Bridging Explicit and Implicit Deep Generative Models via Neural Stein Estimators

Qitian Wu, Rui Gao, Hongyuan Zha

Shanghai Jiao Tong University
University of Texas, Austin
The Chinese University of Hong Kong, Shenzhen

echo740@sjtu.edu.cn

October 18, 2021

Deep Generative Models

- *Explicit model* assumes a (unnormalized) density for each instance x. E.g., the energy model considers a Gibbs distribution,

$$p_\phi(x) = \frac{\exp(-E_\phi(x))}{Z_\phi}, \quad Z_\phi = \int_x \exp(-E_\phi(x))dx,$$

  where $E_\phi(x)$ denotes energy for x and Z_ϕ is a normalizing constant. The energy model explicitly defines a distribution $\mathbb{P}_E$ propotional to $\exp(-E_\phi(x))$.

- *Implicit model* targets a mapping from an easy-to-sample noise z to generated sample x',

$$x' = G(z), \quad z \sim P_0,$$

  where G is called generator and it implicitly defines a distribution $\mathbb{P}_G$ over x. One typical example is Generative Adversarial Network (GAN).

Generative Models

- Explicit models allow a (unnormalized) density value, but often hard to train and sample from (due to the intractable constant). Also, it might not be able to capture the complex structure of data while maintaining tractability.
- Implicit models are flexible in training and easy to sample from. However, it often encounters with practical issues, e.g., unstability of training and local optima (missing some modes in data or generating modes out of data).
- Some situations need both of the worlds: i) sample evaluation for generator, ii) detection of out-of-distribution samples in training, iii) training explicit model using insufficient samples.

Stein Discrepancy

- Assume $q(x)$ a continuously differentiable density supported on $\mathcal{X} \subset \mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ a smooth vector function. Then we have the property (when f satifies some mild conditions):

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

  which is called *Stein Identity*.
- *Stein discrepancy* between $\mathbb{P} : p(\mathrm{x})$ and $\mathbb{Q} : q(\mathrm{x})$ is defined as

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

  where f is *Stein critic* that exploits over function space $\mathcal{F}$ and δ denotes a transformation to scalar value (one typical specification is trace). The definition does not require normalizing constant for $p(x)$.
- *Kernel Stein Discrepancy:* If $\mathcal{F}$ is a unit ball in a Reproducing Kernel Hilbert Space (RKHS) with a positive definite kernel function $k(x, x')$, then the supremum would have a close form.

Wasserstein Metric

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

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

  where the maximization is over all 1-Lipschitz continuous functions.
- The Wasserstein metric inspires Wasserstein GAN (WGAN):

$$\min_G \max_D \ \mathbb{E}_{x\sim\mathbb{P}_{data}}[D(x)] - \mathbb{E}_{z\sim\mathbb{P}_0}[D(G(z))] - \lambda(\|\nabla_x D(x)\| - 1)^2,$$

  where the last term is for gradient penalty.

Stein Bridging

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

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

  where $\lambda \geq 0$ is a weight coefficient.

- To better jointly learn the two models, we incorporate the objective another term:

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

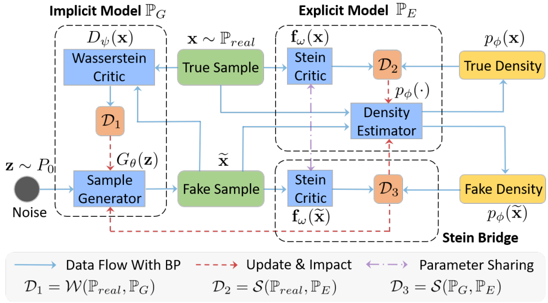
  where $\lambda_1, \lambda_2 \geq 0$ are weight coefficients. We call the third term as *Stein bridge* that measures the closeness between the explicit unnormalized density $\mathbb{P}_E$ and the implicit distribution $\mathbb{P}_G$.

Stein Bridging (Cont.)

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

Remark 1. The Wasserstein metric can be replaced by other common choices for implicit generative models, such as Jensen-Shannon divergence used in the original GAN.
Remark 2. If the explicit model does not require the normalizing constant and assumes a simple distribution form (e.g., PixelCNN++), the two Stein discrepancies can be replaced by the Kullback-Leibler divergence, which is equivalent to the maximum likelihood estimation.
Remark 3. The choice of the Stein discrepancy for the bridging term $\mathcal{S}(\mathbb{P}_G, \mathbb{P}_E)$ is crucial and cannot be replaced by other statistical distances such as KL divergence, since the data-generating distribution does not have an explicit density form (not even up to a normalizing constant).

Stein Bridging: Model Framework



Implicit Model $\mathbb{P}_G$

Explicit Model $\mathbb{P}_E$

$D_\psi(\mathbf{x})$

Wasserstein Critic

$\mathbf{x} \sim \mathbb{P}_{real}$

True Sample

$\mathbf{f}_\omega(\mathbf{x})$

Stein Critic

$\mathcal{D}_2$

$p_\phi(\mathbf{x})$

True Density

$p_\phi(\cdot)$

Density Estimator

$\mathcal{D}_1$

$G_\theta(\mathbf{z})$

$\widetilde{\mathbf{x}}$

$\mathbf{z} \sim P_0$

Noise

Sample Generator

Fake Sample

Stein Critic

$\mathcal{D}_3$

Fake Density

$\mathbf{f}_\omega(\widetilde{\mathbf{x}})$

$p_\phi(\widetilde{\mathbf{x}})$

Stein Bridge

Data Flow With BP

Update & Impact

Parameter Sharing

$\mathcal{D}_1 = \mathcal{W}(\mathbb{P}_{real}, \mathbb{P}_G)$

$\mathcal{D}_2 = \mathcal{S}(\mathbb{P}_{real}, \mathbb{P}_E)$

$\mathcal{D}_3 = \mathcal{S}(\mathbb{P}_G, \mathbb{P}_E)$

When using Wasserstein metric and Stein discrepancy, the objective can be equivalently written as

$$\min_\theta \min_\phi \max_\psi \max_\pi \mathbb{E}_{x\sim\mathbb{P}_{data}}[D_\psi(x)] - \mathbb{E}_{z\sim P_0}[D_\psi(G_\theta(z))]$$

$$+\lambda_1 \mathbb{E}_{x\sim\mathbb{P}_{data}}[\mathcal{A}_{p_\phi}[f_\pi(x)]] + \lambda_2 \mathbb{E}_{z\sim P_0}[\mathcal{A}_{p_\phi}[f_\pi(G_\theta(z))]],$$

where G_θ, p_ϕ, D_ψ and f_π are generator, energy estimator, Wasserstein critic and Stein critic (shared by two Stein terms), respectively.

Training Algorithm

```
REQUIRE: observed training samples $\{\mathrm{x}\} \sim \mathbb{P}_{real}$.
REQUIRE: $\theta_0$, $\phi_0$, $\psi_0$, $\pi_0$, initial parameters for generator, estimator, Wasserstein critic and Stein critic models respectively. $B = 100$, batch size. $n_d = 5$, $n_c = 5$.
While not converged:
  For $n = 1, \cdots, n_d$:
    Update Wasserstein critic $\psi$ with a mini-batch.
  For $n = 1, \cdots, n_c$:
    Update Stein critic $\pi$ with a mini-batch.
  Update the energy estimator $\phi$ with a mini-batch.
  Update the sample generator $\theta$ with a mini-batch.
OUTPUT: trained sample generator $G_\theta(\mathrm{z})$ and energy estimator $p_\phi(\mathrm{x})$.
```

Regularization Effects by Stein Bridge

Theorem

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

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

where $\|D\|_{H^{-1}(\mathbb{P}_E;k)}$ denotes a kernel Sobolev dual norm with kernel $k(x,x')$.

- The regularization term would penalize the non-smoothness of the Wasserstein critic D, which is in the same spirit of gradient-based penalty.
- If $k(x,x') = \mathbb{I}(x = x')$ and $\mathbb{E}_{\mathbb{P}_E}[D] = 0$, then

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

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

Regularization Effects by Stein Bridge (Cont.)

Theorem

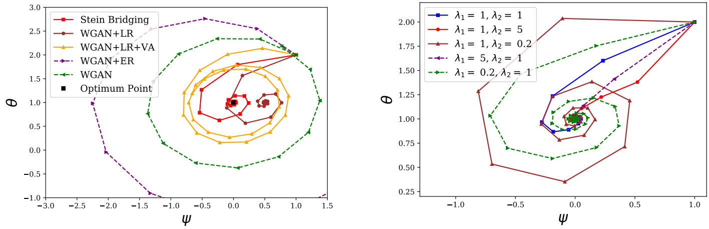
Assume $\{\mathbb{P}_G\}_G$ exhausts all continuous probability distributions, and the Stein class defining the Stein discrepancy is compact (in some linear topological space). Then our problem is equivalent to

$$\min_E \max_f \{\lambda_1 \mathcal{S}(\mathbb{P}_{data}, \mathbb{P}_E) + \lambda_2 \mathbb{E}_{x \sim \mathbb{P}_{data}} [M_{\lambda_2 \mathcal{A}_{\mathbb{P}_E} f}(x)]\},$$

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

Theorem 2 shows that the Stein Bridge plays as a smoothness regularization on the Stein critic f, which smoothens the Stein operator $\mathcal{A}_{\mathbb{P}_E} f$ and further encourages the energy model to seek more modes in data, thus helping alleviate the *mode-collapse issue*.

Stabilize Training

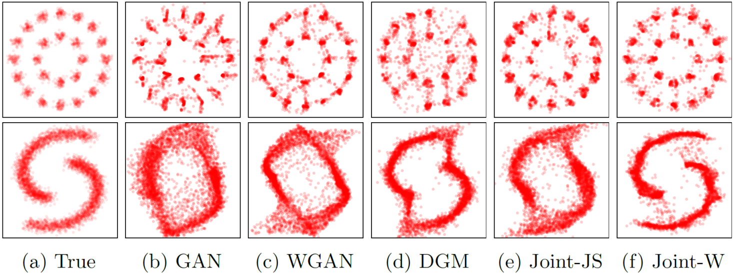


(a) Numerical iterations of different models

(b) Stein Bridging with different weight parameters

- The Wasserstein GAN suffers from ocsillating behavior instead of convergence. Likelihood regularization can help for convergence, but would change the optimum point (leads to a biased model distribution).
- Stein Bridging can guarantee convergence without changing the global optimum.
- Different weights would impact the convergence speed in training.

Quality of Generated Samples


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

- Datasets: i) Two-Circle (24 Gaussian components whose centers locate at two circles); ii) Two-Spiral (100 Gaussian components distributed on two spiral-shaped curves).
- Joint-JS and Joint-W can generate more high-quality samples compared with GAN, WGAN and Deep Directed Generative (DGM) Model [Kim & Bengio et al., 2017].

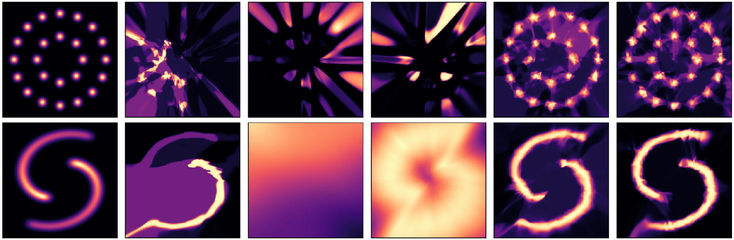
Quality of Generated Samples

Table. Inception scores of generated samples on CIFAR-10.

CIFAR-10 (Unconditional)	
Method	Score
WGAN-GP	6.80
WGAN+LR	6.89
WGAN+ER	6.99
WGAN+VA	6.95
DGM	4.79
Joint-W(ours)	**7.18**

WGAN+LR, WGAN+ER and WGAN+VA are for WGAN with likelihood regularization, entropy regularization, (oscillating) variational annealing regularization [Tao et al., 2019] (that combines the above two), respectively.

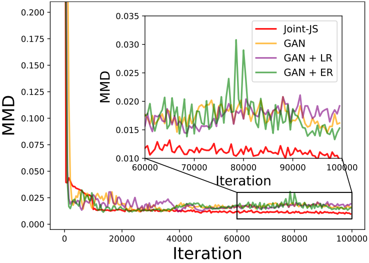
Accuracy of Estimated Energy

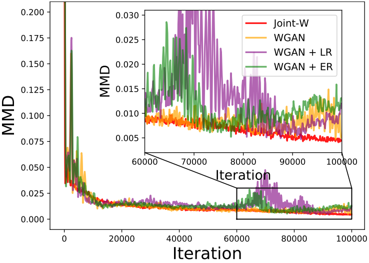


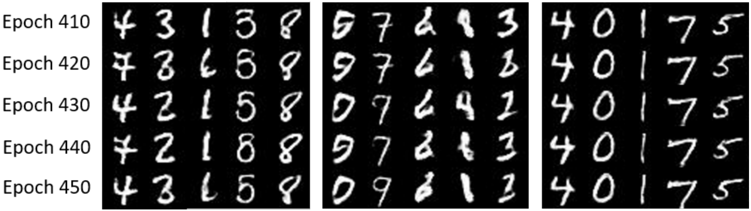
(a) True (b) DEM (c) EGAN (d) DGM (e) Joint-JS (f) Joint-W

- Deep energy model (DEM) (trained by Stein discrepancy) fails to capture all the modes in data (mode-collapse).
- Energy-based GAN (EGAN) [Dai et al., 2017] degrades to a nearly uniform distribution on Two-Spiral dataset.
- Our models manage to exactly fit the ground-truth distributions.

Training Stability





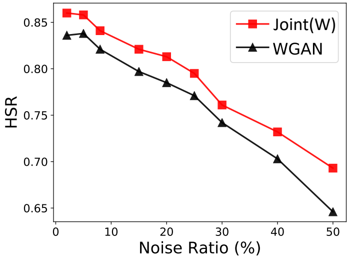


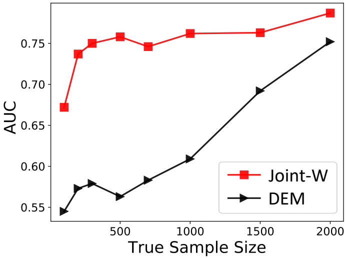
WGAN-GP

WGAN+VA

Joint-W

Applications of Stein Bridging





- The explict energy model can help to detect noised samples in training and further guides the implicit generator to focus more on the in-distribution samples.
- The implicit generator can help to augment the dataset by generating samples, which addresses the difficulty for training of explicit model with insufficient samples.

Conclusion and Future Direction

In this work, we propose a principle approach, Stein Bridging, using Stein discrepancy to jointly train explicit and implicit generative models. There are some interesting directions for future works:

- Consider some induction bias for one generative model and use joint training to propagate such effect to another model.
- Study the training dynamic of joint model and incorporate some adaptive control strategy. Indeed, we empirically find that gradually increasing the weight for Stein bridge in training can provide better performance.
- Consider information-sharing mechanism between two models in training. For example, use the estimated energy as a 'score' that measures the quality of a sample and re-weight the input samples for implicit model.