Bridging Maximum Likelihood and Adversarial Learning via α-Divergence

Miaoyun Zhao,* Yulai Cong,* Shuyang Dai, Lawrence Carin
Department of Electrical and Computer Engineering, Duke University

Abstract

Maximum likelihood (ML) and adversarial learning are two popular approaches for training generative models, and from many perspectives these techniques are complementary. ML learning encourages the capture of all data modes, and it is typically characterized by stable training. However, ML learning tends to distribute probability mass diffusely over the data space, $e.g.$, yielding blurry synthetic images. Adversarial learning is well known to synthesize highly realistic natural images, despite practical challenges like mode dropping and delicate training. We propose an α-Bridge to unify the advantages of ML and adversarial learning, enabling the smooth transfer from one to the other via the α-divergence. We reveal that generalizations of the α-Bridge are closely related to approaches developed recently to regularize adversarial learning, providing insights into that prior work, and further understanding of why the α-Bridge performs well in practice.

1 Introduction

Given observed data samples, a well-known task concerns fitting a generative model to the unknown underlying data distribution. Two popular approaches for that task are classical maximum likelihood (ML) learning and recently-developed adversarial learning. Both of these approaches are equivalent to minimizing a corresponding divergence between the model distribution and the data distribution (Bishop 2006; Goodfellow et al. 2014).

ML learning seeks to find model parameters that maximize the log-likelihood of the model over the observed data samples, which is equivalent to minimizing the forward Kullback-Leibler (KL) divergence between the model distribution and the data distribution (McLachlan and Krishnan 2007). When considering models with latent variables, variational inference (VI) (Jordan et al. 1999; Blei and Jordan 2006) is an important class of approximate ML learning, in which a variational expression constitutes a lower bound on the log-likelihood, and learning proceeds by seeking to maximize this bound. There has been significant recent work on utilizing neural networks within VI (Kingma and Welling 2014; Dai and Wipf 2019).

Because of the properties of the forward KL, ML learning tends to associate positive mass with each data sample, forming a zero-avoiding phenomenon (Minka and others 2005). Accordingly, all data modes are “covered” by the model distribution; by contrast, adversarial learning is often characterized by a mode-dropping phenomenon (Srivastava et al. 2017). Another advantage of ML learning (forward KL) is that its training procedure is typically much more stable than that of adversarial learning. The instability of adversarial learning is in part because the mode dropping may vary as a function of learning iteration. On the other hand, the zero-avoiding phenomenon of ML learning may loosely distribute probability mass among data modes. An example consequence is that generative models trained with ML tend to generate blurry images (Goodfellow et al. 2014; Larsen et al. 2015; Arjovsky, Chintala, and Bottou 2017). Adversarially-learned models, by contrast, are capable of synthesizing highly realistic natural images (Goodfellow et al. 2014; Nowozin, Cseke, and Tomioka 2016; Zhang et al. 2018; Gulrajani et al. 2017; Brock, Donahue, and Simonyan 2019).

The original generative adversarial network (GAN) minimizes the Jensen-Shannon (JS) divergence between the model distribution and that of the data (Goodfellow et al. 2014). In (Nowozin, Cseke, and Tomioka 2016) it was shown that learning based on minimizing any f-divergence can be formulated as an adversarial learning objective (with the JS divergence as a special case). In this paper, we focus on the reverse KL divergence as in (Li et al. 2019) because (i) it naturally relates to ML learning (by reversing the KL); (ii) (Lucic et al. 2018) showed that most GANs with the same budget can reach similar performance with enough hyperparameter optimization and random restarts; (iii) adversarial learning with the f-divergence (Nowozin, Cseke, and Tomioka 2016) reduces to estimating a log-likelihood ratio between the true and model distributions, and the reverse-KL is as good as any other f-divergence choice for this purpose (Li et al. 2019); and (iv) forward and reverse KL divergences are two ends of the α-Bridge developed in this paper.

It is interesting to note that ML learning (based on the forward KL) and adversarial learning (with the reverse

*Correspondence to: Miaoyun Zhao miaoyun9zhao@gmail.com and Yulai Cong yulaicong@gmail.com.

Table 1: Comparing maximum likelihood and adversarial learning.

Method / Property	Maximum Likelihood (Forward KL)	Adversarial (Reverse KL)
Mode covering (zero-avoiding)	✓	✗
Stable training	✓	✗
Inference	✓	✗
Realistic generated samples	✗	✓

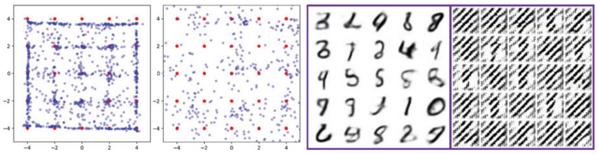


Figure 1: Demonstration of adversarial learning forgetting the information learned/initialized by ML learning on 25-Gaussians (the first two) and MNIST (the last two). From left to right are the snapshots of ML initialization and 20 following iterations of adversarial learning, respectively. See Appendix D for details.

KL as an important example) seem to have complementary advantages and disadvantages (Nguyen et al. 2017), as shown in Table 1. To unify their advantages, an intuitive approach would directly combine them. However, as stated in (Larsen et al. 2015; Mathieu et al. 2016) and empirically shown in (Zhang et al. 2019), such a naive method does not work well. Another intuitive approach would, for example, directly initialize the reverse-KL-based adversarial learning with the parameters learned from ML learning. For the second approach, empirical results in Figure 1 indicate that catastrophic forgetting (Kirkpatrick et al. 2017; Liang et al. 2018) happens when adversarially finetuning the ML-learned parameters. Appendix F discusses/compares other potential approaches to combine adversarial and ML learning. To unify the advantages from ML and adversarial learning in a principled way, we propose a novel α-Bridge, via the α-divergence, to smoothly connect the forward and reverse KL, through which one can transfer the advantages from one to the other. In addition to the practical value of the α-Bridge, our subsequent analysis on the α-divergence is deemed an important methodological perspective on how ML and adversarial learning are related and may be linked.

The main contributions of this paper are as follows. (i) An α-Bridge is proposed to connect the forward and reverse KL in a principled manner, which can be interpreted as a novel way to "bridge" the two research fields of ML and adversarial learning. (ii) The gradient of the α-divergence is shown to have two equivalent expressions, one that utilizes the gradient information from ML learning (forward KL), while the other uses the gradient information from adversarial learning (reverse KL). (iii) The twin gradients of α-divergence have complimentary variance properties, α-Bridge elegantly combines the advantages of both and manages a low Monte Carlo (MC) variance along the varying of α. (iv) Two generalizations of our α-Bridge are revealed, that are closely related to CycleGAN (Zhu et al. 2017) and ALICE (Li et al. 2017), two methods for regularizing (stabilizing) adversarial learning. (v) It is demonstrated empirically that the proposed α-Bridge is capable of benefiting from the advantages of ML learning, transferring information from ML to adversarial learning, and is capable of transplanting the variational posterior in ML learning into an inference arm for adversarial learning.

2 Preliminaries

Given observed data $\boldsymbol{x}$, drawn from unknown underlying data distribution $q(\boldsymbol{x})$, and a parameterized model distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ with parameters $\boldsymbol{\theta}$, the task is to learn $\boldsymbol{\theta}^*$ so that $p_{\boldsymbol{\theta}^*}(\boldsymbol{x})$ best fits the observed data, or identically $p_{\boldsymbol{\theta}^*}(\boldsymbol{x})$ is closest to $q(\boldsymbol{x})$. For that task, two popular research fields include ML learning (with "closeness" of $p_{\boldsymbol{\theta}^*}(\boldsymbol{x})$ and $q(\boldsymbol{x})$ quantified via the forward KL) and adversarial learning (with the reverse KL as an important example of how "closeness" is measured).

2.1 Maximum Likelihood Learning (Forward KL)

A classic method to match a model $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ to the data distribution $q(\boldsymbol{x})$ is ML learning (or maximum likelihood estimation), namely,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{q(\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x})] = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{D}_{\mathrm{KL}}[q(\boldsymbol{x}) \| p_{\boldsymbol{\theta}}(\boldsymbol{x})], \tag{1}$$

where $\mathcal{D}_{\mathrm{KL}}[q(\boldsymbol{x})\|p_{\boldsymbol{\theta}}(\boldsymbol{x})] = \mathbb{E}_{q(\boldsymbol{x})}[\log q(\boldsymbol{x}) - \log p_{\boldsymbol{\theta}}(\boldsymbol{x})]$ is the forward KL. The gradient wrt $\boldsymbol{\theta}$ is

$$\nabla_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}[q(\boldsymbol{x}) \| p_{\boldsymbol{\theta}}(\boldsymbol{x})] = \mathbb{E}_{q(\boldsymbol{x})}[-\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(x)]. \tag{2}$$

For more modeling capacity, it is often convenient to define $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ as the marginal of some parameterized joint distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$, with latent variable $\boldsymbol{z}$. Although $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is usually intractable, variational inference (Jordan et al. 1999; Kingma and Welling 2014; Blei, Kucukelbir, and McAuliffe 2017) seeks to solve the ML learning in (1) via maximizing the evidence lower bound (ELBO)

$$\mathrm{ELBO}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q(\boldsymbol{x}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\big], \tag{3}$$

where $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ is the variational approximation with parameters $\boldsymbol{\phi}$, and the bound is tight when $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. The gradient wrt $\boldsymbol{\theta}$ becomes

$$\nabla_{\boldsymbol{\theta}} \mathrm{ELBO}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q(\boldsymbol{x}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[-\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})].$$

In practice, $\mathbb{E}_{q(\boldsymbol{x})}[\cdot]$ are approximated as averages over a finite set of observed samples.

2.2 Adversarial Learning (Reverse KL)

Recent progress has resulted in many techniques for adversarial training of generative models (Goodfellow et al. 2014; Gulrajani et al. 2017; Nowozin, Cseke, and Tomioka 2016; Brock, Donahue, and Simonyan 2019). The original GAN (Goodfellow et al. 2014) seeks to solve

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\beta}} \mathbb{E}_{q(\boldsymbol{x})}[\log \sigma(f_{\boldsymbol{\beta}}(\boldsymbol{x}))] + \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}[\log(1 - \sigma(f_{\boldsymbol{\beta}}(\boldsymbol{x})))], \tag{4}$$

where $\sigma(f_{\boldsymbol{\beta}}(\boldsymbol{x})) \triangleq D_{\boldsymbol{\beta}}(\boldsymbol{x})$ is called the discriminator, $\sigma(a) = 1/[1+\exp(-a)]$, and samples are drawn from $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ by the generative process

$$\boldsymbol{x} \sim \delta(\boldsymbol{x}|G_{\boldsymbol{\theta}}(\boldsymbol{z})), \boldsymbol{z} \sim p(\boldsymbol{z}), \tag{5}$$

where $\delta(\boldsymbol{x}|\boldsymbol{a})$ is the Dirac delta function located at $\boldsymbol{a}$, $G_{\boldsymbol{\theta}}(\boldsymbol{z})$ is called the generator, and $p(\boldsymbol{z})$ is an easy-to-sample distribution. It is shown (Goodfellow et al. 2014) that the optimal $\boldsymbol{\beta}^*$ for (4) satisfies

$$f_{\boldsymbol{\beta}^*}(\boldsymbol{x}) = \log q(\boldsymbol{x}) - \log p_{\boldsymbol{\theta}}(\boldsymbol{x}). \tag{6}$$

Accordingly, (4) seeks to minimize the Jensen-Shannon (JS) divergence for parameters $\boldsymbol{\theta}$ (Goodfellow et al. 2014).

Alternatively, one could also consider a similar GAN objective based on the reverse KL divergence $\mathcal{D}_{\mathrm{KL}}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})]$ (Nowozin, Cseke, and Tomioka 2016; Li et al. 2019), on which we focus in this paper; as discussed in the Introduction, many GANs are closely related (Nowozin, Cseke, and Tomioka 2016) and can reach similar performance with the same budget (Lucic et al. 2018), and therefore a focus on the reverse KL is not deemed particularly limiting. The log-ratio estimate in (6) is exploited both in the reverse KL and its gradient as

$$\begin{aligned} &\mathcal{D}_{\mathrm{KL}}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})] = \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}[-f_{\boldsymbol{\beta}^*}(\boldsymbol{x})] \\ &\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\mathrm{KL}}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})] = \\ &\qquad \mathbb{E}_{p(\boldsymbol{z})\delta(\boldsymbol{x}|G_{\boldsymbol{\theta}}(\boldsymbol{z}))}\big[-[\nabla_{\boldsymbol{\theta}}G_{\boldsymbol{\theta}}(\boldsymbol{z})][\nabla_{\boldsymbol{x}}f_{\boldsymbol{\beta}^*}(\boldsymbol{x})]\big]. \end{aligned} \tag{7}$$

3 Connecting Maximum Likelihood and Adversarial Learning via α-Bridge

Maximum likelihood and adversarial learning have many *complementary* strengths and weaknesses, motivating development of a method that achieves their principled integration. Toward that end, we propose what we term an α-Bridge, designed using the α-divergence (Cichocki and Amari 2010). The α-Bridge smoothly connects the forward and reverse KL divergences, making it possible to transfer advantages from one to the other.

Given model distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ and the underlying data distribution $q(\boldsymbol{x})$, the α-divergence measuring the dissimilarity between these two distributions is defined as

$$\mathcal{D}_{\alpha}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})] = \frac{1}{\alpha(1-\alpha)}\Big[1 - \int p_{\boldsymbol{\theta}}(\boldsymbol{x})^{\alpha}q(\boldsymbol{x})^{1-\alpha}d\boldsymbol{x}\Big]. \tag{8}$$

The α-divergence has many attractive properties (Cichocki and Amari 2010), for example, (i) it is unique (Amari 2009); (ii) $\lim_{\alpha\to 0}\mathcal{D}_{\alpha}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})] = \mathcal{D}_{\mathrm{KL}}[q(\boldsymbol{x})\|p_{\boldsymbol{\theta}}(\boldsymbol{x})]$; (iii) $\lim_{\alpha\to 1}\mathcal{D}_{\alpha}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})] = \mathcal{D}_{\mathrm{KL}}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})]$; and (iv) the α-divergence is a continuous function of α. These properties motivate development of a smooth "bridge" via the α-divergence, named the α-Bridge, to continuously transfer between forward and reverse KL. Before discussing the proposed α-Bridge in detail, below we first reveal its key foundation, in the context of this paper: the α-divergence has two equivalent expressions for its gradient, which utilize the gradient information either from the forward or reverse KL.

3.1 Twin Gradients of α-Divergence

Given the α-divergence defined in (8), with straightforward derivation, we have

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})] = \\ &\qquad \frac{1}{1-\alpha}\Big[-\int p_{\boldsymbol{\theta}}(\boldsymbol{x})^{\alpha-1}q(\boldsymbol{x})^{1-\alpha}\nabla_{\boldsymbol{\theta}}p_{\boldsymbol{\theta}}(\boldsymbol{x})d\boldsymbol{x}\Big]. \end{aligned} \tag{9}$$

An interesting fact of (9) is that one can turn it into an expectation-based expression wrt either the data distribution $q(\boldsymbol{x})$ or the model one $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, resulting in two different expressions for the same gradient (see Appendix A for details). By forming expectations wrt $q(\boldsymbol{x})$, we have

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})] = \\ &\frac{1}{1-\alpha}\mathbb{E}_{q(\boldsymbol{x})}\Big[-\Big[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{q(\boldsymbol{x})}\Big]^{\alpha}\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{x})\Big] \triangleq \nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{F}, \end{aligned} \tag{10}$$

where $\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{F}$ is used for brevity. The gradient information from the forward KL in (2) serves as a building block for (10). However, it is *different* from the direct gradient of the forward KL in that $\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{F}$ has an adaptive ratio-related weight term $\frac{1}{1-\alpha}\big[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{q(\boldsymbol{x})}\big]^{\alpha}$ within the expectation (when $\alpha \to 0^{+}$ this term vanishes, leading to the gradient of the forward KL in that limit). For the gradient expression related to $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ modeled in (5), we have

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})] \triangleq \nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{R} = \\ &\mathbb{E}_{p(\boldsymbol{z})\delta(\boldsymbol{x}|G_{\boldsymbol{\theta}}(\boldsymbol{z}))}\Big[[\nabla_{\boldsymbol{\theta}}G_{\boldsymbol{\theta}}(\boldsymbol{z})]\Big[\frac{q(\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})}\Big]^{1-\alpha}\Big[\nabla_{\boldsymbol{x}}\log\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{q(\boldsymbol{x})}\Big]\Big]. \end{aligned} \tag{11}$$

Similarly we use $\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{R}$ for brevity. Compared to (7), $\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{R}$ utilizes the gradient information from the reverse KL, with another adaptive weighting term $\big[\frac{q(\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})}\big]^{1-\alpha}$ (which vanishes in the limit $\alpha \to 1^{-}$, yielding the gradient of the reverse KL in that limit). For more general model $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ beyond (5), the GO gradient (Cong et al. 2019) can be utilized to calculate $\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{R}$.

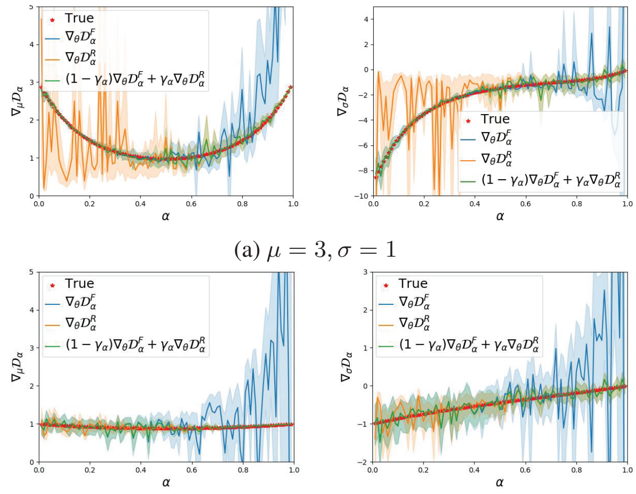
It is important to note that $\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{F}$ and $\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{R}$ are two *equivalent* gradient expressions for the same objective $\mathcal{D}_{\alpha}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})]$, even though they utilize different gradient information (accordingly different MC variance properties as detailed below) from the forward and reverse KL, respectively. Thus, we call them the twin gradients of α-divergence. In the limits on α, the former is associated with the forward KL and the latter with the reverse KL, but for $\alpha \in (0, 1)$ the twin gradients are *not* associated with either; this explains why the proposed α-Bridge in Sec. 3.2 is different from a (possibly convex) combination of the forward and reverse KL.

Since $\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{F}$ and $\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{R}$ are equivalent expressions for $\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})]$, any convex combination of them remains an unbiased gradient estimator, which may be interpreted as exploiting the information from one side to regularize the other side. We propose to use an α-related dynamic combination as

$$\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})] = (1-\gamma_{\alpha})\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{F} + \gamma_{\alpha}\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\alpha}^{R}, \tag{12}$$

where γ_{α} is specified as a smooth increasing function[1] of

[1] It is consistent with the instinct that, as smoothly transferring from the forward to reverse KL, the used information from the forward/reverse KL should smoothly decrease/increase correspondingly. Appendix E shows a series of experiments demonstrating several intuitive choices for γ_{α}. We empirically find that the sigmoid-like function $\gamma_{\alpha} = \frac{\sigma(c\alpha+d)-\sigma(d)}{\sigma(c+d)-\sigma(d)}$ (with hyperparameters c, d) works well. Accordingly, we use such γ_{α} in our experiments and leave as future research how to optimally choose γ_{α}.



(a) $\mu = 3, \sigma = 1$

(b) $\mu = 1, \sigma = 1$

Figure 2: Illustration of different MC variance properties of different gradient estimators of $\nabla_{\{\mu,\sigma\}}\mathcal{D}_\alpha[\mathcal{N}(x;\mu,\sigma^2)||\mathcal{N}(x;0,1)]$ for $\alpha \in (0,1)$. 1 MC sample is used to estimate the gradient. The results are based on 100 random trials.

α satisfying $\gamma_0 = 0$, $\gamma_1 = 1$, ensuring equation (12) *exactly* recovers the gradient of the forward/reverse KL when $\alpha = 0/\alpha = 1$. Such a γ_α is motivated by the smoothness of the α-divergence. When $\alpha \to 0$ the α-divergence smoothly approaches the forward KL with increasingly-similar gradients; intuitively to calculate the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{D}_\alpha[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})]$, one should prefer $\nabla_{\boldsymbol{\theta}}\mathcal{D}_\alpha^F$ more as it uses the ML gradient information. Similarly, $\nabla_{\boldsymbol{\theta}}\mathcal{D}_\alpha^R$ is preferred when $\alpha \to 1$ as it uses the adversarial gradient information and the α-divergence now smoothly approaches the reverse KL. With a simple example, Figure 2 confirms that intuition by showing that the twin gradients $\nabla_{\boldsymbol{\theta}}\mathcal{D}_\alpha^F$ and $\nabla_{\boldsymbol{\theta}}\mathcal{D}_\alpha^R$ have complementary variance properties, the former/latter having lower MC variance when $\alpha \to 0/\alpha \to 1$. Figure 2 also shows that combining the twin gradients as in (12) unifies the advantages from both sides and presents a better gradient estimator with lower MC variance for $\alpha \in (0,1)$. The twin gradients can be interpreted as control variants to each other. This is the foundation of our paper, which is further exploited in the following to develop our α-Bridge. We are taking the convex combination of two different forms of the *same* gradient, which is *distinct* from just taking a convex combination of the *different* gradients from the forward and reverse KL.

3.2 α-Bridge via Twin Gradients

Based on the twin gradients discussed above, we propose a novel α-Bridge to dynamically transfer between forward KL (ML learning) and reverse KL (adversarial learning), so as to unify the advantages from both ends[2]. In this paper, we are motivated by applications associated with GAN, with a goal

[2]See Appendix H for detailed discussions on other potential generalizations of α-Bridge.

Algorithm 1 α-Bridge (from forward to reverse KL)

Input: Data samples $\boldsymbol{x}_i \sim q(\boldsymbol{x})$, an implicit model $p_{\boldsymbol{\theta}}(\boldsymbol{x})$
Output: $\boldsymbol{\theta}^*$ such that $p_{\boldsymbol{\theta}^*}(\boldsymbol{x})$ is closest to the underlying data distribution $q(\boldsymbol{x})$

 # Step I: ML learning (forward KL, $\alpha = 0$)

1: ML learning for the generator parameter $\boldsymbol{\theta}$ with the gradient in (14). Maximizing the ELBO in (3) for training the variational parameters $\boldsymbol{\phi}$. Pretrain the discriminator parameters $\boldsymbol{\beta}$ with the objective in (4).

 # Step II: Transferring from $\alpha \to 0^+$ to $\alpha \to 1^-$

2: **for** α gradually increasing from 0^+ to 1^- **do**
3: Train $\boldsymbol{\theta}$ by minimizing the α-divergence $\mathcal{D}_\alpha[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})]$ with the gradient in (15);
4: Train $\boldsymbol{\phi}$ by maximizing the ELBO in (3);
5: Train $\boldsymbol{\beta}$ with the objective in (4);
6: **end for**

 # Step III: Adversarial learning (reverse KL, $\alpha = 1$)

7: Refine $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and $\boldsymbol{\beta}$ with the objectives in (7), (3), and (4), respectively.

of generating realistic samples from our model. Accordingly, we set our α-Bridge to transfer from the forward KL to the reverse KL, in order to gradually transfer the advantages (see Table 1) of ML learning to adversarial learning. Specifically, we propose to train $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ via the α-Bridge with the following three successive steps.

In Step I, we adopt ML learning (forward KL, $\alpha = 0$) for efficient initialization thanks to its mode-covering and stable-training properties. One can skip this step if pretrained models from ML learning are available. From the perspective of practical implementation, one often need to approximately calculate the gradient in (2), as $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ may be intractable for example for the implicit model in (5). For this issue, we first add small Gaussian noise[3] on top of the generative process of $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ to form a semi-implicit surrogate model (Yin and Zhou 2018) $\tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x}) : \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}|\boldsymbol{x}', \sigma^2\mathbf{I}), \boldsymbol{x}' \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}')$, for which we have $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \lim_{\sigma^2 \to 0} \nabla_{\boldsymbol{\theta}} \log \tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x})$. Observing that $\tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x})$ is equivalent to

$$\tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x}) : \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}|G_{\boldsymbol{\theta}}(\boldsymbol{z}), \sigma^2\mathbf{I}), \boldsymbol{z} \sim p(\boldsymbol{z}), \tag{13}$$

which has computable joint distribution $\tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$, we then use the ELBO technique to get

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &\approx \nabla_{\boldsymbol{\theta}} \log \tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x}) \\ &= \mathbb{E}_{q(\boldsymbol{x})q_{\boldsymbol{\phi}^*}(\boldsymbol{z}|\boldsymbol{x})}\big[\nabla_{\boldsymbol{\theta}} \log \tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})\big], \end{aligned} \tag{14}$$

with an additional variational inference arm.

In the middle Step II, we continue the training of $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ by gradually changing α from 0^+ to 1^-, so as to transfer what's learned during Step I to the next Step III (reverse-KL-based adversarial learning, $\alpha = 1$). The gradient of the α-divergence in (12) is used during training. The same techniques discussed above is adopted to calculate the $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ term within $\nabla_{\boldsymbol{\theta}}\mathcal{D}_\alpha^F$ in (10). To calculate the

[3]We need not to add the noise if $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is modeled as (13) in the first place, for example to take into consideration the widely-existing observation noise of data. See Appendix B for details.

density-ratio-related terms in both $\nabla_{\boldsymbol{\theta}}\mathcal{D}_\alpha^F$ and $\nabla_{\boldsymbol{\theta}}\mathcal{D}_\alpha^R$ (see (10) and (11)), we follow the common practice in the GAN literature to solve (4) for $f_{\boldsymbol{\beta}^*}(\boldsymbol{x})$ in (6). Accordingly, we have $\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{q(\boldsymbol{x})} = e^{-f_{\boldsymbol{\beta}^*}(\boldsymbol{x})}$, $\nabla_{\boldsymbol{x}} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{q(\boldsymbol{x})} = -\nabla_{\boldsymbol{x}} f_{\boldsymbol{\beta}^*}(\boldsymbol{x})$, and

$$
\begin{aligned}
&\nabla_{\boldsymbol{\theta}}\mathcal{D}_\alpha[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})] \approx \\
&\frac{1-\gamma_\alpha}{1-\alpha}\mathbb{E}_{q(\boldsymbol{x})q_{\boldsymbol{\phi}^*}(\boldsymbol{z}|\boldsymbol{x})}\big[-e^{-\alpha f_{\boldsymbol{\beta}^*}(\boldsymbol{x})}\nabla_{\boldsymbol{\theta}}\log\tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\big]+ \\
&\gamma_\alpha\mathbb{E}_{p(\boldsymbol{z})\delta(\boldsymbol{x}|G_{\boldsymbol{\theta}}(\boldsymbol{z}))}\Big[-[\nabla_{\boldsymbol{\theta}}G_{\boldsymbol{\theta}}(\boldsymbol{z})]e^{(1-\alpha)f_{\boldsymbol{\beta}^*}(\boldsymbol{x})}\big[\nabla_{\boldsymbol{x}}f_{\boldsymbol{\beta}^*}(\boldsymbol{x})\big]\Big],
\end{aligned}
\tag{15}
$$

which combines the gradient information from ML and adversarial learning with automatic weights related to both α and the GAN discriminator.

Finally in Step III, we use the zero-forcing reverse-KL-based adversarial learning ($\alpha = 1$) to continually refine the generator parameters $\boldsymbol{\theta}$ and the discriminator parameters $\boldsymbol{\phi}$ using (7) and (4), respectively. The corresponding training process is summarized in Algorithm 1.

3.3 Connections to Prior GAN-Learning Regularization

Considering the aforementioned twin gradients and the α-Bridge, we next present an interpretation of the gradient in (15), with which we reveal two generalizations that are highly related to CycleGAN (Zhu et al. 2017) and ALICE (Li et al. 2017). Details are given in Appendix C. With $\boxed{\boldsymbol{x}}$ denoting the stop-gradient operator[4], the gradient in (15) can be reformulated as

$$
\begin{aligned}
&\nabla_{\boldsymbol{\theta}}\mathcal{D}_\alpha[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})] \approx \\
&\nabla_{\boldsymbol{\theta}}\left[\begin{aligned}&\frac{1-\gamma_\alpha}{1-\alpha}\mathbb{E}_{q(\boldsymbol{x})q_{\boldsymbol{\phi}^*}(\boldsymbol{z}|\boldsymbol{x})}\Big[\frac{e^{-\alpha f_{\boldsymbol{\beta}^*}(\boldsymbol{x})}}{2\sigma^2}\|\boldsymbol{x}-G_{\boldsymbol{\theta}}(\boldsymbol{z})\|_2^2\Big]\\ &+\gamma_\alpha\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}\big[-e^{(1-\alpha)f_{\boldsymbol{\beta}^*}(\boxed{\boldsymbol{x}})}f_{\boldsymbol{\beta}^*}(\boldsymbol{x})\big]\end{aligned}\right],
\end{aligned}
\tag{16}
$$

where the first term can be interpreted as weighted half cycle-consistency (Li et al. 2017; Zhu et al. 2017; Kim et al. 2017), and the second one is related to the reverse-KL-based adversarial learning. Based on the interpretation in (16), one can readily verify (see Appendix C) that by generalizing the α-Bridge derivations as in (16) to consider both marginals

$$
\begin{aligned}
&\mathcal{D}_\alpha[p_{\boldsymbol{\theta}}(\boldsymbol{x})\|q(\boldsymbol{x})]+\mathcal{D}_\alpha[q_{\boldsymbol{\phi}}(\boldsymbol{z})\|p(\boldsymbol{z})] \\
&\overset{\nabla}{\approx}\left[\begin{aligned}&\frac{1-\gamma_\alpha}{1-\alpha}\mathbb{E}_{q(\boldsymbol{x})q_{\boxed{\boldsymbol{\phi}}}(\boldsymbol{z}|\boldsymbol{x})}\Big[\frac{e^{-\alpha f_{\boldsymbol{\beta}^*}(\boldsymbol{x})}}{2\sigma^2}\|\boldsymbol{x}-G_{\boldsymbol{\theta}}(\boldsymbol{z})\|_2^2\Big]\\ &+\frac{1-\gamma_\alpha}{1-\alpha}\mathbb{E}_{p(\boldsymbol{z})p_{\boxed{\boldsymbol{\theta}}}(\boldsymbol{x}|\boldsymbol{z})}\Big[\frac{e^{-\alpha g_{\boldsymbol{\gamma}^*}(\boldsymbol{z})}}{2\sigma^2}\|\boldsymbol{z}-E_{\boldsymbol{\phi}}(\boldsymbol{x})\|_2^2\Big]\\ &+\gamma_\alpha\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}\Big[-e^{(1-\alpha)f_{\boldsymbol{\beta}^*}(\boxed{\boldsymbol{x}})}f_{\boldsymbol{\beta}^*}(\boldsymbol{x})\Big]\\ &+\gamma_\alpha\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\Big[-e^{(1-\alpha)g_{\boldsymbol{\gamma}^*}(\boxed{\boldsymbol{z}})}g_{\boldsymbol{\gamma}^*}(\boldsymbol{z})\Big]\end{aligned}\right],
\end{aligned}
$$

where $\overset{\nabla}{\approx}$ means both sides have approximately equal gradients mimicking (16), and $g_{\boldsymbol{\gamma}^*}(\boldsymbol{z}) = \log p(\boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z})$ corresponds to the optimal discriminator in the $\boldsymbol{z}$ space. Similarly, by considering both joint distributions

$$
\begin{aligned}
&\mathcal{D}_\alpha[p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\|q_{\boxed{\boldsymbol{\phi}}}(\boldsymbol{x},\boldsymbol{z})]+\mathcal{D}_\alpha[q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})\|p_{\boxed{\boldsymbol{\theta}}}(\boldsymbol{x},\boldsymbol{z})] \\
&\overset{\nabla}{\approx}\left[\begin{aligned}&\frac{1-\gamma_\alpha}{1-\alpha}\mathbb{E}_{q_{\boxed{\boldsymbol{\phi}}}(\boldsymbol{x},\boldsymbol{z})}\big[-e^{-\alpha h_{\boldsymbol{\eta}^*}(\boldsymbol{x},\boldsymbol{z})}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\big]\\ &+\frac{1-\gamma_\alpha}{1-\alpha}\mathbb{E}_{p_{\boxed{\boldsymbol{\theta}}}(\boldsymbol{x},\boldsymbol{z})}\Big[-e^{\alpha h_{\boldsymbol{\eta}^*}(\boldsymbol{x},\boldsymbol{z})}\log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\Big]\\ &+\gamma_\alpha\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}\big[-e^{(1-\alpha)h_{\boldsymbol{\eta}^*}(\boxed{\boldsymbol{x}},\boldsymbol{z})}h_{\boldsymbol{\eta}^*}(\boldsymbol{x},\boldsymbol{z})\big]\\ &+\gamma_\alpha\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})}\big[e^{-(1-\alpha)h_{\boldsymbol{\eta}^*}(\boldsymbol{x},\boxed{\boldsymbol{z}})}h_{\boldsymbol{\eta}^*}(\boldsymbol{x},\boldsymbol{z})\big]\end{aligned}\right],
\end{aligned}
$$

one can generalize the α-Bridge to a model very much resembling ALICE (Li et al. 2017). We believe that the connections revealed above, and the techniques developed earlier, may be helpful for constituting a foundation that unifies ML learning, adversarial learning, and intuitive (regularization) properties like cycle-consistency.

4 Related Work

Motivated by the complementary properties of ML and adversarial learning, many methods have been considered for combining these two popular research fields, to unify their advantages. A direct combination of the VAE with GAN objectives was considered in (Larsen et al. 2015), only to "observe the devil in the details" during model development and training. Accordingly gradients were heuristically controlled in back-propagation. It is also stated in (Mathieu et al. 2016) that naively combining those two objectives unstabilizes the system and does not lead to perceptually better generation, which is consistent with the empirical results from (Zhang et al. 2019). The principle combination of ML and adversarial learning deserves a thorough exploration. Instead of directly combining their objectives, the α-Bridge dynamically transfers (information) between both sides to bypass the unstable problem. Many other works combining ML and adversarial learning were motivated differently. On the one hand, with the target of ML learning unchanged, (Makhzani et al. 2015; Mescheder, Nowozin, and Geiger 2017) exploited GAN techniques to better handle the KL term between the prior and posterior of the latent variables, within the ELBO. On the other hand, keeping the target of adversarial learning, a variational auto-encoder/autoencoder was used as a building block within GAN discriminators, mainly for stabilizing training (Berthelot, Schumm, and Metz 2017; Ulyanov, Vedaldi, and Lempitsky 2018). A symmetric KL divergence was exploited to build objectives (Pu et al. 2017; Chen et al. 2018). Since those methods employed discriminators to estimate/replace the ratios within both the forward and reverse KL, the likelihood (gradient) information from forward KL was ignored. By comparison, the α-Bridge has the advantage of benefiting from the gradient information from ML learning. Although combining ML and adversarial learning is enticing, no previous work has achieved this in a principled manner. The proposed α-Bridge seeks to fill this gap.

[4] tf.stop_gradient/torch.no_grad in TensorFlow/PyTorch.

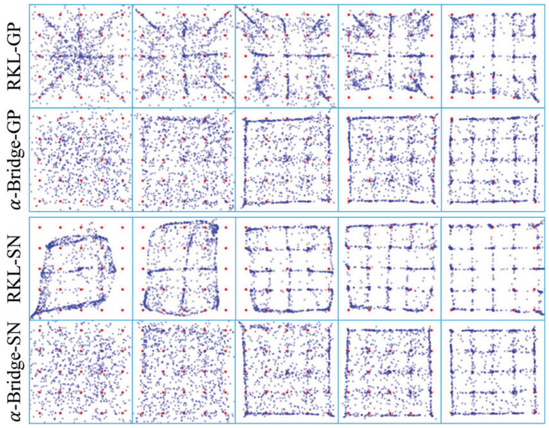


Figure 3: Demonstration on the dynamic evolution of the generated samples (blue) from the compared methods during training. Columns correspond to 1K, 2K, 4K, 6K, and 10K iterations. Real data samples are shown in red.

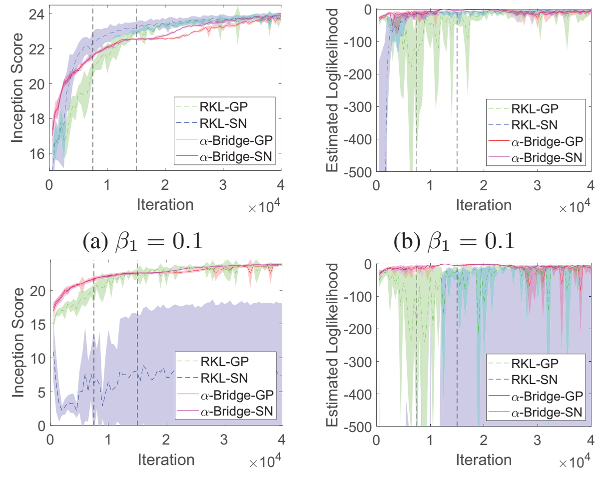
5 Experiments

We demonstrate the proposed α-Bridge from three perspectives. First we show that the α-Bridge, dynamically transferring advantages from ML to adversarial learning, exhibits a more stable training with improved robustness to hyperparameters (this is expected because of the aforementioned discussions of control variants interpretation and connections to prior GAN regularization methods). We then show that the α-Bridge is capable of smoothly transferring the information learned during ML learning to adversarial learning, circumventing the forgetting issue shown in Figure 1. Finally we highlight the versatility of the α-Bridge, by showing its capability in transplanting the variational approximation within ML learning into an inference arm for adversarial learning. See Appendix G for the detailed experimental settings and the corresponding analysis/discussions.

5.1 Stability and Robustness

The 25-Gaussians example from (Tao et al. 2018) is adopted, where the data are generated from a 2D Gaussian mixture model with 25 components, as shown in Figure 3. For direct comparison, reverse-KL-based GANs are chosen as baselines, with recent techniques to stabilize their training, *i.e.,* gradient penalty (GP) (Mescheder, Geiger, and Nowozin 2018) and spectral normalization (SN) (Miyato et al. 2018). Note it is shown in (Lucic et al. 2018) that most GANs with the same budgets can reach similar performance with enough hyperparameter optimization and random restarts. Thus reverse-KL-based GANs further stabilized by GP/SN are considered as fairly good baselines (named as RKL-GP and RKL-SN, respectively)[5]. The inception score (IS) (Salimans et al. 2016) and the log-likelihood estimated with kernel density estimation (Parzen 1962) are used to measure the plausibility of generated samples and the data-mode-covering level of learned models, respectively. Baseline methods are carefully

[5]Empirically, RKL-GP and RKL-SN show comparable/better results than WGAN-GP (Gulrajani et al. 2017) on 25-Gaussians.



(a) $\beta_1 = 0.1$ (b) $\beta_1 = 0.1$

(c) $\beta_1 = 0.5$ (d) $\beta_1 = 0.5$

Figure 4: Quantitative performance of the methods along the training process. Inception score (a)/(c) and estimated log-likelihood (b)/(d) when Adam (Kingma and Ba 2014) hyperparameter $\beta_1 = 0.1$ / $\beta_1 = 0.5$. Higher is better for both metrics. The curves are calculated over 10 random trials. Two vertical dashed lines are used to indicate the three steps of the α-Bridge.

tuned with their best settings adopted for fair comparison (see Appendix G.2).

Figure 4 shows the results of the considered methods. It is clear that α-Bridge, thanks to its smooth transferring nature, is capable of benefiting from the advantages of ML learning, resulting in more stable training (see Figures 4a and 4c) while keeping most data modes covered (see Figures 4b and 4d). Comparing Figures 4a-4b to Figures 4c-4d shows α-Bridge is relatively more robust to hyperparameters than baseline methods (see Appendix G.2 for more details). Figure 3 shows one training curve of the compared methods, highlighting α-Bridge's ability to benefit from the advantages of ML learning. To address the concern of how α-Bridge performs on real datasets, we conduct another experiment on CIFAR10 (Krizhevsky and Hinton 2009) and observe an improved performance of (IS, FID(Heusel et al. 2017))=$(7.225, 28.083)$ over $(6.558, 33.707)$ of the vanilla DCGAN baseline (see Appendix G.4 for more results).

5.2 Smooth Transfer of Information from ML to Adversarial Learning

Besides inheriting the advantages of ML learning, another advantage of the α-Bridge is a smooth transfer of the information learned during ML to adversarial learning. For an explicit demonstration, we run α-Bridge on the MNIST (LeCun et al. 1998) and CelebA (Liu et al. 2015) datasets, and present the generated samples along the training process, as shown in Figure 5. ML learning, *i.e.,* Step I of Algorithm 1, provides fairly good initialization on both datasets; thanks to the zero-avoiding nature of ML learning, one might anticipate an initialization covering all data modes, similar to the phenomena observed in Figure 3. When it comes to the trans-

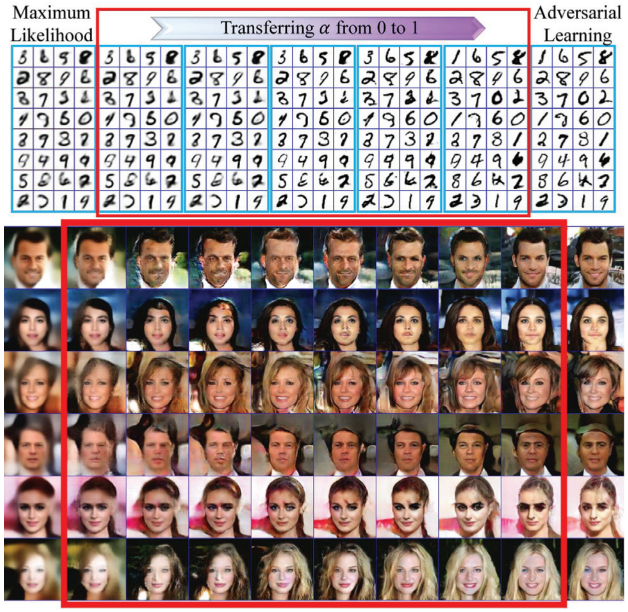


Figure 5: Random samples generated along the training process of the α-bridge on MNIST (top) and CelebA (bottom). Note most information is transferred from ML to adversarial learning, such as the class, rotation, and style of generated digits, and the basic tone, gender, expression, pose of the head, hair style of generated faces.

ferring Step II, α-Bridge smoothly inherits what's learned in ML learning at the beginning, and then gradually adds more detailed information (such as sharper edges on MNIST digits and clearer background on CelebA faces) to generate increasingly realistic images. After the transferring Step II, one observes generated images exhibiting the features from adversarial learning, whose image quality is further refined by the adversarial Step III of Algorithm 1. By reviewing the whole process shown in Figure 5, one observes that α-Bridge is capable of smoothly transferring most information from ML to adversarial learning, in contrast to the forgetting shown in Figure 1.

5.3 Transplanting ML Variational Posterior into Inference Arm for Adversarial Learning

In addition to inheriting the advantages and information from ML learning, we find that the smooth dynamical training of Algorithm 1 also enables α-Bridge to transplant the variational approximation within ML learning into an inference arm for adversarial learning. Such a capacity is appealing because it enables exploiting the generative power of GANs for various practical applications. See Appendix G.6 for technical details.

To verify the effectiveness of the transplanted inference arm, Figure 6 (top) shows the encoder-decoder reconstruction for the generated fake images. It is apparent that the reconstructions are fairly good, confirming the effectiveness of the inference arm. One can also exploit that arm for manipulation of GAN generated images, as shown in Figure 6

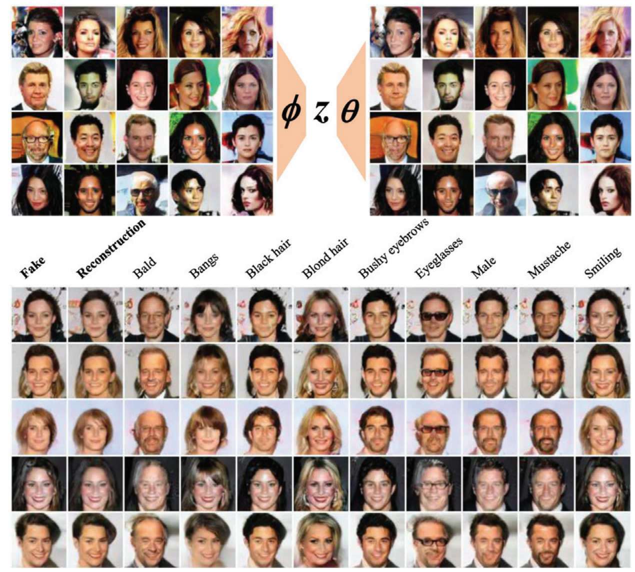


Figure 6: Using the inference arm transplanted by α-Bridge to reconstruct (top) and manipulate (bottom) GAN generated images. $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ denote the inference arm $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and the generator $G_{\boldsymbol{\theta}}(\boldsymbol{z})$, respectively.

(bottom). Detailed implementations for reconstruction and manipulation are given in Appendix G.6. It is clear that with this inference arm, one can modify the semantic concepts of the generated images like bangs, hair, gender, *etc.* Such capacity is valuable for transferring the generative power of GANs to various down-steaming tasks.

6 Conclusions

Motivated by the fact that maximum likelihood (ML) and adversarial learning have complementary characteristics, we have proposed a novel α-Bridge, constituted via the α-divergence, to unify their advantages in a principled manner. Our α-Bridge has as its foundation newly recognized twin gradients of the α-divergence, one of which utilizes the gradient information from the ML (forward KL) perspective, and the other from the adversarial learning (reverse KL) perspective. We also have revealed two generalizations of α-Bridge that closely resemble CycleGAN (Zhu et al. 2017) and ALICE (Li et al. 2017).

Acknowledgments. The research was supported in part by DARPA, DOE, NIH, NSF and ONR. The Titan Xp GPU used was donated by the NVIDIA Corporation.

References

Amari, S. 2009. α-divergence is unique, belonging to both f-divergence and bregman divergence classes. *IEEE Transactions on Information Theory* 55(11):4925–4931.

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

Berthelot, D.; Schumm, T.; and Metz, L. 2017. BEGAN: boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*.

Bishop, C. 2006. *Pattern Recognition and Machine Learning*. Springer.

Blei, D. M., and Jordan, M. I. 2006. Variational inference for Dirichlet process mixtures. *Bayesian analysis* 1(1):121–143.

Blei, D.; Kucukelbir, A.; and McAuliffe, J. 2017. Variational inference: A review for statisticians. *JASA* 112(518):859–877.

Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*.

Chen, L.; Dai, S.; Pu, Y.; Zhou, E.; Li, C.; Su, Q.; Chen, C.; and Carin, L. 2018. Symmetric variational autoencoder and connections to adversarial learning. In *AISTATS*, 661–669.

Cichocki, A., and Amari, S. 2010. Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities. *Entropy* 12(6):1532–1568.

Cong, Y.; Zhao, M.; Bai, K.; and Carin, L. 2019. GO gradient for expectation-based objectives. In *ICLR*.

Dai, B., and Wipf, D. 2019. Diagnosing and enhancing VAE models. In *ICLR*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved training of wasserstein GANs. In *NIPS*, 5767–5777.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NIPS*, 6626–6637.

Jordan, M.; Ghahramani, Z.; Jaakkola, T.; and Saul, L. 1999. An introduction to variational methods for graphical models. *Machine learning* 37(2):183–233.

Kim, T.; Cha, M.; Kim, H.; Lee, J.; and Kim, J. 2017. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 1857–1865.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *PNAS* 201611835.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Larsen, A.; Sønderby, S.; Larochelle, H.; and Winther, O. 2015. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Li, C.; Liu, H.; Chen, C.; Pu, Y.; Chen, L.; Henao, R.; and Carin, L. 2017. Alice: Towards understanding adversarial learning for joint distribution matching. In *NIPS*, 5501–5509.

Li, C.; Bai, K.; Li, J.; Wang, G.; Chen, C.; and Carin, L. 2019. Adversarial learning of a sampler based on an unnormalized distribution. In *AISTATS*.

Liang, K.; Li, C.; Wang, G.; and Carin, L. 2018. Generative adversarial network training is a continual learning problem. *arXiv preprint arXiv:1811.11083*.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In *ICCV*, 3730–3738.

Lucic, M.; Kurach, K.; Michalski, M.; Gelly, S.; and Bousquet, O. 2018. Are GANs created equal? a large-scale study. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *NeurIPS*. Curran Associates, Inc. 700–709.

Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; and Frey, B. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Mathieu, M.; Zhao, J.; Zhao, J.; Ramesh, A.; Sprechmann, P.; and LeCun, Y. 2016. Disentangling factors of variation in deep representation using adversarial training. In *NIPS*, 5040–5048.

McLachlan, G., and Krishnan, T. 2007. *The EM algorithm and extensions*, volume 382. John Wiley & Sons.

Mescheder, L.; Geiger, A.; and Nowozin, S. 2018. Which training methods for GANs do actually converge? In *ICML*, 3478–3487.

Mescheder, L.; Nowozin, S.; and Geiger, A. 2017. Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *ICML*, 2391–2400. JMLR. org.

Minka, T., et al. 2005. Divergence measures and message passing. Technical report, Technical report, Microsoft Research.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Nguyen, T.; Le, T.; Vu, H.; and Phung, D. 2017. Dual discriminator generative adversarial nets. In *NIPS*, 2670–2680.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-GAN: Training generative neural samplers using variational divergence minimization. In *NIPS*, 271–279.

Parzen, E. 1962. On estimation of a probability density function and mode. *The annals of mathematical statistics* 33(3):1065–1076.

Pu, Y.; Wang, W.; Henao, R.; Chen, L.; Gan, Z.; Li, C.; and Carin, L. 2017. Adversarial symmetric variational autoencoder. In *NIPS*, 4330–4339.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *NIPS*, 2234–2242.

Srivastava, A.; Valkov, L.; Russell, C.; Gutmann, M. U.; and Sutton, C. 2017. Veegan: Reducing mode collapse in GANs using implicit variational learning. In *NIPS*, 3308–3318.

Tao, C.; Chen, L.; Henao, R.; Feng, J.; and Carin, L. 2018. Chi-square generative adversarial network. In *ICML*, 4894–4903.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2018. It takes (only) two: Adversarial generator-encoder networks. In *AAAI*.

Yin, M., and Zhou, M. 2018. Semi-implicit variational inference. *arXiv preprint arXiv:1805.11183*.

Zhang, H.; Goodfellow, I.; Metaxas, D.; and Odena, A. 2018. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*.

Zhang, M.; Bird, T.; Habib, R.; Xu, T.; and Barber, D. 2019. Training generative latent models by variational f-divergence minimization.

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