STEIN BRIDGING: ENABLING MUTUAL REINFORCEMENT BETWEEN EXPLICIT AND IMPLICIT GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep generative models are generally categorized into explicit models and implicit models. The former defines an explicit density form, whose normalizing constant is often unknown; while the latter, including generative adversarial networks (GANs), generates samples without explicitly defining a density function. In spite of substantial recent advances demonstrating the power of the two classes of generative models in many applications, both of them, when used alone, suffer from respective limitations and drawbacks. To mitigate these issues, we propose Stein Bridging, a novel joint training framework that connects an explicit density estimator and an implicit sample generator with Stein discrepancy. We show that the Stein Bridge induces new regularization schemes for both explicit and implicit models. Convergence analysis and extensive experiments demonstrate that the Stein Bridging i) improves the stability and sample quality of the GAN training, and ii) facilitates the density estimator to seek more modes in data and alleviate the mode-collapse issue. Additionally, we discuss several applications of Stein Bridging and useful tricks in practical implementation used in our experiments.

1 INTRODUCTION

Deep generative model, as a powerful unsupervised framework for learning the distribution of high-dimensional multi-modal data, has been extensively studied in recent literature. Typically, there are two types of generative models (Goodfellow et al., 2014). Explicit models define an explicit (unnormalized) density function, while implicit models learn to sample from the distribution without explicitly define a density function.

Explicit models have wide applications in undirected graphical models (LeCun et al., 2006; Salakhutdinov & Hinton, 2009; Hinton et al., 2006; Ngiam et al., 2011), random graph theory (Robins et al., 2007), energy-based reinforcement learning (Haarnoja et al., 2017), etc. However, the unknown normalizing constant makes the model hard to train and sample from, and the explicit models might not be able to capture the complex structure of true samples while maintaining tractability. In contrast, implicit models are more flexible in training and easy to sample from, and in particular, generative adversarial networks (GANs) have shown great power in learning representations of images, natural languages, graphs, etc. (Goodfellow et al., 2014; Radford et al., 2016; Arjovsky et al., 2017; Brock et al., 2019). Nevertheless, due to the minimax game between generator and discriminator/critic in GANs, the training process often suffers from instability, and produces undesirable samples often associated with missing modes in data or generating extra modes out of data. More discussion on related work is in Appendix A.

There are situations where we need both an explicit density and a flexible implicit sampler. For sample evaluation, it is not enough to merely distinguish samples from real to faked one, and one may also expect to provide fine-grained evaluation on generated samples, where the energy values given by the explicit models can be a good metric (Dai et al., 2017). Another situation is outlier detection. Implicit models often leverage all true samples (possibly mixed with corrupted samples) as true examples for training. To make up for the issue, explicit models could help to detect out-of-distribution samples via the estimated densities (Zhai et al., 2016). Also, when given insufficient observed samples, explicit models may fail to capture an accurate distribution, in which case implicit model may help with data augmentation and facilitate training for density estimation. These situations motivate...
us to combine both of the worlds in an effective way so as to make the two models compensate and reinforce each other.

In this work, we aim at jointly learning explicit and implicit generative models. In our framework, an implicit energy model is used to estimate the unnormalized densities of true samples via minimizing a Stein discrepancy; in the meantime, an explicit generator model is exploited to minimize the Wasserstein metric (or Jensen-Shannon divergence) between distributions of true and generated samples. On top of these, another Stein discrepancy, acting as a bridge between implicit generated samples and explicit estimated densities, is introduced and pushes the two models to achieve a consensus. We show that the Stein bridge allows the two generative models to reinforce each other by imposing new regularizations on both models, which help the generator to output high-quality samples and facilitate the energy model to avoid mode-collapse. Moreover, we show that the joint training helps to stabilize GAN training via a convergence analysis. Extensive experiments on various tasks verify our theoretical findings as well as demonstrate the superiority of proposed methods compared with existing deep energy models and GAN-based models.

2 BACKGROUND

In this section, we briefly provide some technical background used in our model.

Energy Model. The energy model assigns each data $x$ with a scalar energy value $E_\phi(x)$, where $E_\phi(\cdot)$ is called the energy function and is parameterized by $\phi$. The model is expected to assign low energy to true samples according to a Gibbs distribution $p_\phi(x) = \exp\{-E_\phi(x)/Z_\phi\}$, where $Z_\phi$ is a normalizing constant dependent on $\phi$. The normalizing term $Z_\phi$ is often hard to compute, making the training intractable, and various methods are proposed to detour such term (see Appendix A).

Stein Discrepancy. Stein discrepancy [Gorham & Mackey, 2015; Liu et al., 2016; Chwialkowski et al., 2016; Oates et al., 2017] is a measure of closeness between two probability distributions that does not require the knowledge for the normalizing constant of one of the compared distributions. Let $P$ and $Q$ be two probability distributions on $X \subset \mathbb{R}^d$, and assume $Q$ has a density $q$. The Stein discrepancy $S(P, Q)$ is defined as

$$S(P, Q) := \sup_{f \in \mathcal{F}} \mathbb{E}_{x \sim P} [A_\phi f(x)] := \sup_{f \in \mathcal{F}} \left\{ \mathbb{E}_{x \sim P} [\text{tr} (\nabla_x \log q(x) f(x)^T + \nabla_x f(x))] \right\},$$

where $\mathcal{F}$ is often chosen to be a Stein class (see, e.g., Definition 2.1 in Liu et al. (2016)), or a unit ball in some reproducing kernel Hilbert space (RKHS) with a positive definite kernel $k$. The latter is referred to as Kernel Stein Discrepancy (KSD). We refer to Appendix B for more details.

Wasserstein Metric. Wasserstein metric is suitable for measuring distances between two distributions with non-overlapping supports [Arjovsky et al., 2017]. The Wasserstein-1 metric between distributions $P$ and $Q$ is defined as

$$\mathcal{W}(P, Q) := \min_{\gamma \sim \mathcal{P}(P \times Q)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|],$$

where the maximization is over all joint distributions with marginals $P$ and $Q$. By Kantorovich–Rubinstein duality, it has a dual representation

$$\mathcal{W}(P, Q) := \max_D \left\{ \mathbb{E}_{x \sim P} [D(x)] - \mathbb{E}_{y \sim Q} [D(y)] \right\},$$

where the maximization is over all 1-Lipschitz continuous functions.

Sobolev space and Sobolev dual norm. Use $L^2$ to denote the canonical Hilbert space on $\mathbb{R}^d$ equipped with an inner product $\langle u, v \rangle_{L^2} := \int_{\mathbb{R}^d} u v dx$. The Sobolev space $H^1$ is defined as the closure of $C_0^\infty$, the set of smooth functions on $\mathbb{R}^d$ with compact support, with respect to the norm $\|u\|_{H^1} := (\int_{\mathbb{R}^d} (u^2 + \|\nabla u\|^2_2) dx)^{1/2}$. For $v \in L^2$, its Sobolev dual norm $\|v\|_{H^{-1}}$ is defined by [Evans & Society, 2010]

$$\|v\|_{H^{-1}} := \sup_{u \in H^1} \left\{ \langle v, u \rangle_{L^2} : \int_{\mathbb{R}^d} \|\nabla u\|^2_2 dx \leq 1 \right\}.$$

Therefore, $\|\cdot\|_{H^{-1}}$ can be viewed as a measure of smoothness, which measures the similarity (in terms of largest $L^2$-norm) between $v$ and a subset of smooth functions in $H^1$.

3 PROPOSED MODEL

In this section, we formulate our model, Stein Bridging, and highlight its regularization effects.

}\end{document}
3.1 MODEL FORMULATION

We denote by $\mathbb{P}_{\text{real}}$ the underlying real distribution from which the data $\{x\}$ are sampled. We simultaneously learn two generative models – one explicit and one implicit – that represent estimates of $\mathbb{P}_{\text{real}}$. The explicit generative model has an explicit probability density $\mathbb{P}_E$ proportional to $\exp(-E(x))$, where $E$ is referred to as an energy function. The implicit generative model transforms an easy-to-sample random noise $z$ with distribution $\mathbb{P}_0$ via a generator $G$ to a generated sample $\tilde{x} = G(z)$ with distribution $\mathbb{P}_G$. We use the Stein discrepancy as a measure of closeness between the explicit density $\mathbb{P}_E$ and the real distribution $\mathbb{P}_{\text{real}}$, and use the Wasserstein metric as a measure of closeness between the implicit distribution $\mathbb{P}_G$ and $\mathbb{P}_{\text{real}}$.

To jointly learn the two generative models $\mathbb{P}_G$ and $\mathbb{P}_E$, arguably the most straightforward approach is to minimize the sum of the Stein discrepancy and the Wasserstein metric:

$$
\min_{E,G} \mathcal{W}(\mathbb{P}_{\text{real}}, \mathbb{P}_G) + \lambda S(\mathbb{P}_{\text{real}}, \mathbb{P}_E),
$$

where $\lambda \geq 0$ is a weight coefficient. However, this approach appears no different than learning the two generative models separately. To better train the model, we incorporate the objective another term $S(\mathbb{P}_G, \mathbb{P}_E)$ – called the Stein bridge – that measures the closeness between the explicit density $\mathbb{P}_E$ and the implicit distribution $\mathbb{P}_G$:

$$
\min_{E,G} \mathcal{W}(\mathbb{P}_{\text{real}}, \mathbb{P}_G) + \lambda_1 S(\mathbb{P}_{\text{real}}, \mathbb{P}_E) + \lambda_2 S(\mathbb{P}_G, \mathbb{P}_E),
$$

where $\lambda_1, \lambda_2 \geq 0$ are weight coefficients. Although the Stein bridge might seem redundant mathematically, we show that it helps regularize the models in Section 3.2.

The Wasserstein term in (3) is implemented using its equivalent dual representation (2). The two Stein terms in (3) can be implemented using (1) with either a Stein critic parameterized by a neural network, or the Kernel Stein Discrepancy. To reduce the computational cost, the two Stein critics share their parameters, namely, kernels or neural networks. A scheme of our framework is presented in Fig. 1. We also discuss some related works that attempt to combine both of the worlds (such as energy-based GAN, contrastive learning and cooperative learning) in Appendix A.3 and highlight the difference between our method and theirs in terms of the objective in Table 1.

Remark. In general, we can also choose other statistical distances in (3) to measure closeness between probability distributions. For example, the Wasserstein metric $\mathcal{W}(\mathbb{P}_{\text{real}}, \mathbb{P}_G)$ can be replaced by other common choices for implicit generative models, such as Jensen-Shannon divergence used in the original GAN paper (Goodfellow et al., 2014). If the normalizing constant of $\mathbb{P}_E$ is known or easy to calculate, one can replace the Stein discrepancy by the Kullback-Leibler divergence, which is equivalent to the maximum likelihood estimation. We present details for model specifications in various forms and training algorithm in Appendix E.2.

![Figure 1: Model framework for Stein Bridging which jointly train an implicit sample generator and an explicit density estimator via a Stein bridge.](image)
3.2 Regularization Effects by Virtue of The Stein Bridge

The intuitive motivation of the Stein bridge term in (3) is to push the two models to achieve a consensus. In this subsection, we theoretically show that the Stein bridge allows the two models to reinforce each other by imposing regularizations on the critics.

3.2.1 Kernel Sobolev dual norm regularization on the Wasserstein critic

We first show the regularization effect of the Stein bridge on the Wasserstein critic. Fixing the energy function $E$, consider the max-min problem over the Wasserstein critic $D$ and the generator $G$:

$$\max_D \min_G \{ E_{\mathbb{Y} \sim \mathbb{P}_G} [D(y)] - E_{\mathbb{X} \sim \mathbb{P}_{\text{real}}} [D(x)] + \lambda_2 S(\mathbb{P}_G, \mathbb{P}_E) \}$$

We define the kernel Sobolev dual norm as

$$\| D \|_{H^{-1}(\mathbb{P}; k)} := \sup_{u \in C_0} \{ \langle D, u \rangle_{L^2(\mathbb{P})} : E_{\mathbb{X}, \mathbb{X}' \sim \mathbb{P}} [\nabla u(x) \cdot \nabla u(x')] \leq 1 \},$$

which can be viewed as a kernel generalization of the Sobolev dual norm defined in Section 2, which reduces to the Sobolev dual norm when $k(x, x') = \mathbb{I}(x = x')$ and $\mathbb{P}$ being the Lebesgue measure.

Assuming that $\{ \mathbb{P}_G \}_G$ exhausts all probability distributions, we have the following result.

**Theorem 1.** Formally, problem (4) is equivalent to

$$\max_D \left\{ E_{\mathbb{Y} \sim \mathbb{P}_G} [D(y)] - E_{\mathbb{X} \sim \mathbb{P}_{\text{real}}} [D(x)] - \frac{1}{2\lambda_2} \inf_{t \in \mathbb{R}} \| D - t \|_{H^{-1}(\mathbb{P}; k)} + O(1/\lambda_2^2) \right\} + O(1)$$

Note that if $D$ is an optimal Wasserstein critic in (2), so does $D - r$, $r \in \mathbb{R}$. This is consistent to the penalty term $\inf_{t \in \mathbb{R}} \| D - t \|_{H^{-1}(\mathbb{P}; k)}$, since the penalties are identical for $D$ and $D - r$. According to Section 2, the regularization term would penalize the non-smoothness of the Wasserstein critic $D$, which is in the same spirit of gradient-based penalty (e.g., Gulrajani et al. 2017; Roth et al. 2017), but with a new way to encouraging smoothness.

Another way to interpret the Sobolev dual norm penalty is by observing that if $k(x, x') = \mathbb{I}(x = x')$, and $E_{\mathbb{P}_E} [D - t] = 0$ (Villani 2008), then

$$\| D - t \|_{H^{-1}(\mathbb{P}_E; k)} = \lim_{\epsilon \to 0} \frac{\mathcal{W}_2((1 + \epsilon(D - t))\mathbb{P}_E, \mathbb{P}_E)}{\epsilon},$$

where $\mathcal{W}_2$ denotes the 2-Wasserstein metric. Therefore, the regularization ensures that $D$ would not change suddenly on the high-density region of $\mathbb{P}_E$, and the explicit model reinforces the learning of the Wasserstein critic.

3.2.2 Lipschitz Regularization on the Stein critic

We next investigate how the Stein bridge helps to regularize the Stein critic. Recall that the two Stein terms in (3) share the same Stein critic. Fixing the energy function $E$, consider the max-min problem over the Stein critic $f$ and the generator $G$:

$$\max_f \min_G \{ \lambda_1 E_{\mathbb{Y} \sim \mathbb{P}_{\text{real}}} [A_{\mathbb{P}_E} f(y)] + \lambda_2 E_{\mathbb{Y} \sim \mathbb{P}_G} [A_{\mathbb{P}_E} f(y)] + \mathcal{W}(\mathbb{P}_{\text{real}}, \mathbb{P}_G) \}. \tag{5}$$

Assuming that $\{ \mathbb{P}_G \}_G$ exhausts all probability distributions, we have the following result.

**Theorem 2.** Problem (5) is equivalent to

$$\max_f \{ (\lambda_1 + \lambda_2) E_{\mathbb{Y} \sim \mathbb{P}_{\text{real}}} [A_{\mathbb{P}_E} f(y)] : \text{Lip}(A_{\mathbb{P}_E} f) \leq 1/\lambda_2 \},$$

where $\text{Lip}(A_{\mathbb{P}_E} f)$ denotes the Lipschitz constant of the function $A_{\mathbb{P}_E} f$.

Theorem 2 shows that the Stein bridge, together with the Wasserstein metric $\mathcal{W}(\mathbb{P}_{\text{real}}, \mathbb{P}_G)$, plays as a smoothness regularization on the Stein critic $f$ via the constraint $\text{Lip}(\text{tr}(f^T \nabla \log \mathbb{P}_E + \nabla f)) \leq 1/\lambda_2$. The regularization will penalize large variation of values given by Stein operators on adjacent instances and further encourage the energy model to seek more modes in data instead of focusing on some dominated modes, thus helping to alleviate the mode-collapse issue. To the best of our knowledge, this suggests a novel regularization scheme for Stein-based GAN.
4 Convergence Analysis

In Section 3.2, we justify Stein Bridging by showing the regularization effects. In this section, we further show that it could help to stabilize GAN training with local convergence guarantee. To this end, we first compare the behaviors of WGAN, likelihood- and entropy-regularized WGAN, and our Stein Bridging under SGD via an easy to comprehend toy example. Then we give a formal result that interprets why the introduction of density estimator could stabilize GAN training and help for convergence.

4.1 Analysis of a Linear System

The training for minimax game in GAN is difficult. When using traditional gradient methods, the training would suffer from some oscillatory behaviors [Goodfellow (2017); Liang & Stokes (2019)]. In order to better understand the optimization behaviors, we first study a one-dimensional linear system that provides some insights on this problem. Note that such toy example (or a similar one) is also utilized by Gidel et al. (2019) and Nagarajan & Kolter (2017) to shed lights on the instability of WGAN training.

Consider a linear critic $D_c(x) = \psi x$ and generator $G_\theta(z) = \theta z$. Then the Wasserstein GAN objective can be written as a constrained bilinear problem: $\min_{\theta, \psi} \max_{|\psi| \leq 1} \mathbb{E}[x] - \psi \mathbb{E}[z]$, which could be further simplified as an unconstrained version (the behaviors could be generalized to multi-dimensional cases [Gidel et al. (2019)]):

$$\min_{\theta} \max_{\psi} \psi - \psi \cdot \theta.$$  \hspace{1cm} (6)

Unfortunately, such simple objective cannot guarantee convergence by traditional gradient methods like SGD with alternate updating $\theta_{k+1} = \theta_k - \eta \psi_k$, $\psi_{k+1} = \psi_k + \eta \theta_{k+1}$. Such optimization would suffer from an oscillatory behavior, i.e., the updated parameters go around the optimum point ($[\theta^*, \psi^*] = [1, 0]$) forming a circle without converging to the centrality, which is shown in Fig. 2(a).

A recent study in Liang & Stokes (2019) theoretically show that such oscillation is due to the interaction term in (6).

One solution to the instability of GAN training is to add (likelihood) regularization, which has been widely studied by recent literatures [Warde-Farley & Bengio (2017); Li & Turner (2018)]. With regularization term, the objective changes into $\min_\theta \max_{|\psi| \leq 1} \psi \mathbb{E}[x] - \psi \mathbb{E}[z] - \lambda \mathbb{E} \log \mu(\theta z)$, where $\mu(\cdot)$ denotes the likelihood function and $\lambda$ is a hyperparameter. A recent study [Liao et al. (2019)] proves that when $\lambda < 0$ (likelihood-regularization), the extra term is equivalent to maximizing sample evidence, helping to stabilize GAN training; when $\lambda > 0$ (entropy-regularization), the extra term maximizes sample entropy, which encourages diversity of generator. Here we consider a Gaussian likelihood function for generated sample $x'$, $\mu(x') = \exp(-\frac{1}{2}(x'-b)^2)$ which is up to a constant, and then the objective becomes (see Appendix D.1 for details):

$$\min_{\theta} \max_{\psi} \psi - \psi \cdot \theta - \lambda (\theta^2 - \theta).$$  \hspace{1cm} (7)

![Figure 2: Numerical iterations for SGD training. (a) Comparison of WGAN, likelihood-regularized WGAN (WGAN+LR), variational annealing for WGAN+LR (WGAN+LR+VA), entropy-regularized WGAN (WGAN+ER) and our Stein Bridging. (b) Stein Bridging with different $\lambda_1$ and $\lambda_2$.](image)

The above system would converge with $\lambda < 0$ and diverge with $\lambda > 0$ in gradient-based optimization, shown in Fig. 2(a). Another issue of likelihood-regularization is that the extra term changes the optimum point and makes the model converge to a biased distribution, as proved by Tao et al. (2019). In this case, one can verify that the optimum point becomes $[\psi^*, \theta^*] = [-\lambda, 1]$, resulting a bias. To avoid this issue, Tao et al. (2019) proposes to temporally decrease $|\lambda|$ through training. However, such method would also be stuck in oscillation when $|\lambda|$ gets close to zero as is shown in Fig. 2(a).

1 Our theoretical discussions focus on WGAN, and we also compare with original GAN in the experiments.

2 Here, we adopt the most widely used alternate updating strategy. The simultaneous updating, i.e., $\theta_{k+1} = \theta_k - \eta \psi_k$ and $\psi_{k+1} = \psi_k + \eta \theta_k$, would diverge in this case.
Finally, let us consider our proposed model. We also simplify the density estimator as a basic energy model $p_\theta(x) = \exp(\frac{1}{2}x^2 + \phi x)$ whose score function is $\nabla_x \log p_\theta(x) = x + \phi$. Then if we specify the two Stein discrepancies in \cite{2019arXiv190612981C} as KSD, we have the objective,
\begin{equation}
\min_\theta \max_\psi \min_\phi \psi - \psi \cdot \theta + \frac{\lambda_1}{2} (1 + \phi)^2 + \frac{\lambda_2}{2} (\theta + \phi)^2.
\end{equation}

Interestingly, one can verify that for $\forall \lambda_1, \lambda_2$, the optimum point remains the same $[\psi^*, \theta^*, \phi^*] = [0, 1, -1]$. Then we show that the optimization can guarantee convergence to $[\psi^*, \theta^*, \phi^*]$.

**Proposition 1.** Using alternate SGD for (8) geometrically decreases the square norm $N_t = |\psi|^2 + |\theta - 1|^2 + |\phi + 1|^2$, for any $0 < \eta < 1$ with $\lambda_1 = \lambda_2 = 1$,
\begin{equation}
N_{t+1} = (1 - \eta^2(1 - \eta)^2)N_t.
\end{equation}

In Fig. 2a), we can see that Stein Bridging achieves a good convergence to the right optimum. Compared with \cite{2019arXiv190612981C}, the objective (8) adds a new bilinear term $\phi \cdot \theta$, which acts like a connection between the two generator and estimator, and two other quadratic terms, which help to push the values to decrease through training. The added terms and the original terms in (8) cooperate to guarantee convergence to a unique optimum. (More discussions in Appendix D.1).

We further generalize the analysis to multi-dimensional bilinear system $F(\psi, \theta) = \theta^\top A \psi - b^\top \theta - c^\top \psi$ which is extensively used by researchers for analysis of GAN stability \cite{Goodfellow2017, GempMahadevan18, LiangStokes19, Gidel2019}. For any bilinear system, with the added term $H(\phi, \theta) = \frac{1}{2}(\theta + \phi)^\top B (\theta + \phi)$ where $B = (AA^\top)^{\frac{1}{2}}$ to the objective, we can prove that i) the optimum point remains the same as the original system (Proposition 2) and ii) using alternate SGD algorithm for the new objective can guarantee convergence (Theorem 4). The results are given in Appendix D.3.

### 4.2 Local Convergence for a General Model

To study the convergence for Stein Bridging, we proceed to consider a general optimization objective
\begin{equation}
\min_\theta \max_\psi \min_\phi L(\theta, \psi, \phi),
\end{equation}

where \(L(\theta, \psi, \phi) = F(\theta, \psi) + H(\theta, \phi)\), and $\omega_f = [\theta, \psi]$ and $\omega_h = [\theta, \phi]$ (\(\theta\) is a shared parameter set). Use $\omega^* = [\theta^*, \psi^*, \phi^*]$ to denote the optimum point of $L$ and $\omega^*_f = [\theta^*, \psi^*], \omega^*_h = [\theta^*, \phi^*]$ represent the optimum points of $F$ and $H$ respectively. Define $\Omega_f = \Omega_\theta \times \Omega_\phi$ and $\Omega_h = \Omega_\theta \times \Omega_\phi$, where $\Omega_\theta, \Omega_\phi, \Omega_\phi$ denote constraint sets for $\theta, \psi, \phi$ respectively. Function $H$ is $\mu$-strongly convex, and $F$ is $\mu$-strongly convex for $\theta$ and $\mu$-strongly concave for $\psi$ (see Appendix E.4 for definition of strongly convex condition). Here we define $b(\omega_h) = \nabla_\theta H + \nabla_\phi H, f(\omega_f) = \nabla_\theta F - \nabla_\psi F$, and then we have the following theorem.

**Theorem 3.** If $F$ is $\mu$-strongly convex-concave and $H$ is $\mu$-strongly convex, we can leverage the alternate SGD algorithm, i.e.
\begin{equation}
\omega_{h}^{t+1} = P_{\Omega_h}(\omega_{h}^{t+1/2} - \eta b(\omega_{h}^{t+1/2})),
\end{equation}
\begin{equation}
\omega_{f}^{t+1} = P_{\Omega_f}(\omega_{f}^{t+1/2} - \eta f(\omega_{f}^{t+1/2})),
\end{equation}

where $\omega_{h}^{t+1/2} = [\theta^{t+1/2}, \psi^{t+1/2}], \omega_{h}^{t+1} = [\theta^{t+1/2}, \psi^{t+1}], \omega_{f}^{t+1/2} = [\theta^{t+1/2}, \phi^{t}], \omega_{f}^{t+1} = [\theta^{t+1}, \phi^{t+1}], \nabla_\theta H + \nabla_\phi H, f(\omega_f) = \nabla_\theta F - \nabla_\psi F$, and $P_{\Omega}(\omega) = \arg \min_{\omega \in \Omega} ||\omega - \omega^*||_2$ denotes the projection mapping to $\Omega$. Then we can achieve the convergence by using $\frac{1}{2\mu} < \eta < \frac{1}{\mu}$.

Theorem 3 shows that Stein Bridging could converge to at least a local optimum. Due to the unknown and intricate landscape of deep neural networks, the global optimization and convergence analysis for GAN has remained as an unexplored problem. Despite the fact that strong convexity assumption cannot be guaranteed with deep neural networks, the optimization could converge to a stable point once there exists a local region that satisfies the strongly convex conditions. In the experiments, we will empirically compare the training stability of each method on various datasets to validate our theoretical discussions.
5 Experiments

In this section, we conduct experiments to verify the effectiveness of proposed method from multi-faceted views. First, we select three tasks with different evaluation metrics in Section 5.1, 5.2 and 5.3. Then we further discuss some applications of joint training as well as some useful tricks in Section 5.4, 5.5 and 5.6. The codes will be released later.

We consider two synthetic datasets with mixtures of Gaussian distributions: Two-Circle and Two-Spiral. The first one is composed of 24 Gaussian mixtures that lie in two circles. Such dataset is extended from the 8-Gaussian-mixture scenario which is widely used in previous GAN papers and is more difficult, so that we can use it to test the quality of generated samples and mode coverage of learned density. The second synthetic dataset consists of 100 Gaussian mixtures whose centers are densely arranged on two centrally symmetrical spiral-shaped curves. This dataset can be used to examine the power of generative model on complicated data distributions. The ground-truth distributions and samples are shown in Fig. 3 (a) and Fig. 4 (a). Furthermore, we also apply the methods to MNIST and CIFAR datasets which require the model to deal with high-dimensional data. In each dataset, we use observed samples as input of the model and leverage them to train the generators and the estimators. The details for each dataset are reported in Appendix E.1.

In our experiments, we also replace the Wasserstein metric in (3) by JS divergence. To well distinguish different specifications, we term the model Joint-W if using Wasserstein metric and Joint-JS if using JS divergence in this section. We consider several competitors. First, for implicit generative models, we consider vanilla GAN, WGAN-GP (Gulrajani et al. (2017)), likelihood-regularized GAN/WGAN-GP (short as GAN+LR/WGAN+LR), entropy-regularized GAN/WGAN-GP (short as GAN+ER/WGAN+ER) and a recently proposed variational annealing regularization (Tao et al. (2019)) for GAN (short as GAN+VA/WGAN+VA) to compare the quality of generated samples. We employ the denoising auto-encoder to estimate the gradient for regularization penalty, which is proposed by Alain & Bengio (2014) and utilized by Tao et al. (2019). Second, for explicit density models, we consider Deep Energy Model (DEM) which is optimized based on Stein discrepancy, and energy-based GAN (EGAN) (Dai et al. (2017)). Besides, we also compare with Deep Directed Generative (DGM) Model (Kim & Bengio (2017)) which adopts contrastive divergence to unite sample generator and density estimator. See Appendix A for brief introduction of these methods and Appendix E.3 for implementation details for each method.

5.1 Sample Quality of Implicit Model

Calibrating explicit density model with implicit generator is expected to improve the quality of generated samples. In Fig. 3 and Fig. 4, we show the results of different generators in Two-Circle and Two-Spiral datasets. As we can see, in Two-Circle, there are a large number of generated samples...
given by GAN, WGAN-GP and DGM (the worst one in this case) locating between two Gaussian
components, and the boundary for each component is not distinguishable. Since the ground-truth
densities of regions between two components are very low, such generated samples possess low-
quality, which depicts that these models capture the combinations of two dominated features (i.e.,
models) in the data but such combination does not make sense in practice. By contrast, Joint-JS and
Joint-W could alleviate such issue, reduce the low-quality samples and produce more distinguish-
able boundaries for components. In Two-Spiral, similarly, the generated samples given by GAN
and WGAN-GP form a circle instead of two spirals while the samples of DGM ‘link’ two spirals.
Joint-JS manages to focus more on true high densities compared to GAN and Joint-W provides the
best results. To quantitatively measure the sample quality, we adopt two metrics: Maximum Mean
Discrepancy (MMD) and High-quality Sample Rate (HSR). The detailed definitions are given in
Appendix E.4 and we report the results in Table 5.

Figure 5: Learning curves of Joint-W (resp. Joint-JS) compared with WGAN (resp. GAN or DC-
GAN) and its regularization-based variants.

We visualize the generated digits/images on MNIST/CIFAR-10 datasets in Fig. 9 and Fig. 10 and
use Inception Score and conditional entropy of predicted classes (CEPC) to measure the sample
quality (See Appendix E.4 for details). As shown in Table 2 Joint-W (resp. Joint-JS) is superior
than WGAN-GP (resp. DCGAN), regularized WGAN (resp. DCGAN) and DGM. The CEPC char-
acterizes how well the picture can be distinguished by a pre-trained classifier, i.e., the quality of
picture, so the results depict that proposed method could give higher-quality generated pictures.

5.2 Density Estimation of Explicit Model

Another advantage of joint learning is that the generator could help the density estimator to capture
more accurate distribution. As shown in Fig 5 both Joint-JS and Joint-W manage to capture all
Gaussian components while other methods miss some of modes. In Fig 4 Joint-JS and Joint-W
exactly fit the ground-truth distribution. By contrast, DEM misses one spiral while EGAN degrades
to a uniform-like distribution. DGM manages to fit two spirals but allocate high densities to regions
that have low densities in the groung-truth distribution. To quantitatively measure the performance,
we introduce three evaluation metrics: KL & JS divergence between the ground-truth and estimated
densities and Area Under the Curve (AUC) for false-positive rate v.s. true-positive rate where we
select points with true high (resp. low) densities as positive (resp. negative) examples. The detailed
information and results are given in Appendix E.4 and Table 5 respectively. The values show that
Joint-W and Joint-JS could provide more accurate density estimation than other competitors.

We also rank the generated digits (and true digits) on MNIST w.r.t the densities given by the energy
model in Fig. [11] Fig. [12] and Fig. [13]. As depicted in the figures, the digits with high densities (or
low densities) given by Joint-JS possess enough diversity (the thickness, the inclination angles as

<table>
<thead>
<tr>
<th>Method</th>
<th>Score</th>
<th>CEPC</th>
<th>Method</th>
<th>Score</th>
<th>CEPC</th>
<th>Method</th>
<th>Score</th>
<th>CEPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCGAN</td>
<td>8.43</td>
<td>0.168</td>
<td>WGAN+GP</td>
<td>7.71</td>
<td>0.256</td>
<td>WGAN+GP</td>
<td>6.80</td>
<td>0.153</td>
</tr>
<tr>
<td>DCGAN+LR</td>
<td>8.40</td>
<td>0.171</td>
<td>WGAN+LR</td>
<td>7.82</td>
<td>0.243</td>
<td>WGAN+LR</td>
<td>6.89</td>
<td>0.154</td>
</tr>
<tr>
<td>DCGAN+ER</td>
<td>8.33</td>
<td>0.179</td>
<td>WGAN+ER</td>
<td>7.75</td>
<td>0.252</td>
<td>WGAN+ER</td>
<td>6.99</td>
<td>0.156</td>
</tr>
<tr>
<td>DCGAN+VA</td>
<td>8.40</td>
<td>0.172</td>
<td>WGAN+VA</td>
<td>7.74</td>
<td>0.254</td>
<td>WGAN+VA</td>
<td>6.95</td>
<td>0.154</td>
</tr>
<tr>
<td>DGM</td>
<td>8.15</td>
<td>0.201</td>
<td>DGM</td>
<td>6.87</td>
<td>0.372</td>
<td>DGM</td>
<td>4.79</td>
<td>0.146</td>
</tr>
<tr>
<td>Joint-JS(ours)</td>
<td><strong>8.53</strong></td>
<td><strong>0.156</strong></td>
<td>Joint-W(ours)</td>
<td><strong>7.90</strong></td>
<td><strong>0.231</strong></td>
<td>Joint-W(ours)</td>
<td><strong>7.11</strong></td>
<td><strong>0.151</strong></td>
</tr>
</tbody>
</table>

Table 2: Inception scores (higher is better) and conditional entropies (short as CEPC and lower is
better) on MNIST and CIFAR-10. † We directly use the best result reported in their paper.
well as the shapes of digits diverses). By constrast, all the digits with high densities given by DGM tend to be thin and digits with low densities are very thick. Also, as for EGAN, digits with high (or low) densities appear to have the same inclination angle (for high densities, ‘1’ keeps straight and ‘9’ ‘leans’ to the left while for low densities, just the opposite). Such phenomenon indicates that DGM and EGAN tend to allocate high (or low) densities to data with certain modes and would miss some modes that possibly possess high densities in ground-truth distributions. Fortunately, our method overcomes the issue and manages to capture complicated distributions.

5.3 Enhancing the Stability of GAN

Our discussions and analysis show that joint training helps to stabilize GAN training. In Fig. 5 we present the learning curves of Joint-W (resp. Joint-JS) compared with WGAN (resp. GAN or DCGAN) and its regularization-based variants on different datasets. One can clearly see from the curves that joint training could reduce the variance of metric values especially during the second half of training. Furthermore, we visualize the generated pictures given by the same noise \( z \) in adjacent epoches in Fig. 6. The results show that Joint-W outputs more stable generation in adjacent epoches while the generated samples given by WGAN-GP and WGAN+VA exhibit an obvious variation. Especially, some digits generated by WGAN-GP and WGAN+VA change from one class to another. Such phenomenon is quite similar to the oscillatory behavior with non-convergence in optimization that we discuss in Section 4.1.

Another issue discussed in Section 4.1 is the bias of model distribution for regularized GAN methods. To quantify this evaluation, we calculate \( l_1 \) and \( l_2 \) distances between the means of 50,000 generated digits (resp. images) and 50,000 true digits (resp. images) in MNIST (reps. CIFAR-10). The results are shown in Table 3. The smaller distances given by Joint-W indicate that it converges to a better local optimum with smaller bias from the original data distribution. Also, in Table 6 (resp. Table 7), we report the distances for digits (resp. images) in each class on MNIST (resp. CIFAR).

5.4 Detecting Out-of-Distribution Samples

The explicit model estimates densities for each sample and one of its applications is to detect outliers in the input data. Here, we adopt CIFAR-10 to measure the ability of our estimator to distinguish the in-distribution samples and (true/false) out-of-distribution samples. We consider four situations and in each case, we consider the test images of CIFAR-10 as positive set (expected to allocate high densities) and construct a negative set (expected to allocate low densities). We let the model output densities for images in two sets, rank them according to the densities and plot the ROC curve for false-positive rate v.s. true-positive rate in Fig. 8. In the first case, we flip each image in the positive set as negative set. Note that such flipped images are not out-of-distribution samples, so the model is expected to allocate high densities to them, i.e., the ROC curve should be close to a straight line from \((0, 0)\) to \((1, 1)\). The results show that Joint-W, EGAN and DEM give the exact results while DGM assigns all flipped images with lower densities, which means that it fails to capture the semantics in images. In the following three cases, we i) generate random noise, ii) average two images with
Table 3: Distances between means of generated digits (resp. images) and ground-truth digits (resp. images) on MNIST (resp. CIFAR-10).

<table>
<thead>
<tr>
<th>Method</th>
<th>MNIST l₁ Dis</th>
<th>MNIST l₂ Dis</th>
<th>CIFAR l₁ Dis</th>
<th>CIFAR l₂ Dis</th>
</tr>
</thead>
<tbody>
<tr>
<td>WGAN-GP</td>
<td>13.80</td>
<td>0.93</td>
<td>80.98</td>
<td>1.72</td>
</tr>
<tr>
<td>WGAN+LR</td>
<td>12.91</td>
<td>0.86</td>
<td>82.96</td>
<td>1.81</td>
</tr>
<tr>
<td>WGAN+ER</td>
<td>12.26</td>
<td>0.77</td>
<td>72.28</td>
<td>1.59</td>
</tr>
<tr>
<td>WGAN+VA</td>
<td>12.38</td>
<td>0.78</td>
<td>69.01</td>
<td>1.53</td>
</tr>
<tr>
<td>DGM</td>
<td>12.12</td>
<td>0.79</td>
<td>179.30</td>
<td>3.95</td>
</tr>
<tr>
<td>Joint-W</td>
<td><strong>11.82</strong></td>
<td><strong>0.73</strong></td>
<td><strong>64.23</strong></td>
<td><strong>1.41</strong></td>
</tr>
</tbody>
</table>

Figure 8: ROC curves for evaluation of outlier detection on CIFAR-10.
REFERENCES


A LITERATURE REVIEWS

We discuss some of related literatures and shed lights on the relationship between our work with others.

A.1 EXPLICIT GENERATIVE MODELS

Explicit generative models are interested in fitting each instance with a scaler density expected to explicitly capture the distribution behind data. Such densities are often up to a constant and called as energy functions which are common in undirected graphical models (LeCun et al. (2006)). Hence, explicit generative models are also termed as energy-based models. An early version of energy-based models is the FRAME (Filters, Random field, And Maximum Entropy) model (Zhu et al. (1997); Wu et al. (2000)). Later on, some works leverage deep neural networks to model the energy function (Ngiam et al. (2011); Xie et al. (2016b)) and pave the way for researches on deep energy model (DEM) (e.g., Liu & Wang (2017); Kim & Bengio (2017); Zhai et al. (2016); Haarnoja et al. (2017); Du & Mordatch (2019); Nijkamp et al. (2019)). Apart from DEM, there are also some other forms of deep explicit models based on restricted Boltzmann machines like deep belief networks (Hinton et al. (2006)) and deep Boltzmann machines (Salakhutdinov & Hinton (2009)).

The normalized constant under the energy function requires an intractable integral over all possible instances, which makes the model hard to learn via Maximum Likelihood Estimation (MLE). To solve this issue, some works propose to approximate the constant by MCMC methods (Geman & Geman (1984); Neal (2011)). However, MCMC requires an inner-loop samples in each training, which induces high computational costs. Another solution is to optimize an alternate surrogate loss function. For example, contrastive divergence (CD) (Liu & Wang (2017)) is proposed to measure how much KL divergence can be improved by running a small numbers of Markov chain steps towards the intractable likelihood, while score matching (SM) (Hyvärinen (2005)) detours the constant by minimizing the distance for gradients of log-likelihoods. Moreover, the intractable normalized constant makes it hard to sample from. To obtain an accurate samples from unnormalized densities, many studies propose to approximate the generation by diffusion-based processes, like generative flow (Nguyen et al. (2017)) and variational gradient descent (Liu & Wang (2016)). Also, a recent work (Hu et al. (2018)) leverages Stein discrepancy to design a neural sampler from unnormalized densities. The fundamental disadvantage of explicit model is that the energy-based learning is difficult to accurately capture the distribution of true samples due to the low manifold of real-world instances (Liu & Wang (2017)).

A.2 IMPLICIT GENERATIVE MODELS

Implicit generative models focus on a generation mapping from random noises to generated samples. Such mapping function is often called as generator and possesses better flexibility compared with explicit models. Two typical implicit models are Variational Auto-Encoder (VAE) (Kingma & Welling (2014)) and Generative Adversarial Networks (GAN) (Goodfellow et al. (2014)). VAE introduces a latent variable and attempts to maximize the variational lower bound for likelihood of joint distribution of latent variable and observable variable, while GAN targets an adversarial game between the generator and a discriminator (or critic in WGAN) that aims at discriminating the generated and true samples. In this paper, we focus on GAN and its variants (e.g., WGAN (Arjovsky et al. (2017)), WGAN-GP (Gulrajani et al. (2017)), DCGAN (Radford et al. (2016)), etc.) as the implicit generative model and we leave the discussions on VAE as future work.

Two important issues concerning GAN and its variants are instability of training and local optima. The typical local optima for GAN can be divided into two categories: mode-collapse (the model fails to capture all the modes in data) and mode-redundance (the model generates modes that do not exist in data). Recently there are many attempts to solve these issues from various perspectives. One perspective is from regularization. Two typical regularization methods are likelihood-based and entropy-based regularization with the prominent examples Warde-Farley & Bengio (2017) and Li & Turner (2018) that respectively leverage denoising feature matching and implicit gradient approximation to enforce the regularization constraints. The likelihood and entropy regularizations could respectively help the generator to focus on data distribution and encourage more diverse samples, and a recent work (Tao et al. (2019)) uses Langevin dynamics to indicate that i) the entropy and likelihood regularizations are equivalent and share an opposite relationship in mathematics, and ii) both regularizations would make the model converge to a surrogate point with a bias from original
data distribution. Then Tao et al. (2019) proposes a variational annealing strategy to empirically unite two regularizations and tackle the biased distributions.

To deal with the instability issue, there are also some recent literatures from optimization perspectives and proposes different algorithms to address the non-convergence of minimax game optimization (for instance, Gemp & Mahadevan (2018), Liang & Stokes (2019), Gidel et al. (2019)). Moreover, the disadvantage of implicit models is the lack of explicit densities over instances, which disables the black-box generator to characterize the distributions behind data.

A.3 Attempts to Combine Both of the Worlds

Recently, there are several studies that attempt to combine explicit and implicit generative models from different ways. For instance, Zhao et al. (2017) proposes energy-based GAN that leverages energy model as discriminator to distinguish the generated and true samples. The similar idea is also used by Kim & Bengio (2017) and Dai et al. (2017) which let the discriminator estimate a scaler energy value for each sample. Such discriminator is optimized to give high energy to generated samples and low energy to true samples while the generator aims at generating samples with low energy. The fundamental difference is that Zhao et al. (2017) and Dai et al. (2017) both aim at minimizing the discrepancy between distributions of generated and true samples while the motivation of Kim & Bengio (2017) is to minimize the KL divergence between estimated densities and true samples. Kim & Bengio (2017) adopts contrastive divergence (CD) to link MLE for energy model over true data with the adversarial training of energy-based GAN. However, both CD-based method and energy-based GAN have limited power for both generator and discriminator. Firstly, if the generated samples resemble true samples, then the gradients for discriminator given by true and generated samples are just the opposite and will counteract each other, and the training will stop before the discriminator captures accurate data distribution. Second, since the objective boils down to minimizing the KL divergence (for Kim & Bengio (2017)) or Wasserstein distance (for Dai et al. (2017)) between model and true distributions, the issues concerning GAN (or WGAN) like training instability and mode-collapse would also bother these methods.

Another way for combination is by cooperative training. Xie et al. (2016a) (and its improved version Xie et al. (2018)) leverages the samples of generator as the MCMC initialization for energy-based model. The synthesized samples produced from finite-step MCMC are closer to the energy model and the generator is optimized to make the finite-step MCMC revise its initial samples. Also, a recent work Du et al. (2018) proposes to regard the explicit model as a teacher net who guides the training of implicit model. The synthesized samples produced from finite-step MCMC are closer to the energy model.

B Background for Stein Discrepancy

Assume $q(x)$ to be a continuously differentiable density supported on $\mathcal{X} \subset \mathbb{R}^d$ and $f(x)$ a smooth vector function. Define $A_q[f(x)] = \nabla_x \log q(x)f(x)^\top + \nabla_x f(x)$ as a Stein operator. If $f$ is a Stein class (satisfying some mild boundary conditions) then we have the following Stein identity property:

$$
E_{x \sim q}[A_q[f(x)]] = E_{x \sim q}[\nabla_x \log q(x)f(x)^\top + \nabla_x f(x)] = 0.
$$

Such property induces the Stein discrepancy between distributions $P : p(x)$ and $Q : q(x)$, $x \in \mathcal{X}$:

$$
S(Q, P) = \sup_{f \in \mathcal{F}} \{E_{x \sim q}[A_p[f(x)]] = \sup_{f \in \mathcal{F}} \{E_{x \sim q}tr(\nabla_x \log p(x)f(x)^\top + \nabla_x f(x))\}, \quad (12)
$$

where $f$ is what we call Stein critic that exploits over function space $\mathcal{F}$ and if $\mathcal{F}$ is large enough then $S(Q, P) = 0$ if and only if $Q = P$. Note that in (12), we do not need the normalized constant for $p(x)$ which enables Stein discrepancy to deal with unnormalized density.

If $\mathcal{F}$ is a unit ball in a Reproducing Kernel Hilbert Space (RKHS) with a positive definite kernel function $k(., .)$, then the supremum in (12) would have a closed form (see Liu et al. (2016), Chwialkowski
Formally, the gap between (14) and (15) is
\[ S_k(Q, P) = E_{x' \sim q}[u_p(x, x')], \]  
where \( u_p(x, x') = \log p(x)^T k(x, x') \log p(x') + \log p(x)^T \nabla_x k(x, x') + \nabla_x k(x, x')^T \log p(x') + t \nabla_x \log p(x') \).

The (13) gives the Kernel Stein Discrepancy (KSD).

\section{Proofs of Results in Section 3.2}

\subsection{Proof of Theorem 1}

\textbf{Proof.} Fixing the Wasserstein critic \( D \), we are going to solve for

\[ \min_G \{ E_{y \sim P_E}[D(y)] + \lambda_2 S(P_G, P_E) \}. \]

By definition, if there exists a \( P_E \)-measure zero set with positive \( \text{supp}(P) \)-measure, then \( S(P, P_E) = \infty \). Hence, to solve (14), it suffices to consider distributions whose support belongs to \( \text{supp}(P_E) \).

If \( C^\infty \) is dense in \( L^2 \), we can restrict to those \( P \)'s that are absolutely continuous with respect to \( P_E \):

\[ \inf_{P \in \mathcal{P}_E} \left\{ E_{x \sim P}[D(x)] + \lambda_2 \cdot E_{x \sim P}[\nabla \log(P(x)/P_E(x))^T k(x, x') \nabla \log(P(x)/P_E(x))'] \right\}. \]

Set \( h(x) := \nabla \log(P(x)/P_E(x)) \), the problem above becomes

\[ \inf_{h \in \mathcal{H}_E, \|h\| = 1} \left\{ E_{P_E}[Dh] + \lambda_2 \cdot E_{x \sim P}[\nabla \log(1 + h(x))^T k(x, x') \nabla \log(1 + h(x))] \right\}. \]

For the minimization problem above, invoking Lagrangian duality gives

\[ \sup_{t \in \mathbb{R}} \min_{h \in \mathcal{H}_E} \left\{ E_{P_E}[D(t - h)] + \lambda_2 \cdot E_{x \sim P}[\nabla \log(1 + h(x))^T k(x, x') \nabla \log(1 + h(x))] \right\}. \]

Applying the approximation \( \log(1 + a) = a + O(a^2) \) to the minimization problem above yields

\[ \inf_{h} \left\{ E_{P_E}[(D - t)h(x)] + \lambda_2 \cdot E_{x \sim P}(1 + h(x))^T k(x, x') \nabla h(x)(1 + h(x)) \right\}. \]

Consider a further approximation

\[ \inf_{h} \left\{ E_{P_E}[(D - t)h] + \lambda_2 \cdot E_{x \sim P}[\nabla h(x)^T k(x, x') \nabla h(x')] \right\}. \]

By definition of \( \|D - t\|_{H^{-1}(P_E; k)} \), the inner infimum equals \( -\frac{1}{4\lambda_2} \|D - t\|_{H^{-1}(P_E; k)} \) and hence (15) equals

\[ -\frac{1}{4\lambda_2} \inf_{t \in \mathbb{R}} \|D - t\|_{H^{-1}(P_E; k)}. \]

Formally, the gap between (14) and (15) is \( O(1/\lambda_2^2) \), which completes the proof. We note that a sufficient condition to make the above formal derivation hold is that \( \mathcal{X} \) is compact and the kernel is bounded, although this can be greatly weakened.

\subsection{Proof for Theorem 2}

\textbf{Proof.} Essentially the result is a consequence of distributionally robust optimization with Wasserstein metric \cite{Gao2016} \cite{Gao2017}. Here we provide a simplified version for completeness. Consider

\[ \min_G \{ \lambda_2 E_{y \sim P_E}[A_{\gamma_E}f(y)] + \mathcal{W}(P_{\text{real}}, P_G) \}. \]

By assumption, using the definition of Wasserstein metric, we write the problem above as

\[ \min_{\gamma} E_{(x, y) \sim \gamma} [\lambda_2 A_{\gamma_E}f(y)] + \|x - y\|, \]

where the minimization is over all joint distributions of \( (x, y) \) with \( x \)-marginal being \( P_{\text{real}} \). Using the law of total expectation, the problem above is equivalent to

\[ \min_{\gamma} E_{x \sim P_{\text{real}}} [E_{y \sim \gamma_x} [\lambda_2 A_{\gamma_E}f(y) + \|x - y\| \mid x]] \]

where the minimization is over \( \gamma_x \), all conditional distributions of \( y \) given \( x \). If \( \text{Lip}(\lambda_2 A_{\gamma_E}f) < 1 \), then the minimal value equals \( -\infty \), otherwise the minimal value equals \( \lambda_2 E_{y \sim P_{\text{real}}} [A_{\gamma_E}f(y)] \).

Hence the proof is completed.
D  PROOFS AND MORE DISCUSSIONS IN SECTION 4

D.1 DETAILS FOR ONE-DIMENSIONAL CASE

For the analysis of 1-dim regularized WGAN in section 3.1.1, we assume a Gaussian likelihood function for generated sample \( x' \), \( \mu(x') = \exp(-\frac{1}{2}(x'-b)^2) \) which is up to a constant. Its parameter can be estimated by \( b = \mathbb{E}[x] \). Then since \( x' = \theta z \), we have \( \mathbb{E}(\log \mu(\theta z)) = -\frac{1}{2}\mathbb{E}[z^2]\theta^2 + \mathbb{E}[z]\mathbb{E}[x]|\theta - \frac{1}{2}\mathbb{E}[x]^2 \). Like the case in WGAN, we consider \( \mathbb{E}[x] = \mathbb{E}[z] = 1 \) and we have \( \mathbb{E}[z^2] = 1 + \mathbb{E}[z] \). Hence, for the analysis on likelihood- (and entropy-) regularized WGAN, we can study the following system:

\[
\min_{\theta} \max_{\psi} \psi - \psi \cdot \theta - \lambda(\theta^2 - \theta).
\]

When \( \lambda = 1 \), the above objective degrades to (6); when \( \lambda < 0 \) (likelihood-regularization), the gradient of regularization term pushes \( \theta \) to shrink, which helps for convergence; when \( \lambda > 0 \) (entropy-regularization), the added term forms an amplifying strength on \( \lambda \) and we have \( \mathbb{E} \).

Interestingly, the added terms \( \frac{\lambda}{2} \frac{1}{2}(\theta + \phi)^2 + \frac{\lambda}{2}(\theta + \phi)^2 \) in (8) and the original terms \( \psi - \psi \cdot \theta \) in WGAN play both necessary roles to guarantee the convergence to the unique optimum points \( [\psi^*, \theta^*, \phi^*] = [0, 1, -1] \). If we remove the critic and optimize \( \theta \) and \( \phi \) with the remaining loss terms, we would find that the training would converge but not necessarily to \( [\psi^*, \theta^*] = [0, 1] \) (since the optimum points are not unique in this case). On the other hand, if we remove the estimator, the system degrades to (6) and would not converge to the unique optimum point \( [\psi^*, \theta^*] = [0, 1] \). If we consider both of the world and optimize three terms together, the training would converge to a unique global optimum \( [\psi^*, \theta^*, \phi^*] = [0, 1, -1] \).

D.2 PROOF FOR PROPOSITION 1

Proof. Instead of directly studying the optimization for (8), we first prove the following problem will converge to the unique optimum,

\[
\min_{\theta} \max_{\psi} \min_{\phi} \theta \psi + \theta \phi + \frac{1}{2} \theta^2 + \phi^2.
\]

Applying alternate SGD we have the following iterations:

\[
\psi_{t+1} = \psi_t + \eta \cdot \theta_t,
\]

\[
\phi_{t+1} = \phi_t - \eta \cdot (\theta_t + 2\psi_t) = (1 - 2\eta)\phi_t - \eta\theta_t,
\]

\[
\theta_{t+1} = \theta_t - \eta(\psi_{t+1} + \phi_{t+1} + \theta_t) = -\eta(1 - 2\eta)\phi_t + (1 - \eta)\theta_t - \eta\psi_t.
\]

Then we obtain the relationship between adjacent iterations:

\[
\begin{bmatrix}
\psi_{t+1} \\
\phi_{t+1} \\
\theta_{t+1}
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & \eta \\
0 & 1 - 2\eta & -\eta \\
-\eta & -\eta(1 - 2\eta) & 1 - \eta
\end{bmatrix}
\cdot
\begin{bmatrix}
\psi_t \\
\phi_t \\
\theta_t
\end{bmatrix} = M \cdot
\begin{bmatrix}
\psi_t \\
\phi_t \\
\theta_t
\end{bmatrix}
\]

We further calculate the eigenvalues for matrix \( M \) and have the following equations (assume the eigenvalue as \( \lambda \)):

\[
(\lambda - 1)^3 + 3\eta(\lambda - 1)^2 + 2\eta^2(1 + \eta)(\lambda - 1) + 2\eta^3 = 0.
\]

One can verify that the solutions to the above equation satisfy \( |\lambda| < \sqrt{(1 - \eta + \eta^2)(1 + \eta - \eta^2)} \).

Then we have the following relationship

\[
\left\| \begin{bmatrix}
\psi_{t+1} \\
\phi_{t+1} \\
\theta_{t+1}
\end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix}
\psi_t & \phi_t & \theta_t
\end{bmatrix} \cdot M \cdot \begin{bmatrix}
\psi_t \\
\phi_t \\
\theta_t
\end{bmatrix} \right\|_2^2 \leq \lambda_m^2 \cdot \left\| \begin{bmatrix}
\psi_t \\
\phi_t \\
\theta_t
\end{bmatrix} \right\|_2^2
\]

where \( \lambda_m \) denotes the eigenvalue with the maximum absolute value of matrix \( M \). Hence, we have

\[
\psi_{t+1}^2 + \phi_{t+1}^2 + \theta_{t+1}^2 \leq (1 - \eta + \eta^2)(1 + \eta - \eta^2)|\psi_t^2 + \phi_t^2 + \theta_t^2|.
\]

We proceed to replace \( \psi, \phi \) and \( \theta \) in (16) by \( \psi', \phi' \) and \( \theta' \) respectively and conduct a change of variable: let \( \theta' = 1 - \theta \) and \( \phi' = -1 - \phi \). Then we get the conclusion in the proposition.
D.3 Generalization to Bilinear Systems

Our analysis in the one-dimension case inspires us that we can add affiliated variable to modify the objective and stabilize the training for general bilinear system. The bilinear system is of wide interest for researchers focusing on stability of GAN training (Goodfellow (2017); Liang & Stokes (2019); Gidel et al. (2019); Gemp & Mahadevan (2018)). The general bilinear function can be written as

\[ F(\psi, \theta) = \theta^\top A \psi - b^\top \theta - c^\top \psi, \]

where \( \psi, \theta \) are both \( r \)-dimensional vectors and the objective is \( \min_{\theta} \max_{\psi} F(\psi, \theta) \) which can be seen as a basic form of various GAN objectives. Unfortunately, if we directly use simultaneous (resp. alternate) SGD to optimize such objectives, one can obtain divergence (resp. fluctuation.) To solve the issue, some recent papers propose several optimization algorithms, like extrapolation (Liang & Stokes (2019)). Also, Liang & Stokes (2019) shows that it is the interaction term which generates non-zero values for \( \nabla_{\theta}\phi F \) and \( \nabla_{\phi}\theta F \) that leads to such instability of training. Different from previous works that focused on algorithmic perspective, we propose to add new affiliated variables which modify the objective function and allow the SGD algorithm to achieve convergence without changing the optimum points.

Based on the minimax objective of (17) we add affiliated \( r \)-dimensional variable \( \phi \) (corresponding to the estimator in our model) the original system and tackle the following problem:

\[ \min_{\theta} \max_{\psi} \min_{\phi} F(\psi, \theta) + \alpha H(\phi, \theta), \]

where \( H(\phi, \theta) = \frac{1}{2}(\theta + \phi)^\top B(\theta + \phi) \), \( B = (AA^\top)^{\frac{1}{2}} \) and \( \alpha \) is a non-negative constant. Theoretically, the new problem keeps the optimum points of (17) unchanged. Let \( L(\psi, \phi, \theta) = F(\psi, \theta) + \alpha G(\phi, \theta) \)

**Proposition 2.** Assume the optimum point of \( \min_{\theta} \max_{\psi} \min_{\phi} F(\psi, \theta) \) are \([\psi^*, \theta^*]\), then the optimum points of (18) would be \([\psi^*, \theta^*, \phi^*]\) where \( \phi^* = -\theta^* \).

**Proof.** The condition tells us that \( \nabla_{\theta} F(\psi^*, \theta) = 0 \) and \( \nabla_{\psi} F(\psi^*, \theta) = 0 \). Then we derive the gradients for \( L(\psi, \phi, \theta) \),

\[ \nabla_{\psi} L(\psi^*, \phi, \theta) = \nabla_{\theta} F(\psi^*, \theta) = 0, \]

\[ \nabla_{\theta} L(\psi, \phi, \theta^*) = \nabla_{\theta} F(\psi, \theta^*) + \nabla_{\theta} H(\phi, \theta^*) = \frac{1}{2}(B + B^\top)(\theta^* + \phi), \]

\[ \nabla_{\phi} L(\psi, \phi, \theta) = \nabla_{\phi} H(\phi, \theta) = \frac{1}{2}(B + B^\top)(\phi + \theta). \]

Combining (20) and (21) we get \( \phi^* = -\theta^* \). Hence, the optimum point of (18) is \([\psi^*, \theta^*, \phi^*]\) where \( \phi^* = -\theta^* \). \( \square \)

The advantage of the new problem is that it can be solved by SGD algorithm and guarantees convergence theoretically. We formulate the results in the following theorem.

**Theorem 4.** For problem \( \min_{\theta} \max_{\psi} \min_{\phi} L(\psi, \phi, \theta) \) using alternate SGD algorithm, i.e.,

\[ \psi_{t+1} = \psi_t + \eta \nabla_{\psi} L(\theta_t, \psi_t, \phi_t), \]

\[ \phi_{t+1} = \phi_t - \eta \nabla_{\phi} L(\theta_t, \psi_{t+1}, \phi_t), \]

\[ \theta_{t+1} = \theta_t - \eta \nabla_{\theta} L(\theta_t, \psi_{t+1}, \phi_{t+1}), \]

we can achieve convergence to \([\psi^*, \theta^*, \phi^*]\) where \( \phi^* = -\theta^* \) with at least linear rate of \((1 - \eta_1 + \eta_2^2)(1 + \eta_2 - \eta_1^2)\) where \( \eta_1 = \eta \sigma_{\min}, \eta_2 = \eta \sigma_{\max} \) and \( \sigma_{\min} \) (resp. \( \sigma_{\max} \)) denotes the maximum (resp. minimum) singular value of matrix \( A \).

To prove Theorem 3, we can prove a more general argument.
Lemma 1. If we consider any first-order optimization method on \([18]\), i.e.,
\[
\psi_{t+1} \in \psi_0 + \text{span}(L(\psi_0, \phi, \theta), \cdots, F(\psi_t, \phi, \theta)), \forall t \in \mathbb{N},
\]
\[
\phi_{t+1} \in \psi_0 + \text{span}(L(\psi, \phi_0, \theta), \cdots, L(\psi, \phi_t, \theta)), \forall t \in \mathbb{N},
\]
\[
\theta_{t+1} \in \psi_0 + \text{span}(L(\psi, \phi, \theta_0), \cdots, L(\psi, \phi, \theta_t)), \forall t \in \mathbb{N},
\]
Then we have
\[
\tilde{\psi}_t = V^T(\psi_t - \psi^*), \quad \tilde{\phi}_t = U^T(\phi_t - \phi^*), \quad \tilde{\theta}_t = U^T(\theta_t - \theta^*),
\]
where \(U\) and \(V\) are the singular vectors decomposed by matrix \(A\) using SVD decomposition, i.e., \(A = UDV^\top\) and the triple \(([\tilde{\psi}_t]), ([\tilde{\phi}_t]), ([\tilde{\theta}_t])\) \(\leq t \leq T\) follows the update rule with step size \(\sigma, \eta\) as the same optimization method on a unidimensional problem
\[
\min_{\psi} \max_{\phi} \min_{\theta} \psi^2 + \theta^2 + \frac{1}{2} \phi^2,
\]
with step size \(\eta\), where \(\sigma_i\) denotes the \(i\)-th singular value on the diagonal of \(D\).

Proof. The proof is extended from the proof of Lemma 3 in Gidel et al. (2019). The general class of first-order optimization methods derive the following updations:
\[
\psi_{t+1} = \psi_0 + \sum_{s=0}^{t+1} \rho_{st}(A^\top \theta_s - c) = \psi_0 + \sum_{s=0}^{t+1} \rho_{st}A^\top(\theta_s - \theta^*),
\]
\[
\phi_{t+1} = \phi_0 + \frac{1}{2} \sum_{s=0}^{t+1} \delta_{st}(B + B^\top)(\theta_s + \phi_s),
\]
\[
\theta_{t+1} = \theta_0 + \sum_{s=0}^{t+1} \mu_{st}[A(\psi_s - \psi^*) + \frac{1}{2}(B + B^\top)(\theta_s + \phi_s)],
\]
where \(\rho_{st}, \delta_{st}, \mu_{st} \in \mathbb{R}\) depend on specific optimization method (for example, in SGD, \(\rho_{st} = \delta_{st} = \mu_{st}\) remain as a non-zero constant for all \(t\) and other coefficients are zero).

Using SVD \(A = UDV^\top\) and the fact \(\theta^* = -\phi^*\), \(B = (UDD^\top U^\top) = D\), we have
\[
V^T(\psi_{t+1} - \psi^*) = V^T(\psi_0 - \psi^*) + \sum_{s=0}^{t+1} \rho_{st}D^\top U^T(\theta_s - \theta^*)
\]
\[
U^T(\phi_{t+1} - \phi^*) = U^T(\phi_0 - \phi^*) + \sum_{s=0}^{t+1} \delta_{st}U^T D(\theta_s - \theta^*) + U^D(\phi_s - \phi^*),
\]
\[
U^T(\theta_{t+1} - \theta^*) = U^T(\theta_0 - \theta^*) + \sum_{s=0}^{t+1} \rho_{st}[DV^T(\psi_s - \psi^*) + U^T D(\theta_s - \theta^*) + U^T D(\phi_s - \phi^*)],
\]
and equivalently,
\[
\tilde{\psi}_{t+1} = \tilde{\psi}_0 + \sum_{s=0}^{t+1} \rho_{st}D^\top \tilde{\theta}_t, \quad \tilde{\phi}_t = \tilde{\phi}_0 + \sum_{s=0}^{t+1} \delta_{st}D(\tilde{\theta}_t + \tilde{\phi}_t),
\]
\[
\tilde{\theta}_{t+1} = \tilde{\theta}_0 + \sum_{s=0}^{t+1} \rho_{st}D(\tilde{\psi}_t + \tilde{\phi}_t + \tilde{\phi}_t).
\]

Note that \(D\) is a rectangular matrix with non-zero elements on a diagonal block of size \(r\). Hence, the above \(r\)-dimensional problem can be reduced to \(r\) unidimensional problems:
\[
[\tilde{\psi}_{t+1}]_i = [\tilde{\psi}_0]_i + \sum_{s=0}^{t+1} \rho_{st}\sigma_i(\tilde{\theta}_t)_i, \quad [\tilde{\phi}_t]_i = [\tilde{\phi}_0]_i + \sum_{s=0}^{t+1} \delta_{st}\sigma_i(\tilde{\theta}_t)_i + [\tilde{\phi}_t],
\]
\[
[\tilde{\theta}_{t+1}]_i = [\tilde{\theta}_0]_i + \sum_{s=0}^{t+1} \rho_{st}\sigma_i([\tilde{\psi}_t]_i + [\tilde{\phi}_t]_i + [\tilde{\phi}_t]).
\]
The above iterations can be conducted independently in each dimension where the optimization in \(i\)-th dimension follows the same updating rule with step size \(\sigma, \eta\) as problem in (23).
Furthermore, since problem (23) can achieve convergence with a linear rate of \((1-\eta+\eta^2)(1+\eta-\eta^2)\) using alternate SGD (the proof is similar to that of (16)), the multi-dimensional problem in (18) can achieve convergence by SGD with at least a rate of \((1-\eta_1+\eta^2_2)(1+\eta_2-\eta^2_2)\) where \(\eta_1 = \eta \sigma_{\text{max}}\), \(\eta_2 = \eta \sigma_{\text{min}}\) and \(\sigma_{\text{max}} \) (resp. \(\sigma_{\text{min}}\)) denotes the maximum (resp. minimum) singular value of matrix \(A\). We conclude the proof for Theorem 4.

Theorem 4 suggests that the added term \(H(\phi, \theta)\) with affiliated variables \(\phi\) could help the SGD algorithm achieve convergence to the the same optimum points as directly optimizing \(F(\psi, \theta)\).

Our method is related to consensus optimization algorithm (Liang & Stokes (2019)) which adds a regularization term \(\|\nabla_\theta F(\psi, \theta)\| + \|\nabla_\psi F(\psi, \theta)\|\) to (17) resulting extra quadratic terms for \(\theta\) and \(\psi\). The disadvantage of such method is the requirement of Hessian matrix of \(F(\psi, \theta)\) which is computational expensive for high-dimensional data. By contrast, our solution only requires the first-order derivatives.

D.4 Strongly Convexity

In section 3.1.2, we assume \(H(\theta, \phi)\) as a \(\mu\)-strongly convex function which indicates that it satisfies the conditions:

\[
(\nabla_\theta H(\theta, \cdot) - \nabla_\theta H(\theta', \cdot)) \cdot (\theta - \theta') \geq \mu \|\theta - \theta'\|^2_2, \forall \theta, \theta' \in \Theta,
\]

\[
(\nabla_\phi H(\cdot, \phi) - \nabla_\phi H(\cdot, \phi')) \cdot (\phi - \phi') \geq \mu \|\phi - \phi'\|^2_2, \forall \phi, \phi' \in \Phi.
\]

Bedises, \(F(\theta, \psi)\) is \(\mu\)-strongly convex for \(\theta\) and \(\mu\)-strongly concave for \(\psi\) so it satisfies:

\[
(\nabla_\theta F(\theta, \cdot) - \nabla_\theta F(\theta', \cdot)) \cdot (\theta - \theta') \geq \mu \|\theta - \theta'\|^2_2, \forall \theta, \theta' \in \Theta,
\]

\[
(\nabla_\psi F(\cdot, \psi) - \nabla_\psi F(\cdot, \psi')) \cdot (\psi - \psi') \geq \mu \|\psi - \psi'\|^2_2, \forall \psi, \psi' \in \Psi.
\]

In section 3.1.2, we also define \(h(\omega_h) = \nabla_\theta H + \nabla_\phi H\) and \(f(\omega_f) = \nabla_\theta F - \nabla_\psi F\), so the above condition can be written in a more compact form,

\[
(h(\omega_h) - h(\omega'_h)) \cdot (\omega_h - \omega'_h) \geq \mu \|\omega_h - \omega'_h\|^2_2, \forall \omega_h, \omega'_h \in \Omega_h,
\]

\[
(f(\omega_f) - f(\omega'_f)) \cdot (\omega_f - \omega'_f) \geq \mu \|\omega_f - \omega'_f\|^2_2, \forall \omega_f, \omega'_f \in \Omega_f.
\]

D.5 Proof for Theorem 2

The proof relies on two lemmas,

**Lemma 2.** For any \(\omega \in \Omega\) and \(\omega^+ = P_{\Omega}(\omega + u)\), then we have \(\|\omega^+ - \omega\|^2_2 \leq u^\top (\omega^+ - \omega)\).

**Proof.** Since \(\omega^+\) is a projection of \(\omega + u\) on a convex set \(\Omega\), we have \(\omega^+ - (\omega + u) = (\omega^+ - \omega) - u \geq 0\). Rearranging the above inequality one can easily get the lemma. \(\square\)

**Lemma 3.** If function \(\Phi(\omega)\) is \(\mu\)-strongly convex, we have

\[
\mu \|\omega - \omega^*\|^2_2 \leq \nabla \Phi(\omega)^\top (\omega - \omega^*).
\]

Similarly, if \(\Phi(\omega)\) is \(\mu\)-strongly concave, we have \(\mu \|\omega - \omega^*\|^2_2 \leq \nabla \Phi(\omega)^\top (\omega - \omega^*)\).

**Proof.** By optimality of \(\omega^*\), we have \(\nabla \Phi(\omega^*)^\top (\omega - \omega^*) \geq 0\). Since \(\Phi\) is \(\mu\)-convex, we can further derive

\[
\mu \|\omega - \omega^*\|^2_2 \leq \nabla \Phi(\omega^*)^\top (\omega - \omega^*) + \mu \|\omega - \omega^*\|^2_2 \leq \nabla \Phi(\omega)^\top (\omega - \omega^*).\]

\(\square\)
Proof. (Proof for Theorem 3) We apply Lemma 2 to (11) with $(\omega, \mu, \omega^+)$ = $(\omega_f^{t+1/2}, \eta f(\omega_f^{t+1/2}), \omega_f^{t+1})$ and we have
\[
\|\omega_f^{t+1} - \omega_f^{t+1/2}\| \leq -\eta f(\omega_f^{t+1/2})^\top (\omega_f^{t+1} - \omega_f^{t+1/2})
\] (25)
Then we have
\[
\|\omega_f^{t+1} - \omega_f^*\|^2 = \|\omega_f^{t+1/2} - \omega_f^* + \omega_f^* - \omega_f^{t+1/2}\|^2 \\
\leq 2\|\omega_f^{t+1/2} - \omega_f^*\|^2 + 2\|\omega_f^* - \omega_f^{t+1/2}\|^2 \quad (by \quad \|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2) \tag{26}
\]
According to Lemma 3, we have
\[
2\eta f(\omega_f^{t+1/2})^\top (\omega_f^{t+1/2} - \omega_f^*) \geq 2\eta \mu \|\omega_f^{t+1/2} - \omega_f^*\|^2.
\] (27)
Plug (27) into (26) and we get
\[
\|\omega_f^{t+1} - \omega_f^*\|^2 \leq (2 - 2\eta \mu)\|\omega_f^{t+1/2} - \omega_f^*\|^2.
\] (28)
The above inequality is equivalent to
\[
\|\theta^{t+1} - \theta^*\|^2 + \|\psi^{t+1} - \psi^*\|^2 \leq (2 - 2\eta \mu)(\|\theta^{t+1/2} - \theta^*\|^2 + \|\psi^{t} - \psi^*\|^2). \tag{29}
\]
Similarly, one can obtain
\[
\|\omega_h^{t+1} - \omega_h^*\|^2 \leq (2 - 2\eta \mu)\|\omega_h^{t+1/2} - \omega_h^*\|^2,
\] (30)
i.e.,
\[
\|\theta^{t+1/2} - \theta^*\|^2 + \|\phi^{t+1} - \phi^*\|^2 \leq (2 - 2\eta \mu)(\|\theta^{t} - \theta^*\|^2 + \|\phi^{t} - \phi^*\|^2). \tag{31}
\]
Combining (29) and (31) we have
\[
\|\omega^{t+1} - \omega^*\|^2 = \|\theta^{t+1} - \theta^*\|^2 + \|\psi^{t+1} - \psi^*\|^2 + \|\phi^{t+1} - \phi^*\|^2 \\
\leq (1 - 2\eta \mu)\|\theta^{t} - \theta^*\|^2 + (2 - 2\eta \mu)\|\psi^{t+1} - \psi^*\|^2 + (2 - 2\eta \mu)\|\phi^{t+1} - \phi^*\|^2 \tag{32}
\]
Hence, if $\frac{\eta}{2\mu} < \eta < \frac{1}{\mu}$ we have $0 < 2 - 2\eta \mu < 1$ and $\|\omega^{t} - \omega^*\|^2 \leq (2 - 2\eta \mu)^t \|\omega^0 - \omega^*\|^2$.

E DETAILS FOR EXPERIMENT SETUP

E.1 SYNTHETIC DATASETS

We provide the details for two synthetic datasets. The Two-Circle dataset consists of 24 Gaussian mixtures where 8 of them are located in an inner circle with radius $r_1 = 4$ and 16 of them lie in an outer circle with radius $r_2 = 8$. For each Gaussian component, the covariance matrix is $\begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix}$ and the mean value is $[r_1 \cos t, r_1 \sin t]$, where $t = \frac{2\pi k}{8}$, $k = 1, \ldots, 8$, for the inner circle, and $[r_2 \cos t, r_2 \sin t]$, where $t = \frac{2\pi k}{16}$, $k = 1, \ldots, 16$ for the outer circle. We sample $N_1 = 2000$ points as true observed samples for model training. In section 5.5, we consider noised data scenario. In this case, we randomly add $n$ noise points sampled from Gaussian distribution $\mathcal{N}(0, \sigma \mathbf{I})$ where $\sigma = 2$ to the original true samples. Here we set $n = [40, 100, 160, 300, 400, 600, 800, 1000]$.

The Two-Spiral dataset contains 100 Gaussian mixtures whose centers locate on two spiral-shaped curves. For each Gaussian component, the covariance matrix is $\begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$ and the mean value is $[-c_1 \cos c_4, c_4 \sin c_4]$, where $c_1 = \frac{2\pi}{5} + \text{linspace}(0, 0.5, 50) \cdot 2\pi$, for one spiral, and $[c_2 \cos c_3, -c_3 \sin c_2]$, where $c_2 = \frac{2\pi}{7} + \text{linspace}(0, 0.5, 50) \cdot 2\pi$ for another spiral. We sample $N_2 = 5000$ points as true observed samples. In section 5.5, we consider insufficient data scenario. In this case, the sample size $N_2$ is reduced to $[100, 200, 300, 500, 700, 1000, 2000]$. 
Under review as a conference paper at ICLR 2020

We can write (3) as a more general form

\[ \min_{\theta, \phi} \max_\pi \max_{\psi} \max_{\pi} \mathbb{E}_{x \sim P_{data}} \left[ d_\phi(x) - \mathbb{E}_{x \sim \psi} \left[ d_\phi(G_\theta(z)) \right] \right] + \lambda_1 \mathbb{E}_{x \sim P_{data}} \left[ A_{P_\psi} \left[ f_\pi(x) \right] \right] + \lambda_2 \mathbb{E}_{z \sim \psi} \left[ A_{P_\psi} \left[ f_\pi(G_\theta(z)) \right] \right] \]

We specify \( D_1 \) as Wasserstein distance or JS divergence in our paper and for \( D_2 \) and \( D_3 \) we consider the general Stein discrepancy or kernel Stein discrepancy. Here we use \( \mathcal{W} \), \( \mathcal{J} \) to denote Wasserstein distance and JS divergence respectively, and \( S, S_k \) to represent general Stein discrepancy and kernel Stein discrepancy respectively. We omit the gradient penalty term for Wasserstein distance here but use it in experiments.

### E.2 Model Specifications and Training Algorithm

In different tasks, we consider different model specifications in order to meet the demand of capacity as well as test the effectiveness under various settings. Our proposed framework [3] adopts Wasserstein distance for the first term and two Stein discrepancies for the second and the third terms. We can write [3] as a more general form

\[ \min_{\theta, \phi} \max_\pi \max_{\psi} \max_{\pi} \mathbb{E}_{x \sim P_{data}} \left[ d_\phi(x) - \mathbb{E}_{x \sim \psi} \left[ d_\phi(G_\theta(z)) \right] \right] + \lambda_1 \mathbb{E}_{x \sim P_{data}} \left[ A_{P_\psi} \left[ f_\pi(x) \right] \right] + \lambda_2 \mathbb{E}_{z \sim \psi} \left[ A_{P_\psi} \left[ f_\pi(G_\theta(z)) \right] \right] \]

where \( D_1, D_2, D_3 \) denote three general discrepancy measures for distributions. As stated in our remark, \( D_1 \) can be specified as arbitrary discrepancy measures for implicit generative models. Here we also use JS divergence, the objective for vanilla GAN. To well distinguish them, we call the model using Wasserstein distance (resp. JS divergence) as Joint-W (resp. Joint-JS) in our experiments. On the other hand, the two Stein discrepancies in [3] can be specified by KSD (as defined by \( S_k \) in [2]) or general Stein discrepancy with an extra critic (as defined by \( S \) in [1]). Hence, the two specifications for \( D_1 \) and the two for \( D_2 (D_3) \) compose four different combinations in total, and we organize the objectives in each case in Table 4.

In our experiments, we use KSD with RBF kernels for \( D_2 \) and \( D_3 \) in Joint-W and Joint-JS on two synthetic datasets. For MNIST with conditional training (given the digit class as model input), we also use KSD with RBF kernels. For MNIST and CIFAR with unconditional training (the class is not given as known information), we find that KSD cannot provide desirable results so we adopt general Stein discrepancy for higher model capacity.

The objectives in Table 4 appear to be computationally expensive. In the worst case (using general Stein discrepancy), there are two minimax operations where one is from GAN or WGAN and one is from Stein discrepancy estimation. To guarantee training efficiency, we alternatively update the generator, estimator, Wasserstein critic and Stein critic over the parameters \( \theta, \phi, \psi \) and \( \pi \) respectively. Specifically, in one iteration, we optimize the generator over \( \theta \) and the estimator over \( \phi \) with one step respectively, and then optimize the Wasserstein critic over \( \psi \) with \( n_d \) steps and the Stein critic over \( \pi \) with \( n_c \) steps. Such training approach guarantees the same time complexity order of proposed method as that of GAN or WGAN, and the training time for our model can be bounded within constant times the time for training GAN model. In our experiment, we set \( n_d = n_c = 5 \) and empirically find that our model Stein Bridging would be two times slower than WGAN on average. We present the training algorithm for Stein Bridging in Algorithm 1.

### E.3 Implementation Details

We give the information of network architectures and hyper-parameter settings for our model as well as each competitor in our experiments.

<table>
<thead>
<tr>
<th>( D_1 )</th>
<th>( D_2 )</th>
<th>( D_3 )</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{W} )</td>
<td>( S )</td>
<td>( S )</td>
<td>( \min_{\theta} \min_{\phi} \max_{\pi} \max_{\psi} \mathbb{E}<em>{x \sim P</em>{data}} [d_\phi(x)] - \mathbb{E}<em>{x \sim P</em>{data}} [d_\phi(G_\theta(z))] + \lambda_1 \mathbb{E}<em>{x \sim P</em>{data}} [A_{P_\psi} [f_\pi(x)]] + \lambda_2 \mathbb{E}<em>{z \sim P</em>{data}} [A_{P_\psi} [f_\pi(G_\theta(z))]] )</td>
</tr>
<tr>
<td>( \mathcal{W} )</td>
<td>( S_k )</td>
<td>( S_k )</td>
<td>( \min_{\theta} \min_{\phi} \max_{\pi} \max_{\psi} \mathbb{E}<em>{x \sim P</em>{data}} [d_\phi(x)] - \mathbb{E}<em>{x \sim P</em>{data}} [d_\phi(G_\theta(z))] + \lambda_1 \mathbb{E}<em>{x \sim P</em>{data}} [A_{P_\psi} [f_\pi(x)]] + \lambda_2 \mathbb{E}<em>{z \sim P</em>{data}} [A_{P_\psi} [f_\pi(G_\theta(z))]] )</td>
</tr>
<tr>
<td>( \mathcal{J} )</td>
<td>( S )</td>
<td>( S )</td>
<td>( \min_{\theta} \min_{\phi} \max_{\pi} \max_{\psi} \mathbb{E}<em>{x \sim P</em>{data}} [\log(d_\phi(x))] + \mathbb{E}<em>{x \sim P</em>{data}} [\log(1 - d_\phi(G_\theta(z)))] + \lambda_1 \mathbb{E}<em>{x \sim P</em>{data}} [A_{P_\psi} [f_\pi(x)]] + \lambda_2 \mathbb{E}<em>{z \sim P</em>{data}} [A_{P_\psi} [f_\pi(G_\theta(z))]] )</td>
</tr>
<tr>
<td>( \mathcal{J} )</td>
<td>( S_k )</td>
<td>( S_k )</td>
<td>( \min_{\theta} \min_{\phi} \max_{\pi} \max_{\psi} \mathbb{E}<em>{x \sim P</em>{data}} [\log(d_\phi(x))] + \mathbb{E}<em>{x \sim P</em>{data}} [\log(1 - d_\phi(G_\theta(z)))] + \lambda_1 \mathbb{E}<em>{x \sim P</em>{data}} [A_{P_\psi} [f_\pi(x)]] + \lambda_2 \mathbb{E}<em>{z \sim P</em>{data}} [A_{P_\psi} [f_\pi(G_\theta(z))]] )</td>
</tr>
</tbody>
</table>

Table 4: Objectives for different specifications of \( D_1(P_{real}, P_G), D_2(P_{real}, P_E) \) and \( D_3(P_G, P_E) \).
Algorithm 1: Training Algorithm for Stein Bridging

1. **REQUIRE**: observed training samples \( \{x\} \sim P_{\text{real}}. \)
2. **REQUIRE**: \( \theta_0, \theta_0, \psi, \pi_0 \), initial parameters for generator, estimator, Wasserstein critic and Stein critic models respectively. \( \alpha^E = 0.0002, \beta_1^E = 0.9, \beta_2^E = 0.999 \), Adam hyper-parameters for explicit models, \( \alpha^I = 0.0002, \beta_1^I = 0.5, \beta_2^I = 0.999 \), Adam hyper-parameters for implicit models. \( \lambda_1 = 1, \lambda_2 \), weights for \( D_2 \) and \( D_3 \) (we suggest increasing \( \lambda_2 \) from 0 to 1 through training). \( n_d = 5, n_e = 5 \) number of iterations for Wasserstein critic and Stein critic, respectively, before one iteration for generator and estimator. \( B = 100 \), batch size.

3. **while not converged**
   
   4. **for** \( n = 1, \cdots, n_d \) **do**
      
      5. Sample \( B \) true samples \( \{x_i\}_{i=1}^B \) from \( \{x\} \);
      6. Sample \( B \) random noise \( \{z_i\}_{i=1}^B \sim P_0 \) and obtain generated samples \( \tilde{x}_i = G_\theta(z_i) ; \)
      7. \( \mathcal{L}_{\text{dis}} = \frac{1}{B} \sum_{i=1}^B \psi_k(\tilde{x}_i) - \psi_k(x_i) - \lambda(\|\nabla_{\tilde{x}_i} \psi_k(\tilde{x}_i)\| - 1)^2 \) // the last term is for gradient penalty in WGAN-GP where \( \tilde{x}_i = \epsilon_i x_i + (1 - \epsilon_i) \tilde{x}_i, \epsilon_i \sim U(0,1); \)
      8. \( \psi_{k+1} \leftarrow \text{Adam}(\mathcal{L}_{\text{dis}}, \psi_k, \alpha^I, \beta_1^I, \beta_2^I) \) // update the Wasserstein critic;
      9. **for** \( n = 1, \cdots, n_e \) **do**
      
      10. Sample \( B \) true samples \( \{x_i\}_{i=1}^B \) from \( \{x\} \);
      11. Sample \( B \) random noise \( \{z_i\}_{i=1}^B \sim P_0 \) and obtain generated samples \( \tilde{x}_i = G_\theta(z_i) ; \)
      12. \( \mathcal{L}_{\text{critic}} = \frac{1}{B} \sum_{i=1}^B \lambda_1 A_{\phi}(x_i) + \lambda_2 A_{\phi}(\hat{x}_i); \)
      13. \( \phi_{k+1} \leftarrow \text{Adam}(\mathcal{L}_{\text{critic}}, \phi_k, \alpha^E, \beta_1^E, \beta_2^E) \) // update the Stein critic;
      14. Sample \( B \) random noise \( \{z_i\}_{i=1}^B \sim P_0 \) and obtain generated samples \( \tilde{x}_i = G_\theta(z_i) ; \)
      15. \( \mathcal{L}_{\text{est}} = \frac{1}{B} \sum_{i=1}^B \lambda_1 A_{\phi}(f_\pi(x_i)) + \lambda_2 A_{\phi}(f_\pi(\tilde{x}_i)); \)
      16. \( \phi_{k+1} \leftarrow \text{Adam}(\mathcal{L}_{\text{est}}, \phi_k, \alpha^E, \beta_1^E, \beta_2^E) \) // update the density estimator;
      17. \( \mathcal{L}_{\text{gen}} = \frac{1}{B} \sum_{i=1}^B -d_\psi(\tilde{x}_i) + \lambda_2 A_{\phi}(f_\pi(\tilde{x}_i)); \)
      18. \( \theta_{k+1} \leftarrow \text{Adam}(\mathcal{L}_{\text{gen}}, \theta_k, \alpha^I, \beta_1^I, \beta_2^I) \) // update the sample generator;

19. **OUTPUT**: trained sample generator \( G_\theta(x) \) and density estimator \( p_\phi(x) \).

The energy function is often parametrized as a sum of multiple experts (Hinton (1999)) and each expert can have various function forms depending on the distributions. If using sigmoid distribution, the energy function becomes (see section 2.1 in Kim & Bengio (2017) for details)

\[
E_\phi(x) = \sum_i \log(1 + e^{-\langle \mathbf{W}_i n(x_i) + b_i \rangle}),
\]

where \( n(x) \) maps input \( x \) to a feature vector and could be specified as a deep neural network, which corresponds to deep energy model (Ngiam et al. (2011)).

For synthetic datasets, we set the noise dimension as 4. All the generators are specified as a three-layer fully-connected (FC) neural network with neuron size \( 4 \rightarrow 128 \rightarrow 128 \rightarrow 2 \), and all the Wasserstein critics (or the discriminators in JS-divergence-based GAN) are also a three-layer FC network with neuron size \( 2 \rightarrow 128 \rightarrow 128 \rightarrow 1 \). For the estimators, we set the expert number as 4 and the feature function \( n(x) \) is a FC network with neuron size \( 2 \rightarrow 128 \rightarrow 128 \rightarrow 4 \). Then in the last layer we sum the outputs from each expert as the energy value \( E(x) \). The activation units are searched within \([\text{LeakyReLU, tanh, sigmoid, softplus}]\). The learning rate \([1e^{-6}, 1e^{-5}, 1e^{-4}, 1e^{-3}, 1e^{-2}]\) and the batch size \([50, 100, 150, 200]\). The gradient penalty weight for WGAN is searched in \([0, 0.1, 1, 10, 100]\).

For MNIST dataset, we set the noise dimension as 100. All the critics/discriminators are implemented as a four-layer network where the first two layers adopt convolution operations with filter size 5 and stride \([2, 2]\) and the last two layers are FC layers. The size for each layer is \( 1 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 1 \). All the generators are implemented as a four-layer networks where the first two layers are FC and the last two adopt deconvolution operations with filter size 5 and stride \([2, 2]\). The size for each layer is \( 100 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 1 \). For the estimators, we consider the expert number as 128 and the feature function is the same as the Wasserstein critic except that the size of last layer is 128. Then we sum the outputs from each expert as the energy
Table 5: Quantitative results including MMD (lower is better), HSR (higher is better) as the metrics for quality of generated samples and KLD (lower is better), JSD (lower is better), AUC (higher is better) as the metrics for accuracy of estimated densities on Two-Circle and Two-Spiral datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Two-Circle</th>
<th>Two-Spiral</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MMD</td>
<td>HSR</td>
</tr>
<tr>
<td>GAN</td>
<td>0.0033</td>
<td>0.772</td>
</tr>
<tr>
<td>GAN+LR</td>
<td>0.0106</td>
<td>0.391</td>
</tr>
<tr>
<td>GAN+ER</td>
<td>0.0103</td>
<td>0.428</td>
</tr>
<tr>
<td>GAN+VA</td>
<td>0.0118</td>
<td>0.295</td>
</tr>
<tr>
<td>WGAN-GP</td>
<td>0.0010</td>
<td>0.841</td>
</tr>
<tr>
<td>WGAN+LR</td>
<td>0.0013</td>
<td>0.840</td>
</tr>
<tr>
<td>WGAN+ER</td>
<td>0.0008</td>
<td>0.830</td>
</tr>
<tr>
<td>WGAN+VA</td>
<td>0.0016</td>
<td>0.835</td>
</tr>
<tr>
<td>DEM</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>EGAN</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DGM</td>
<td>0.0040</td>
<td>0.774</td>
</tr>
<tr>
<td>Joint-JS</td>
<td>0.0037</td>
<td>0.883</td>
</tr>
<tr>
<td>Joint-W</td>
<td>0.0007</td>
<td>0.844</td>
</tr>
</tbody>
</table>

value. The activation units are searched within [ReLU, LeakyReLU, tanh]. The learning rate \([2 \times 10^{-5}, 2 \times 10^{-4}, 2 \times 10^{-3}, 2 \times 10^{-2}]\) and the batch size \([32, 64, 100, 128] \). The gradient penalty weight for WGAN is searched in \([1, 10, 100, 1000]\).

For CIFAR dataset, we adopt the same architecture as DCGAN for critics and generators. As for the estimator, the architecture of feature function is the same as the critics except the last year where we set the expert number as 128 and sum each output as the output energy value. The architectures for Stein critic are the same as Wasserstein critic for both MNIST and CIFAR datasets.

E.4 Evaluation Metrics

We adopt some quantitative metrics to evaluate the performance of each method on different tasks. In section 4.1, we use two metrics to test the sample quality: Maximum Mean Discrepancy (MMD) and High-quality Sample Rate (HSR). MMD measures the discrepancy between two distributions \(X\) and \(Y\), 

\[
MMD(X,Y) = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) - \frac{1}{m} \sum_{j=1}^{m} \Phi(y_j)
\]

where \(x_i\) and \(y_j\) denote samples from \(X\) and \(Y\) respectively and \(\Phi\) maps each sample to a RKHS. Here we use RBF kernel and calculate MMD between generated samples and true samples. HSR statistics the rate of high-quality samples over all generated samples. For Two-Circle dataset, we define the generated points whose distance from the nearest Gaussian component is less than \(\sigma_1\) as high-quality samples. We generate 2000 points in total and statistic HSR. For Two-Spiral dataset, we set the distance threshold as \(5\sigma_2\) and generate 5000 points to calculate HSR.

As for Inception Score and CEPC. For MNIST, we pre-train a classifier for 10 digits which can provide the test accuracy up to 99% for calculation of scores. The conditional entropy of predicted classes (CEPC) for given samples is defined as 

\[
H(y|x) \approx \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{10} p(y_k|x_i) \log p(y_k|x_i)
\]

where \(x_i\) is a generated instance and \(y\) denotes the predicted class given \(x\) from a pre-trained classifier. CEPC measures how well a given sample can be classified into a right class, i.e. the quality of such sample. For CIFAR, we use the Inception V3 Network in Tensorflow as pre-trained classifier.

In section 4.2, we use three metrics to characterize the performance for density estimation: KL divergence, JS divergence and AUC. We divide the map into a 300 meshgrid, calculate the density values of each point given by the estimators and compute the KL and JS divergences between estimated density and ground-truth density. Besides, we select the centers of each Gaussian components as positive examples (expected to have high densities) and randomly sample 10 points within a circle around each center as negative examples (expected to have relatively low densities) and rank them according to the densities given by the model. Then we obtain the area under the curve (AUC) for false-positive rate v.s. true-positive rate.
Table 6: $l_1$ and $l_2$ distances between means of true digits and generated digits in each class on MNIST.

<table>
<thead>
<tr>
<th>Class</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>WGAN-GP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>20.3</td>
<td>11.4</td>
<td>14.3</td>
<td>14.8</td>
<td>13.5</td>
<td>13.3</td>
<td>13.8</td>
<td>11.0</td>
<td>13.0</td>
<td>12.3</td>
</tr>
<tr>
<td>$l_2$</td>
<td>1.74</td>
<td>1.07</td>
<td>0.82</td>
<td>0.98</td>
<td>0.83</td>
<td>0.68</td>
<td>0.95</td>
<td>0.62</td>
<td>0.82</td>
<td>0.75</td>
</tr>
<tr>
<td>WGAN+LR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>13.8</td>
<td>5.9</td>
<td>13.6</td>
<td>19.1</td>
<td>11.8</td>
<td>18.3</td>
<td>10.7</td>
<td>11.5</td>
<td>14.0</td>
<td>9.9</td>
</tr>
<tr>
<td>$l_2$</td>
<td>0.80</td>
<td>0.34</td>
<td>0.84</td>
<td>1.81</td>
<td>0.65</td>
<td>1.37</td>
<td>0.62</td>
<td>0.70</td>
<td>0.90</td>
<td>0.57</td>
</tr>
<tr>
<td>WGAN+ER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>16.1</td>
<td>8.9</td>
<td>11.7</td>
<td>14.2</td>
<td>12.3</td>
<td>10.8</td>
<td>13.9</td>
<td>11.4</td>
<td>12.1</td>
<td>10.9</td>
</tr>
<tr>
<td>$l_2$</td>
<td>1.20</td>
<td>0.74</td>
<td>0.54</td>
<td>0.86</td>
<td>0.73</td>
<td>0.54</td>
<td>0.97</td>
<td>0.69</td>
<td>0.72</td>
<td>0.63</td>
</tr>
<tr>
<td>WGAN+VA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>16.3</td>
<td>7.1</td>
<td>13.7</td>
<td>13.7</td>
<td>11.9</td>
<td>13.2</td>
<td>13.6</td>
<td>11.2</td>
<td>12.1</td>
<td>10.6</td>
</tr>
<tr>
<td>$l_2$</td>
<td>1.12</td>
<td>0.35</td>
<td>0.81</td>
<td>0.85</td>
<td>0.69</td>
<td>0.76</td>
<td>1.04</td>
<td>0.71</td>
<td>0.74</td>
<td>0.71</td>
</tr>
<tr>
<td>DGM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>22.2</td>
<td>10.9</td>
<td>12.7</td>
<td>10.2</td>
<td>10.8</td>
<td>9.0</td>
<td>9.5</td>
<td>10.9</td>
<td>12.7</td>
<td>11.7</td>
</tr>
<tr>
<td>$l_2$</td>
<td>1.41</td>
<td>0.83</td>
<td>0.81</td>
<td>0.65</td>
<td>0.67</td>
<td>0.56</td>
<td>0.66</td>
<td>0.67</td>
<td>0.88</td>
<td>0.76</td>
</tr>
<tr>
<td>Joint-W</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>14.1</td>
<td>7.5</td>
<td>14.3</td>
<td>12.9</td>
<td>11.1</td>
<td>11.0</td>
<td>13.7</td>
<td>9.7</td>
<td>12.0</td>
<td>11.5</td>
</tr>
<tr>
<td>$l_2$</td>
<td>0.89</td>
<td>0.47</td>
<td>0.93</td>
<td>0.73</td>
<td>0.55</td>
<td>0.51</td>
<td>1.06</td>
<td>0.53</td>
<td>0.70</td>
<td>0.97</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Class</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>WGAN-GP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>80.8</td>
<td>82.7</td>
<td>40.2</td>
<td>69.3</td>
<td>44.7</td>
<td>59.2</td>
<td>77.6</td>
<td>107.7</td>
<td>50.8</td>
<td>89.3</td>
</tr>
<tr>
<td>$l_2$</td>
<td>1.75</td>
<td>1.84</td>
<td>0.92</td>
<td>1.57</td>
<td>1.04</td>
<td>1.40</td>
<td>1.78</td>
<td>2.32</td>
<td>1.78</td>
<td>1.92</td>
</tr>
<tr>
<td>WGAN+LR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>78.4</td>
<td>79.2</td>
<td>73.8</td>
<td>86.0</td>
<td>75.8</td>
<td>77.2</td>
<td>106.7</td>
<td>103.0</td>
<td>56.5</td>
<td>92.3</td>
</tr>
<tr>
<td>$l_2$</td>
<td>1.63</td>
<td>1.76</td>
<td>1.59</td>
<td>1.88</td>
<td>1.68</td>
<td>1.74</td>
<td>2.36</td>
<td>2.23</td>
<td>1.24</td>
<td>2.00</td>
</tr>
<tr>
<td>WGAN+ER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>75.5</td>
<td>64.0</td>
<td>100.0</td>
<td>65.0</td>
<td>58.5</td>
<td>69.1</td>
<td>74.5</td>
<td>81.8</td>
<td>62.5</td>
<td>71.3</td>
</tr>
<tr>
<td>$l_2$</td>
<td>1.56</td>
<td>1.45</td>
<td>2.04</td>
<td>1.43</td>
<td>1.35</td>
<td>1.57</td>
<td>1.67</td>
<td>1.82</td>
<td>1.40</td>
<td>1.58</td>
</tr>
<tr>
<td>WGAN+VA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>60.9</td>
<td>70.0</td>
<td>79.4</td>
<td>62.7</td>
<td>63.0</td>
<td>73.9</td>
<td>76.2</td>
<td>77.2</td>
<td>59.8</td>
<td>66.4</td>
</tr>
<tr>
<td>$l_2$</td>
<td>1.32</td>
<td>1.55</td>
<td>1.68</td>
<td>1.39</td>
<td>1.42</td>
<td>1.63</td>
<td>1.70</td>
<td>1.77</td>
<td>1.33</td>
<td>1.48</td>
</tr>
<tr>
<td>DGM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>167.8</td>
<td>185.0</td>
<td>149.4</td>
<td>250.1</td>
<td>105.3</td>
<td>134.0</td>
<td>223.8</td>
<td>197.3</td>
<td>148.3</td>
<td>231.7</td>
</tr>
<tr>
<td>$l_2$</td>
<td>3.67</td>
<td>4.14</td>
<td>3.15</td>
<td>5.41</td>
<td>2.39</td>
<td>3.04</td>
<td>4.68</td>
<td>4.51</td>
<td>3.24</td>
<td>5.25</td>
</tr>
<tr>
<td>Joint-W</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_1$</td>
<td>59.3</td>
<td>58.1</td>
<td>77.3</td>
<td>54.8</td>
<td>58.1</td>
<td>65.1</td>
<td>63.9</td>
<td>82.8</td>
<td>59.1</td>
<td>63.2</td>
</tr>
<tr>
<td>$l_2$</td>
<td>1.26</td>
<td>1.30</td>
<td>1.60</td>
<td>1.23</td>
<td>1.28</td>
<td>1.44</td>
<td>1.80</td>
<td>1.27</td>
<td>1.27</td>
<td>1.43</td>
</tr>
</tbody>
</table>
Figure 9: Generated digits given by Joint-W on MNIST.

Figure 10: Generated images given by Joint-W on CIFAR.

Figure 11: The generated digits (and real digits) with the highest densities and the lowest densities given by Joint-W.
Figure 12: The generated digits (and real digits) with the highest densities and the lowest densities given by DGM.

Figure 13: The generated digits (and real digits) with the highest densities and the lowest densities given by EGAN.