Stein Bridging: Enabling Mutual Reinforcement between Explicit and Implicit Generative Models

Qitian Wu[1], Rui Gao[2], Hongyuan Zha[3,4] *
[1]Shanghai Jiao Tong University
[2]University of Texas at Austin
[3]The Chinese University of Hong Kong (Shenzhen)
[4]Georgia Institute of Technology

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 [16]. 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 [26, 41, 21, 34], random graph theory [39], energy-based reinforcement learning [19], 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 pariticular, generative adverarial networks (GANs) have shown great power in learning representations of images, natural languages, graphs, etc. [16, 38, 2, 4]. 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 [6]. 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 [49]. 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

*Email: `echo740@sjtu.edu.cn, rui.gao@mccombs.utexas.edu, zha@cc.gatech.edu`. Part of the work was done when Qitian Wu and Rui Gao were visiting The Chinese University of Hong Kong (Shenzhen). Hongyuan Zha is on leave from Georgia Institute of Technology.

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 explicit energy model is used to estimate the unnormalized densities of true samples via minimizing a Stein discrepancy; in the meantime, an implicit 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.

2.1 Energy Model.

The energy model assigns each data $\mathbf{x}$ with a scalar energy value $E_\phi(\mathbf{x})$, where $E_\phi(\cdot)$ is called the energy function and is parameterized by ϕ. The model is expected to assign low energy to true samples according to a Gibbs distribution $p_\phi(\mathbf{x}) = \exp\{-E_\phi(\mathbf{x})\}/Z_\phi$, where Z_ϕ is a normalizing constant dependent on ϕ.

The energy function is often parametrized as a sum of multiple experts [20] 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 [24] for details)

$$E_\phi(\mathbf{x}) = \sum_i \log(1 + e^{-(\mathbf{W}_i n(\mathbf{x}) + b_i)}), \tag{1}$$

where $n(\mathbf{x})$ maps input $\mathbf{x}$ to a feature vector and could be specified as a deep neural network, which corresponds to deep energy model [34].

The normalizing term Z_ϕ is often hard to compute, making the training intractable, and various methods are proposed to detour such term (see Appendix A).

2.2 Stein Discrepancy.

Stein discrepancy [17, 29, 5, 37] measures the difference between two distributions. Assume $q(\mathbf{x})$ to be a continuously differentiable density supported on $\mathcal{X} \subset \mathbb{R}^d$ and $\mathbf{f}(\mathbf{x})$ a smooth vector function. Define $\mathcal{A}_q\mathbf{f}(\mathbf{x}) = \nabla_\mathbf{x} \log q(\mathbf{x})\mathbf{f}(\mathbf{x})^\top + \nabla_\mathbf{x}\mathbf{f}(\mathbf{x})$ as a Stein operator. If $\mathbf{f}$ is a Stein class (satisfying some mild boundary conditions) then we have the following Stein identity property:

$$\mathbb{E}_{\mathbf{x}\sim q}[A_q\mathbf{f}(\mathbf{x})] = \mathbb{E}_{\mathbf{x}\sim q}[\nabla_\mathbf{x} \log q(\mathbf{x})\mathbf{f}(\mathbf{x})^\top + \nabla_\mathbf{x}\mathbf{f}(\mathbf{x})] = 0.$$

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

$$\mathcal{S}(\mathbb{Q}, \mathbb{P}) = \sup_{\mathbf{f}\in\mathcal{F}}\{\mathbb{E}_{\mathbf{x}\sim q}[A_p\mathbf{f}(\mathbf{x})] = \sup_{\mathbf{f}\in\mathcal{F}}\{\mathbb{E}_{\mathbf{x}\sim q} tr(\nabla_\mathbf{x} \log p(\mathbf{x})\mathbf{f}(\mathbf{x})^\top + \nabla_\mathbf{x}\mathbf{f}(\mathbf{x}))\}, \tag{2}$$

where $\mathbf{f}$ is what we call *Stein critic* that exploits over function space $\mathcal{F}$ and if $\mathcal{F}$ is large enough then $\mathcal{S}(\mathbb{Q}, \mathbb{P}) = 0$ if and only if $\mathbb{Q} = \mathbb{P}$. Note that in (2), we do not need the normalized constant for $p(\mathbf{x})$ 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 $k(\cdot, \cdot)$, then the supremum in (2) would have a close form (see [29] for more details):

$$\mathcal{S}_K(\mathbb{Q}, \mathbb{P}) = \mathbb{E}_{x,x'\sim q}[u_p(x, x')], \tag{3}$$

where $u_p(x, x') = \log p(\mathbf{x})^\top k(x, x') \log p(\mathbf{x}') + \log p(\mathbf{x})^\top \nabla_x k(x, x') + \nabla_x k(x, x')^\top \log p(\mathbf{x}') + tr(\nabla_{x,x'} k(x, x'))$. The (3) gives the Kernel Stein Discrepancy (KSD). One can refer to a recent work [3] for some important properties about (kernel) Stein discrepancy.

2.3 Wasserstein Metric.

Wasserstein metric is suitable for measuring distances between two distributions with non-overlapping supports [2]. The Wasserstein-1 metric between distributions $\mathbb{P}$ and $\mathbb{Q}$ is defined as $\mathcal{W}(\mathbb{P},\mathbb{Q}) := \min_\gamma \ \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma}[\|\mathbf{x}-\mathbf{y}\|]$, where the minimization is over all joint distributions with marginals $\mathbb{P}$ and $\mathbb{Q}$. By Kantorovich-Rubinstein duality, it has a dual representation

$$\mathcal{W}(\mathbb{P},\mathbb{Q}) := \max_{D} \ \{\mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{y}\sim\mathbb{Q}}[D(\mathbf{y})]\}\,, \tag{4}$$

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

2.4 Sobolev space and Sobolev dual norm.

Use L^2 to denote the canonical Hilbert space on $\mathbb{R}^d$ equipped with an inner product $\langle u,v\rangle_{L^2} := \int_{\mathbb{R}^d} uv d\mathbf{x}$. The Sobolev space H^1 is defined as the closure of C_0^∞, the set of smooth functions on $\mathbb{R}^d$ with compact support, with respect to the norm $\|u\|_{H^1} := \big(\int_{\mathbb{R}^d}(u^2+\|\nabla u\|_2^2)d\mathbf{x}\big)^{1/2}$. For $v\in L^2$, its Sobolev dual norm $\|v\|_{H^{-1}}$ is defined by [9]

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

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.

3.1 Model Formulation

We denote by $\mathbb{P}_{\text{real}}$ the underlying real distribution from which the data $\{\mathbf{x}\}$ 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 $\exp(-E(\mathbf{x}))$, where E is referred to as an energy function. The implicit generative model transforms an easy-to-sample random noise $\mathbf{z}$ with distribution P_0 via a generator G to a generated sample $\widetilde{x}=G(\mathbf{z})$ 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:

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

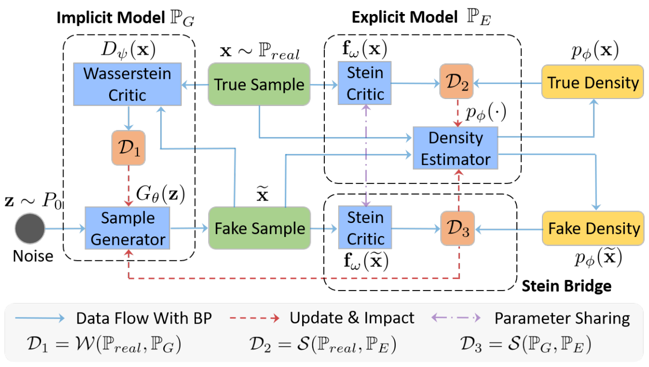
where $\lambda\ge 0$ is a weight coefficient. However, this approach appears no different than learning the two generative models separately. To better train the model, we incorporate the objective another term $\mathcal{S}(\mathbb{P}_G,\mathbb{P}_E)$ – called the *Stein bridge* – that measures the closeness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

$$\min_{E,G} \ \mathcal{W}(\mathbb{P}_{\text{real}},\mathbb{P}_G)+\lambda_1\mathcal{S}(\mathbb{P}_{\text{real}},\mathbb{P}_E)+\lambda_2\mathcal{S}(\mathbb{P}_G,\mathbb{P}_E), \tag{5}$$

where $\lambda_1,\lambda_2\ge 0$ 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 (5) is implemented using its equivalent dual representation (4). The two Stein terms in (5) can be implemented using (2) 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 (5) 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 [16]. 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 D.2.



Implicit Model $\mathbb{P}_G$
Explicit Model $\mathbb{P}_E$
$D_\psi(\mathbf{x})$
Wasserstein Critic
$\mathbf{x} \sim \mathbb{P}_{real}$
True Sample
$\mathbf{f}_\omega(\mathbf{x})$
Stein Critic
$\mathcal{D}_2$
$p_\phi(\mathbf{x})$
True Density
$p_\phi(\cdot)$
Density Estimator
$\mathcal{D}_1$
$\mathbf{z} \sim P_0$
Noise
$G_\theta(\mathbf{z})$
Sample Generator
$\tilde{\mathbf{x}}$
Fake Sample
Stein Critic
$\mathbf{f}_\omega(\tilde{\mathbf{x}})$
$\mathcal{D}_3$
Fake Density
$p_\phi(\tilde{\mathbf{x}})$
Stein Bridge
Data Flow With BP
Update & Impact
Parameter Sharing
$\mathcal{D}_1 = \mathcal{W}(\mathbb{P}_{real}, \mathbb{P}_G)$
$\mathcal{D}_2 = \mathcal{S}(\mathbb{P}_{real}, \mathbb{P}_E)$
$\mathcal{D}_3 = \mathcal{S}(\mathbb{P}_G, \mathbb{P}_E)$

Figure 1: Model framework for *Stein Bridging* which jointly train an implicit sample generator and an explicit density estimator via a Stein bridge.

Model	Objective
GAN	$\mathcal{D}_1$
Energy Model	$\mathcal{D}_2$
Energy-based GAN [50, 6]	$\mathcal{D}_1$
Contrastive Learning [24, 31]	$\mathcal{D}_2$
Cooperative Learning [47, 46]	$\mathcal{D}_2 + \mathcal{D}_3$
Stein Bridging (ours)	$\mathcal{D}_1 + \mathcal{D}_2 + \mathcal{D}_3$

Table 1: Comparison of objectives between different generative models, where $\mathcal{D}_1 := \mathcal{D}_1(\mathbb{P}_{\text{real}}, \mathbb{P}_G)$, $\mathcal{D}_2 := \mathcal{D}_2(\mathbb{P}_{\text{real}}, \mathbb{P}_E)$ and $\mathcal{D}_3 := \mathcal{D}_3(\mathbb{P}_G, \mathbb{P}_E)$ denote general statistical distances between two distributions.

3.2 Regularization Effects by Virtue of The Stein Bridge

The intuitive motivation of the Stein bridge term in (5) 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:

$$\max_D \min_G \left\{ \mathbb{E}_{\mathbf{y}\sim\mathbb{P}_G}[D(\mathbf{y})] - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{real}}}[D(\mathbf{x})] + \lambda_2 \mathcal{S}(\mathbb{P}_G, \mathbb{P}_E) \right\} \tag{6}$$

We define the *kernel Sobolev dual norm* as

$$\|D\|_{H^{-1}(\mathbb{P};k)} := \sup_{u \in C_0^\infty} \left\{ \langle D, u \rangle_{L^2(\mathbb{P})} : \ \mathbb{E}_{\mathbf{x},\mathbf{x}'\sim\mathbb{P}}[\nabla u(\mathbf{x})^\top k(\mathbf{x}, \mathbf{x}') \nabla u(\mathbf{x}')] \le 1 \right\},$$

which can be viewed as a kernel generalization of the Sobolev dual norm defined in Section 2.4, which reduces to the Sobolev dual norm when $k(\mathbf{x}, \mathbf{x}') = \mathbb{I}(\mathbf{x} = \mathbf{x}')$ 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* (6) *is equivalent to*

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

Note that if D is an optimal Wasserstein critic in (4), so does $D - r$, $r \in \mathbb{R}$. This is consistent to the penalty term $\inf_{t\in\mathbb{R}} \|D - t\|_{H^{-1}(\mathbb{P}_E;k)}$, since the penalties are identical for D and $D - r$. According to Section 2.4, 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., [18, 40]), but with a new way to encouraging smoothness.

Another way to interpret the Sobolev dual norm penalty is by observing that if $k(\mathbf{x}, \mathbf{x}') = \mathbb{I}(\mathbf{x} = \mathbf{x}')$, and $\mathbb{E}_{\mathbb{P}_E}[D - t] = 0$ [43], 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 (5) share the same Stein critic. Fixing the energy function E, consider the max-min problem over the Stein critic $\mathbf{f}$ and the generator G:

$$\max_{\mathbf{f}} \min_{G} \{\lambda_1 \mathbb{E}_{\mathbf{y}\sim\mathbb{P}_{\text{real}}}[\mathcal{A}_{\mathbb{P}_E}\mathbf{f}(\mathbf{y})] + \lambda_2 \mathbb{E}_{\mathbf{y}\sim\mathbb{P}_G}[\mathcal{A}_{\mathbb{P}_E}\mathbf{f}(\mathbf{y})] + \mathcal{W}(\mathbb{P}_{\text{real}}, \mathbb{P}_G)\}. \tag{7}$$

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

Theorem 2. *Problem* (7) *is equivalent to*

$$\max_{\mathbf{f}} \{(\lambda_1 + \lambda_2)\mathbb{E}_{\mathbf{y}\sim\mathbb{P}_{\text{real}}}[\mathcal{A}_{\mathbb{P}_E}\mathbf{f}(\mathbf{y})] : \ \text{Lip}(\mathcal{A}_{\mathbb{P}_E}\mathbf{f}) \leq 1/\lambda_2\},$$

where $\text{Lip}(\mathcal{A}_{\mathbb{P}_E}\mathbf{f})$ *denotes the Lipschitz constant of the function* $\mathcal{A}_{\mathbb{P}_E}\mathbf{f}$.

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 $\mathbf{f}$ via the constraint $\text{Lip}(\text{tr}(\mathbf{f}^\top \nabla \log \mathbb{P}_E + \nabla \mathbf{f})) \leq 1/\lambda_2$. 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 stablize 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 [15, 28]. In order to better understand the optimization behaviors, we first study a one-dimension linear system that provides some insights on this problem. Note that such toy example (or a similar one) is also utilized by [14, 32] to shed lights on the instability of WGAN training[1]. Consider a linear critic $D_\psi(x) = \psi x$ and generator $G_\theta(z) = \theta x$. Then the Wasserstein GAN objective can be written as a constrained bilinear problem: $\min_\theta \max_{|\psi|\leq 1} \psi\mathbb{E}[x] - \psi\theta\mathbb{E}[z]$, which could be further simplified as an unconstrained version (the behaviors could be generalized to multi-dimensional cases [14]):

$$\min_{\theta} \max_{\psi} \psi - \psi \cdot \theta. \tag{8}$$

Unfortunately, such simple objective cannot guarantee convergence by traditional gradient methods like SGD with alternate updating[2]: $\theta_{k+1} = \theta_k - \eta\psi_k,$, $\psi_{k+1} = \psi_k + \eta\theta_{k+1}$. 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 [28] theoretically show that such oscillation is due to the interaction term in (8).

One solution to the instability of GAN training is to add (likelihood) regularization, which has been widely studied by recent literatures [44, 27]. With regularization term, the objective changes into

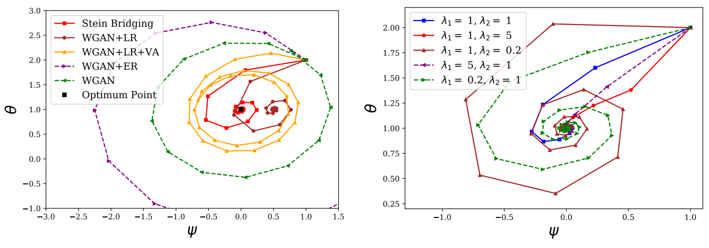
[1]Our theoretical discussions focus on WGAN, and we also compare with original GAN in the experiments.

[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.

$\min_\theta \max_{|\psi|\leq 1} \psi\mathbb{E}[x] - \psi\theta\mathbb{E}[z] - \lambda\mathbb{E}[\log\mu(\theta z)]$, where $\mu(\cdot)$ denotes the likelihood function and λ is a hyperparameter. A recent study [42] 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 C.1 for details):

$$\min_\theta \max_\psi \psi - \psi\cdot\theta - \lambda(\theta^2 - \theta). \tag{9}$$

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 [42]. In this case, one can verify that the optimum point becomes $[\psi^*, \theta^*] = [-\lambda, 1]$, resulting a bias. To avoid this issue, [42] 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).



(a) (b)

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 λ_1 and λ_2.

Finally, let us consider our proposed model. We also simplify the density estimator as a basic energy model $p_\phi(x) = \exp(\frac{1}{2}x^2 + \phi x)$ whose score function is $\nabla_x \log p_\phi(x) = x + \phi$. Then if we specify the two Stein discrepancies in (5) as KSD, we have the objective,

$$\min_\theta \max_\psi \min_\phi \psi - \psi\cdot\theta + \frac{\lambda_1}{2}(1+\phi)^2 + \frac{\lambda_2}{2}(\theta+\phi)^2. \tag{10}$$

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 (10) geometrically decreases the square norm* $N_t = |\psi^t|^2 + |\theta - 1|^2 + |\phi + 1|^2$, *for any* $0 < \eta < 1$ *with* $\lambda_1 = \lambda_2 = 1$,

$$N_{t+1} = (1 - \eta^2(1-\eta)^2)N_t. \tag{11}$$

In Fig. 2(a), we can see that Stein Bridging achieves a good convergence to the right optimum. Compared with (8), the objective (10) 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 (10) cooperate to guarantee convergence to a unique optimum. (More discussions in Appendix C.1).

We further generalize the analysis to multi-dimensional bilinear system $F(\boldsymbol{\psi}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top\mathbf{A}\boldsymbol{\psi} - \mathbf{b}^\top\boldsymbol{\theta} - \mathbf{c}^\top\boldsymbol{\psi}$ which is extensively used by researches for analysis of GAN stability [15, 13, 28, 14]. For any bilinear system, with the added term $H(\boldsymbol{\phi}, \boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta} + \boldsymbol{\phi})^\top\mathbf{B}(\boldsymbol{\theta} + \boldsymbol{\phi})$ where $\mathbf{B} = (\mathbf{A}\mathbf{A}^\top)^{\frac{1}{2}}$ 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 C.3.

4.2 Local Convergence for a General Model

To study the convergence for Stein Bridging, we proceed to consider a general optimization objective

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\psi}} \min_{\boldsymbol{\phi}} L(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}),$$

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

Theorem 3. *If F is μ-strongly convex-concave and H is μ-strongly convex, we can leverage the alternate SGD algorithm, i.e.*

$$\boldsymbol{\omega}_h^{t+1} = P_{\Omega_h}(\boldsymbol{\omega}_h^{t+1/2} - \eta h(\boldsymbol{\omega}_h^{t+1/2})), \tag{12}$$

$$\boldsymbol{\omega}_f^{t+1} = P_{\Omega_f}(\boldsymbol{\omega}_f^{t+1/2} - \eta f(\boldsymbol{\omega}_f^{t+1/2})), \tag{13}$$

where $\boldsymbol{\omega}_h^{t+1/2} = [\boldsymbol{\theta}^t, \boldsymbol{\psi}^t]$, $\boldsymbol{\omega}_h^{t+1} = [\boldsymbol{\theta}^{t+1/2}, \boldsymbol{\psi}^{t+1}]$, $\boldsymbol{\omega}_f^{t+1/2} = [\boldsymbol{\theta}^{t+1/2}, \boldsymbol{\phi}^t]$, $\boldsymbol{\omega}_f^{t+1} = [\boldsymbol{\theta}^{t+1}, \boldsymbol{\phi}^{t+1}]$, and $P_\Omega(\boldsymbol{\omega}) = \arg\min_{\boldsymbol{\omega}' \in \Omega} \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2^2$ denotes the projection mapping to Ω. 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[3] 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.

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 D.1.

In our experiments, we also replace the Wasserstein metric in (5) 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 valina GAN, WGAN-GP [18], 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 [42] 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 [1] and utilized by [42]. Second, for explicit density models, we consider Deep Energy Model (DEM) which is optimized based on Stein discrepancy, and energy-based GAN (EGAN) [6]. Besides, we also compare with Deep Directed Generative (DGM) Model [24] which adopts contrastive divergence to unite sample generator and density estimator. See Appendix A for brief introduction of these methods and Appendix D.3 for implementation details for each method.

[3]The reproducible codes are available at `https://github.com/echo740/SteinBridging`

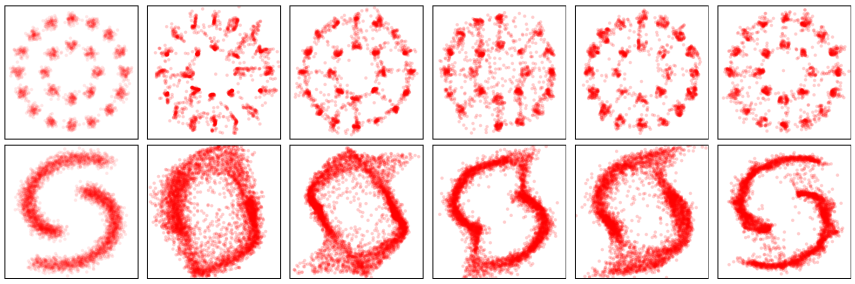

(a) True (b) GAN (c) WGAN (d) DGM (e) Joint-JS (f) Joint-W

Figure 3: (a) True samples and (b)∼(f) generated samples produced by the generators of different methods on Two-Circle (upper line) and Two-Spiral (bottom line) datasets.

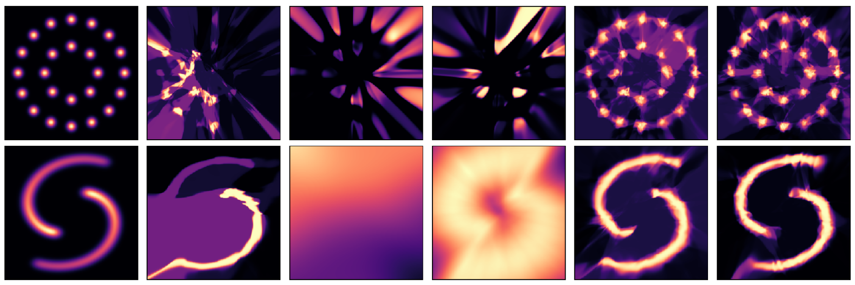

(a) True (b) DEM (c) EGAN (d) DGM (e) Joint-JS (f) Joint-W

Figure 4: (a) True densities and (b)∼(f) estimated densities given by the estimators of different methods on Two-Circle (upper line) and Two-Spiral (bottom line) datasets.

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., modes) 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 distinguishable 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 D.4 and we report the results in Table 5.

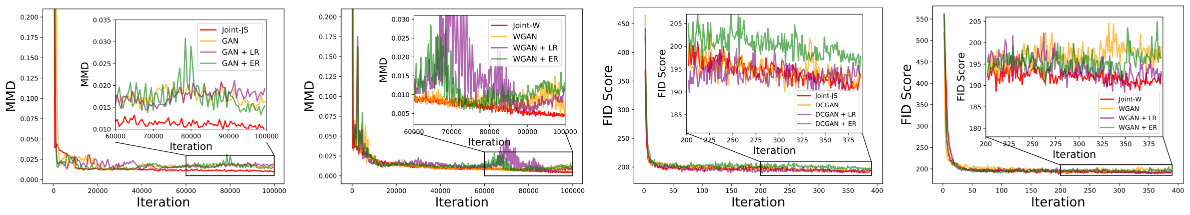
We visualize the generated digits/images on MNIST/CIFAR-10 datasets in Fig. 9 and Fig. 10 and use Inception Score and conditional entropy of predicted classes (CEPC) to measure the sample quality (See Appendix D.4 for details). As shown in Table 2, Joint-W (resp. Joint-JS) is superior than WGAN-GP (resp. DCGAN), regularized WGAN (resp. DCGAN) and DGM. The CEPC characterizes 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 3, 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 D.4 and Table 5 respectively. The values show that Joint-W and Joint-JS could provide more accurate density estimation than other competitors.

MNIST (Conditional)			MNIST (Unconditional)			CIFAR-10 (Unconditional)		
Method	Score	CEPC	Method	Score	CEPC	Method	Score	CEPC
DCGAN	8.43	0.168	WGAN-GP	7.71	0.256	WGAN-GP	6.80	0.153
DCGAN+LR	8.40	0.171	WGAN+LR	7.82	0.243	WGAN+LR	6.89	0.154
DCGAN+ER	8.33	0.179	WGAN+ER	7.75	0.252	WGAN+ER	6.99	0.156
DCGAN+VA	8.40	0.172	WGAN+VA	7.74	0.254	WGAN+VA	6.95†	0.154
DGM	8.15	0.201	DGM	6.87	0.372	DGM	4.79	0.146
Joint-JS(ours)	**8.53**	**0.156**	Joint-W(ours)	**7.90**	**0.231**	Joint-W(ours)	**7.11**	**0.151**

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.



(a) Two-Circle (b) Two-Spiral (c) MNIST (Condition) (d) MNIST (Uncondition)

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

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 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 50000 generated digits (resp. images) and 50000 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)

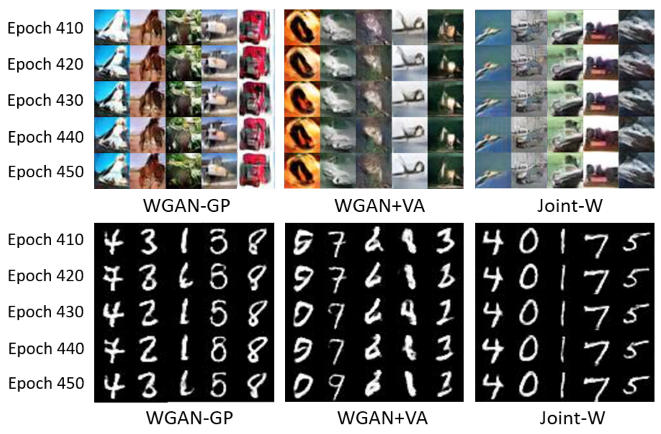
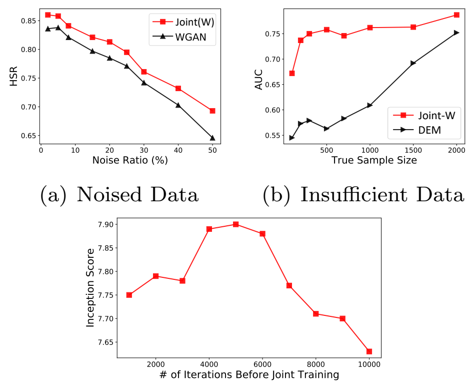


Figure 6: Generated digits (resp. images) given by the same noise z in adjacent training epoches on MNIST (reps. CIFAR) dataset.



(a) Noised Data (b) Insufficient Data

(c) Warm up on CIFAR-10

Figure 7: Joint-W with (a) noised data, (b) insufficient data and (c) 'warm up' iterations before joint training.

	MNIST		CIFAR	
Method	l_1 Dis	l_2 Dis	l_1 Dis	l_2 Dis
WGAN-GP	13.80	0.93	80.98	1.72
WGAN+LR	12.91	0.86	82.96	1.81
WGAN+ER	12.26	0.77	72.28	1.59
WGAN+VA	12.38	0.78	69.01	1.53
DGM	12.12	0.79	179.30	3.95
Joint-W	**11.82**	**0.73**	**64.23**	**1.41**

Table 3: Distances between means of generated digits (resp. images) and ground-truth digits (resp. images) on MNIST (resp. CIFAR-10).

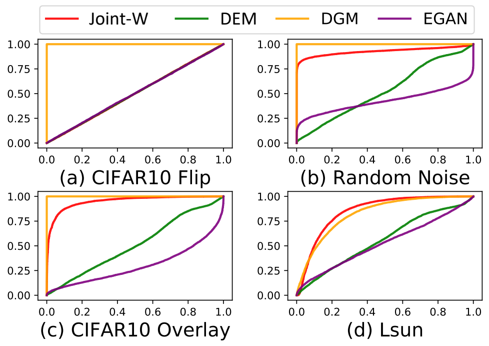


Figure 8: ROC curves for evaluation of outlier detection on CIFAR-10.

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 different CIFAR classes, and iii) adopt Lsun Bedroom dataset as the negative set, respectively. In these situations, the model is expected to distinguish the images in two sets. The results in Fig. 8 show that DGM provides the best results while the performance of Joint-W is quite close to DGM and much better than DEM and EGAN.

5.5 Addressing Data Insufficiency and Noisy Data

We proceed to test the model performance in some extreme situations where the observed samples are mixed with noises or the observed samples are quite insufficient. The results are presented in Fig. 7(a) where we add different ratios of random noise to the true samples in Two-Circle dataset and Fig. 7(b) where we only sample insufficient data for training in Two-Spiral dataset. The details are in Appendix D.1. The noise in data impacts the performance of WGAN and Joint-W, but comparatively, the performance decline for Joint-W is less insignificant than WGAN, which indicates better robustness of joint training w.r.t noised data. In Fig. 7(b), when the sample size decreases from 2000 to 100, the AUC value of DEM declines dramatically, showing its dependency on sufficient training samples. By contrast, the AUC of Joint-W exhibits a small decline when the sample size is more than 500 and suffers from an obvious decline when it is less than 300. Such phenomenon demonstrates lower sensitivity of joint training to observed sample size.

5.6 When to Start Joint Learning

In our experiment, we also observe an interesting phenomenon: the performance achieved at convergence would be better if we start joint training after some iterations with independent training for the generator and the estimator. In other words, at the beginning, we could set $\lambda_2 = 0$ (or some very small values) in (5) and after some iterations set it as a normal level. We report the inception scores on MNIST with different numbers of iterations for independent training in Fig. 7(c) where we can see that the score firstly goes up and then goes down when we increase iterations for independent training. Such phenomenon is quite similar to the 'warm up' trick used for training deep networks where one can use small learning rates at iterations in the begining and amplify its value for further training. One intuitive reason behind this phenomenon is that at the beginning, both the generator and estimator are weak and if we minimize the discrepancy between them at this point, they would possibly constrain each other and get limited in some bad local optima. When they become strong enough after some training iterations, uniting them through joint training would help them compensate and reinforce each other as our discussions.

6 Conclusions

In this paper, we aim at uniting the training for implicit generative model (represented by GAN) and explicit generative model (represented by a deep energy-based model). Besides two loss terms for GAN and energy-based model, we introduce the third loss characterized via Stein discrepancy between the generator in GAN and the energy-based model. Theoretically, we show that joint training could i) help to stablize GAN training and facilitate its convergence, and ii) enforcing dual regularization effects on both models and help to escape from local optima in optimization. We also conduct extensive experiments with different tasks and application senarios to verify our theoretical findings as well as demonstrate the superiority of our method compared with various GAN models and deep energy-based models.

References

[1] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *J. Mach. Learn. Res.*, 15(1):3563–3593, 2014.

[2] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[3] Alessandro Barp, François-Xavier Briol, Andrew B. Duncan, Mark A. Girolami, and Lester W. Mackey. Minimum stein discrepancy estimators. *CoRR*, abs/1906.08283, 2019.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[5] Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A kernel test of goodness of fit. In *ICML*, pages 2606–2615, 2016.

[6] Zihang Dai, Amjad Almahairi, Philip Bachman, Eduard H. Hovy, and Aaron C. Courville. Calibrating energy-based generative adversarial networks. In *ICLR*, 2017.

[7] Chao Du, Kun Xu, Chongxuan Li, Jun Zhu, and Bo Zhang. Learning implicit generative models by teaching explicit ones. *CoRR*, abs/1807.03870, 2018.

[8] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *CoRR*, abs/1903.08689, 2019.

[9] L.C. Evans and American Mathematical Society. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 2010.

[10] Rui Gao, Xi Chen, and Anton J Kleywegt. Wasserstein distributional robustness and regularization in statistical learning. *arXiv preprint arXiv:1712.06050*, 2017.

[11] Rui Gao and Anton J Kleywegt. Distributionally robust stochastic optimization with wasserstein distance. *arXiv preprint arXiv:1604.02199*, 2016.

[12] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, 1984.

[13] Ian Gemp and Sridhar Mahadevan. Global convergence to the equilibrium of gans using variational inequalities. *CoRR*, abs/1808.01531, 2018.

[14] Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. In *ICLR*, 2019.

[15] Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017.

[16] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[17] Jackson Gorham and Lester Mackey. Measuring sample quality with stein's method. In *Advances in Neural Information Processing Systems*, pages 226–234, 2015.

[18] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *NIPS*, pages 5767–5777, 2017.

[19] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *ICML*, pages 1352–1361, 2017.

[20] Geoffrey E. Hinton. Product of experts. In *ICANN'99 Artificial Neural Networks*, 1999.

[21] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[22] Tianyang Hu, Zixiang Chen, Hanxi Sun, Jincheng Bai, Mao Ye, and Guang Cheng. Stein neural sampler. *CoRR*, abs/1810.03545, 2018.

[23] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6:695–709, 2005.

[24] Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. In *ICLR*, 2017.

[25] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[26] Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. *Predicting Structured Data, MIT Press*, 2006.

[27] Yingzhen Li and Richard E. Turner. Gradient estimators for implicit models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[28] Tengyuan Liang and James Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *AISTATS*, pages 907–915, 2019.

[29] Qiang Liu, Jason D. Lee, and Michael I. Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *ICML*, pages 276–284, 2016.

[30] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *NIPS*, pages 2370–2378, 2016.

[31] Qiang Liu and Dilin Wang. Learning deep energy models: Contrastive divergence vs. amortized MLE. *CoRR*, abs/1707.00797, 2017.

[32] Vaishnavh Nagarajan and J. Zico Kolter. Gradient descent GAN optimization is locally stable. In *NIPS*, pages 5585–5595, 2017.

[33] Radford M. Neal. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Handbook of Markov Chain Monte Carlo*, 2, 2011.

[34] Jiquan Ngiam, Zhenghao Chen, Pang Wei Koh, and Andrew Y. Ng. Learning deep energy models. In *ICML*, pages 1105–1112, 2011.

[35] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *CVPR*, pages 3510–3520, 2017.

[36] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Yingnian Wu. On learning non-convergent non-persistent short-run mcmc toward energy-based model. *CoRR*, abs/1904.09770, 2019.

[37] Chris J Oates, Mark Girolami, and Nicolas Chopin. Control functionals for monte carlo integration. *Journal of the Royal Statistical Society, Series B*, 2017.

[38] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[39] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p^*) models for social networks. *Social Networks*, 29(2):173–191, 2007.

[40] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in neural information processing systems*, pages 2018–2028, 2017.

[41] Ruslan Salakhutdinov and Geoffrey E. Hinton. Deep boltzmann machines. In *AISTATS*, pages 448–455, 2009.

[42] Chenyang Tao, Shuyang Dai, Liqun Chen, Ke Bai, Junya Chen, Chang Liu, Ruiyi Zhang, Georgiy V. Bobashev, and Lawrence Carin. Variational annealing of gans: A langevin perspective. In *ICML*, pages 6176–6185, 2019.

[43] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[44] David Warde-Farley and Yoshua Bengio. Improving generative adversarial networks with denoising feature matching. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[45] Ying Nian Wu, Song Chun Zhu, and Xiuwen Liu. Equivalence of julesz ensembles and FRAME models. *International Journal of Computer Vision*, 38(3):247–265, 2000.

[46] Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. Cooperative learning of energy-based model and latent variable model via MCMC teaching. In *AAAI*, pages 4292–4301, 2018.

[47] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of descriptor and generator networks. *CoRR*, abs/1609.09408, 2016.

[48] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. A theory of generative convnet. In *ICML*, pages 2635–2644, 2016.

[49] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *ICML*, pages 1100–1109, 2016.

[50] Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun. Energy-based generative adversarial networks. In *ICLR*, 2017.

[51] Song Chun Zhu, Ying Nian Wu, and David Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, 1997.

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 [26]. 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 [51, 45]. Later on, some works leverage deep neural networks to model the energy function [34, 48] and pave the way for researches on deep energy model (DEM) (e.g., [31, 24, 49, 19, 8, 36]). Apart from DEM, there are also some other forms of deep explicit models based on restricted Boltzmann machines like deep belief networks [21] and deep Boltzmann machines [41].

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 [12, 33]. 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) [31] 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) [23] 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 [35] and variational gradient descent [30]. Also, a recent work [22] 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 [31].

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) [25] and Generative Adversarial Networks (GAN) [16]. 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 [2], WGAN-GP [18], DCGAN [38], 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 [44] and [27] 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 [42] 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 [42] 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, [13, 28, 14]). 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, [50] proposes energy-based GAN that leverages energy model as discriminator to distinguish the generated and true samples. The similar idea is also used by [24] and [6] 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 [50] and [6] both aim at minimizing the discrepancy between distributions of generated and true samples while the motivation of [24] is to minimize the KL divergence between estimated densities and true samples. [24] 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 discriminitor captures accurate data distribution. Second, since the objective boils down to minimizing the KL divergence (for [24]) or Wasserstein distance (for [6]) 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. [47] (and its improved version [46]) 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 [7] proposes to regard the explicit model as a teacher net who guides the training of implicit generator as a student net to produce samples that could overcome the mode-collapse issue. The main drawback of cooperative training is that they indirectly optimize the discrepancy between the generator and data distribution via the energy model as a 'mediator', which leads to a fact that once the energy model gets stuck in a local optimum (e.g., mode-collapse or mode-redundance) the training for the generator would be affected. In other words, the training for two models would constrain rather than exactly compensate each other. In Table 1, we do a high-level comparison among the above-mentioned generative models w.r.t the objectives. Different from other methods, our model considers three discrepancies simultaneously as a triangle to jointly train the generator and the estimator, enabling them to compensate and reinforce each other.

B Proofs of Results in Section 3.2

B.1 Proof of Theorem 1

Proof. Fixing the Wasserstein critic D, we are going to solve for

$$\min_G \{\mathbb{E}_{\mathbf{y}\sim\mathbb{P}_G}[D(\mathbf{y})] + \lambda_2 \mathcal{S}(\mathbb{P}_G, \mathbb{P}_E)\}. \tag{14}$$

By definition, if there exists a $\mathbb{P}_E$-measure zero set with positive supp($\mathbb{P}$)-measure, then $\mathcal{S}(\mathbb{P}, \mathbb{P}_E) = \infty$. Hence, to solve (14), it suffices to consider distributions whose support belongs to supp($\mathbb{P}_E$). Since C^∞ is dense in L^2, we can restrict to those $\mathbb{P}$'s that are absolutely continuous with respect to $\mathbb{P}_E$:

$$\inf_{\mathbb{P}} \left\{\mathbb{E}_{\mathbb{P}}[D] + \lambda_2 \cdot \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[\nabla \log(\mathbb{P}(\mathbf{x})/\mathbb{P}_E(\mathbf{x}))^\top k(\mathbf{x}, \mathbf{x}')\nabla \log(\mathbb{P}(\mathbf{x}')/\mathbb{P}_E(\mathbf{x}'))]\right\}.$$

Set $h(\mathbf{x}) := \mathbb{P}(\mathbf{x})/\mathbb{P}_E(\mathbf{x}) - 1$, the problem above becomes

$$\mathbb{E}_{\mathbb{P}_E}[D] + \min_{h:\ \mathbb{E}_{\mathbb{P}_E}[h]=0} \left\{\mathbb{E}_{\mathbb{P}_E}[Dh] + \lambda_2 \cdot \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[\nabla \log(1 + h(\mathbf{x}))^\top k(\mathbf{x}, \mathbf{x}')\nabla \log(1 + h(\mathbf{x}'))]\right\}.$$

For the minimization problem above, invoking Lagrangian duality gives

$$\sup_{t\in\mathbb{R}} \min_h \left\{\mathbb{E}_{\mathbb{P}_E}[(D - t)h] + \lambda_2 \cdot \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[\nabla \log(1 + h(\mathbf{x}))^\top k(\mathbf{x}, \mathbf{x}')\nabla \log(1 + h(\mathbf{x}'))]\right\}.$$

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

$$\inf_h \left\{\mathbb{E}_{\mathbb{P}_E}[(D - t)h] + \lambda_2 \cdot \mathbb{E}_{\mathbf{x},\mathbf{x}'\sim\mathbb{P}_E}[(1 + h(\mathbf{x})))\nabla h(\mathbf{x})^\top k(\mathbf{x}, \mathbf{x}')\nabla h(\mathbf{x}')(1 + h(\mathbf{x}'))]\right\}.$$

Consider a further approximation

$$\inf_h \left\{ \mathbb{E}_{\mathbb{P}_E}[Dh] + \lambda_2 \cdot \mathbb{E}_{\mathbf{x},\mathbf{x}' \sim \mathbb{P}_E}[\nabla h(\mathbf{x})^\top k(\mathbf{x},\mathbf{x}') \nabla h(\mathbf{x}')] \right\}. \tag{15}$$

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

$$-\frac{1}{4\lambda_2} \inf_{t\in\mathbb{R}} \|D - t\|_{H^{-1}(\mathbb{P}_E;k)}.$$

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.

□

B.2 Proof for Theorem 2

Proof. Essentially the result is a consequence of distributionally robust optimization with Wasserstein metric [11, 10]. Here we provide a simplified version for completeness. Consider

$$\min_G \{\lambda_2 \mathbb{E}_{\mathbf{y}\sim\mathbb{P}_G}[\mathcal{A}_{\mathbb{P}_E}\mathbf{f}(\mathbf{y})] + \mathcal{W}(\mathbb{P}_{\text{real}}, \mathbb{P}_G)\}.$$

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

$$\min_\gamma \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma} \left[\lambda_2 \mathcal{A}_{\mathbb{P}_E}\mathbf{f}(\mathbf{y})] + ||\mathbf{x} - \mathbf{y}||\right],$$

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

$$\begin{aligned} &\min_{\{\gamma_\mathbf{x}\}_\mathbf{x}} \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{real}}} \left[\mathbb{E}_{\mathbf{y}\sim\gamma_\mathbf{x}} \left[\lambda_2 \mathcal{A}_{\mathbb{P}_E}\mathbf{f}(\mathbf{y}) + ||\mathbf{x} - \mathbf{y}|| \mid \mathbf{x}\right]\right] \\ &= \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{real}}} \left[\min_{\gamma_\mathbf{x}} \mathbb{E}_{\mathbf{y}\sim\gamma_\mathbf{x}} \left[\lambda_2 \mathcal{A}_{\mathbb{P}_E}\mathbf{f}(\mathbf{y}) + ||\mathbf{x} - \mathbf{y}|| \mid \mathbf{x}\right]\right] \end{aligned}$$

where the minimization is over $\gamma_\mathbf{x}$, all conditional distributions of $\mathbf{y}$ given $\mathbf{x}$. If $\text{Lip}(\lambda_2 \mathcal{A}_{\mathbb{P}_E}\mathbf{f}) < 1$, then the minimal value equals $-\infty$, otherwise the minimal value equals $\lambda_2 \mathbb{E}_{\mathbf{y}\sim\mathbb{P}_{\text{real}}}[\mathcal{A}_{\mathbb{P}_E}\mathbf{f}(\mathbf{y})]$. Hence the proof is completed. □

C Proofs and More Discussions in Section 4

C.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 $\mathbb{E}[x] = \mathbb{E}[z] = 1$. Assume $\text{Var}[z] = 1$ 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 (8); when $\lambda < 0$ (likelihood-regularization), the the gradient of regularization term pushes θ to shrink, which helps for convergence; when $\lambda > 0$ (entropy-regularization), the added term forms an amplifiying strength on θ and leads to divergence.

Interestingly, the added terms $\frac{\lambda_1}{2}(1 + \phi)^2 + \frac{\lambda_2}{2}(\theta + \phi)^2$ in (10) 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 θ and ϕ 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 (8) 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 $[\psi^*, \theta^*, \phi^*] = [0, 1, -1]$.

C.2 Proof for Proposition 1

Proof. Instead of directly studying the optimization for (10), we first prove the following problem will converge to the unique optimum,

$$\min_{\theta} \max_{\psi} \min_{\phi} \theta\psi + \theta\phi + \frac{1}{2}\theta^2 + \phi^2. \tag{16}$$

Applying alternate SGD we have the following iterations:

$$\psi_{t+1} = \psi_t + \eta * \theta_t,$$

$$\phi_{t+1} = \phi_t - \eta * (\theta_t + 2\phi_t) = (1 - 2\eta)\phi_t - \eta\theta_t,$$

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

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 λ):

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

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

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

where λ_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 ψ, ϕ and θ in (16) by ψ', ϕ' and θ' respectively and conduct a change of variable: let $\theta' = 1 - \theta$ and $\phi' = -1 - \phi$. Then we get the conclusion in the proposition.

□

C.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 [15, 28, 14, 13]. The general bilinear function can be written as

$$F(\boldsymbol{\psi}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{A}\boldsymbol{\psi} - \mathbf{b}^\top\boldsymbol{\theta} - \mathbf{c}^\top\boldsymbol{\psi}, \tag{17}$$

where $\boldsymbol{\psi}, \boldsymbol{\theta}$ are both r-dimensional vectors and the objective is $\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\psi}} F(\boldsymbol{\psi}, \boldsymbol{\theta})$ 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 from the past [14], crossing the curl [13] and consensus optimization [28]. Also, [28] shows that it is the interaction term which generates non-zero values for $\nabla_{\boldsymbol{\theta}\boldsymbol{\psi}}F$ and $\nabla_{\boldsymbol{\psi}\boldsymbol{\theta}}F$ 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 $\boldsymbol{\phi}$ (corresponding to the estimator in our model) the original system and tackle the following problem:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\psi}} \min_{\boldsymbol{\phi}} F(\boldsymbol{\psi}, \boldsymbol{\theta}) + \alpha H(\boldsymbol{\phi}, \boldsymbol{\theta}), \tag{18}$$

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

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

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

$$\nabla_{\boldsymbol{\psi}} L(\boldsymbol{\psi}^*, \boldsymbol{\phi}, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\psi}^*, \boldsymbol{\theta}) = 0, \tag{19}$$

$$\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\psi}, \boldsymbol{\phi}, \boldsymbol{\theta}^*) = \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\psi}, \boldsymbol{\theta}^*) + \nabla_{\boldsymbol{\theta}} H(\boldsymbol{\phi}, \boldsymbol{\theta}^*) = \frac{1}{2}(\mathbf{B} + \mathbf{B}^\top)(\boldsymbol{\theta}^* + \boldsymbol{\phi}), \tag{20}$$

$$\nabla_{\boldsymbol{\phi}} L(\boldsymbol{\psi}, \boldsymbol{\phi}, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\phi}} H(\boldsymbol{\phi}, \boldsymbol{\theta}) = \frac{1}{2}(\mathbf{B} + \mathbf{B}^\top)(\boldsymbol{\phi} + \boldsymbol{\theta}), \tag{21}$$

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

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* $\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\psi}} \min_{\boldsymbol{\phi}} L(\boldsymbol{\psi}, \boldsymbol{\phi}, \boldsymbol{\theta})$ *using alternate SGD algorithm, i.e.,*

$$\begin{aligned} \boldsymbol{\psi}_{t+1} &= \boldsymbol{\psi}_t + \eta \nabla_{\boldsymbol{\psi}} L(\boldsymbol{\theta}_t, \boldsymbol{\psi}_t, \boldsymbol{\phi}_t), \\ \boldsymbol{\phi}_{t+1} &= \boldsymbol{\phi}_t - \eta \nabla_{\boldsymbol{\phi}} L(\boldsymbol{\theta}_t, \boldsymbol{\psi}_{t+1}, \boldsymbol{\phi}_t), \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_t, \boldsymbol{\psi}_{t+1}, \boldsymbol{\phi}_{t+1}), \end{aligned} \tag{22}$$

we can achieve convergence to $[\boldsymbol{\psi}^*, \boldsymbol{\theta}^*, \boldsymbol{\phi}^*]$ *where* $\boldsymbol{\phi}^* = -\boldsymbol{\theta}^*$ *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* σ_{min} *(resp.* σ_{max}*) denotes the maximum (resp. minimum) singular value of matrix* $\mathbf{A}$.

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.,*

$$\boldsymbol{\psi}_{t+1} \in \boldsymbol{\psi}_0 + span(L(\boldsymbol{\psi}_0, \boldsymbol{\phi}, \boldsymbol{\theta}), \cdots, F(\boldsymbol{\psi}_t, \boldsymbol{\phi}, \boldsymbol{\theta})), \forall t \in \mathbb{N},$$

$$\boldsymbol{\phi}_{t+1} \in \boldsymbol{\psi}_0 + span(L(\boldsymbol{\psi}, \boldsymbol{\phi}_0, \boldsymbol{\theta}), \cdots, L(\boldsymbol{\psi}, \boldsymbol{\phi}_t, \boldsymbol{\theta})), \forall t \in \mathbb{N},$$

$$\boldsymbol{\theta}_{t+1} \in \boldsymbol{\psi}_0 + span(L(\boldsymbol{\psi}, \boldsymbol{\phi}, \boldsymbol{\theta}_0), \cdots, L(\boldsymbol{\psi}, \boldsymbol{\phi}, \boldsymbol{\theta}_t)), \forall t \in \mathbb{N},$$

Then we have

$$\widetilde{\boldsymbol{\psi}_t} = \mathbf{V}^\top(\boldsymbol{\psi}_t - \boldsymbol{\psi}^*), \quad \widetilde{\boldsymbol{\phi}_t} = \mathbf{U}^\top(\boldsymbol{\phi}_t - \boldsymbol{\phi}^*), \quad \widetilde{\boldsymbol{\theta}_t} = \mathbf{U}^\top(\boldsymbol{\theta}_t - \boldsymbol{\theta}^*),$$

where $\mathbf{U}$ *and* $\mathbf{V}$ *are the singular vectors decomposed by matrix* $\mathbf{A}$ *using SVD decomposition, i.e.,* $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ *and the triple* $([\widetilde{\boldsymbol{\psi}_t}]_i, [\widetilde{\boldsymbol{\phi}_t}]_i, [\widetilde{\boldsymbol{\theta}_t}]_i)_{1 \le i \le r}$ *follows the update rule with step size* $\sigma_i \eta$ *as the same optimization method on a unidimensional problem*

$$\min_{\theta} \max_{\psi} \min_{\phi} \theta\psi + \theta\phi + \frac{1}{2}\theta^2 + \frac{1}{2}\phi^2, \tag{23}$$

with step size η, *where* σ_i *denotes the i-th singular value on the diagonal of* $\mathbf{D}$.

Proof. The proof is extended from the proof of Lemma 3 in [14]. The general class of first-order optimization methods derive the following updations:

$$\boldsymbol{\psi}_{t+1} = \boldsymbol{\psi}_0 + \sum_{s=0}^{t+1} \rho_{st}(\mathbf{A}^\top \boldsymbol{\theta}_s - \mathbf{c}) = \psi_0 + \sum_{s=0}^{t+1} \rho_{st} \mathbf{A}^\top(\boldsymbol{\theta}_s - \boldsymbol{\theta}^*),$$

$$\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_0 + \frac{1}{2}\sum_{s=0}^{t+1} \delta_{st}(\mathbf{B} + \mathbf{B}^\top)(\boldsymbol{\theta}_s + \boldsymbol{\phi}_s),$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_0 + \sum_{s=0}^{t+1} \mu_{st}[\mathbf{A}(\boldsymbol{\psi}_s - \boldsymbol{\psi}^*) + \frac{1}{2}(\mathbf{B} + \mathbf{B}^\top)(\boldsymbol{\theta}_s + \boldsymbol{\phi}_s)],$$

where $\rho_{st}, \delta_{st}, \mu_{st} \in \mathbb{R}$ depend on specific optimization method (for example, in SGD, $\rho_{tt} = \delta_{tt} = \mu_{tt}$ remain as a non-zero constant for $\forall t$ and other coefficients are zero).

Using SVD $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ and the fact $\theta^* = -\phi^*$, $\mathbf{B} = (\mathbf{U}\mathbf{D}\mathbf{D}^\top\mathbf{U}^\top) = \mathbf{D}$, we have

$$\mathbf{V}^\top(\boldsymbol{\psi}_{t+1} - \boldsymbol{\psi}^*) = \mathbf{V}^\top(\boldsymbol{\psi}_0 - \boldsymbol{\psi}^*) + \sum_{s=0}^{t+1} \rho_{st}\mathbf{D}^\top\mathbf{U}^\top(\boldsymbol{\theta}_s - \boldsymbol{\theta}^*)$$

$$\mathbf{U}^\top(\boldsymbol{\phi}_{t+1} - \boldsymbol{\phi}^*) = \mathbf{U}^\top(\boldsymbol{\phi}_0 - \boldsymbol{\phi}^*) + \sum_{s=0}^{t+1} \delta_{st}\mathbf{U}^\top\mathbf{D}(\boldsymbol{\theta}_s - \boldsymbol{\theta}^*) + \mathbf{U}^\top\mathbf{D}(\boldsymbol{\phi}_s - \boldsymbol{\phi}^*),$$

$$\mathbf{U}^\top(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*) = \mathbf{U}^\top(\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*) + \sum_{s=0}^{t+1} \rho_{st}[\mathbf{D}\mathbf{V}^\top(\boldsymbol{\psi}_s - \boldsymbol{\psi}^*) + \mathbf{U}^\top\mathbf{D}(\boldsymbol{\theta}_s - \boldsymbol{\theta}^*) + \mathbf{U}^\top\mathbf{D}(\boldsymbol{\phi}_s - \boldsymbol{\phi}^*)],$$

and equivalently,

$$\widetilde{\boldsymbol{\psi}}_{t+1} = \widetilde{\boldsymbol{\psi}}_0 + \sum_{s=0}^{t+1} \rho_{st}\mathbf{D}^\top\widetilde{\boldsymbol{\theta}}_t, \quad \widetilde{\boldsymbol{\phi}}_t = \widetilde{\boldsymbol{\phi}}_0 + \sum_{s=0}^{t+1} \delta_{st}\mathbf{D}(\widetilde{\boldsymbol{\theta}}_t + \widetilde{\boldsymbol{\phi}}_t),$$

$$\widetilde{\boldsymbol{\theta}}_{t+1} = \widetilde{\boldsymbol{\theta}}_0 + \sum_{s=0}^{t+1} \rho_{st}\mathbf{D}(\widetilde{\boldsymbol{\psi}}_t + \widetilde{\boldsymbol{\theta}}_t + \widetilde{\boldsymbol{\phi}}_t).$$

Note that $\mathbf{D}$ 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:

$$[\widetilde{\boldsymbol{\psi}}_{t+1}]_i = [\widetilde{\boldsymbol{\psi}}_0]_i + \sum_{s=0}^{t+1} \rho_{st}\sigma_i[\widetilde{\boldsymbol{\theta}}_t]_i, \quad [\widetilde{\boldsymbol{\phi}}_t]_i = [\widetilde{\boldsymbol{\phi}}_0]_i + \sum_{s=0}^{t+1} \delta_{st}\sigma_i([\widetilde{\boldsymbol{\theta}}_t]_i + [\widetilde{\boldsymbol{\phi}}_t]_i),$$

$$[\widetilde{\boldsymbol{\theta}}_{t+1}]_i = [\widetilde{\boldsymbol{\theta}}_0]_i + \sum_{s=0}^{t+1} \rho_{st}\sigma_i([\widetilde{\boldsymbol{\psi}}_t]_i + [\widetilde{\boldsymbol{\theta}}_t]_i + [\widetilde{\boldsymbol{\phi}}_t]_i).$$

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 $\sigma_i\eta$ as problem in (23). $\square$

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 $(1-\eta_1+\eta_2^2)(1+\eta_2-\eta_1^2)$ where $\eta_1 = \eta\sigma_{max}$, $\eta_2 = \eta\sigma_{min}$ and σ_{max} (resp. σ_{min}) denotes the maximum (resp. minimum) singular value of matrix $\mathbf{A}$. We conclude the proof for Theorem 4.

Theorem 4 suggests that the added term $H(\boldsymbol{\phi}, \boldsymbol{\theta})$ with affiliated variables ϕ could help the SGD algorithm achieve convergence to the the same optimum points as directly optimizing $F(\boldsymbol{\psi}, \boldsymbol{\theta})$. Our method is related to consensus optimization algorithm [28] which adds a regularization term $\|\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\psi}, \boldsymbol{\theta})\| + \|\nabla_{\boldsymbol{\psi}} F(\boldsymbol{\psi}, \boldsymbol{\theta})\|$ to (17) resulting extra quadratic terms for $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. The disadvantage of such method is the requirement of Hessian matrix of $F(\boldsymbol{\psi}, \boldsymbol{\theta})$ which is computational expensive for high-dimensional data. By contrast, our solution only requires the first-order derivatives.

C.4 Strongly Convexity

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

$$(\nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta}, \cdot) - \nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta}', \cdot))^\top(\boldsymbol{\theta} - \boldsymbol{\theta}') \geq \mu\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2, \forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Omega_{\boldsymbol{\theta}},$$

$$(\nabla_{\boldsymbol{\phi}} H(\cdot, \boldsymbol{\phi}) - \nabla_{\boldsymbol{\phi}} H(\cdot, \boldsymbol{\phi}'))^\top(\boldsymbol{\phi} - \boldsymbol{\phi}') \geq \mu\|\boldsymbol{\phi} - \boldsymbol{\phi}'\|_2^2, \forall \boldsymbol{\phi}, \boldsymbol{\phi}' \in \Omega_{\boldsymbol{\phi}}.$$

Bedises, $F(\boldsymbol{\theta}, \boldsymbol{\psi})$ is μ-strongly convex for $\boldsymbol{\theta}$ and μ-strongly concave for ψ so it satisfies:

$$(\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}, \cdot) - \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}', \cdot))^\top(\boldsymbol{\theta} - \boldsymbol{\theta}') \geq \mu\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2, \forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Omega_{\boldsymbol{\theta}},$$

$$(\nabla_{\boldsymbol{\psi}} F(\cdot, \boldsymbol{\psi}') - \nabla_{\boldsymbol{\psi}} F(\cdot, \boldsymbol{\psi}))^\top(\boldsymbol{\psi} - \boldsymbol{\psi}') \geq \mu\|\boldsymbol{\psi} - \boldsymbol{\psi}'\|_2^2, \forall \boldsymbol{\psi}, \boldsymbol{\psi}' \in \Omega_{\boldsymbol{\psi}}.$$

In section 3.1.2, we also define $h(\boldsymbol{\omega}_h) = \nabla_{\boldsymbol{\theta}} H + \nabla_{\boldsymbol{\phi}} H$ and $f(\boldsymbol{\omega}_f) = \nabla_{\boldsymbol{\theta}} F - \nabla_{\boldsymbol{\psi}} F$, so the above condition can be written in a more compact form,

$$(h(\boldsymbol{\omega}_h) - h(\boldsymbol{\omega}_h'))^\top(\boldsymbol{\omega}_h - \boldsymbol{\omega}_h') \geq \mu\|\boldsymbol{\omega}_h - \boldsymbol{\omega}_h'\|_2^2, \forall \boldsymbol{\omega}_h, \boldsymbol{\omega}_h' \in \Omega_h,$$

$$(f(\boldsymbol{\omega}_f) - f(\boldsymbol{\omega}_f'))^\top(\boldsymbol{\omega}_f - \boldsymbol{\omega}_f') \geq \mu\|\boldsymbol{\omega}_f - \boldsymbol{\omega}_f'\|_2^2, \forall \boldsymbol{\omega}_f, \boldsymbol{\omega}_f' \in \Omega_f.$$

C.5 Proof for Theorem 2

The proof relies on two lemmas,

Lemma 2. *For any $\boldsymbol{\omega} \in \Omega$ and $\boldsymbol{\omega}^+ = P_\Omega(\boldsymbol{\omega} + \boldsymbol{u})$, then we have*

$$\|\boldsymbol{\omega}^+ - \boldsymbol{\omega}\|_2^2 \leq u^\top(\boldsymbol{\omega}^+ - \boldsymbol{\omega}).$$

Proof. Since $\boldsymbol{\omega}^+$ is a projection of $\boldsymbol{\omega} + \boldsymbol{u}$ on a convex set Ω, we have

$$(\boldsymbol{\omega}^+ - (\boldsymbol{\omega} + \boldsymbol{u}))^\top(\boldsymbol{\omega}^+ - \boldsymbol{\omega}) \leq 0. \tag{24}$$

Rearranging the above inequality one can easily get the lemma. □

Lemma 3. *If function $\Phi(\boldsymbol{\omega})$ is μ-strongly convex, we have*

$$\mu\|\boldsymbol{\omega} - \boldsymbol{\omega}^*\|_2^2 \leq \nabla\Phi(\boldsymbol{\omega})^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*).$$

Similarly, if $\Phi(\boldsymbol{\omega})$ is μ-strongly concave, we have $\mu(\boldsymbol{\omega} - \boldsymbol{\omega}^*) \leq \nabla - \Phi(\boldsymbol{\omega})^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*)$.

Proof. By optimality of $\boldsymbol{\omega}^*$, we have

$$\nabla\Phi(\boldsymbol{\omega}^*)^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*) \geq 0.$$

Since Φ is μ-convex, we can further derive

$$\mu\|\boldsymbol{\omega} - \boldsymbol{\omega}^*\|_2^2 \leq \nabla\Phi(\boldsymbol{\omega}^*)^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*) + \mu\|\boldsymbol{\omega} - \boldsymbol{\omega}^*\|_2^2 \leq \nabla\Phi(\boldsymbol{\omega})^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*).$$

□

Proof. (Proof for Theorem 3) We apply Lemma 2 to (13) with $(\boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{\omega}^+) = (\boldsymbol{\omega}_f^{t+1/2}, -\eta f(\boldsymbol{\omega}_f^{t+1/2}), \boldsymbol{\omega}_f^{t+1})$ and we have

$$\|\boldsymbol{\omega}_f^{t+1} - \boldsymbol{\omega}_f^{t+1/2}\| \leq -\eta f(\boldsymbol{\omega}_f^{t+1/2})^\top(\boldsymbol{\omega}_f^{t+1} - \boldsymbol{\omega}_f^{t+1/2}) \tag{25}$$

Then we have

$$\begin{aligned}
\|\boldsymbol{\omega}_f^{t+1} - \boldsymbol{\omega}_f^*\|_2^2 &= \|\boldsymbol{\omega}_f^{t+1/2} - \boldsymbol{\omega}_f^* + \boldsymbol{\omega}_f^{t+1} - \boldsymbol{\omega}_f^{t+1/2}\|_2^2 \\
&\leq 2\|\boldsymbol{\omega}_f^{t+1/2} - \boldsymbol{\omega}_f^*\|_2^2 + 2\|\boldsymbol{\omega}_f^{t+1} - \boldsymbol{\omega}_f^{t+1/2}\|_2^2 \quad (\textit{by} \quad \|a+b\|_2^2 \leq 2\|a\|_2^2 + 2\|b\|_2^2) \\
&\leq 2\|\boldsymbol{\omega}_f^{t+1/2} - \boldsymbol{\omega}_f^*\|_2^2 - 2\eta f(\boldsymbol{\omega}_f^{t+1/2})^\top(\boldsymbol{\omega}_f^{t+1} - \boldsymbol{\omega}_f^{t+1/2}).
\end{aligned} \tag{26}$$

According to Lemma 3, we have

$$2\eta f(\boldsymbol{\omega}_f^{t+1/2})^\top(\boldsymbol{\omega}_f^{t+1/2} - \boldsymbol{\omega}_f^*) \geq 2\eta\mu\|\boldsymbol{\omega}_f^{t+1/2} - \boldsymbol{\omega}_f^*\|_2^2. \tag{27}$$

Plug (27) into (26) and we get

$$\|\boldsymbol{\omega}_f^{t+1} - \boldsymbol{\omega}_f^*\|_2^2 \leq (2 - 2\eta\mu)\|\boldsymbol{\omega}_f^{t+1/2} - \boldsymbol{\omega}_f^*\|_2^2. \tag{28}$$

The above inequality is equivalent to

$$\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^*\|_2^2 + \|\boldsymbol{\psi}^{t+1} - \boldsymbol{\psi}^*\|_2^2 \leq (2 - 2\eta\mu)(\|\boldsymbol{\theta}^{t+1/2} - \boldsymbol{\theta}^*\|_2^2 + \|\boldsymbol{\psi}^t - \boldsymbol{\psi}^*\|_2^2). \tag{29}$$

Similarly, one can obtain

$$\|\boldsymbol{\omega}_h^{t+1} - \boldsymbol{\omega}_h^*\|_2^2 \leq (2 - 2\eta\mu)\|\boldsymbol{\omega}_h^{t+1/2} - \boldsymbol{\omega}_h^*\|_2^2, \tag{30}$$

i.e.,

$$\|\boldsymbol{\theta}^{t+1/2} - \boldsymbol{\theta}^*\|_2^2 + \|\boldsymbol{\phi}^{t+1} - \boldsymbol{\phi}^*\|_2^2 \leq (2 - 2\eta\mu)(\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|_2^2 + \|\boldsymbol{\phi}^t - \boldsymbol{\phi}^*\|_2^2). \tag{31}$$

Combining (29) and (31) we have

$$\begin{aligned}
\|\boldsymbol{\omega}^{t+1} - \boldsymbol{\omega}^*\|_2^2 &= \|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^*\|_2^2 + \|\boldsymbol{\psi}^{t+1} - \boldsymbol{\psi}^*\|_2^2 + \|\boldsymbol{\phi}^{t+1} - \boldsymbol{\phi}^*\|_2^2 \\
&\leq (1 - 2\eta\mu)\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|_2^2 + (2 - 2\eta\mu)\|\boldsymbol{\psi}^{t+1} - \boldsymbol{\psi}^*\|_2^2 + (2 - 2\eta\mu)\|\boldsymbol{\phi}^{t+1} - \boldsymbol{\phi}^*\|_2^2 \\
&\leq (2 - 2\eta\mu)\|\boldsymbol{\omega}^t - \boldsymbol{\omega}^*\|_2^2.
\end{aligned} \tag{32}$$

Hence, if $\frac{1}{2\mu} < \eta < \frac{1}{\mu}$ we have $0 < 2 - 2\eta\mu < 1$ and $\|\boldsymbol{\omega}^t - \boldsymbol{\omega}^*\|_2^2 \leq (2 - 2\eta\mu)^t\|\boldsymbol{\omega}^0 - \boldsymbol{\omega}^*\|_2^2$ □

D Details for Experiment Setup

D.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 $\begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix} = \sigma_1 \mathbf{I}$ and the mean value is $[r_1 \cos t, r_1 \sin t]$, where $t = \frac{2\pi \cdot k}{8}$, $k = 1, \cdots, 8$, for the inner circle, and $[r_2 \cos t, r_2 \sin t]$, where $t = \frac{2\pi \cdot k}{16}$, $k = 1, \cdots, 16$ 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 $\mathcal{N}(\mathbf{0}, \sigma_0 \mathbf{I})$ where $\sigma_0 = 2$ 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 $\begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} = \sigma_2 \mathbf{I}$ and the mean value is $[-c_1 \cos c_1, c_1 \sin c_1]$, where $c_1 = \frac{2\pi}{3} + linspace(0, 0.5, 50) \cdot 2\pi$, for one spiral, and $[c_2 \cos c_2, -c_2 \sin c_2]$, where $c_2 = \frac{2\pi}{3} + linspace(0, 0.5, 50) \cdot 2\pi$ 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 $[100, 200, 300, 500, 700, 1000, 2000]$.

D.2 Model Specifications and Training Algorithm

In different tasks, we consider different model specifications in order to meet the demand of capacify as well as test the effectiveness under various settings. Our proposed framework (5) adopts Wasserstein distance for the first term and two Stein discrepancies for the second and the third terms. We can write (5) as a more general form

$$\min_{\theta, \phi} \mathcal{D}_1(\mathbb{P}_{\text{real}}, \mathbb{P}_G) + \lambda_1 \mathcal{D}_2(\mathbb{P}_{\text{real}}, \mathbb{P}_E) + \lambda_2 \mathcal{D}_3(\mathbb{P}_G, \mathbb{P}_E), \tag{33}$$

where $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ denote three general discrepancy measures for distributions. As stated in our remark, $\mathcal{D}_1$ can be specified as arbitrary discrepancy measures for implicit generative models. Here we also use JS divergence, the objective for valina 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 (5) can be specified by KSD (as defined by $\mathcal{S}_k$ in (3)) or general Stein discrepancy with an extra critic (as defined by $\mathcal{S}$ in (2)). Hence, the two specifications for $\mathcal{D}_1$ and the two for $\mathcal{D}_2$ ($\mathcal{D}_3$) 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 $\mathcal{D}_2$ and $\mathcal{D}_3$ 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 comutationally 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 θ, ϕ, ψ and π respectively. Specifically, in one iteration, we optimize the generator over θ and the estimator over ϕ with one step respectively, and then optimize the Wasserstein critic over ψ with n_d steps and the Stein critic over π 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.

D.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.

$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	Objective
$\mathcal{W}$	$\mathcal{S}$	$\mathcal{S}$	$\min_\theta \min_\phi \max_\psi \max_\pi \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{data}}[d_\psi(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p_0}[d_\psi(G_\theta(\mathbf{z}))]$ $+\lambda_1 \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{data}}[\mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\mathbf{x})]] + \lambda_2 \mathbb{E}_{\mathbf{z}\sim {\mid}_0}[\mathcal{A}_{p_\phi}[\mathbf{f}_\pi(G_\theta(\mathbf{z}))]]$
$\mathcal{W}$	$\mathcal{S}_k$	$\mathcal{S}_k$	$\min_\theta \min_\phi \max_\psi \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{data}}[d_\psi(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p_0}[d_\psi(G_\theta(\mathbf{z}))]$ $+\lambda_1 \mathbb{E}_{\mathbf{x},\mathbf{x}'\sim\mathbb{P}_{data}}[u_{p_\phi}(x, x')] + \lambda_2 \mathbb{E}_{\mathbf{z},\mathbf{z}'\sim p_0}[u_{p_\phi}(G_\theta(\mathbf{z}), G_\theta(\mathbf{z}'))]$
$\mathcal{JS}$	$\mathcal{S}$	$\mathcal{S}$	$\min_\theta \min_\phi \max_\psi \max_\pi \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_r}[\log(d_\psi(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}\sim p_0}[\log(1 - d_\psi(G_\theta(\mathbf{z})))]$ $+\lambda_1 \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{data}}[\mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\mathbf{x})]] + \lambda_2 \mathbb{E}_{\mathbf{z}\sim {\mid}_0}[\mathcal{A}_{p_\phi}[\mathbf{f}_\pi(G_\theta(\mathbf{z}))]]$
$\mathcal{JS}$	$\mathcal{S}_k$	$\mathcal{S}_k$	$\min_\theta \min_\phi \max_\psi \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_r}[\log(d_\psi(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}\sim p_0}[\log(1 - d_\psi(G_\theta(\mathbf{z})))]$ $+\lambda_1 \mathbb{E}_{\mathbf{x},\mathbf{x}'\sim\mathbb{P}_{data}}[u_{p_\phi}(x, x')] + \lambda_2 \mathbb{E}_{\mathbf{z},\mathbf{z}'\sim p_0}[u_{p_\phi}(G_\theta(\mathbf{z}), G_\theta(\mathbf{z}'))]$

Table 4: Objectives for different specifications of $\mathcal{D}_1(\mathbb{P}_{\text{real}}, \mathbb{P}_G)$, $\mathcal{D}_2(\mathbb{P}_{\text{real}}, \mathbb{P}_E)$ and $\mathcal{D}_3(\mathbb{P}_G, \mathbb{P}_E)$. We specify $\mathcal{D}_1$ as Wasserstein distance or JS divergence in our paper and for $\mathcal{D}_2$ and $\mathcal{D}_3$ we consider the general Stein discrepancy or kernel Stein discrepancy. Here we use $\mathcal{W}$, $\mathcal{JS}$ to denote Wasserstein distance and JS divergence respectively, and $\mathcal{S}$, $\mathcal{S}_k$ 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.

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 $4 - 128 - 128 - 2$, and all the Wasserstein critics (or the discriminators in JS-divergence-based GAN) are also a three-layer FC network with neuron size $2 - 128 - 128 - 1$. For the estimators, we set the expert number as 4 and the feature function $n(\mathbf{x})$ is a FC network with neuron size $2 - 128 - 128 - 4$. Then in the last layer we sum the outputs from each expert as the energy value $E(\mathbf{x})$. The activation units are searched within $[LeakyReLU, tanh, sigmoid, softplus]$. 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 $1-64-128-256-1$. 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 $100 - 256 - 128 - 64 - 1$. 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 value. The activation units are searched within $[ReLU, LeakyReLU, tanh]$. The learning rate $[2e-5, 2e-4, 2e-3, 2e-2]$ 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.

D.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, $MMD(X, Y) = \|\frac{1}{n}\sum_{i=1}^{n}\Phi(x_i) - \frac{1}{m}\sum_{j=1}^{m}\Phi(y_i)\|$ where x_i and y_j denote samples from X and Y respectively and Φ 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-Cirlce dataset, we define the generated points whose distance from the nearest Gaussian component is less than σ_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 $\mathbb{H}(y|x) \approx \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{10} p(y_k|x_i)\log p(y_k|x_i)$ where x is a generated instance and y denotes the predicted class given x from a pre-trained classifier. CEPC

Algorithm 1: Training Algorithm for Stein Bridging

```
1  REQUIRE: observed training samples $\{\mathbf{x}\} \sim \mathbb{P}_{real}$.
2  REQUIRE: $\theta_0$, $\phi_0$, $\psi_0$, $\pi_0$, 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 $\mathcal{D}_2$ and $\mathcal{D}_3$ (we suggest increasing $\lambda_2$ from 0 to 1 through
     training). $n_d = 5$, $n_c = 5$ 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 do
4      for $n = 1, \cdots, n_d$ do
5          Sample $B$ true samples $\{\mathbf{x}_i\}_{i=1}^B$ from $\{\mathbf{x}\}$;
6          Sample $B$ random noise $\{\mathbf{z}_i\}_{i=1}^B \sim P_0$ and obtain generated samples $\widetilde{\mathbf{x}}_\mathbf{i} = G_\theta(\mathbf{z}_i)$ ;
7          $\mathcal{L}_{dis} = \frac{1}{B}\sum_{i=1}^B d_\psi(\mathbf{x}_i) - d_\psi(\widetilde{\mathbf{x}}_\mathbf{i}) - \lambda(\|\nabla_{\hat{\mathbf{x}}_\mathbf{i}} d_\psi(\hat{\mathbf{x}}_\mathbf{i})\| - 1)^2$ // the last term is for gradient
             penalty in WGAN-GP where $\hat{\mathbf{x}}_\mathbf{i} = \epsilon_i\mathbf{x}_i + (1-\epsilon_i)\widetilde{\mathbf{x}}_\mathbf{i}, \epsilon_i \sim U(0,1)$;
8          $\psi_{k+1} \leftarrow Adam(-\mathcal{L}_{dis}, \psi_k, \alpha^I, \beta_1^I, \beta_2^I)$// update the Wasserstein critic;
9      for $n = 1, \cdots, n_c$ do
10         Sample $B$ true samples $\{\mathbf{x}_i\}_{i=1}^B$ from $\{\mathbf{x}\}$;
11         Sample $B$ random noise $\{\mathbf{z}_i\}_{i=1}^B \sim P_0$ and obtain generated samples $\widetilde{\mathbf{x}}_\mathbf{i} = G_\theta(\mathbf{z}_i)$ ;
12         $\mathcal{L}_{critic} = \frac{1}{B}\sum_{i=1}^B \lambda_1\mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\mathbf{x})] + \lambda_2\mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\widetilde{\mathbf{x}}_\mathbf{i})]$;
13         $\pi_{k+1} \leftarrow Adam(-\mathcal{L}_{critic}, \pi_k, \alpha^E, \beta_1^E, \beta_2^E)$// update the Stein critic;
14     Sample $B$ random noise $\{\mathbf{z}_i\}_{i=1}^B \sim P_0$ and obtain generated samples $\widetilde{\mathbf{x}}_\mathbf{i} = G_\theta(\mathbf{z}_i)$ ;
15     $\mathcal{L}_{est} = \frac{1}{B}\sum_{i=1}^B \lambda_1\mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\mathbf{x})] + \lambda_2\mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\widetilde{\mathbf{x}}_\mathbf{i})]$;
16     $\phi_{k+1} \leftarrow Adam(\mathcal{L}_{est}, \phi_k, \alpha^E, \beta_1^E, \beta_2^E)$// update the density estimator;
17     $\mathcal{L}_{gen} = \frac{1}{B}\sum_{i=1}^B -d_\psi(\widetilde{\mathbf{x}}_\mathbf{i}) + \lambda_2\mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\widetilde{\mathbf{x}}_\mathbf{i})]$;
18     $\theta_{k+1} \leftarrow Adam(\mathcal{L}_{gen}, \theta_k, \alpha^I, \beta_1^I, \beta_2^I)$// update the sample generator;
19 OUTPUT: trained sample generator $G_\theta(\mathbf{z})$ and density estimator $p_\phi(\mathbf{x})$.
```

measures how well a given sample can be classfied 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.

	Two-Cirlce					Two-Spiral				
Method	MMD	HSR	KLD	JSD	AUC	MMD	HSR	KLD	JSD	AUC
GAN	0.0033	0.772	-	-	-	0.0082	0.583	-	-	-
GAN+LR	0.0106	0.391	-	-	-	0.0068	0.821	-	-	-
GAN+ER	0.0103	0.428	-	-	-	0.0071	0.780	-	-	-
GAN+VA	0.0118	0.295	-	-	-	0.0085	0.761	-	-	-
WGAN-GP	0.0010	0.841	-	-	-	0.0090	0.697	-	-	-
WGAN+LR	0.0013	0.840	-	-	-	0.0095	0.607	-	-	-
WGAN+ER	0.0008	0.830	-	-	-	0.0182	0.730	-	-	-
WGAN+VA	0.0016	0.835	-	-	-	0.0159	0.618	-	-	-
DEM	-	-	2.036	0.431	0.683	-	-	1.206	0.315	0.640
EGAN	-	-	3.350	0.474	0.616	-	-	1.916	0.445	0.499
DGM	0.0040	0.774	2.272	0.445	0.600	0.0019	0.833	1.725	0.414	0.589
Joint-JS	0.0037	**0.883**	1.104	0.297	**0.962**	0.0031	0.717	0.655	0.193	0.808
Joint-W	**0.0007**	0.844	**1.030**	**0.281**	0.961	**0.0003**	**0.909**	**0.364**	**0.110**	**0.810**

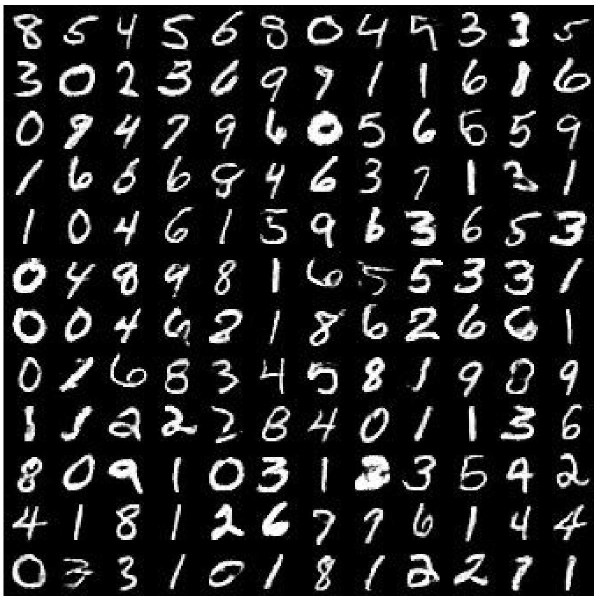
Table 5: Quantitative results including MMD (lower is better), HSR (higher is better) as the metrics for quality of generated samples and KLD (lower is better), JSD (lower is better), AUC (higher is better) as the metrics for accuracy of estimated densities on Two-Circle and Two-Spiral datasets.

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

Table 6: $l1$ and $l2$ distances between means of true digits and generated digits in each class on MNIST.

Class		'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'
WGAN-GP	l_1	80.8	82.7	40.2	69.3	44.7	59.2	77.6	107.7	50.81	89.3
	l_2	1.75	1.84	0.92	1.57	1.04	1.40	1.78	2.32	1.78	1.92
WGAN+LR	l_1	78.4	79.2	73.8	86.0	75.8	77.2	106.7	103.0	56.5	92.3
	l_2	1.63	1.76	1.59	1.88	1.68	1.74	2.36	2.23	1.24	2.00
WGAN+ER	l_1	75.5	64.0	100.0	65.0	58.5	69.1	74.5	81.8	62.5	71.3
	l_2	1.56	1.45	2.04	1.43	1.35	1.57	1.67	1.82	1.40	1.58
WGAN+VA	l_1	60.9	70.0	79.4	62.7	63.0	73.9	76.2	77.2	59.8	66.4
	l_2	1.32	1.55	1.68	1.39	1.42	1.63	1.70	1.77	1.33	1.48
DGM	l_1	167.8	185.0	149.4	250.1	105.3	134.0	223.8	197.3	148.3	231.7
	l_2	3.67	4.14	3.15	5.41	2.39	3.04	4.68	4.51	3.24	5.25
Joint-W	l_1	59.3	58.1	77.3	54.8	58.1	65.1	63.9	82.8	59.1	63.2
	l_2	1.26	1.30	1.60	1.23	1.28	1.44	1.44	1.80	1.27	1.43

Table 7: $l1$ and $l2$ distances between means of true images and generated images in each class on CIFAR. (Class '0', '1', '2', '3', '4', '5', '6', '7', '8' and '9' stand for 'airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship' and 'truck' respectively.)


(a) Randomly sampled over all digits

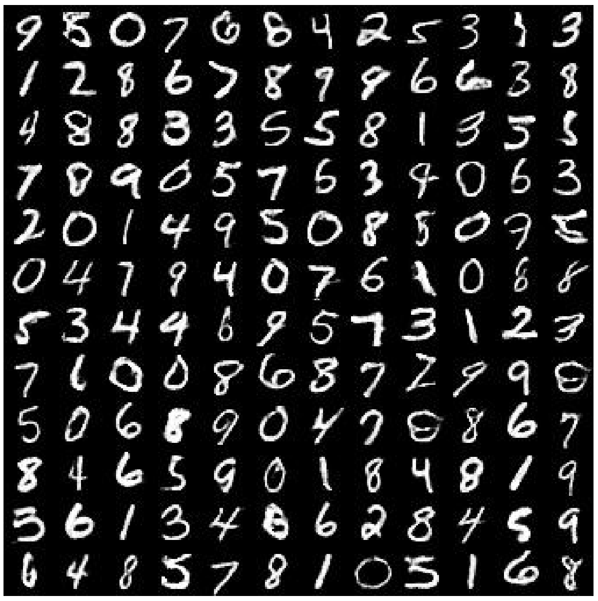
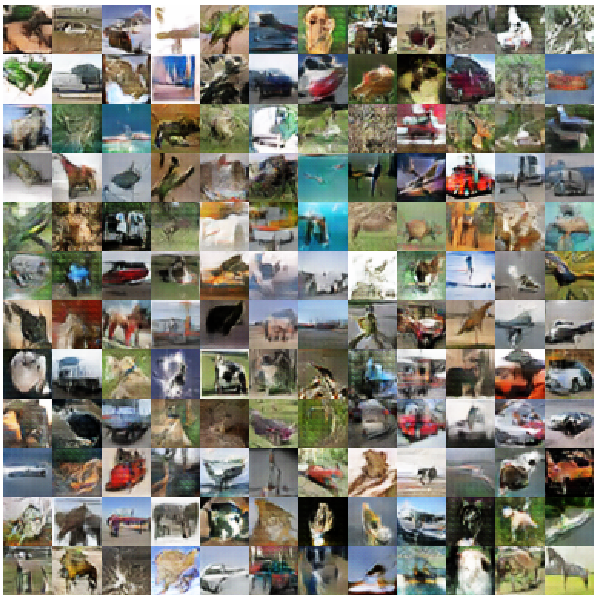

(b) Randomly sampled over digits with top 50% densities

Figure 9: Generated digits given by Joint-W on MNIST.


(a) Randomly sampled over all images

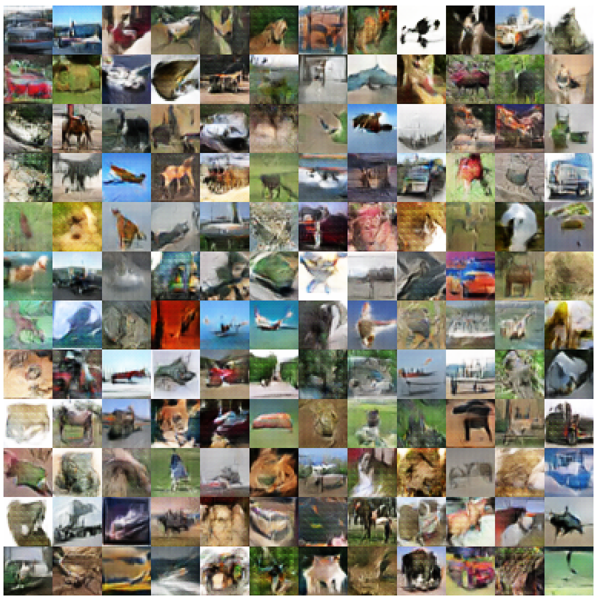
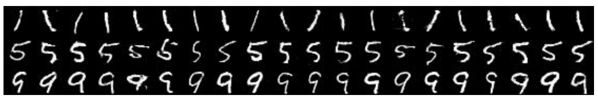
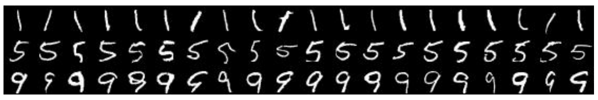

(b) Randomly sampled over images with top 50% densities

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


(a) Generated digits with highest densities

(b) Generated digits with lowest densities


(c) Real digits with highest densities

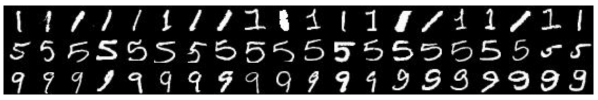
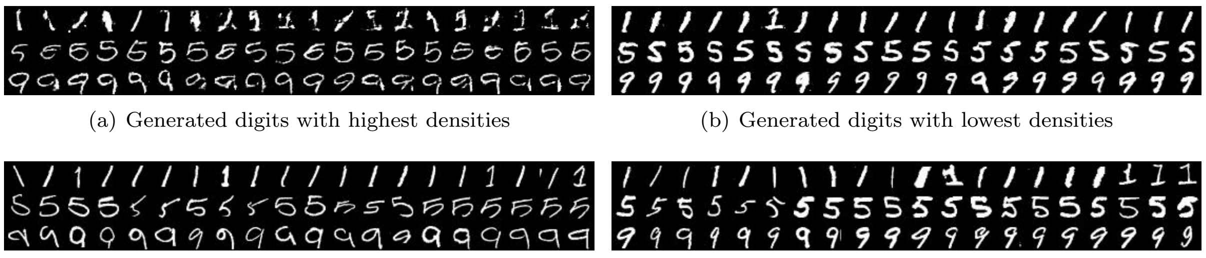

(d) Real digits with lowest densities

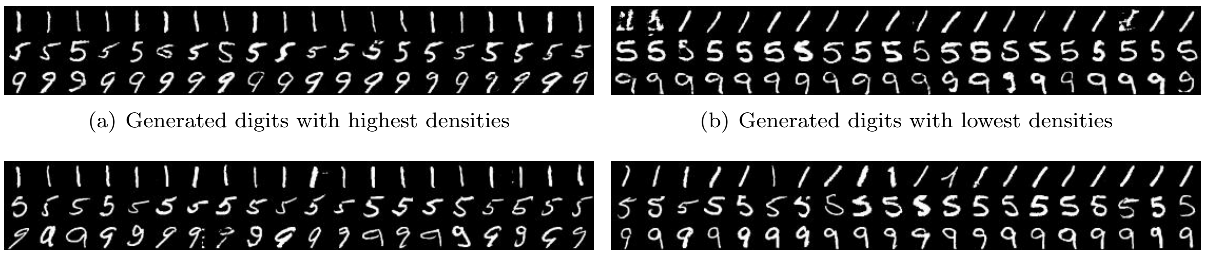
Figure 11: The generated digits (and real digits) with the highest densities and the lowest densities given by Joint-W.



(a) Generated digits with highest densities (b) Generated digits with lowest densities

(c) Real digits with highest densities (d) Real digits with lowest densities

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



(a) Generated digits with highest densities (b) Generated digits with lowest densities

(c) Real digits with highest densities (d) Real digits with lowest densities

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