

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

Qitian Wu¹ Rui Gao² Hongyuan Zha^{3,4}

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), targets a mapping from random noise to generated 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 (unnormalized) density estimator and an implicit sample generator with Stein discrepancy. We show that the Stein bridge enables mutual reinforcement between two models by inducing dual regularization effects. Convergence analysis and extensive experiments demonstrate that Stein Bridging i) improves training stability and sample quality of GAN, and ii) facilitates the density estimator to seek more modes in data and alleviate mode-collapse issue. Additionally, we discuss several applications of Stein Bridging and useful tricks in practical implementation.

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: explicit and implicit (Goodfellow et al., 2014). Explicit models define a (unnormalized) density function, while implicit models learn a mapping from easy-to-sample random noise to generated samples that have a hidden complex distribution without an explicit density form.

Both generative models have shown great power in various applications, but when used alone, they suffer from respective limitations. Explicit models have wide applications in undirected graphical models (LeCun et al., 2006; Salakhutdinov & Hinton, 2009; Hinton et al., 2006; Ngiam et al., 2011), random graph theory (Robins et al., 2007), energy-based reinforcement learning (Haarnoja et al., 2017), etc. However, the unknown normalizing constant makes the model hard to train and sample from, and explicit models may not be able to capture the complex structure of true samples while maintaining tractability. In contrast, implicit models are more flexible in training and easy to sample from, and in particular, generative adversarial networks (GANs) are powerful in learning representations of images, natural languages, graphs, etc. (Goodfellow et al., 2014; Radford et al., 2016; Arjovsky et al., 2017; Brock et al., 2019). Nevertheless, due to the minimax game between generator and discriminator/critic in GANs, the training process often suffers from instability, and produces undesirable samples that are often associated with missing modes in data or generating extra modes out of data. In view of this, it is natural to build a unified framework that takes advantage of the two models and lets them *compensate* each other.

Furthermore, there are many situations where an explicit (unnormalized) density estimator and a flexible implicit sampler could *reinforce* each other's training given effective information sharing. First, the density values given by explicit models can be a good metric that measures quality of samples (Dai et al., 2017), and such evaluation can be used for scoring generated samples given by implicit model or detecting outliers and noise in input true samples (Zhai et al., 2016). Second, the generated samples by implicit model could help to augment dataset especially when given insufficient true samples that would possibly make explicit

¹Shanghai Jiao Tong University ²University of Texas at Austin ³The Chinese University of Hong Kong (Shenzhen) ⁴Georgia Institute of Technology. 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. Correspondence to: Qitian Wu <echo740@sjtu.edu.cn>, Rui Gao <rui.gao@mcombs.utexas.edu>, Hongyuan Zha <zha@cc.gatech.edu>.

model fail to capture an accurate distribution. Third, in some specific tasks, we need to add some induction bias to the model but in most cases it is easy for one model to incorporate the induction bias while harder for another. For example, if we aim to obtain a certain type of generated samples, it would be difficult to mathematically enforce some constraints on implicit model, but we can consider a truncated density or energy function for explicit model. In this situation, the explicit model can guide the implicit one to generate samples that meet the requirement through joint training.

Motivated by discussions above, in this paper, we propose a joint learning framework that enables mutual *compensation* and *reinforcement* between 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 generated samples and estimated densities, is introduced and pushes the two models to achieve a consensus.

Theoretically, we show that the Stein bridge allows the two generative models to facilitate each other's training by imposing new regularizations on both models, which, in specific, help the generator to output high-quality samples and guide the energy model to avoid mode-collapse. We also prove that the joint training helps to stabilize GAN training via a convergence analysis. Moreover, extensive experiments on various tasks verify our theoretical findings as well as demonstrate significant improvement of inception score on CIFAR-10 compared with WGAN-GP and its improved variants. We also discuss three useful applications of Stein Bridging for i) detecting noise and outliers in data, ii) addressing data insufficiency and iii) enforcing induction bias for generative models.

Our contributions are summarized as follows: i) **New Methodology:** We propose a new joint training framework that unifies explicit and implicit generative models via Stein's method; ii) **Theoretical Insights:** We theoretically show that proposed model can enable mutual reinforcement between two models via dual regularizations and also stabilize the training; iii) **Multifaceted Experiments:** We apply our model to various datasets, compare it with several strong competitors and demonstrate its effectiveness and superiority on multiple tasks.

2. Background

In this section, we briefly provide some technical background related to our model.

Energy Model. The energy model assigns each data $\mathbf{x} \in \mathbb{R}^d$ with a scalar energy value $E_\phi(\mathbf{x})$, where $E_\phi(\cdot)$ is called 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 of ϕ . 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).

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

$$\begin{aligned} \mathcal{S}(\mathbb{P}, \mathbb{Q}) &:= \sup_{\mathbf{f} \in \mathcal{F}} \mathbb{E}_{\mathbf{x} \sim \mathbb{P}}[\mathcal{A}_{\mathbb{Q}} \mathbf{f}(\mathbf{x})] \\ &:= \sup_{\mathbf{f} \in \mathcal{F}} \{\Gamma(\mathbb{E}_{\mathbf{x} \sim \mathbb{P}}[\nabla_{\mathbf{x}} \log q(\mathbf{x}) \mathbf{f}(\mathbf{x})^\top + \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x})])\}, \end{aligned} \quad (1)$$

where \mathcal{F} is often chosen to be a Stein class (see, e.g., Definition 2.1 in Liu et al. (2016)), $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ is a vector-valued function called *Stein critic* and ϕ is an operation that transforms a $d \times d'$ matrix into a scalar value. One common choice of Γ is trace operator when $d' = d$. One can also use other forms for Γ , like matrix norm when $d' \neq d$ (Liu et al., 2016). If \mathcal{F} is a unit ball in some reproducing kernel Hilbert space (RKHS) with a positive definite kernel k , it induces Kernel Stein Discrepancy (KSD). More details are provided in Appendix B.

Wasserstein Metric. Wasserstein metric is suitable for measuring distances between two distributions with non-overlapping supports (Arjovsky et al., 2017). The Wasserstein-1 metric between distributions \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 with respect to γ is over all joint distributions with marginals \mathbb{P} and \mathbb{Q} . By Kantorovich-Rubinstein duality, $\mathcal{W}(\mathbb{P}, \mathbb{Q})$ has a dual representation

$$\mathcal{W}(\mathbb{P}, \mathbb{Q}) := \max_D \{\mathbb{E}_{\mathbf{x} \sim \mathbb{P}}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{y} \sim \mathbb{Q}}[D(\mathbf{y})]\}, \quad (2)$$

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

Sobolev space and Sobolev dual norm. Let $L^2(\mathbb{P})$ be the Hilbert space on \mathbb{R}^d equipped with an inner product $\langle u, v \rangle_{L^2(\mathbb{P})} := \int_{\mathbb{R}^d} uv d\mathbb{P}(\mathbf{x})$. The (weighted) 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 norm $\|u\|_{H^1} := (\int_{\mathbb{R}^d} (u^2 + \|\nabla u\|_2^2) d\mathbb{P}(\mathbf{x}))^{1/2}$, where \mathbb{P} has a density. For $v \in L^2$, its Sobolev dual norm $\|v\|_{H^{-1}}$ is defined by (Evans & Society, 2010)

$$\|v\|_{H^{-1}} := \sup_{u \in H^1} \left\{ \langle v, u \rangle_{L^2} : \int_{\mathbb{R}^d} \|\nabla u\|_2^2 d\mathbb{P}(\mathbf{x}) \leq 1, \int_{\mathbb{R}^d} u(\mathbf{x}) d\mathbb{P}(\mathbf{x}) = 0 \right\}.$$

The constraint $\int_{\mathbb{R}^d} u(\mathbf{x}) d\mathbf{x} = 0$ is necessary to guarantee the finiteness of the supremum, and the supremum can be equivalently taken over C_0^∞ .

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}_{real} underlying real distribution from which data $\{\mathbf{x}\}$ are sampled. We simultaneously learn two generative models – one explicit and one implicit – that represent estimates of \mathbb{P}_{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 $\tilde{\mathbf{x}} = G(\mathbf{z})$ with distribution \mathbb{P}_G . We use the Stein discrepancy as a measure of closeness between the explicit unnormalized density \mathbb{P}_E and the real distribution \mathbb{P}_{real} , and use the Wasserstein metric as a measure of closeness between the implicit distribution \mathbb{P}_G and \mathbb{P}_{real} .

To jointly learn the two generative models \mathbb{P}_G and \mathbb{P}_E , arguably the most straightforward way 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 $\lambda \geq 0$ is a weight coefficient. However, this approach appears no different than learning the two generative models separately. To achieve information sharing between two models, we incorporate another term $\mathcal{S}(\mathbb{P}_G, \mathbb{P}_E)$ – called the *Stein bridge* – that measures the closeness between the explicit unnormalized density \mathbb{P}_E and the implicit distribution \mathbb{P}_G :

$$\min_{E, G} \mathcal{W}(\mathbb{P}_{\text{real}}, \mathbb{P}_G) + \lambda_1 \mathcal{S}(\mathbb{P}_{\text{real}}, \mathbb{P}_E) + \lambda_2 \mathcal{S}(\mathbb{P}_G, \mathbb{P}_E), \quad (3)$$

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

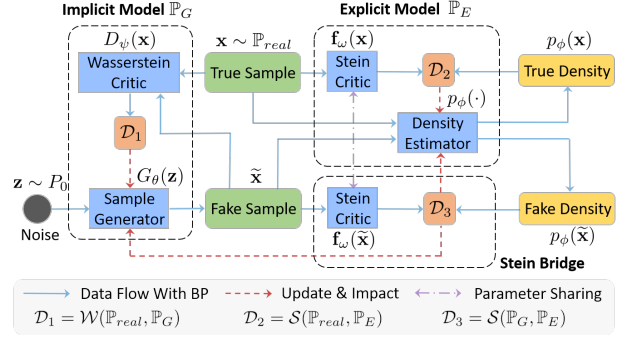


Figure 1. Model framework for *Stein Bridging* which jointly train an implicit sample generator and an explicit (unnormalized) density estimator via a Stein bridge.

The Wasserstein term in (3) is implemented using its equivalent dual representation in (2). The two Stein terms in (3) can be implemented using (1) with either a Stein critic parameterized by a neural network, or the Kernel Stein Discrepancy. 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.

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

3.2. Regularization Effects of Stein Bridge

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

3.2.1. KERNEL SOBOLEV DUAL NORM REGULARIZATION ON WASSERSTEIN CRITIC

Define the *kernel Sobolev dual norm* as

$$\|D\|_{H^{-1}(\mathbb{P}; k)} := \sup_{u \in C_0^\infty} \{ \langle D, u \rangle_{L^2(\mathbb{P})} : \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \mathbb{P}} [\nabla u(\mathbf{x})^\top k(\mathbf{x}, \mathbf{x}') \nabla u(\mathbf{x}')] \leq 1, \mathbb{E}_{\mathbb{P}}[h] = 0 \}.$$

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

Theorem 1. *Assume that $\{\mathbb{P}_G\}_G$ exhausts all continuous probability distributions and \mathcal{S} is chosen as kernel Stein discrepancy. Then problem (3) is equivalent to*

$$\min_E \max_D \left\{ \mathbb{E}_{\mathbf{y} \sim \mathbb{P}_E} [D(\mathbf{y})] - \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\text{real}}} [D(\mathbf{x})] - \frac{1}{4\lambda_2} \|D\|_{H^{-1}(\mathbb{P}_E; k)} \right\}.$$

The kernel Sobolev norm regularization penalizes large variation of the Wasserstein critic D . Particularly, observe that (Villani, 2008) if $k(\mathbf{x}, \mathbf{x}') = \mathbb{I}(\mathbf{x} = \mathbf{x}')$ and $\mathbb{E}_{\mathbb{P}_E} [D] = 0$, then

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

where \mathcal{W}_2 denotes the 2-Wasserstein metric. Hence, the Sobolev dual norm regularization ensures D to not change suddenly on high-density region of \mathbb{P}_E , and thus reinforces the learning of the Wasserstein critic. Stein bridge penalizes large variation of the Wasserstein critic, in the same spirit but of different form comparing to gradient-based penalty (e.g., (Gulrajani et al., 2017; Roth et al., 2017)).

3.2.2. LIPSCHITZ REGULARIZATION ON STEIN CRITIC

We next investigate how the Stein bridge helps to regularize the Stein critic. We have the following result.

Theorem 2. *Assume $\{\mathbb{P}_G\}_G$ exhausts all continuous probability distributions, and the Stein class defining the Stein discrepancy is compact (in some linear topological space). Then problem (3) is equivalent to*

$$\min_E \left\{ \lambda_1 \mathcal{S}(\mathbb{P}_{\text{real}}, \mathbb{P}_E) + \lambda_2 \max_{\mathbf{f}} \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\text{real}}} [(\mathcal{A}_{\mathbb{P}_E} \mathbf{f})_{\lambda_2}(\mathbf{x})] \right\},$$

where $(\mathcal{A}_{\mathbb{P}_E} \mathbf{f})_{\lambda_2}(\cdot)$ denotes the (generalized) Moreau-Yosida regularization of the function $\mathcal{A}_{\mathbb{P}_E} \mathbf{f}$ with parameter λ_2 , i.e., $(\mathcal{A}_{\mathbb{P}_E} \mathbf{f})_{\lambda_2}(\mathbf{x}) = \min_{\mathbf{y} \in \mathcal{X}} \{ \mathcal{A}_{\mathbb{P}_E} \mathbf{f}(\mathbf{y}) + \frac{1}{\lambda_2} \|\mathbf{x} - \mathbf{y}\| \}$.

Note that $(\mathcal{A}_{\mathbb{P}_E} \mathbf{f})_{\lambda_2}$ is Lipschitz continuous with constant $1/\lambda_2$. Hence, the Stein bridge, together with the Wasserstein metric $\mathcal{W}(\mathbb{P}_{\text{real}}, \mathbb{P}_G)$, plays as a Lipschitz regularization on the Stein critic \mathbf{f} via Moreau-Yosida regularization. By smoothing the Stein critic, Stein bridge further encourages the energy model to seek more modes in data instead of focusing on some dominated modes, thus alleviating mode-collapse issue. This suggests a novel regularization scheme for Stein-based GAN.

4. Convergence Analysis

In Section 3.2, we justify Stein Bridging by showing its regularization effects. In this section, we further show that

it could help 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 (unnormalized) density estimator can stabilize GAN training and help for convergence.

4.1. Analysis of a Linear System

The training for minimax game in GAN is difficult. When using traditional gradient methods, the training would suffer from some oscillatory behaviors (Goodfellow, 2017; Liang & Stokes, 2019). In order to better understand the optimization behaviors, we first study a one-dimension linear system that provides some insights on this problem. Such toy example (or a similar one) is also utilized by (Gidel et al., 2019; Nagarajan & Kolter, 2017) to shed lights on the instability of WGAN training¹. Consider a linear critic $D_\psi(x) = \psi x$ and generator $G_\theta(z) = \theta z$. Then the Wasserstein GAN objective can be written as a constrained bilinear problem: $\min_\theta \max_{|\psi| \leq 1} \psi \mathbb{E}[x] - \psi \theta \mathbb{E}[z]$, which could be further simplified as an unconstrained version (the behaviors can be generalized to multi-dimensional cases (Gidel et al., 2019)):

$$\min_\theta \max_\psi \psi - \psi \cdot \theta. \quad (4)$$

Unfortunately, such simple objective cannot guarantee convergence by traditional gradient methods like SGD with alternate updating²: $\theta_{k+1} = \theta_k + \eta \psi_k$, $\psi_{k+1} = \psi_k + \eta(1 - \theta_{k+1})$. Such optimization would suffer from an oscillatory behavior, i.e., the updated parameters go around the optimum point $([\psi^*, \theta^*] = [0, 1])$ forming a circle without converging to the centrality, which is shown in Fig. 2(a). A recent study in (Liang & Stokes, 2019) theoretically show that such oscillation is due to the interaction term in (4).

One solution to the instability of GAN training is to add (likelihood) regularization, which has been widely studied by recent literatures (Warde-Farley & Bengio, 2017; Li & Turner, 2018). With regularization term, the objective changes into $\min_\theta \max_{|\psi| \leq 1} \psi \mathbb{E}[x] - \psi \theta \mathbb{E}[z] - \lambda \mathbb{E}[\log \mu(\theta z)]$, where $\mu(\cdot)$ denotes the likelihood function and λ is a hyperparameter. A recent study (Tao et al., 2019) proves that when $\lambda < 0$ (likelihood-regularization), the extra term is equivalent to maximizing sample evidence, helping to stabilize GAN training; when $\lambda > 0$ (entropy-regularization), the extra term maximizes sample entropy, which encourages diversity of generator. Here we con-

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

²Here, we adopt the most widely used alternate updating strategy. The simultaneous updating, i.e., $\theta_{k+1} = \theta_k + \eta \psi_k$ and $\psi_{k+1} = \psi_k + \eta(1 - \theta_k)$, would diverge in this case.

sider a Gaussian likelihood function for generated sample x' , $\mu(x') = \exp(-\frac{1}{2}(x' - b)^2)$ which is up to a constant, and the objective becomes (see Appendix D.1 for details):

$$\min_{\theta} \max_{\psi} \psi - \psi \cdot \theta - \lambda(\theta^2 - \theta). \quad (5)$$

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

Finally, 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 $\nabla_x \log p_{\phi}(x) = -x - \phi$. Then if we specify the two Stein discrepancies in (3) as KSD, the objective is (see Appendix D.1 for details),

$$\min_{\theta} \max_{\psi} \min_{\phi} \psi - \psi \cdot \theta + \frac{\lambda_1}{2}(1 + \phi)^2 + \frac{\lambda_2}{2}(\theta + \phi)^2. \quad (6)$$

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

Proposition 1. *Using alternate SGD for (6) geometrically decreases the square norm $N_t = |\psi^t|^2 + |\theta - 1|^2 + |\phi + 1|^2$, for any $0 < \eta < 1$ with $\lambda_1 = \lambda_2 = 1$,*

$$N_{t+1} = (1 - \eta^2(1 - \eta)^2)N_t. \quad (7)$$

As shown in Fig. 2(a), Stein Bridging achieves a good convergence to the right optimum. Compared with (4), the objective (6) adds a new bilinear term $\phi \cdot \theta$, which acts like a connection between the generator and estimator, and two other quadratic terms, which help to penalize the increasing of values through training. The added terms and original terms in (6) cooperate to guarantee convergence to a unique optimum. (More discussions in Appendix D.1).

We further generalize the analysis to multi-dimensional bilinear system $F(\psi, \theta) = \theta^{\top} \mathbf{A} \psi - \mathbf{b}^{\top} \theta - \mathbf{c}^{\top} \psi$ which is extensively used by researches for analysis of GAN stability (Goodfellow, 2017; Gemp & Mahadevan, 2018; Liang & Stokes, 2019; Gidel et al., 2019). For any bilinear system, with added term $H(\phi, \theta) = \frac{1}{2}(\theta + \phi)^{\top} \mathbf{B}(\theta + \phi)$ 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 D.3.

4.2. Local Convergence for a General Model

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

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

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

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

$$\omega_h^{t+1} = P_{\Omega_h}(\omega_h^{t+1/2} - \eta h(\omega_h^{t+1/2})), \quad (8)$$

$$\omega_f^{t+1} = P_{\Omega_f}(\omega_f^{t+1/2} - \eta f(\omega_f^{t+1/2})), \quad (9)$$

where $\omega_h^{t+1/2} = [\theta^t, \psi^t]$, $\omega_h^{t+1} = [\theta^{t+1/2}, \psi^{t+1}]$, $\omega_f^{t+1/2} = [\theta^{t+1/2}, \phi^t]$, $\omega_f^{t+1} = [\theta^{t+1}, \phi^{t+1}]$, and $P_{\Omega}(\omega) = \arg \min_{\omega' \in \Omega} \|\omega - \omega'\|_2^2$ denotes projection mapping to Ω . Then convergence can be achieved with $\frac{1}{2\mu} < \eta < \frac{1}{\mu}$.

Theorem 3 indicates that Stein Bridging converges 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 open problem. Despite the fact that strong convexity assumption cannot be guaranteed with deep neural networks, the optimization can converge to a stable point once there exists a local region that satisfies the strongly convex conditions. In the experiments, we will empirically compare the training stability of each method on various datasets to validate our theoretical discussions.

5. Experiments

In this section, we conduct experiments to verify the effectiveness of proposed method from multifaceted views. The codes are released in <https://github.com/echo740/SteinBridging>.

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

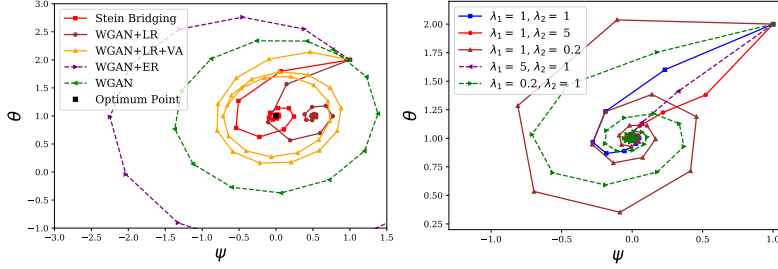


Figure 2. Numerical iterations for SGD training. (a) Comparison of WGAN, likelihood-regularized WGAN (WGAN+LR), variational annealing for WGAN+LR (WGAN+LR+VA), entropy-regularized WGAN (WGAN+ER) and our Stein Bridging. (b) Stein Bridging with different λ_1 and λ_2 .

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

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

5.1. Sample Quality of Implicit Model

Calibrating explicit (unnormalized) 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. In Two-Circle, there are a large number of gen-

Table 1. Inception scores on MNIST and CIFAR-10. † We directly use the best result reported in their paper.

Method	MNIST	CIFAR-10
WGAN-GP	7.71	6.80
WGAN+LR	7.82	6.89
WGAN+ER	7.75	6.99
WGAN+VA	7.74	6.95†
DGM	6.87	4.79
Joint-W(ours)	7.90	7.18

erated samples given by GAN, WGAN-GP and DGM 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 data but such combination makes no sense in practice. By contrast, Joint-JS and Joint-W could alleviate such issue, reduce the low-quality samples and produce more distinguishable boundaries. 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 Maximum Mean Discrepancy (MMD) and High-quality Sample Rate (HSR). The detailed definitions are given in Appendix E.4 and we report results in Table 4.

We visualize generated digits/images on MNIST/CIFAR-10 in Fig. 8 and Fig. 9 and use Inception Score to measure the sample quality. As shown in Table 1, Joint-W is superior than WGAN-GP, regularized WGAN and DGM. 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

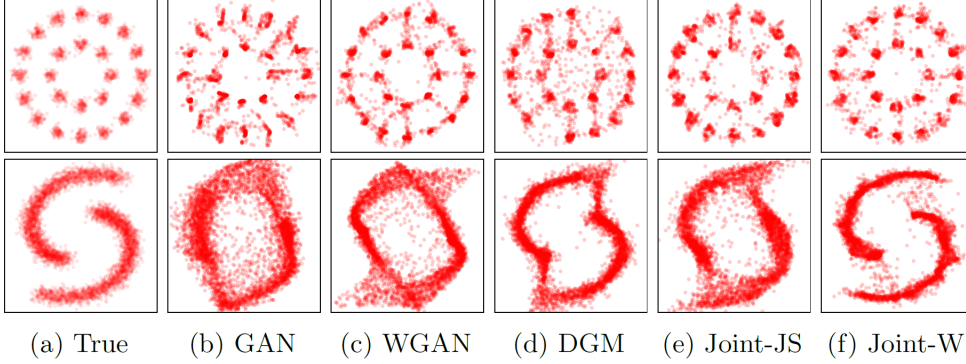


Figure 3. Comparison for sample quality of implicit model over synthetic datasets. (a) True samples from real distribution and (b)~(f) generated samples produced by the generators of different methods on Two-Circle (upper line) and Two-Spiral (bottom line) datasets.

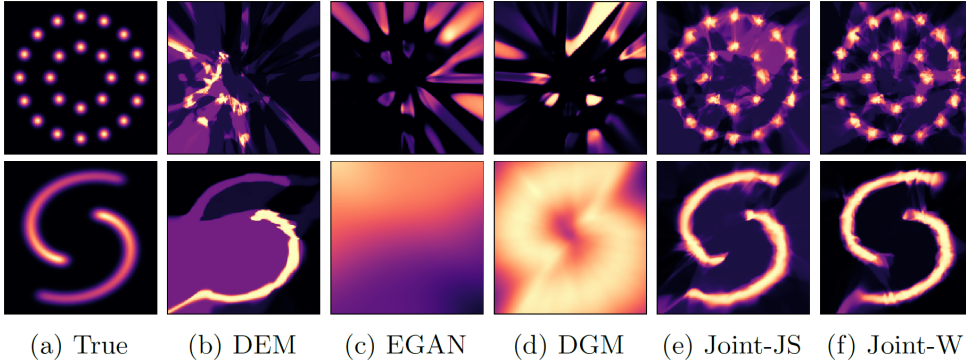


Figure 4. Comparison for density estimation performance of explicit model over synthetic datasets. (a) True densities of real distribution and (b)~(f) estimated densities given by the estimators of different methods on Two-Circle (upper line) and Two-Spiral (bottom line) datasets.

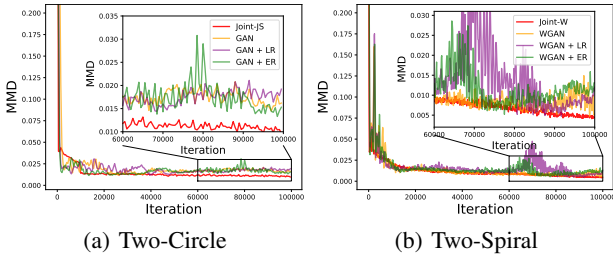


Figure 5. Learning curves of Joint-W (resp. Joint-JS) compared with WGAN (resp. GAN) and its regularization-based variants.

other methods miss some of modes. In Fig 4, Joint-JS and Joint-W exactly fit the ground-truth distribution. Nevertheless, 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 ground-truth distribution. As quantitative comparison, we study three evaluation metrics: KL & JS divergence and Area Under the Curve (AUC). The detailed information and results are given in Appendix E.4 and Table 4 respectively. The values show that Joint-W and Joint-JS provide better estimation than all competitors.

We also rank generated digits (and true digits) on MNIST w.r.t densities given by the energy model in Fig. 10, Fig. 11 and Fig. 12. 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 diversifies). By contrast, 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) and its regularization-based variants on different datasets. The curves depict that joint training could reduce the variance of metric values especially during the second half of training. Furthermore, we visualize generated samples given by the same noise z in adjacent epochs in Fig. 6. The results show that Joint-W outputs more stable generation in adjacent epochs while 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 that we discuss in Section 4.1.



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

Another issue mentioned 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 (resp. CIFAR-10). The results are reported in Table 3. The smaller distances given by Joint-W indicate that it converges to a better point with smaller bias from true data distribution. Also, in Table 5 (resp. Table 6), we report distances for digits (resp. images) in each class on MNIST (resp. CIFAR).

5.4. Applications of Stein Bridging

As further discussions, we study three applications for joint training model. The details are in Appendix E.1

Detecting Noise and Outliers. We first study the impact of noised samples in data. The results are presented in Fig. 7(a) where we add different ratios of random noise to the true samples in Two-Circle dataset. 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.

Addressing Data Insufficiency. Then we study the impact of different numbers of observed samples and only sample insufficient data for training in Two-Spiral dataset. As shown 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 its lower sensitivity to observed sample size.

Enforcing Induction Bias. As discussed in introduction part, joint models can also help to add induction bias to

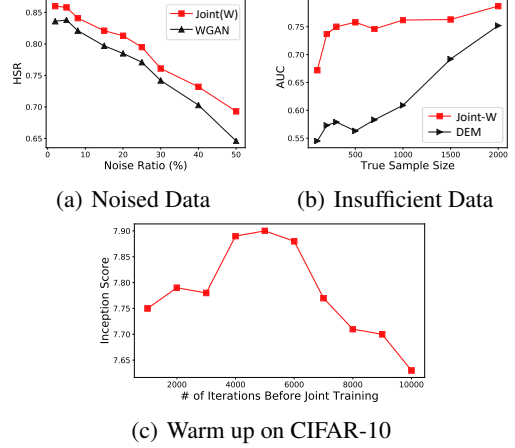


Figure 7. Joint-W with (a) noised data, (b) insufficient data and (c) ‘warm up’ iterations before joint training.

generative models when one aims to control what samples are generated from the model. To validate this argument, we change the specification of energy model to PixelCNN++ which is easy to control pixel-level features of generated images in order to add induction bias. In CIFAR-10, we manage to achieve inception score 7.25 with a small improvement compared to using original energy model.

5.5. 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 for independent training of two models. At the beginning, we set $\lambda_2 = 0$ (or a very small value) in (3) and after some iterations set it on a normal level. We report 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. Such phenomenon is quite similar to the ‘warm up’ trick used for training deep networks where one uses small learning rates at 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, enabling information sharing between them 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 stabilize 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 on different tasks and applications 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

- Alain, G. and Bengio, Y. What regularized auto-encoders learn from the data-generating distribution. *J. Mach. Learn. Res.*, 15(1):3563–3593, 2014.
- Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *ICML*, pp. 214–223, 2017.
- Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.
- Chwialkowski, K., Strathmann, H., and Gretton, A. A kernel test of goodness of fit. In *ICML*, pp. 2606–2615, 2016.
- Dai, Z., Almahairi, A., Bachman, P., Hovy, E. H., and Courville, A. C. Calibrating energy-based generative adversarial networks. In *ICLR*, 2017.
- Du, C., Xu, K., Li, C., Zhu, J., and Zhang, B. Learning implicit generative models by teaching explicit ones. *CoRR*, abs/1807.03870, 2018.
- Du, Y. and Mordatch, I. Implicit generation and generalization in energy-based models. *CoRR*, abs/1903.08689, 2019.
- Evans, L. and Society, A. M. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 2010. ISBN 9781470411442. URL <https://books.google.com/books?id=UL9WtAEACAAJ>.
- Gao, R. and Kleywegt, A. J. Distributionally robust stochastic optimization with wasserstein distance. *arXiv preprint arXiv:1604.02199*, 2016.
- Gao, R., Chen, X., and Kleywegt, A. J. Wasserstein distributional robustness and regularization in statistical learning. *arXiv preprint arXiv:1712.06050*, 2017.
- Geman, S. and Geman, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, 1984.
- Gemp, I. and Mahadevan, S. Global convergence to the equilibrium of gans using variational inequalities. *CoRR*, abs/1808.01531, 2018.
- Gidel, G., Berard, H., Vignoud, G., Vincent, P., and Lacoste-Julien, S. A variational inequality perspective on generative adversarial networks. In *ICLR*, 2019.
- Goodfellow, I. J. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In *NIPS*, pp. 2672–2680, 2014.
- Gorham, J. and Mackey, L. Measuring sample quality with stein’s method. In *Advances in Neural Information Processing Systems*, pp. 226–234, 2015.
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *NIPS*, pp. 5767–5777, 2017.
- Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *ICML*, pp. 1352–1361, 2017.
- Hinton, G. E. Product of experts. In *ICANN99 Artificial Neural Networks*, 1999.
- Hinton, G. E., Osindero, S., and Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.
- Hu, T., Chen, Z., Sun, H., Bai, J., Ye, M., and Cheng, G. Stein neural sampler. *CoRR*, abs/1810.03545, 2018.
- Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6:695–709, 2005.
- Kim, T. and Bengio, Y. Deep directed generative models with energy-based probability estimation. In *ICLR*, 2017.
- Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *ICLR*, 2014.
- LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. J. A tutorial on energy-based learning. *Predicting Structured Data*, MIT Press, 2006.
- Li, Y. and Turner, R. E. 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.

- Liang, T. and Stokes, J. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *AISTATS*, pp. 907–915, 2019.
- Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *NIPS*, pp. 2370–2378, 2016.
- Liu, Q. and Wang, D. Learning deep energy models: Contrastive divergence vs. amortized MLE. *CoRR*, abs/1707.00797, 2017.
- Liu, Q., Lee, J. D., and Jordan, M. I. A kernelized stein discrepancy for goodness-of-fit tests. In *ICML*, pp. 276–284, 2016.
- Nagarajan, V. and Kolter, J. Z. Gradient descent GAN optimization is locally stable. In *NIPS*, pp. 5585–5595, 2017.
- Neal, R. M. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Handbook of Markov Chain Monte Carlo*, 2, 2011.
- Ngiam, J., Chen, Z., Koh, P. W., and Ng, A. Y. Learning deep energy models. In *ICML*, pp. 1105–1112, 2011.
- Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A., and Yosinski, J. Plug & play generative networks: Conditional iterative generation of images in latent space. In *CVPR*, pp. 3510–3520, 2017.
- Nijkamp, E., Hill, M., Zhu, S.-C., and Wu, Y. On learning non-convergent non-persistent short-run mcmc toward energy-based model. *CoRR*, abs/1904.09770, 2019.
- Oates, C. J., Girolami, M., and Chopin, N. Control functionals for monte carlo integration. *Journal of the Royal Statistical Society, Series B*, 2017.
- Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.
- Robins, G., Pattison, P., Kalish, Y., and Lusher, D. An introduction to exponential random graph (p^*) models for social networks. *Social Networks*, 29(2):173–191, 2007.
- Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. Stabilizing training of generative adversarial networks through regularization. In *Advances in neural information processing systems*, pp. 2018–2028, 2017.
- Salakhutdinov, R. and Hinton, G. E. Deep boltzmann machines. In *AISTATS*, pp. 448–455, 2009.
- Shapiro, A., Dentcheva, D., and Ruszczyński, A. *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.
- Tao, C., Dai, S., Chen, L., Bai, K., Chen, J., Liu, C., Zhang, R., Bobashev, G. V., and Carin, L. Variational annealing of gans: A langevin perspective. In *ICML*, pp. 6176–6185, 2019.
- Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.
- Warde-Farley, D. and Bengio, Y. 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.
- Weaver, N. *Lipschitz algebras*. World Scientific, 1999.
- Wu, Y. N., Zhu, S. C., and Liu, X. Equivalence of julesz ensembles and FRAME models. *International Journal of Computer Vision*, 38(3):247–265, 2000.
- Xie, J., Lu, Y., Zhu, S., and Wu, Y. N. Cooperative training of descriptor and generator networks. *CoRR*, abs/1609.09408, 2016a.
- Xie, J., Lu, Y., Zhu, S., and Wu, Y. N. A theory of generative convnet. In *ICML*, pp. 2635–2644, 2016b.
- Xie, J., Lu, Y., Gao, R., and Wu, Y. N. Cooperative learning of energy-based model and latent variable model via MCMC teaching. In *AAAI*, pp. 4292–4301, 2018.
- Zhai, S., Cheng, Y., Lu, W., and Zhang, Z. Deep structured energy based models for anomaly detection. In *ICML*, pp. 1100–1109, 2016.
- Zhao, J. J., Mathieu, M., and LeCun, Y. Energy-based generative adversarial networks. In *ICLR*, 2017.
- Zhu, S. C., Wu, Y. N., and Mumford, D. 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 (unnormalized) density expected to explicitly capture the distribution behind data. Such densities are often up to a constant and called as energy functions which are common in undirected graphical models (LeCun et al., 2006). Hence, explicit generative models are also termed as energy-based models. An early version of energy-based models is the FRAME (Filters, Random field, And Maximum Entropy) model (Zhu et al., 1997; Wu et al., 2000). Later on, some works leverage deep neural networks to model the energy function ((Ngiam et al., 2011; Xie et al., 2016b)) and pave the way for researches on deep energy model (DEM) (e.g., (Liu & Wang, 2017; Kim & Bengio, 2017; Zhai et al., 2016; Haarnoja et al., 2017; Du & Mordatch, 2019; Nijkamp et al., 2019)). Apart from DEM, there are also some other forms of deep explicit models based on restricted Boltzmann machines like deep belief networks (Hinton et al., 2006) and deep Boltzmann machines (Salakhutdinov & Hinton, 2009).

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

A.2. Implicit Generative Models

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

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

To deal with the instability issue, there are also some recent literatures from optimization perspectives and proposes differ-

ent algorithms to address the non-convergence of minimax game optimization (for instance, (Gemp & Mahadevan, 2018; Liang & Stokes, 2019; Gidel et al., 2019)). Moreover, the disadvantage of implicit models is the lack of explicit densities over instances, which disables the black-box generator to characterize the distributions behind data.

A.3. Attempts to Combine Both of the Worlds

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

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

Assume $q(\mathbf{x})$ to be a continuously differentiable density supported on $\mathcal{X} \subset \mathbb{R}^d$ and $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ 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:

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

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

$$\mathcal{S}(\mathbb{Q}, \mathbb{P}) = \sup_{\mathbf{f} \in \mathcal{F}} \{\mathbb{E}_{\mathbf{x} \sim q}[\mathcal{A}_p[\mathbf{f}(\mathbf{x})]]\} = \sup_{\mathbf{f} \in \mathcal{F}} \{\Gamma(\mathbb{E}_{\mathbf{x} \sim q}[\nabla_{\mathbf{x}} \log p(\mathbf{x}) \mathbf{f}(\mathbf{x})^\top + \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x})])\}, \quad (10)$$

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 (1), 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 (1) would have a close form (see (Liu et al., 2016; Chwialkowski et al., 2016; Oates et al., 2017) for more details):

$$\mathcal{S}_K(\mathbb{Q}, \mathbb{P}) = \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim q}[u_p(\mathbf{x}, \mathbf{x}')], \quad (11)$$

where $u_p(\mathbf{x}, \mathbf{x}') = \nabla_{\mathbf{x}} \log p(\mathbf{x})^\top k(\mathbf{x}, \mathbf{x}') \nabla_{\mathbf{x}} \log p(\mathbf{x}') + \nabla_{\mathbf{x}} \log p(\mathbf{x})^\top \nabla_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}') + \nabla_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}')^\top \nabla_{\mathbf{x}} \log p(\mathbf{x}') + \text{tr}(\nabla_{\mathbf{x}, \mathbf{x}'} k(\mathbf{x}, \mathbf{x}'))$. The (11) gives the Kernel Stein Discrepancy (KSD).

C. Proofs of Results in Section 3.2

C.1. Proof of Theorem 1

Proof. Applying Kantorovich's duality on $\mathcal{W}(\mathbb{P}_G, \mathbb{P}_r)$ and using the exhaustiveness assumption on the generator, we rewrite the problem as

$$\min_{E, \mathbb{P}} \max_D \{ \mathbb{E}_{\mathbb{P}}[D] - \mathbb{E}_{\mathbb{P}_{\text{real}}}[D] + \lambda_1 \mathcal{S}(\mathbb{P}_{\text{real}}, \mathbb{P}_E) + \lambda_2 \mathcal{S}(\mathbb{P}, \mathbb{P}_E) \}, \quad (12)$$

where the minimization with respect to E is over all energy functions, the minimization with respect to \mathbb{P} is over all probability distributions with continuous density, and the maximization with respect to D is over all 1-Lipschitz continuous functions. Recall the definition of kernel Stein discrepancy

$$\mathcal{S}(\mathbb{P}, \mathbb{P}_E) = \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \mathbb{P}} [(\nabla_x \log d\mathbb{P}/d\mathbb{P}_E(\mathbf{x}))^\top k(\mathbf{x}, \mathbf{x}') \nabla_x \log d\mathbb{P}/d\mathbb{P}_E(\mathbf{x}')],$$

where $d\mathbb{P}/d\mathbb{P}_E$ is the Radon-Nikodym derivative. Observe that $\mathcal{S}(\mathbb{P}, \mathbb{P}_E)$ is infinite if \mathbb{P} is not absolutely continuous with respect to \mathbb{P}_E . Hence, to minimize the objective of (12), it suffices to consider those \mathbb{P} 's that are absolutely continuous with respect to \mathbb{P}_E . Introducing a variable replacement $h(\mathbf{x}) := d\mathbb{P}/d\mathbb{P}_E(\mathbf{x}) - 1$, then problem (12) becomes

$$\begin{aligned} \min_{E, h} \max_D \Big\{ & \mathbb{E}_{\mathbb{P}_E}[(1+h)D] - \mathbb{E}_{\mathbb{P}_{\text{real}}}[D] + \lambda_1 \mathcal{S}(\mathbb{P}_{\text{real}}, \mathbb{P}_E) \\ & + \lambda_2 \cdot \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \mathbb{P}} [\nabla_x \log(1+h(\mathbf{x}))^\top k(\mathbf{x}, \mathbf{x}') \nabla_x \log(1+h(\mathbf{x}'))] \Big\}, \end{aligned} \quad (13)$$

where the minimization with respect to h is over all $L^1(\mathbb{P}_E)$ functions with \mathbb{P}_E -expectation zero.

Fixing E , we claim that we can swap \min_h and \max_D . Indeed, without loss of generality, we can restrict D to be such that $D(\mathbf{x}_0) = 0$ for some element \mathbf{x}_0 , as a constant shift does not change the value of $\mathbb{E}_{\mathbb{P}_E}[(1+h)D] - \mathbb{E}_{\mathbb{P}_{\text{real}}}[D]$. The set of Lipschitz functions that vanish at \mathbf{x}_0 is a Banach space, and the set of 1-Lipschitz functions is compact (Weaver, 1999). Moreover, $L^1(\mathbb{P}_E)$ is also a Banach space. The above verifies the condition of Sion's minimax theorem, and thus the claim is proved.

Swapping \min_h and \max_D in (13) and fixing E and D , we consider

$$\begin{aligned} & \min_{h: \mathbb{E}_{\mathbb{P}_E}[h]=0} \{ \mathbb{E}_{\mathbb{P}_E}[hD] + \lambda_2 \cdot \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \mathbb{P}_E} [\nabla_x \log(1+h(\mathbf{x}))^\top k(\mathbf{x}, \mathbf{x}') \nabla_x \log(1+h(\mathbf{x}'))] \} \\ &= \min_{h: \mathbb{E}_{\mathbb{P}_E}[h]=0} \left\{ \mathbb{E}_{\mathbb{P}_E}[hD] + \lambda_2 \cdot \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \mathbb{P}_E} \left[\frac{\nabla_x h(\mathbf{x})^\top}{1+h(\mathbf{x})} k(\mathbf{x}, \mathbf{x}') \frac{\nabla_x h(\mathbf{x}')}{1+h(\mathbf{x}')} \right] \right\} \\ &= \min_{h: \mathbb{E}_{\mathbb{P}_E}[h]=0} \left\{ \mathbb{E}_{\mathbb{P}_E}[hD] + \lambda_2 \cdot \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \mathbb{P}_E} [\nabla_x h(\mathbf{x})^\top k(\mathbf{x}, \mathbf{x}') \nabla_x h(\mathbf{x}')] \right\}, \end{aligned}$$

where the first equality follows from the chain rule of the derivative, and the second equality follows from a change of measure $d\mathbb{P} = (1+h)d\mathbb{P}_E$. Introducing an auxiliary variable r so that r^2 is an upper bound of $\mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \mathbb{P}_E} [\nabla_x h(\mathbf{x})^\top k(\mathbf{x}, \mathbf{x}') \nabla_x h(\mathbf{x}')]$, we have that

$$\begin{aligned} & \min_{h: \mathbb{E}_{\mathbb{P}_E}[h]=0} \{ \mathbb{E}_{\mathbb{P}_E}[hD] + \lambda_2 \cdot \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \mathbb{P}_E} [\nabla_x h(\mathbf{x})^\top k(\mathbf{x}, \mathbf{x}') \nabla_x h(\mathbf{x}')] \} \\ &= \min_{r \geq 0} \min_{h: \mathbb{E}_{\mathbb{P}_E}[h]=0} \{ \mathbb{E}_{\mathbb{P}_E}[hD] + \lambda_2 r^2 : \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \mathbb{P}_E} [\nabla_x h(\mathbf{x})^\top k(\mathbf{x}, \mathbf{x}') \nabla_x h(\mathbf{x}')] \leq r^2 \} \\ &= \min_{r \geq 0} \min_{h: \mathbb{E}_{\mathbb{P}_E}[h]=0} \{ r \mathbb{E}_{\mathbb{P}_E}[hD] + \lambda_2 r^2 : \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \mathbb{P}_E} [\nabla_x h(\mathbf{x})^\top k(\mathbf{x}, \mathbf{x}') \nabla_x h(\mathbf{x}')] \leq 1 \} \\ &= \min_{r \geq 0} \left\{ \lambda_2 r^2 - r \|D\|_{H^{-1}(\mathbb{P}_E; k)} \right\} \\ &= -\frac{1}{4\lambda_2} \|D\|_{H^{-1}(\mathbb{P}_E; k)}, \end{aligned}$$

where the first equality holds because the minimization over r forces $\mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim \mathbb{P}_E} [\nabla_x h(\mathbf{x})^\top k(\mathbf{x}, \mathbf{x}') \nabla_x h(\mathbf{x}')] = r^2$ at optimal; the second equality follows from a change of variable from h to rh ; and the third equality follows from the definition of the kernel Sobolev dual norm. Plugging back in (13) yields the ideal result. \square

C.2. Proof for Theorem 2

Proof. Applying the definition of Stein discrepancy on $\mathcal{S}(\mathbb{P}_E, \mathbb{P}_G)$ and under the exhaustiveness assumption of G , we rewrite the problem as

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

where the minimization with respect to E is over the set of all energy functions; the minimization with respect to \mathbb{P} is over all continuous distributions; and the maximization with respect to \mathbf{f} is over the Stein class for \mathbb{P}_E . Let us fix E . Using a similar argument as in the proof of Theorem 1, it suffices to restrict \mathbb{P} on the set of distributions that are absolutely continuous with respect to \mathbb{P}_E , which can be identified as the set of $L^1(\mathbb{P}_E)$ functions with \mathbb{P}_E -mean zero and is thus Banach. Together with the compactness assumption of the Stein class, using Sion's minimax theorem, we can swap the minimization over \mathbb{P} and the maximization over \mathbf{f} . Now, fixing further \mathbf{f} , consider

$$\min_{\mathbb{P}} \{ \lambda_2 \mathbb{E}_{\mathbf{y} \sim \mathbb{P}} [\mathcal{A}_{\mathbb{P}_E} \mathbf{f}(\mathbf{y})] + \mathcal{W}(\mathbb{P}_{\text{real}}, \mathbb{P}) \}. \quad (14)$$

Recall the definition of Wasserstein metric

$$\mathcal{W}(\mathbb{P}_{\text{real}}, \mathbb{P}) = \min_{\gamma} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} [\|\mathbf{x} - \mathbf{y}\|],$$

where the minimization is over all joint distributions of (\mathbf{x}, \mathbf{y}) with \mathbf{x} -marginal \mathbb{P}_{real} and \mathbf{y} -marginal \mathbb{P} . We rewrite problem (14) as

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

where γ has marginals \mathbb{P}_{real} and \mathbb{P} . Since \mathbb{P} is unconstrained, the above problem is further equivalent to

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

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

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

where the minimization in the first line of the equation is over $\gamma_{\mathbf{x}}$, the set of all conditional distributions of \mathbf{y} given \mathbf{x} where \mathbf{x} is over the support $\text{supp } \mathbb{P}_{\text{real}}$ of \mathbb{P}_{real} ; the exchanging of \min and \mathbb{E} in the first equality follows from the interchangeability principle (Shapiro et al., 2009); the second equality holds because the infimum can be restricted to the set of point masses. Finally, the original problem is equivalent to

$$\min_E \max_{\mathbf{f}} \left\{ \lambda_1 \mathcal{S}(\mathbb{P}_{\text{real}}, \mathbb{P}_E) + \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\text{real}}} \left[\min_{\mathbf{y} \in \mathcal{X}} \{ \lambda_2 \mathcal{A}_{\mathbb{P}_E} \mathbf{f}(\mathbf{y}) + \|\mathbf{x} - \mathbf{y}\| \} \right] \right\}.$$

Therefore, the proof is completed using the definition of Moreau-Yosida regularization. \square

D. Proofs and More Discussions in Section 4

D.1. Details for One-Dimensional Case

For the analysis of 1-dim regularized WGAN in section 3.1.1, we assume a Gaussian likelihood function for 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 for generated sample $x' = \theta z$, 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 (4); 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 amplifying strength on θ and leads to divergence.

We proceed to consider 1-dim case of Stein Bridging with energy model $p_\phi(x) = \exp(-\frac{1}{2}x^2 - \phi x)$. If using KSD with kernel $k(x_1, x_2) = \mathbb{I}(x_1 = x_2)$, then $\mathcal{S}(\mathbb{P}_{real}, \mathbb{P}_E) = \mathbb{E}_{x_1, x_2}[(\nabla_{x_1} \log p_\phi(x_1) - \nabla_{x_1} \log \mu(x_1))k(x_1, x_2)(\nabla_{x_2} \log p_\phi(x_2) - \nabla_{x_2} \log \mu(x_2))] = \mathbb{E}_x[(\nabla_x \log p_\phi(x) - \nabla_x \log \mu(x))^2] = (\phi + \mathbb{E}[x])^2$. Similarly, one can obtain $\mathcal{S}(\mathbb{P}_G, \mathbb{P}_E) = (\phi + \theta \mathbb{E}[z])^2$. Therefore we arrive at the objective in (6)

$$\min_{\theta} \max_{\psi} \min_{\phi} \psi - \psi \cdot \theta + \frac{\lambda_1}{2}(1 + \phi)^2 + \frac{\lambda_2}{2}(\theta + \phi)^2. \quad (15)$$

Interestingly, the added terms $\frac{\lambda_1}{2}(1 + \phi)^2 + \frac{\lambda_2}{2}(\theta + \phi)^2$ in (6) 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 (4) and would not converge to the unique optimum point $[\psi^*, \theta^*] = [0, 1]$. If we consider both of the world and optimize three terms together, the training would converge to a unique global optimum $[\psi^*, \theta^*, \phi^*] = [0, 1, -1]$.

D.2. Proof for Proposition 1

Proof. Instead of directly studying the optimization for (6), we first prove the following problem will converge to the unique optimum,

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

Applying alternate SGD we have the following iterations:

$$\begin{aligned} \psi_{t+1} &= \psi_t + \eta * \theta_t, \\ \phi_{t+1} &= \phi_t - \eta * (\theta_t + 2\phi_t) = (1 - 2\eta)\phi_t - \eta\theta_t, \\ \theta_{t+1} &= \theta_t - \eta(\psi_{t+1} + \phi_{t+1} + \theta_t) = -\eta(1 - 2\eta)\phi_t + (1 - \eta)\theta_t - \eta\psi_t. \end{aligned}$$

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.

□

D.3. Generalization to Bilinear Systems

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

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

where ψ, θ are both r -dimensional vectors and the objective is $\min_{\theta} \max_{\psi} F(\psi, \theta)$ which can be seen as a basic form of various GAN objectives. Unfortunately, if we directly use simultaneous (resp. alternate) SGD to optimize such objectives, one can obtain divergence (resp. fluctuation). To solve the issue, some recent papers propose several optimization algorithms, like extrapolation from the past ((Gidel et al., 2019)), crossing the curl ((Gemp & Mahadevan, 2018)) and consensus optimization ((Liang & Stokes, 2019)). Also, (Liang & Stokes, 2019) shows that it is the interaction term which generates non-zero values for $\nabla_{\theta\psi} F$ and $\nabla_{\psi\theta} F$ 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 ϕ (corresponding to the estimator in our model) the original system and tackle the following problem:

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned} \psi_{t+1} &= \psi_t + \eta \nabla_{\psi} L(\theta_t, \psi_t, \phi_t), \\ \phi_{t+1} &= \phi_t - \eta \nabla_{\phi} L(\theta_t, \psi_{t+1}, \phi_t), \\ \theta_{t+1} &= \theta_t - \eta \nabla_{\theta} L(\theta_t, \psi_{t+1}, \phi_{t+1}), \end{aligned} \quad (22)$$

we can achieve convergence to $[\psi^*, \theta^*, \phi^*]$ where $\phi^* = -\theta^*$ with at least linear rate of $(1 - \eta_1 + \eta_2^2)(1 + \eta_2 - \eta_1^2)$ where $\eta_1 = \eta \sigma_{\min}$, $\eta_2 = \eta \sigma_{\max}$ and 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.,

$$\begin{aligned} \psi_{t+1} &\in \psi_0 + \text{span}(L(\psi_0, \phi, \theta), \dots, F(\psi_t, \phi, \theta)), \forall t \in \mathbb{N}, \\ \phi_{t+1} &\in \psi_0 + \text{span}(L(\psi, \phi_0, \theta), \dots, L(\psi, \phi_t, \theta)), \forall t \in \mathbb{N}, \end{aligned}$$

$$\boldsymbol{\theta}_{t+1} \in \psi_0 + \text{span}(L(\psi, \phi, \boldsymbol{\theta}_0), \dots, L(\psi, \phi, \boldsymbol{\theta}_t)), \forall t \in \mathbb{N},$$

Then we have

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

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 $([\tilde{\boldsymbol{\psi}}_t]_i, [\tilde{\boldsymbol{\phi}}_t]_i, [\tilde{\boldsymbol{\theta}}_t]_i)_{1 \leq i \leq r}$ follows the update rule with step size $\sigma_i \eta$ as the same optimization method on a unidimensional problem

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

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 (Gidel et al., 2019). The general class of first-order optimization methods derive the following updations:

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

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

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

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 $\boldsymbol{\theta}^* = -\boldsymbol{\phi}^*$, $\mathbf{B} = (\mathbf{U}\mathbf{D}\mathbf{D}^\top\mathbf{U}^\top) = \mathbf{D}$, 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,

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

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

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:

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

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

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 $H(\phi, \theta)$ with affiliated variables ϕ could help the SGD algorithm achieve convergence to the same optimum points as directly optimizing $F(\psi, \theta)$. Our method is related to consensus optimization algorithm ((Liang & Stokes, 2019)) which adds a regularization term $\|\nabla_{\theta} F(\psi, \theta)\| + \|\nabla_{\psi} F(\psi, \theta)\|$ to (17) resulting extra quadratic terms for θ and ψ . The disadvantage of such method is the requirement of Hessian matrix of $F(\psi, \theta)$ which is computational expensive for high-dimensional data. By contrast, our solution only requires the first-order derivatives.

D.4. Strongly Convexity

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

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

Besides, $F(\theta, \psi)$ is μ -strongly convex for θ and μ -strongly concave for ψ so it satisfies:

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

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

$$\begin{aligned} (h(\omega_h) - h(\omega'_h))^{\top} (\omega_h - \omega'_h) &\geq \mu \|\omega_h - \omega'_h\|_2^2, \forall \omega_h, \omega'_h \in \Omega_h, \\ (f(\omega_f) - f(\omega'_f))^{\top} (\omega_f - \omega'_f) &\geq \mu \|\omega_f - \omega'_f\|_2^2, \forall \omega_f, \omega'_f \in \Omega_f. \end{aligned}$$

D.5. Proof for Theorem 2

The proof relies on two lemmas,

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

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

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

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

Rearranging the above inequality one can easily get the lemma. \square

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

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

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

Proof. By optimality of ω^* , we have

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

Since Φ is μ -convex, we can further derive

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

\square

Proof. (Proof for Theorem 3) We apply Lemma 2 to (9) with $(\omega, \mathbf{u}, \omega^+) = (\omega_f^{t+1/2}, -\eta f(\omega_f^{t+1/2}), \omega_f^{t+1})$ and we have

$$\|\omega_f^{t+1} - \omega_f^{t+1/2}\| \leq -\eta f(\omega_f^{t+1/2})^\top (\omega_f^{t+1} - \omega_f^{t+1/2}) \quad (25)$$

Then we have

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

According to Lemma 3, we have

$$2\eta f(\omega_f^{t+1/2})^\top (\omega_f^{t+1/2} - \omega_f^*) \geq 2\eta\mu\|\omega_f^{t+1/2} - \omega_f^*\|_2^2. \quad (27)$$

Plug (27) into (26) and we get

$$\|\omega_f^{t+1} - \omega_f^*\|_2^2 \leq (2 - 2\eta\mu)\|\omega_f^{t+1/2} - \omega_f^*\|_2^2. \quad (28)$$

The above inequality is equivalent to

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

Similarly, one can obtain

$$\|\omega_h^{t+1} - \omega_h^*\|_2^2 \leq (2 - 2\eta\mu)\|\omega_h^{t+1/2} - \omega_h^*\|_2^2, \quad (30)$$

i.e.,

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

Combining (29) and (31) we have

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

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

E. Details for Experiment Setup

E.1. Synthetic Datasets

We provide the details for two synthetic datasets. The Two-Circle dataset consists of 24 Gaussian mixtures where 8 of them are located in an inner circle with radius $r_1 = 4$ and 16 of them lie in an outer circle with radius $r_2 = 8$. For each Gaussian component, the covariance matrix is $\begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix} = \sigma_1 \mathbf{I}$ and the mean value is $[r_1 \cos t, r_1 \sin t]$, where $t = \frac{2\pi \cdot k}{8}$, $k = 1, \dots, 8$, for the inner circle, and $[r_2 \cos t, r_2 \sin t]$, where $t = \frac{2\pi \cdot k}{16}$, $k = 1, \dots, 16$ for the outer circle. We sample $N_1 = 2000$ points as true observed samples for model training. In section 5.4, 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 $c_1 = \frac{2\pi}{3} + \text{linspace}(0, 0.5, 50) \cdot 2\pi$, for one spiral, and $[c_2 \cos c_2, -c_2 \sin c_2]$, where $c_2 = \frac{2\pi}{3} + \text{linspace}(0, 0.5, 50) \cdot 2\pi$ for another spiral. We sample $N_2 = 5000$ points as true observed samples. In section 5.4, we consider insufficient data scenario. In this case, the sample size N_2 is reduced to $[100, 200, 300, 500, 700, 1000, 2000]$.

Table 2. 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.

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

E.2. Model Specifications and Training Algorithm

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

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

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 vanilla GAN. To well distinguish them, we call the model using Wasserstein distance (resp. JS divergence) as Joint-W (resp. Joint-JS) in our experiments. On the other hand, the two Stein discrepancies in (3) can be specified by KSD (as defined by \mathcal{S}_k in (11)) or general Stein discrepancy with an extra critic (as defined by \mathcal{S} in (1)). 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 2.

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 2 appear to be computationally expensive. In the worst case (using general Stein discrepancy), there are two minimax operations where one is from GAN or WGAN and one is from Stein discrepancy estimation. To guarantee training efficiency, we alternatively update the generator, estimator, Wasserstein critic and Stein critic over the parameters 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.

E.3. Implementation Details

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

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

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  $n = 1, \dots, n_d$  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  $\tilde{\mathbf{x}}_i = G_\theta(\mathbf{z}_i)$ ;
7      $\mathcal{L}_{dis} = \frac{1}{B} \sum_{i=1}^B d_\psi(\mathbf{x}_i) - d_\psi(\tilde{\mathbf{x}}_i) - \lambda(\|\nabla_{\tilde{\mathbf{x}}_i} d_\psi(\tilde{\mathbf{x}}_i)\| - 1)^2$  // the last term is for gradient penalty in
       WGAN-GP where  $\hat{\mathbf{x}}_i = \epsilon_i \mathbf{x}_i + (1 - \epsilon_i) \tilde{\mathbf{x}}_i, \epsilon_i \sim U(0, 1)$ ;
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  $n = 1, \dots, n_c$  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  $\tilde{\mathbf{x}}_i = G_\theta(\mathbf{z}_i)$ ;
12     $\mathcal{L}_{critic} = \frac{1}{B} \sum_{i=1}^B \lambda_1 \mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\mathbf{x})] + \lambda_2 \mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\tilde{\mathbf{x}}_i)]$ ;
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  $\tilde{\mathbf{x}}_i = G_\theta(\mathbf{z}_i)$ ;
15     $\mathcal{L}_{est} = \frac{1}{B} \sum_{i=1}^B \lambda_1 \mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\mathbf{x})] + \lambda_2 \mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\tilde{\mathbf{x}}_i)]$ ;
16     $\phi_{k+1} \leftarrow Adam(\mathcal{L}_{est}, \phi_k, \alpha^E, \beta_1^E, \beta_2^E)$  // update the density estimator;
17     $\mathcal{L}_{gen} = \frac{1}{B} \sum_{i=1}^B -d_\psi(\tilde{\mathbf{x}}_i) + \lambda_2 \mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\tilde{\mathbf{x}}_i)]$ ;
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  $p_\phi(\mathbf{x})$ .

```

in (Kim & Bengio, 2017) for details)

$$E_\phi(\mathbf{x}) = \sum_i \log(1 + e^{-(\mathbf{W}_i n(\mathbf{x}) + b_i)}), \quad (34)$$

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 ((Ngiam et al., 2011))

When not using KSD, the implementation for Stein critic \mathbf{f} and operation function ϕ in (1) has still remained an open problem. Some existing studies like (Hu et al., 2018) set $d' = 1$ in which situation \mathbf{f} reduces to a scalar-function from d -dimension input to one-dimension scalar value. Such setting can reduce computational cost since large d' could lead to heavy computation for training. Empirically, in our experiments on image dataset, we find that setting $d' = 1$ can provide similar performance to $d' = 10$ or $d' = 100$. Hence, we set $d' = 1$ in our experiment in order for efficiency. Besides, to further reduce computational cost, we let the two Stein critics share the parameters, which empirically provide better performance than two different Stein critics.

Another tricky point is how to design a proper Γ given $d' \neq d$ where the trace operation is not applicable. One simple way is to set Γ as some matrix norms. However, the issue is that using matrix norm would make it hard for SGD learning. The reason is that the Γ and the expectation in (1) cannot exchange the order, in which case there is no unbiased estimation by mini-batch samples for the gradient. Here, we specify Γ as max-pooling over different dimensions of $\mathcal{A}_{p_\phi}[\mathbf{f}_\pi(\mathbf{x})]$, i.e. the gradient would back-propagate through the dimension with largest absolute value at one time. Theoretically, such setting can guarantee the value in each dimension reduces to zero through training and we find it works well in practice.

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

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

Method	MNIST		CIFAR	
	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

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. In other words, we consider $d' = 1$ in (1) and further simply ϕ as an average of each dimension of $\mathbb{E}_{\mathbf{x} \sim \mathbb{P}}[\mathcal{A}_{\mathbb{Q}} f(\mathbf{x})]$. Empirically we found this setting can provide efficient computation and decent performance.

E.4. Evaluation Metrics

We adopt some quantitative metrics to evaluate the performance of each method on different tasks. In section 4.1, we use two metrics to test the sample quality: Maximum Mean Discrepancy (MMD) and High-quality Sample Rate (HSR). MMD measures the discrepancy between two distributions X and Y , $MMD(X, Y) = \|\frac{1}{n} \sum_{i=1}^n \Phi(x_i) - \frac{1}{m} \sum_{j=1}^m \Phi(y_j)\|$ where x_i and y_j denote samples from X and Y respectively and 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-Circle 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 measures how well a given sample can be classified into a right class, i.e. the quality of such sample. For CIFAR, we use the Inception V3 Network in Tensorflow as pre-trained classifier.

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

Table 4. 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.

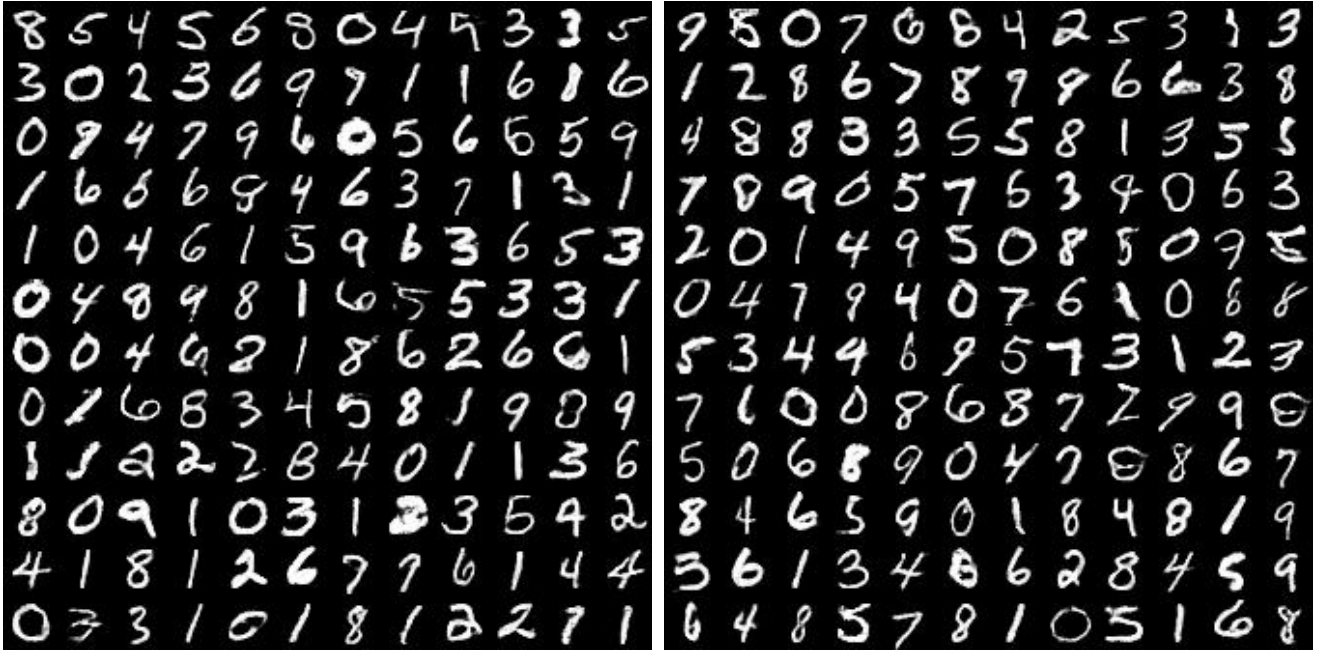
Method	Two-Circle					Two-Spiral				
	MMD	HSR	KLD	JSD	AUC	MMD	HSR	KLD	JSD	AUC
GAN	0.0033	0.772	-	-	-	0.0082	0.583	-	-	-
GAN+LR	0.0106	0.391	-	-	-	0.0068	0.821	-	-	-
GAN+ER	0.0103	0.428	-	-	-	0.0071	0.780	-	-	-
GAN+VA	0.0118	0.295	-	-	-	0.0085	0.761	-	-	-
WGAN-GP	0.0010	0.841	-	-	-	0.0090	0.697	-	-	-
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. l_1 and l_2 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	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. l_1 and l_2 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.)

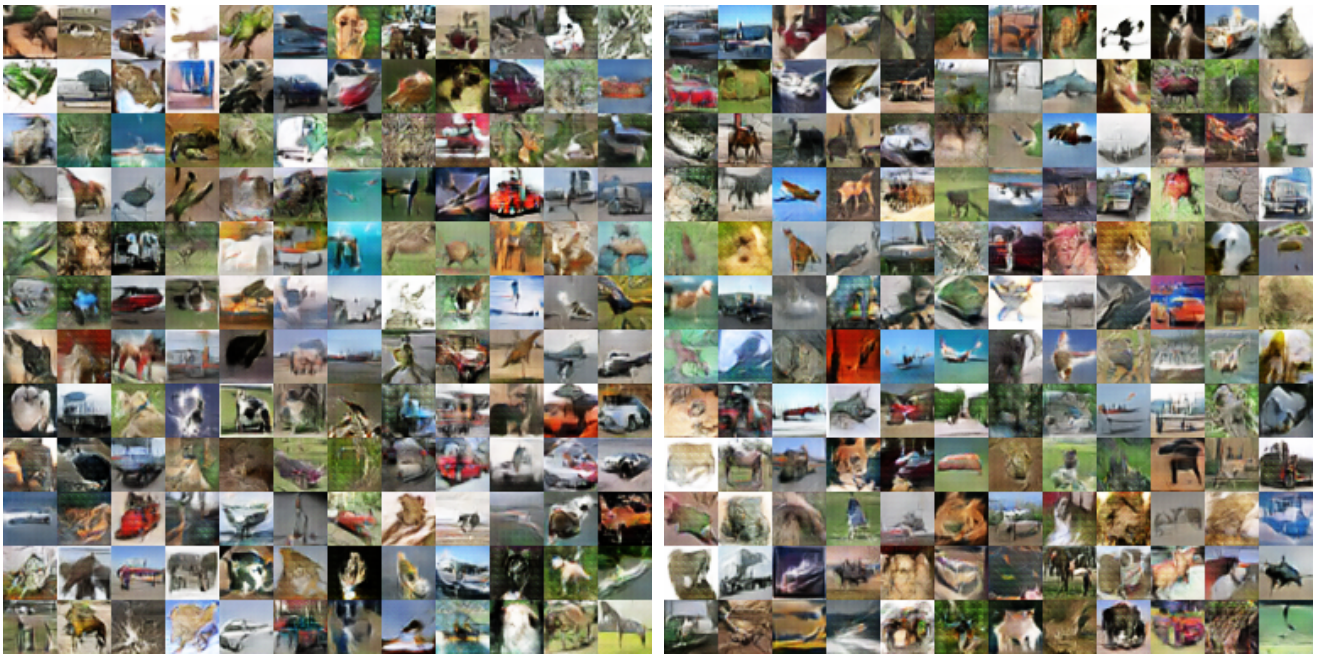
Class		'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'
WGAN-GP	l_1	80.8	82.7	40.2	69.3	44.7	59.2	77.6	107.7	50.81	89.3
	l_2	1.75	1.84	0.92	1.57	1.04	1.40	1.78	2.32	1.78	1.92
WGAN+LR	l_1	78.4	79.2	73.8	86.0	75.8	77.2	106.7	103.0	56.5	92.3
	l_2	1.63	1.76	1.59	1.88	1.68	1.74	2.36	2.23	1.24	2.00
WGAN+ER	l_1	75.5	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



(a) Randomly sampled over all digits

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

Figure 8. Generated digits given by Joint-W on MNIST.



(a) Randomly sampled over all images

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

Figure 9. Generated images given by Joint-W on CIFAR.

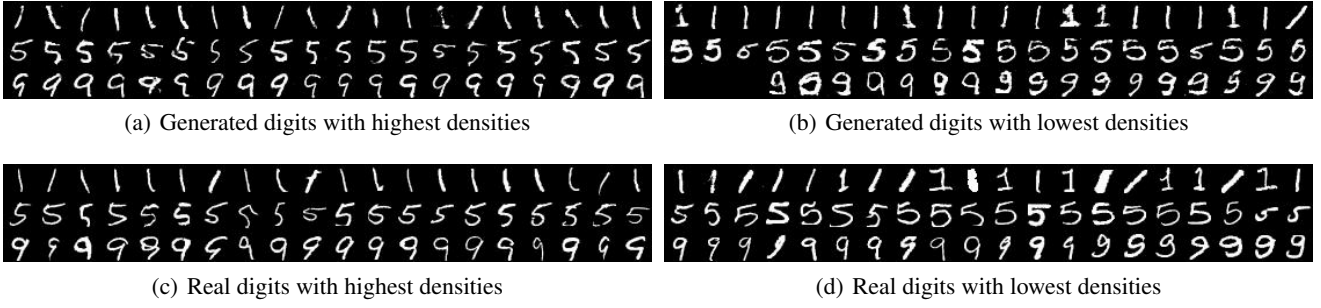


Figure 10. The generated digits (and real digits) with the highest densities and the lowest densities given by Joint-W.

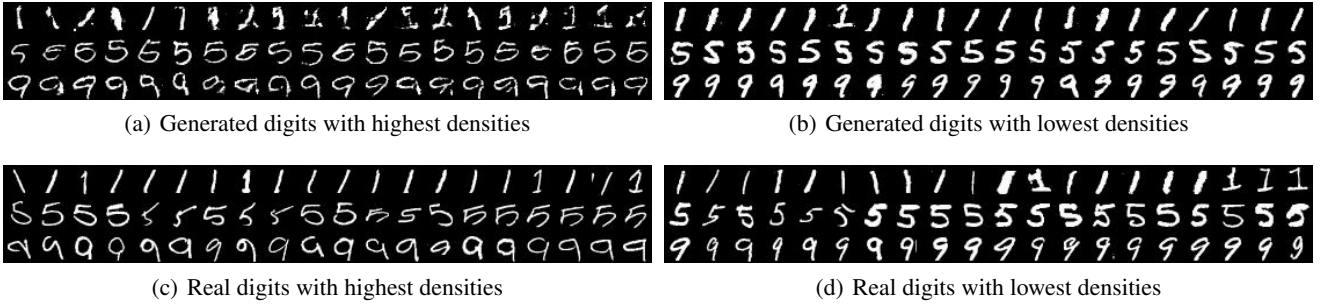


Figure 11. The generated digits (and real digits) with the highest densities and the lowest densities given by DGM.

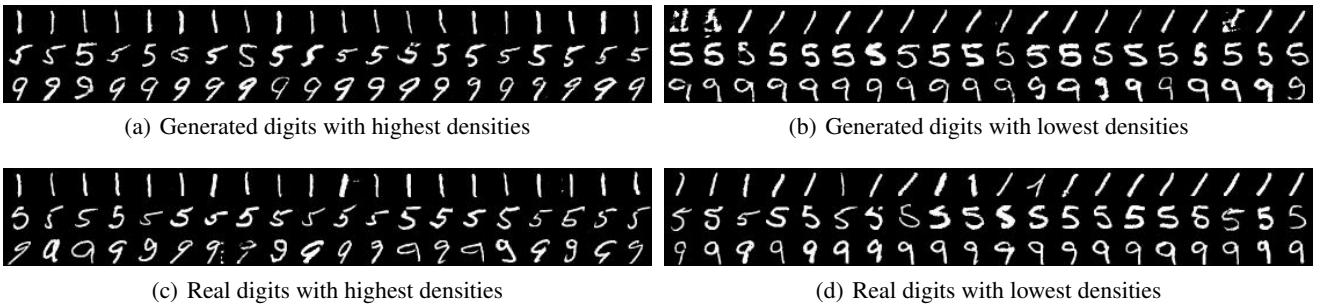


Figure 12. The generated digits (and real digits) with the highest densities and the lowest densities given by EGAN.