. Connecting generative adversarial networks and actor-critic methods. *arXiv preprint arXiv:1610.01945*, 2016.
- [37] S. Purushotham, W. Carvalho, T. Nilanon, and Y. Liu. Variational recurrent adversarial deep domain adaptation. In *ICLR*, 2017.

- [38] L. Qin, Z. Zhang, H. Zhao, Z. Hu, and E. P. Xing. Adversarial connective-exploiting networks for implicit discourse relation classification. *arXiv preprint arXiv:1704.00217*, 2017.
- [39] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [40] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *NIPS*, pages 2226–2234, 2016.
- [41] J. Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.
- [42] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised MAP inference for image super-resolution. *ICLR*, 2017.
- [43] D. Tran, R. Ranganath, and D. M. Blei. Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896*, 2017.
- [44] S. Zhai, Y. Cheng, R. Feris, and Z. Zhang. Generative adversarial networks as variational training of energy based models. *arXiv preprint arXiv:1611.01799*, 2016.

A Adversarial Domain Adaptation (ADA)

ADA aims to transfer prediction knowledge learned from a source domain with labeled data to a target domain without labels, by learning domain-invariant features. Let $D_\phi(\mathbf{x}) = q_\phi(y|\mathbf{x})$ be the domain discriminator. The conventional formulation of ADA is as following:

$$\begin{aligned} \max_D \mathcal{L}_D &= \mathbb{E}_{\mathbf{x}=G(\mathbf{z}), \mathbf{z} \sim p(\mathbf{z}|y=1)} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x}=G(\mathbf{z}), \mathbf{z} \sim p(\mathbf{z}|y=0)} [\log(1 - D(\mathbf{x}))], \\ \max_G \mathcal{L}_G &= \mathbb{E}_{\mathbf{x}=G(\mathbf{z}), \mathbf{z} \sim p(\mathbf{z}|y=1)} [\log(1 - D(\mathbf{x}))] + \mathbb{E}_{\mathbf{x}=G(\mathbf{z}), \mathbf{z} \sim p(\mathbf{z}|y=0)} [\log D(\mathbf{x})]. \end{aligned} \quad (20)$$

Further add the supervision objective of predicting label t in the source domain with a classifier $u(t|\mathbf{x})$:

$$\max_{u,G} \mathcal{L}_{u,G} = \mathbb{E}_{(\mathbf{z},t)} [\log u(t|G(\mathbf{x}))]. \quad (21)$$

We then obtain the conventional formulation of adversarial domain adaptation used or similar in [12, 37, 38, 8].

B Lemma 1

Proof.

$$\begin{aligned} \mathbb{E}_{p(y)} [\mathbb{E}_{p_\theta(\mathbf{x}|y)} [\log q^r(y|\mathbf{x})]] &= \\ - \mathbb{E}_{p(y)} [\text{KL}(p_\theta(\mathbf{x}|y) \| q^r(\mathbf{x}|y)) - \text{KL}(p_\theta(\mathbf{x}|y) \| p_{\theta_0}(\mathbf{x}))], \end{aligned} \quad (22)$$

where

$$\begin{aligned} \mathbb{E}_{p(y)} [\text{KL}(p_\theta(\mathbf{x}|y) \| p_{\theta_0}(\mathbf{x}))] &= \\ p(y=0) \cdot \text{KL} \left(p_\theta(\mathbf{x}|y=0) \| \frac{p_{\theta_0}(\mathbf{x}|y=0) + p_{\theta_0}(\mathbf{x}|y=1)}{2} \right) &+ \\ p(y=1) \cdot \text{KL} \left(p_\theta(\mathbf{x}|y=1) \| \frac{p_{\theta_0}(\mathbf{x}|y=0) + p_{\theta_0}(\mathbf{x}|y=1)}{2} \right). \end{aligned} \quad (23)$$

Taking derivatives w.r.t θ at θ_0 we get

$$\begin{aligned} \nabla_\theta \mathbb{E}_{p(y)} [\text{KL}(p_\theta(\mathbf{x}|y) \| p_{\theta_0}(\mathbf{x}))] |_{\theta=\theta_0} &= \\ \frac{1}{2} \int_{\mathbf{x}} \nabla_\theta p_\theta(\mathbf{x}|y=0) \frac{p_{\theta_0}(\mathbf{x}|y=0) + p_{\theta_0}(\mathbf{x}|y=1)}{2} |_{\theta=\theta_0} &+ \\ \frac{1}{2} \int_{\mathbf{x}} \nabla_\theta p_\theta(\mathbf{x}|y=1) \frac{p_{\theta_0}(\mathbf{x}|y=0) + p_{\theta_0}(\mathbf{x}|y=1)}{2} |_{\theta=\theta_0} & \\ = \nabla_\theta JSD(p_\theta(\mathbf{x}|y=0) \| p_\theta(\mathbf{x})) |_{\theta=\theta_0} \end{aligned} \quad (24)$$

Taking derivatives of the both sides of Eq.(22) at w.r.t θ at θ_0 and plugging the last equation of Eq.(24), we obtain the desired results. \square

C Lemme 2

Proof. For the reconstruction term:

$$\begin{aligned} \mathbb{E}_{p_{\theta_0}(\mathbf{x})} [\mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x},y)q_\eta^r(y|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z},y)]] &= \\ \frac{1}{2} \mathbb{E}_{p_{\theta_0}(\mathbf{x}|y=1)} [\mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x},y=0), y=0 \sim q_\eta^r(y|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z},y=0)]] & \\ + \frac{1}{2} \mathbb{E}_{p_{\theta_0}(\mathbf{x}|y=0)} [\mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x},y=1), y=1 \sim q_\eta^r(y|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z},y=1)]] & \\ = \frac{1}{2} \mathbb{E}_{p_{data}(\mathbf{x})} [\mathbb{E}_{\tilde{q}_\eta(\mathbf{z}|\mathbf{x})} [\log \tilde{p}_\theta(\mathbf{x}|\mathbf{z})]] + const, \end{aligned} \quad (25)$$

where $y = 0 \sim q_\eta^r(y|\mathbf{x})$ means $q_\eta^r(y|\mathbf{x})$ predicts $y = 0$ with probability 1. Note that both $q_\eta(\mathbf{z}|\mathbf{x}, y = 1)$ and $p_\theta(\mathbf{x}|\mathbf{z}, y = 1)$ are constant distributions without free parameters to learn; $q_\eta(\mathbf{z}|\mathbf{x}, y = 0) = \tilde{q}_\eta(\mathbf{z}|\mathbf{x})$, and $p_\theta(\mathbf{x}|\mathbf{z}, y = 0) = \tilde{p}_\theta(\mathbf{x}|\mathbf{z})$.

For the KL prior regularization term:

$$\begin{aligned}
& \mathbb{E}_{p_{\theta_0}(\mathbf{x})} [\text{KL}(q_\eta(\mathbf{z}|\mathbf{x}, y)q_*^r(y|\mathbf{x})\|p(\mathbf{z}|y)p(y))] \\
&= \mathbb{E}_{p_{\theta_0}(\mathbf{x})} \left[\int q_*^r(y|\mathbf{x}) \text{KL}(q_\eta(\mathbf{z}|\mathbf{x}, y)\|p(\mathbf{z}|y)) dy + \text{KL}(q_*^r(y|\mathbf{x})\|p(y)) \right] \\
&= \frac{1}{2} \mathbb{E}_{p_{\theta_0}(\mathbf{x}|y=1)} [\text{KL}(q_\eta(\mathbf{z}|\mathbf{x}, y=0)\|p(\mathbf{z}|y=0)) + \text{const}] + \frac{1}{2} \mathbb{E}_{p_{\theta_0}(\mathbf{x}|y=1)} [\text{const}] \\
&= \frac{1}{2} \mathbb{E}_{p_{data}(\mathbf{x})} [\text{KL}(\tilde{q}_\eta(\mathbf{z}|\mathbf{x})\|\tilde{p}(\mathbf{z}))].
\end{aligned} \tag{26}$$

Combining Eq.(25) and Eq.(26) we recover the conventional VAE objective in Eq.(7) in the paper. \square

D Importance Weighted GANs (IWGAN)

From Eq.(4) in the paper, we can view GANs as maximizing a lower bound of the “marginal log-likelihood”:

$$\begin{aligned}
\log q(y) &= \log \int p_\theta(\mathbf{x}|y) \frac{q^r(y|\mathbf{x})p_{\theta_0}(\mathbf{x})}{p_\theta(\mathbf{x}|y)} d\mathbf{x} \\
&\geq \int p_\theta(\mathbf{x}|y) \log \frac{q^r(y|\mathbf{x})p_{\theta_0}(\mathbf{x})}{p_\theta(\mathbf{x}|y)} d\mathbf{x} \\
&= -\text{KL}(p_\theta(\mathbf{x}|y)\|q^r(\mathbf{x}|y)) + \text{const}.
\end{aligned} \tag{27}$$

We can apply the same importance weighting method as in IWAE [4] to derive a tighter bound.

$$\begin{aligned}
\log q(y) &= \log \mathbb{E} \left[\frac{1}{k} \sum_{i=1}^k \frac{q^r(y|\mathbf{x}_i)p_{\theta_0}(\mathbf{x}_i)}{p_\theta(\mathbf{x}_i|y)} \right] \\
&\geq \mathbb{E} \left[\log \frac{1}{k} \sum_{i=1}^k \frac{q^r(y|\mathbf{x}_i)p_{\theta_0}(\mathbf{x}_i)}{p_\theta(\mathbf{x}_i|y)} \right] \\
&= \mathbb{E} \left[\log \frac{1}{k} \sum_{i=1}^k w_i \right] \\
&:= \mathcal{L}_k(y)
\end{aligned} \tag{28}$$

where we have denoted $w_i = \frac{q^r(y|\mathbf{x}_i)p_{\theta_0}(\mathbf{x}_i)}{p_\theta(\mathbf{x}_i|y)}$. We recover the lower bound of Eq.(27) when setting $k = 1$.

To maximize the importance weighted lower bound, we compute the gradient:

$$\begin{aligned}
\nabla_\theta \mathcal{L}_k(y) &= \nabla_\theta \mathbb{E}_{\mathbf{x}_1, \dots, \mathbf{x}_k} \left[\log \frac{1}{k} \sum_{i=1}^k w_i \right] = \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_k} \left[\nabla_\theta \log \frac{1}{k} \sum_{i=1}^k w(y, \mathbf{x}(\mathbf{z}_i, \boldsymbol{\theta})) \right] \\
&= \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_k} \left[\sum_{i=1}^k \tilde{w}_i \nabla_\theta \log w(y, \mathbf{x}(\mathbf{z}_i, \boldsymbol{\theta})) \right],
\end{aligned} \tag{29}$$

where $\tilde{w}_i = w_i / \sum_{i=1}^k w_i$ are the normalized importance weights. We expand the weight at $\boldsymbol{\theta} = \boldsymbol{\theta}_0$

$$w_i|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} = \frac{q^r(y|\mathbf{x}_i)p_{\theta_0}(\mathbf{x}_i)}{p_{\theta_0}(\mathbf{x}_i|y)} = q^r(y|\mathbf{x}_i) \frac{\frac{1}{2}p_{\theta_0}(\mathbf{x}_i|y=0) + \frac{1}{2}p_{\theta_0}(\mathbf{x}_i|y=1)}{p_{\theta_0}(\mathbf{x}_i|y)}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}. \tag{30}$$

The ratio of $p_{\theta_0}(\mathbf{x}_i|y=0)$ and $p_{\theta_0}(\mathbf{x}_i|y=1)$ is intractable. Using the Bayes’ rule and approximating with the discriminator distribution, we have

$$\frac{p(\mathbf{x}|y=0)}{p(\mathbf{x}|y=1)} = \frac{p(y=0|\mathbf{x})p(y=1)}{p(y=1|\mathbf{x})p(y=0)} \approx \frac{q(y=0|\mathbf{x})}{q(y=1|\mathbf{x})}. \tag{31}$$

Plug Eq.(31) into the above we have

$$w_i|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \approx \frac{q^r(y|\mathbf{x}_i)}{q(y|\mathbf{x}_i)}. \tag{32}$$

In Eq.(29), the derivative $\nabla_{\theta} \log w_i$ is

$$\nabla_{\theta} \log w(y, \mathbf{x}(\mathbf{z}_i, \boldsymbol{\theta})) = \nabla_{\theta} \log q^r(y|\mathbf{x}(\mathbf{z}_i, \boldsymbol{\theta})) + \nabla_{\theta} \log \frac{p_{\theta_0}(\mathbf{x}_i)}{p_{\theta}(\mathbf{x}_i|y)}. \quad (33)$$

Similar to GAN, we omit the second term on the RHS of the equation. Therefore, the resulting update rule of $p_{\theta}(\mathbf{x}|y)$ is

$$\nabla_{\theta} \mathcal{L}_k(y) = \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_k} \left[\sum_{i=1}^k \frac{q^r(y|\mathbf{x}_i)}{q(y|\mathbf{x}_i)} \nabla_{\theta} \log q^r(y|\mathbf{x}(\mathbf{z}_i, \boldsymbol{\theta})) \right] \quad (34)$$

E Experimental Results of SVAE

Table 3 shows the results.

	1%	10%
SVAE	0.9412 \pm .0039	0.9768 \pm .0009
AASVAE	0.9425\pm.0045	0.9797\pm.0010

Table 3: Classification accuracy of semi-supervised VAEs and the adversary activated extension on the MNIST test set, with varying size of real labeled training examples.