Semi-implicit generative model

Anonymous Author(s)
Affiliation
Address
email

Abstract

To combine explicit and implicit generative models, we introduce semi-implicit generator (SIG) as a flexible hierarchical model that can be trained in the maximum likelihood framework. Both theoretically and experimentally, we demonstrate that SIG can generate high quality samples especially when dealing with multi-modality. By introducing SIG as an unbiased regularizer to an generative adversarial network (GAN), we show the interplay between maximum likelihood and adversarial learning can stabilize the adversarial training, resist the notorious mode collapsing problem of GANs, and improve the diversity of generated random samples.

1 Introduction

Generative models consist of a group of fundamental machine learning algorithms that are used to estimate the underlying probability distributions over data manifolds. Promoted by recent development in deep neural networks, deep generative models achieve great success in data simulation, density estimation, missing data imputation, reinforcement learning and are widely utilized for tasks such as image super-resolution, compression and image-to-text translation. The goal of generative models is to minimize the distance between the generative distribution and data distribution under a certain metric or divergence D

$$\min_{\boldsymbol{\phi}} D(P_{data}(\boldsymbol{x})||P_{model}(\boldsymbol{x};\boldsymbol{\phi})) \tag{1}$$

where P_{data} is usually approximated with empirical data distibution $\hat{P}_{data} = \frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{x}_i}$ based on observations $\{\boldsymbol{x}_i\}_{1:N}$.

Depending on the type of $P_{model}(\boldsymbol{x};\boldsymbol{\theta})$, an existing generative model can often be classified as either an explicit generative model or implicit one. The former requires an explicit probability density function (PDF) for P_{model} such that we can both sample data from it and evaluate its likelihood. Examples for explicit generative models include variational auto-encoders [1], PixelRNN [2], Real NVP [3], and many Bayesian hierarchical models such as sigmoid belief net [4]. An explicit generative model has a tractable density that can often be directly optimized by (1). The optimization target is a distance measure with nice geometric properties, which often leads to stable training and theoretically guaranteed convergence. However, the requirement of having a tractable density usually restricts the flexibility of an explicit model, making it hard to scale with increasing data complexity.

An implicit generative model, on the other hand, generates its random samples via a stochastic procedure but may not allow a point-wise evaluable PDF, which often makes a direct optimization in (1) become infeasible. Generative adversarial networks (GANs) [5] tackle this problem by introducing an augmented discriminator and solving a minimax game: a generative network generates random samples by propagating random noises through a deep neural network, whereas a discriminator aims to distinguish the generated samples from true data. Under the condition of having an optimal discriminator, training a vanilla GAN's generator is equivalent to optimizing (1) where D is set as the Jensen-Shannon Divergence. Unfortunately in practice, the overall loss function of GAN is usually

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

non-convex and practitioners have encountered a variety of obstacles such as gradient vanishing, mode collapsing, and high sensitivity to the network architecture [5–8].

To incorporate highly expressive generative model while maintaining a well-behaved optimization objective, we introduce semi-implicit generator (SIG), a Bayesian hierarchical generative model that mixes a specified distribution $P(\boldsymbol{x} \mid \boldsymbol{\theta})$ with an implicit distribution $P_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ where the implicit distribution can be constructed by deterministically transforming random noise $\boldsymbol{z}_i$ to $\boldsymbol{\theta}_i$ using a $\boldsymbol{\phi}$ parameterized deterministic transform as $\boldsymbol{\theta}_i = g_{\boldsymbol{\phi}}(\boldsymbol{z}_i),\ \boldsymbol{z}_i \sim p(\boldsymbol{z})$. Intuitively, $P(\boldsymbol{x} \mid \boldsymbol{\theta})$ can incorporate our prior knowledge on the observed data, such as the data support, while $P_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ can maintain the high expressiveness. With the hierarchical structure, SIG can be directly trained by choosing D as the Kullback-Leibler(KL) divergence and estimating (1) with Monte-Carlo estimation. We show the SIG optimization objective can intrinsically resist the mode-collapse problem. By leveraging adversarial training, we apply SIG as a semi-implicit regularizer to generative adversarial networks, which helps stabilize optimization, significantly mitigates mode collapsing, and generates high quality samples in natural image scenarios.

2 Semi-implicit generator

Defining a family of parametric distribution $P_{model}(\boldsymbol{x}; \boldsymbol{\theta})$, a classic explicit generative model is trained by maximizing the log-likelihood as

$$\max_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \log P_{model}(\boldsymbol{x}_i; \boldsymbol{\theta}), \tag{2}$$

which is identical to minimize cross-entropy $\mathbb{H}(\hat{P}_{\text{data}}, P_{model}) = -\mathbb{E}_{\hat{P}_{data}(\boldsymbol{x})} \log P_{model}(\boldsymbol{x}; \boldsymbol{\theta})$. Assuming $P_{\text{data}} = \hat{P}_{\text{data}}$, which is independent of the optimization parameter $\boldsymbol{\theta}$, minimizing this cross-entropy is equivalent to (1) where D is set as the KL divergence.

Instead of treating $\boldsymbol{\theta}$ as a global optimization parameter, we consider $\boldsymbol{\theta}$ as local random variable generated from distribution $p_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ with parameter $\boldsymbol{\phi}$. Semi-implicit generator (SIG) is defined in a two-stage manner

$$\boldsymbol{x}_i \sim p(\boldsymbol{x} \mid \boldsymbol{\theta}_i),\ \boldsymbol{\theta}_i \sim p_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \tag{3}$$

Marginalizing $\boldsymbol{\theta}_i$ out, we can view the generator as $P_{model}(\boldsymbol{x}_i; \boldsymbol{\phi}) = \int p(\boldsymbol{x}_i \mid \boldsymbol{\theta}_i) p_{\boldsymbol{\phi}}(\boldsymbol{\theta}_i) d\boldsymbol{\theta}_i$. Here $p(\boldsymbol{x}_i \mid \boldsymbol{\theta}_i)$ is required to be explicit but $p_{\boldsymbol{\phi}}(\boldsymbol{\theta}_i)$ can be defined by sampling a random variable $\boldsymbol{z}_i$ from fixed distribution $p(\boldsymbol{z})$ and setting $\boldsymbol{\theta}_i = g_{\boldsymbol{\phi}}(\boldsymbol{z}_i)$, where $g_{\boldsymbol{\phi}} : \mathcal{Z} \to \mathcal{X}$ is a deterministic mapping represented by neural network with parameter $\boldsymbol{\phi}$. Therefore, typically $p_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ cannot be evaluated pointwisely and the marginal $P_{model}(\boldsymbol{x}; \boldsymbol{\phi})$ is implicit. Notice in this setting, $\boldsymbol{\theta}$ is required to be continuous while $\boldsymbol{x}$ can be sampled from discrete distribution with continuous parameters.

Minimizing the cross-entropy as $\mathbb{H}(P_{\text{data}}, P_{model})$ is equivalent to minimizing the KL-divergence with respect to the model parameter as in (1)

$$\min_{\boldsymbol{\phi}} \text{KL}(P_{data}(\boldsymbol{x}) || P_{model}(\boldsymbol{x}; \boldsymbol{\phi})) \tag{4}$$

$$\Leftrightarrow \min_{\boldsymbol{\phi}} \mathbb{H}(P_{data}, P_{model}) = -\mathbb{E}_{P_{data}(\boldsymbol{x})} \log \left(\int p(\boldsymbol{x} \mid \boldsymbol{\theta}) p_{\boldsymbol{\phi}}(\boldsymbol{\theta}) d\boldsymbol{\theta} \right) \tag{5}$$

We show below that SIG can be trained by minimizing an upper bound of the cross entropy in (5).

Lemma 1. *Let us construct an estimator for the cross-entropy $\mathbb{H}(P_{data}, P_{model})$ as*

$$\mathbb{H}_M = -\mathbb{E}_{p_{data}(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M \sim p_{\boldsymbol{\phi}}(\boldsymbol{\theta})} \log \frac{1}{M} \sum_{j=1}^{M} p(\boldsymbol{x} \mid \boldsymbol{\theta}_j), \tag{6}$$

then for all M, $\mathbb{H}(P_{data}, P_{model}) \leq \mathbb{H}_{M+1} \leq \mathbb{H}_M$ and $\mathbb{H}(P_{data}, P_{model}) = \lim_{M \to \infty} \mathbb{H}_M$. When $M = 1$, let $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} -\mathbb{E}_{P_d(\boldsymbol{x})} \log[p(\boldsymbol{x} | \boldsymbol{\theta})]$ then $\mathbb{H}_1 \geq -\mathbb{E}_{p_d(\boldsymbol{x})} \log[p(\boldsymbol{x} | \boldsymbol{\theta}^*)]$ where the equality is true if and only if $p_{\boldsymbol{\phi}}(\boldsymbol{\theta}) = \delta_{\boldsymbol{\theta}^*}(\boldsymbol{\theta})$.

In practice, $\mathbb{H}_M$ is approximated with Monte-Carlo samples as $-\frac{1}{N} \sum_{i=1}^{N} \log \frac{1}{M} \sum_{j=1}^{M} p(\boldsymbol{x}_i \mid \boldsymbol{\theta}_j)$, where $\{\boldsymbol{x}_i\}_{1:N}$ and $\{\boldsymbol{\theta}_j\}_{1:M}$ are two sets of Monte Carlo samples generated from $P_{\text{data}}(\boldsymbol{x})$ and

implicit $P_{\boldsymbol{\phi}}(\boldsymbol{\theta})$, respectively. When $M = 1$, the local $\boldsymbol{\theta}_i$ will degenerate to the same $\boldsymbol{\theta}^*$ and the objective degenerate to (2). To analyze the performance of SIG, we first consider multi-modal data on which popular deep generative models such as GANs often fail due to mode collapsing. For theoretical analysis, we first define a discrete multi-modal space as follows.

Definition 1. *(Discrete multi-modal space) Suppose $(\mathcal{X}, \nu)$ is a metric space with metric $\nu : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}^+$, $\mathcal{X} = \bigcup_{i=1}^{K} U_i$, where $U_i \cap U_j = \emptyset$ for $i \neq j$. Let the distance between two sets be $D(U_i, U_j) = \inf\{\nu(x, y); x \in U_i, y \in U_j\}$ and let the diameter of a set be $d(U) = sup\{\nu(x, y); x, y \in U\}$. Suppose there exists $c_0 > \epsilon_0 > 0$ such that $\min_{i,j} D(U_i, U_j) > c_0$, $\max_i d(U_i) < \epsilon_0$. Then $\mathcal{X} = \bigcup_{k=1}^{K} U_k$ is a discrete multi-modal space under mesure ν.*

Strictly speaking, there could be sub-modes within each U_i, but the above definition emphasizes the existence of multiple separated regions in the support. Since the loss of a deep neural network is a nonconvex problem, finding the global optimality condition for $\boldsymbol{\phi}$ can be difficult [9, 10]. Thanks to the structure of SIG as a two-stage model, assuming the implicit distribution is flexible enough, we can study a simplified optimal assignment problem: assuming that N data points have been sampled from the true data distribution, how to assign M generated data to the neighborhood of the true data such that $\mathbb{H}_M$ defined in (6) is minimized under expectation

$$\min_{\{m_1,\cdots,m_k\}} -\frac{1}{N}\sum_{i=1}^{N} \log \frac{1}{M}\sum_{j=1}^{M} \mathbb{E}_{\boldsymbol{x}_i \in U_{t_i}, \boldsymbol{\theta}_j \in U_{z_j}}[p(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}_j)], \tag{7}$$

where the data are assumed to be generated from a discrete multi-modal space $\mathcal{X} = \bigcup_{i=1}^{K} U_i$, $\boldsymbol{x}_i \in U_{t_i}$, $\boldsymbol{\theta}_j \in U_{z_j}$, $t_i, z_j \in \{1, \ldots, K\}$ and $\{m_k\}_{k=1}^K$ are the number of $\boldsymbol{\theta}$'s that are assigned to be in U_k. Assuming the data distribution is the marginal distribution of a normal-implicit mixture as $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}_i, \sigma^2 I)$, $\boldsymbol{\theta}_i \sim p_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ and U_i are equally spaced, we have the following theorem.

Theorem 1. *(SIG for multi-modal space) Suppose P_{data} is defined on a discrete multi-modal space $\mathcal{X} = \bigcup_{i=1}^{K} U_i$ with l_2-norm. Suppose there are N data points $\boldsymbol{x}_i \sim P_{data}, i = 1, \cdots, N$, among which n_k points belong to U_k. Suppose we need to sample $\boldsymbol{\theta}_j \sim p_{\boldsymbol{\phi}}(\boldsymbol{\theta}), j = 1, \cdots, M$, and m_k denotes the number of $\boldsymbol{\theta}$'s in U_k. Denoting r as a radial basis function (RBF), we let $u = \mathbb{E}[r(\boldsymbol{x}, \boldsymbol{\theta})]$ if $\boldsymbol{x}, \boldsymbol{\theta} \in U_i$, and $v = \mathbb{E}[r(\boldsymbol{x}, \boldsymbol{\theta})]$ if $\boldsymbol{x} \in U_i$, $\boldsymbol{\theta} \in U_j, i \neq j$. Then the objective in* (7) *is convex and the optimum m_k to maximize* (7) *satisfies $\frac{m_k^*}{M} = \frac{n_k}{N} + (\frac{n_k}{N} - \frac{1}{K})\frac{Kv}{(u-v)}$. In particular, $m_k^* \neq 0$ if $n_k > \frac{N}{K}\frac{1}{1+\frac{u-v}{Kv}}$.*

Corollary 1. *Assume $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \sigma^2 I)$ and $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \sigma^2 I)$. Let $u = \mathbb{E}\exp(\frac{-\|\boldsymbol{x}-\boldsymbol{\theta}\|^2}{2\sigma^2})$ if $\boldsymbol{\mu}_x = \boldsymbol{\mu}_{\boldsymbol{\theta}}$ and $v = \mathbb{E}\exp(\frac{-\|\boldsymbol{x}-\boldsymbol{\theta}\|^2}{2\sigma^2})$ if $\|\boldsymbol{\mu}_x - \boldsymbol{\mu}_{\boldsymbol{\theta}}\| = c\sigma$, then $\frac{v}{u-v} = \frac{1}{e^{c^2/6-1}}$.*

The ideal proportion for $\frac{m_k^*}{M}$ would be $\frac{n_k}{N}$, and $(\frac{n_k}{N} - \frac{1}{K})\frac{Kv}{u-v}$ plays the role as bias. In the normal-implicit mixture case, as shown in Corollary 1, if $\boldsymbol{x} \in U_{t_i}$, $\boldsymbol{\theta} \in U_{z_j}$ are approximately normal distributed, $\frac{v}{u-v}$ can be exponentially small for well separated modes. This indicates that SIG has a strong built-in resistance to model collapsing.

There is an interesting connection between SIG and variational auto-encoder (VAE) [1, 11]. VAE tries to maximize the evidence lower bound as ELBO $= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[-\text{KL}(q(\boldsymbol{z} \,|\, \boldsymbol{x})||p_{\text{model}}(\boldsymbol{x}, \boldsymbol{z}))]$, which is the same as maximizing

$$-\text{KL}(q(\boldsymbol{z} \,|\, \boldsymbol{x})p_{\text{data}}(\boldsymbol{x})||p_{\text{model}}(\boldsymbol{x}, \boldsymbol{z})), \tag{8}$$

for which the optimal solution is $q(\boldsymbol{z} \,|\, \boldsymbol{x})p_{\text{data}}(\boldsymbol{x}) = p_{\text{model}}(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x} \,|\, \boldsymbol{z})p(\boldsymbol{z})$. Therefore, VAE imposes the constraint that there exits a recognition network/encoder $q(\boldsymbol{z} \,|\, \boldsymbol{x})$, which is inferred by minimizing the KL divergence from $p_{\text{model}}(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x} \,|\, \boldsymbol{z})p(\boldsymbol{z})$, the joint distribution of the model, to $\hat{p}_{\text{data}}(\boldsymbol{z}, \boldsymbol{x}) = q(\boldsymbol{z} \,|\, \boldsymbol{x})p_{\text{data}}(\boldsymbol{x})$, the joint distribution specified by the data distribution and encoder.

In SIG, we maximize

$$-\text{KL}(p_{\text{data}}(\boldsymbol{x})||\int p(\boldsymbol{x} \,|\, \boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}) = -\text{KL}(\int q(\boldsymbol{z} \,|\, \boldsymbol{x})p_{\text{data}}(\boldsymbol{x})d\boldsymbol{z}||\int p(\boldsymbol{x} \,|\, \boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}), \tag{9}$$

where $q(\boldsymbol{z} \mid \boldsymbol{x})$ can be any valid probability density/mass function. VAE tries to match the joint distribution between the data combined with its encoder and the model, whereas SIG only cares about matching the marginal model distribution and the data distribution. It is clear that SIG does not require a specific encoder structure and hence provides more flexibility.

In experiments, we find that SIG can generate high-quality data samples on relatively simple data manifolds such as MNIST, but observe that the richness of its generated images can be hard to scale well with high data complexity, such as CelebA dataset with 200K 409×687 RGB images. More specifically, when setting $M = 100$, we find the effect of "mode averaging" on generated images for complex data. We suspect that M needs to scale with data complexity such that $\mathbb{H}_M$ is close to $\mathbb{H}(P_{\text{data}}, P_{model})$ and this is the price we pay for SIG to have a stable training with a strong resistance against mode collapsing. While SIG performs well on relatively simple data but suffers from "mode averaging" on complex natural images, the generative adversarial network (GAN) has shown the ability to generate high quality samples with large scale observed data, but suffers from "model collapsing" even on a simple mixture of Gaussians. To benefits from both worlds, we apply SIG as a regularizer in adversarial learning, which can produce realistic samples, while strongly resisting both the mode collapsing and unstable optimization problems that are notorious for the training of GANs.

3 Generative adversarial network with semi-implicit regularizer

Generative adversarial network (GAN) [12] solves a minimax problem

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}))] \tag{10}$$

It is shown in [5, 13] that if the generator loss is changed from $\mathbb{E}_{\boldsymbol{z}}[\log(1 - D(G(\boldsymbol{z}))]$ to $\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \exp(\sigma^{-1}(D(G(\boldsymbol{z}))))$, under the optimal discriminator condition, the generator loss (10) is identical to the SIG loss (4), which means SIG can be considered as training with the GAN's objective, using the optimal discriminator for the update of the generator. The discriminator in GAN can be considered as an augmented part of the model to avoid density evaluation and indirectly feed the information of the real data to optimizing the generator. With the help of the discriminator, the weak fitting of generator to real data brings high expressive samples that go beyond memorizing inputs. However, recently extensive research in both practical experiments [8, 14] and theoretical analysis [15–17] show that the lack of capacity, insufficient training of the discriminator, and the mismatches between the generator and discriminator in both network types and structures are the root causes of a variety of obstacles in GAN training. It also has been observed in [12] and highlighted in [6, 14] that the optimal generator for a fixed discriminator is a sum of delta functions at the $\boldsymbol{x}$'s, where the discriminator assigns the highest value, which eventually collapses the generator to produce a small family of similar samples. In comparison, SIG is trained by maximizing likelihood without using a discriminator, which can be considered as a strong fitting between real data and generated samples directly. This encourages us to combine the two models and apply SIG as a regularizer in a GAN model, which is referred to as GAN-SI.

For GAN-SI, the discriminative loss is

$$\min_{\boldsymbol{\gamma}} -\mathbb{E}_{\boldsymbol{x} \sim P_d} \log D_{\boldsymbol{\gamma}}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})} \log(1 - D_\gamma(T_{\boldsymbol{\phi}}(\boldsymbol{z}))) \tag{11}$$

and generator loss is a linear combination of the original GAN loss and SIG loss as

$$\min_{\boldsymbol{\phi}} -\mathbb{E}_{\boldsymbol{z} \sim g(\boldsymbol{z})} \log D_\gamma(T_{\boldsymbol{\phi}}(\boldsymbol{z})) - \lambda \mathbb{E}_{\boldsymbol{x} \sim P_d} \log \int p(\boldsymbol{x} \mid \boldsymbol{\theta}) p_{\boldsymbol{\phi}}(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{12}$$

where $\boldsymbol{\gamma}$ is the neural network parameter of the discriminator, $T_{\boldsymbol{\phi}}$ is the deterministic transform for the implicit distribution in SIG, $p(\boldsymbol{x} \mid \boldsymbol{\theta})$ is chosen as $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\theta}, \sigma^2 I)$ for image generation, and $\lambda \geq 0$ is a hyperparameter to balance the strength between the GAN and SIG objectives. In practice, we set λ such that the GAN's generator loss and the cross-entropy term in (12) are on the same scale. The neural networks are set according to the DCGAN [8].

Since SIG can be considered as training GAN with a theoretically optimal discriminator, by adjusting λ, we are able to interpolate between the standard GAN training and true generator loss, therefore balancing the discrimination-generalization trade-off in the GAN dynamics [16]. This idea is related to Unrolled GAN [14] in which the discriminator parameter is temporarily updated K times before updating the generator and the look-forwarded discriminator parameters are used to train the current

generator. By adjusting the unrolling steps K, Unrolled GAN can also interpolate between the standard GAN ($K = 0$) and optimal discriminator GAN ($K = \infty$). However in Unrolled GAN, the discriminator for ($K = \infty$) is not the theoretically optimal discriminator but a fully optimized one that is still influenced by the network design and data complexity. The effectivity of Unrolled GAN in improving stability and mode-coverage is explained by the intuition that the training for the generator with looking ahead technique can take into account the discriminator's reaction in the future, thus helping spread the probability mass. But there is no theoretical analysis provided yet. Moreover, the interpolation is non-linear, a few orders of magnitude slower as shown by [18], which makes picking K not easy. Training GAN with a semi-implicit regularizer benefits from both theoretical explanation and low extra computation, and shows the improved performance on reducing mode collapsing and increasing the stability of optimization in multiple experiments.

4 Related work

Using a two-stage model is related to Empirical Bayes (EB) [19, 20]. A Bayesian hierarchical model can be represented as $\boldsymbol{x}_i \sim p(\boldsymbol{x}_i \mid \boldsymbol{\theta}_i), \boldsymbol{\theta}_i \sim p(\boldsymbol{\theta}_i \mid \boldsymbol{\phi}), \boldsymbol{\phi} \sim p(\boldsymbol{\phi})$, where $p(\boldsymbol{\phi})$ is a hyper-prior distribution. In EB, the hyperprior $p(\boldsymbol{\phi})$ is dropped and the data is used to provide information about $\boldsymbol{\phi}$ such that the marginal likelihood $\prod_i p(\boldsymbol{x}_i \mid \boldsymbol{\phi})$ is maximized. Previous learning algorithms for EB are often based on simple methods such as Expectation-Maximization and moment-matching. SIG can be considered as a parametric EB model where the neural network parameters are represented by $\boldsymbol{\phi}$ and the training objective is to find the maximum marginal likelihood (MMLE) solution of $\boldsymbol{\phi}$ [21].

Without an explicit probability density, the evaluation of GAN has been considered challenging. There have been several recent attempts to introduce maximum likelihood to the GAN training [22, 23]. Flow-GAN [23] constructs a generative model based on normalizing flow, which has been proven as an effective way to expand the distribution family in variational inference. Normalizing flow, however, requires the deterministic transformation to be invertible, a constraint that is often too strong to allow it to generate satisfactory random samples by its own. Therefore, its main use is to be combined with GAN to help improve its sample quality.

There has been significant recent interest in improving the vanilla GAN objective. For example, the measure between the data and model distributions can be changed to the KL-divergence [13] or Wasserstein distance [6]; variational divergence estimation and density ratio estimation approaches have been used to extent the measure to a family of f-divergence [24, 25]; a mutual information term has been introduced into the generator loss to enable learning disentangled representation and visual concepts [26]; and based on a heuristic intuition, two regularizers with an auxiliary encoder are introduced to stabilize the training and improve mode-catching, respectively [27].

A variety of GAN research focuses on solving the mode collapse problem via new methodology and/or theoretical analysis. Encoder-decoder GAN architectures, such as MDGAN [27], VEEGAN [18], BiGAN [28], and ALI [29], use an encoding network to learn reversed mapping from the data to noise. The intuition is that training an encoder can force the system to learn meaningful mapping that can transform imbedded codes to data points from different modes. Unrolled GAN [14], as discussed in the previous section, interpolates between the vanilla GAN discriminator and optimal discriminator that resists model collapsing. AdaGAN [30] takes a boosting-like approach which is trained on weighted samples with more weights assigned to missing modes. From a theoretical perspective, it is shown that if the discriminator size is bounded by p, even the generator loss is ϵ close to optimal, the output distribution can be supported only on $\mathcal{O}(p \log p/\epsilon^2)$ images [31]. A simplified GMM-GAN is used to theoretically show that the optimal discriminator dynamics can converge to the ground truth in total variation distance, while a first order approximation of the discriminator leads to unstable GAN dynamics and mode collapsing [15]. A negative conclusion is made that the encoder-decoder training objective cannot learn meaningful latent codes and avoid mode collapsing [32]. These theoretical analyses do support our practice of combining the GAN and SIG objectives.

5 Experiments

In this section, we first demonstrate the stability and mode coverage property of SIG on synthetic datasets. The toy examples show SIG can capture skewness, multimodality, and generate both continuous and discrete random samples that are indistinguishable from the true data. By interpolating

between MLE and adversarial training scheme, we show GAN-SI can balance sample quality and diversity on real dataset. The evaluation criterion of generative model, however, is not straightforward and no single metric is conclusive on its own. Therefore, we exploit multiple metrics to cross validate each other and emphasize quality and diversity separately. We notice the GAN training is sensitive to network structure, hyper-parameters, random initialization, and mini-batch feeding. To make a fair comparison, we share the same network structure between different generative models in each specific experiment setting and do multiple random trials. The results support the theorem that SIG can stably cover multi-modes and training GAN-SI adversarially greatly mitigates mode collapsing in GANs.

5.1 Toy examples

We first show the expressive of SIG with both discrete and continuous true data. For the discrete data, SIG is set as $x \sim \text{Pois}(\theta),\ \theta \sim p_{\boldsymbol{\phi}}(\theta)$ where $p_{\boldsymbol{\phi}}(\theta)$ is implicit distribution generated by mapping from ten dimensional random noise with a two-hidden-layer multi-layer perceptron (MLP). The top left figures correspond to $P_d = \text{NegativeBinomial}(r = 2, p = 0.5) = \int \text{Poisson}(\theta)\text{Gamma}(\theta; 2, 1)d\theta$ and bottom left figures correspond to $P_d = \frac{1}{2}\text{Poisson}(10) + \frac{1}{2}\text{NegativeBinomial}(r = 0.2, p = 0.9)$. For continuous data, SIG is set as $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\theta}, 0.1^2 I), \boldsymbol{\theta} \sim p_{\boldsymbol{\phi}}(\boldsymbol{\theta})$, where the $p_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ is the same as that for the discrete cases. As Figure 1 show, the implicit distribution is able to recover the underlying mixing distribution such that the samples following the marginal distribution can well approximate the true data. Vanilla GAN, as comparison, can only generate samples whose similarity to true data is restricted by the discriminator and cannot recover the original data well.

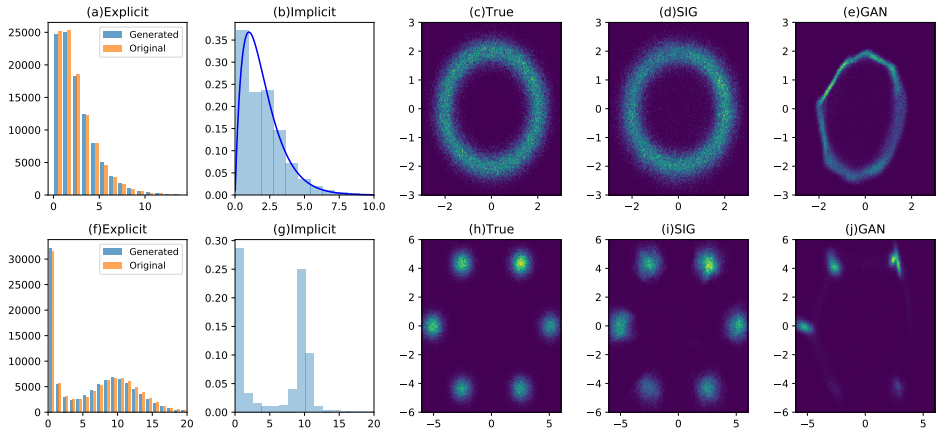


Figure 1: Generated samples from SIG with true data coming from: (a)-(b): Negative-Binomial distribution; the implicit distribution can learn the true mixing distribution $\text{Gamma}(\theta; 2, 1)$; (f)-(g): Mixture of Poisson and Negative-Binomial distribution; (c)-(e): Ring+Gaussian noise; (h)-(j): Gaussian mixture arranged in a ring.

5.2 Mixture of Gaussians

We compare different generative models on a 5×5 Gaussian mixture model. For fair comparison, all the models share the same generative network: a two-hidden-layer MLP with size 100 and rectified linear units (ReLU) activation function. The discriminator for GAN has a fully connected layer with size 100, and the encoder for VAE and VeeGAN is a two-hidden-layer MLP with size 100.

Detecting mode collapsing on a large dataset is challenging but can be accurately measured on synthetic data. To quantitatively evaluate the sample quality, we sample 50,000 points from trained generator and count it as high quality sample if it is within three standard deviation away from any of the mixture component centers. A center that is associated with more than 1000 high quality samples will be counted as a captured mode. The proportion of high quality samples at each mode, together with the proportion of low quality samples, form a 26 dimensional discrete distribution P_g. We calculated the KL divergence $\text{KL}(P_g||P_d)$. All results are reported based on the average and standard error of five independent random trials.

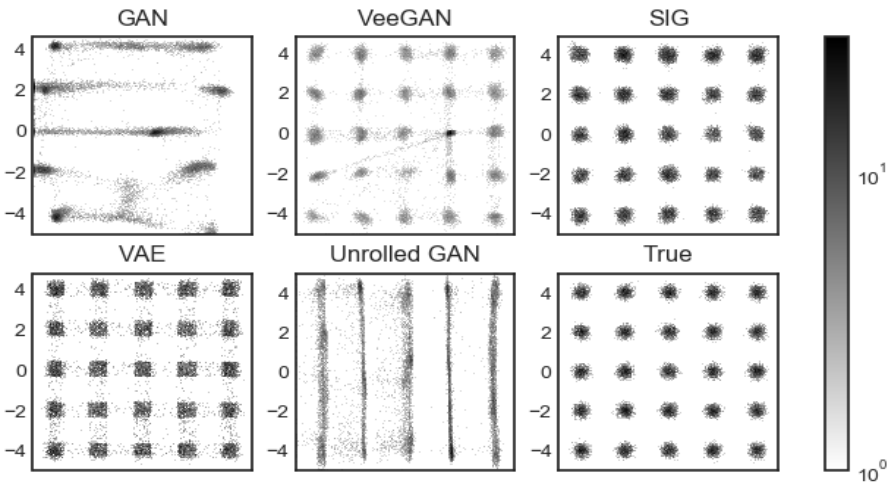


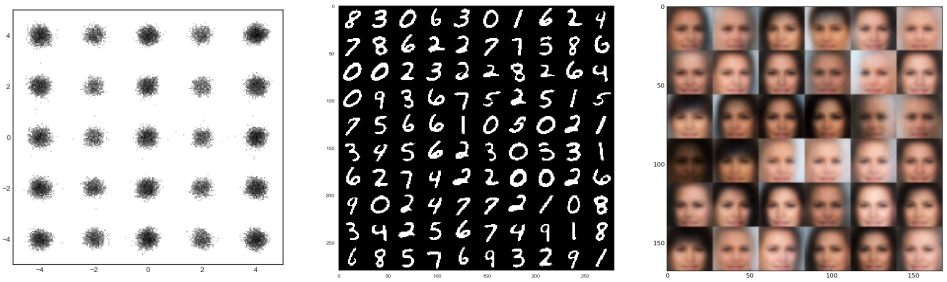
Figure 2: Comparison of generated sample for Gaussian mixture model by vanilla GAN, modified GAN to reduce mode collapsing(Unrolled GAN, VeeGAN), VAE and SIG.

Table 1: Comparison of mode-capturing ability on mixture of Gaussian. 'Modes' is the number of captured modes out of 25. 'KL' is KL($P_{\text{model}}||P_{data}$). For 'Modes' and 'Proportion of high quality samples', the higher the better; for 'KL', the lower the better.

	GAN	VAE	VeeGAN	Unrolled GAN	SIG
Modes	4.0±3.08	25±0	23.2±0.84	6.2±8.6	**25±0**
Proportion of high quality samples	0.36±0.16	0.83±0.02	0.82±0.03	0.42±0.13	**0.91±0.04**
KL	2.87±0.78	0.32±0.07	0.38±0.08	1.97±0.60	**0.14±0.07**

As shown in Table 1, SIG captures all the modes and generates the highest proportion of high quality samples, whose distribution is closest to the ground truth. It also achieves the shortest running time and highest stability using a single neural network.

We notice, however, SIG generalization ability may not scale well with increasing data complexity, as shown in Figure 3. To generate natural images, we train SIG adversarially and notice the proposed GAN-SI can stabilize GAN training and mitigate the mode collapsing problem.



(a) (b) (c)

Figure 3: (a): SIG can generate low complexity data well. The input data is from unevenly distributed GMM, where the components in the 1st, 3rd, and 5th columns has twice more data than the 2nd and 4th. SIG generates samples well aligned with the true distribution. (b): SIG generated MNIST digits. (c): SIG scales not well when the data is as complex as CelebA.

5.3 Stacked MNIST

To measure the performance of combining MLE and adversarial training schemes on discrete multimode data, we stack 3 randomly chosen MNIST images on the RGB color channels to form a

$28 \times 28 \times 3$ image (MNIST-3) [14, 18, 27, 30]. MNIST-3 contains 1000 modes corresponding to 3-digit between 0 and 999. Similar to [14] and [30] we find the missing modes problem of GAN on MNIST3 is sensitive to the network architecture and the randomness in training process due to the instability. Therefore, we choose three different network sizes (denoted as S, M, and L), run each experiment five times and use exactly the same generator and discriminator for DCGAN and DCGAN-SI.

The inception score (IS) [7] is a widely used criterion for GAN evaluation. It is applied to data $\boldsymbol{x}$ with label y using a pre-trained classifier. Low entropy $H(y|x)$ of conditional distribution $p(y \mid \boldsymbol{x})$ and high entropy $H(y)$ of marginal distribution $p(y)$ are considered to represent high image quality and diversity respectively.

$$\text{IS} = \exp(\mathbb{E}[\text{KL}(p(y \mid \boldsymbol{x})||p(y))]) = \exp(H(y) - \mathbb{E}[H(y|x)]) \tag{13}$$

As the IS by itself cannot fully characterize generative model performance [33, 34], we provide more metrics for evaluation: High quality image means the proportion of images that can be classified by the trained classifier with a probability larger than 0.7; Mode is the number of digit triples that have at least one sample; KL is $\text{KL}(P_g||P_d)$ where $P_d = (\frac{1}{1000}, \cdots, \frac{1}{1000})$.

Table 2: High quality image and $\exp(H(y|x))$ reflect sample quality while $\exp(H(y))$, Mode and KL reflect sample diversity. For Inception score, High quality image, $\exp(H(y))$, higher is better; for $\exp(H(y|x))$ and KL, lower is better.

	IS	High quality	$\exp(H(y\|x))$	$\exp(H(y))$	Mode	KL
DCGAN(S)	2.9±0.52	**0.63±0.14**	**1.96±0.32**	5.1±1.19	21.0±8.12	4.99±0.24
DCGAN-SI(S)	**4.33±0.59**	0.6±0.07	2.05±0.2	**8.78±0.41**	**279.2±296.52**	**2.63±1.0**
DCGAN(M)	5.59±0.36	0.7±0.03	1.71±0.09	9.51±0.31	811.8±116.24	0.75±0.35
DCGAN-SI(M)	**5.93±0.47**	**0.72±0.04**	**1.65±0.11**	**9.75±0.11**	**969.0±29.19**	**0.3±0.13**
DCGAN(L)	4.71±1.12	0.67±0.08	1.78±0.17	8.25±1.32	389.8±477.24	2.95±2.33
DCGAN-SI(L)	**6.05±0.68**	**0.73±0.06**	**1.62±0.17**	**9.75±0.12**	**957.0±32.74**	**0.36±0.12**

5.4 Sample quality and diversity on CIFAR10

We test the semi-implicit regularizer on the CIFAR-10 dataset, a widely studied dataset consisting of 50,000 training images with 32×32 pixel from ten categories. The image diversity is high between or within each category. We combine semi-implicit regularizer with two popular GAN frameworks DCGAN [8] and WGAN-GP [35] to balance the quality and diversity of generated samples.

Table 3: Inception scores for models on CIFAR-10

Real data	Unsupervised, standard CNN			
	DCGAN	DCGAN-SI	WGAN-GP	WGAN-GP-SI
11.24 ± .12	6.16 ± .14	6.85± .06	6.43 ± .07	6.67 ± .11

We train each model for 100K iterations with mini-batch size 64. The optimizer is Adam with learning rate 0.0002. The inception model we use is based on pre-trained Inception Model [36] on ImageNet. As shown in Appendix Figure 6, the images generated by DCGAN include duplicated images indicating the existence of mode collapsing which does not seem to happen with regularized DCGAN-SI, and this is reflected in the improvement of inception score as shown in Table 3.

6 Conclusions

We propose semi-implicit generator (SIG) as a flexible and stable generative model. Training under well-understood maximum likelihood framework, SIG is proposed either as a black-box generative model or as unbiased regularizer in adversarial learning. We analyze the inherent mode-capturing mechanism and show its advantage over several state-of-the-art generative methods in reducing mode collapse. Combined with GAN, semi-implicit regularizer provides an interplay between adversarial learning and maximum likelihood inference, leading to a better balance between sample quality and diversity.

References

[1] Diederik P Kingma and Max Welling. Auto-encoding variational Baye. *arXiv preprint arXiv:1312.6114*, 2013.

[2] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[3] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[4] R. M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, pages 71–113, 1992.

[5] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[6] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[7] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, pages 2234–2242, 2016.

[8] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[9] Yi Shang and Benjamin W Wah. Global optimization for neural network training. *Computer*, 29(3):45–54, 1996.

[10] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Global optimality conditions for deep neural networks. *arXiv preprint arXiv:1707.02444*, 2017.

[11] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286, 2014.

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

[13] Ian J Goodfellow. On distinguishability criteria for estimating generative models. *arXiv preprint arXiv:1412.6515*, 2014.

[14] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[15] Jerry Li, Aleksander Madry, John Peebles, and Ludwig Schmidt. Towards understanding the dynamics of generative adversarial networks. *arXiv preprint arXiv:1706.09884*, 2017.

[16] Pengchuan Zhang, Qiang Liu, Dengyong Zhou, Tao Xu, and Xiaodong He. On the discrimination-generalization tradeoff in gans. *arXiv preprint arXiv:1711.02771*, 2017.

[17] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

[18] Akash Srivastava, Lazar Valkoz, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *NIPS*, pages 3310–3320, 2017.

[19] Herbert Robbins et al. An empirical bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1956.

[20] George Casella. An introduction to empirical bayes data analysis. *The American Statistician*, 39(2):83–87, 1985.

[21] Bradley P Carlin and Thomas A Louis. Bayes and empirical Bayes methods for data analysis. *Statistics and Computing*, 7(2):153–154, 1997.

[22] Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. Maximum-likelihood augmented discrete generative adversarial networks. *arXiv preprint arXiv:1702.07983*, 2017.

[23] Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *AAAI*, 2018.

[24] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, pages 271–279, 2016.

[25] Ben Poole, Alexander A Alemi, Jascha Sohl-Dickstein, and Anelia Angelova. Improved generator objectives for gans. *arXiv preprint arXiv:1612.02780*, 2016.

[26] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, pages 2172–2180, 2016.

[27] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*, 2016.

[28] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[29] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[30] Ilya O Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. Adagan: Boosting generative models. In *NIPS*, pages 5430–5439, 2017.

[31] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint arXiv:1703.00573*, 2017.

[32] Sanjeev Arora, Andrej Risteski, and Yi Zhang. Theoretical limitations of encoder-decoder gan architectures. *arXiv preprint arXiv:1711.02651*, 2017.

[33] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[34] Ali Borji. Pros and cons of gan evaluation measures. *arXiv preprint arXiv:1802.03446*, 2018.

[35] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, pages 5769–5779, 2017.

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Semi-implicit generative model: supplementary material

A Proofs

Proof of Lemma 1. Assume integer $K > M > 0$ and $\mathcal{I}$ is the set of all size M subsets of $\{1, \cdots, K\}$. Let I be a discrete uniform random variable that takes outcome $\{i_1, \cdots, i_M\}$ in $\mathcal{I}$ with probability $P(I = \{i_1, \cdots, i_M\}) = \frac{1}{\binom{K}{M}}$. We have $\mathbb{E}_I \frac{1}{M} \sum_{j \in I} p(\boldsymbol{x}|\boldsymbol{\theta}_j) = \frac{1}{K} \sum_{j=1}^K p(\boldsymbol{x}|\boldsymbol{\theta}_j)$. Then we have

$$\begin{aligned}
\mathbb{H}_K &= -\mathbb{E}_{\boldsymbol{x} \sim P_d(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K \sim p_{\boldsymbol{\phi}}(\boldsymbol{\theta})} \log \frac{1}{K} \sum_{j=1}^K p(\boldsymbol{x}|\boldsymbol{\theta}_j) \\
&= -\mathbb{E}_{P_{data}} \mathbb{E}_{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K \sim p_{\boldsymbol{\phi}}(\boldsymbol{\theta})} \log \mathbb{E}_I \frac{1}{M} \sum_{j \in I} p(\boldsymbol{x}|\boldsymbol{\theta}_j) \\
&\leq -\mathbb{E}_{P_{data}} \mathbb{E}_{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K \sim p_{\boldsymbol{\phi}}(\boldsymbol{\theta})} \mathbb{E}_I \log \frac{1}{M} \sum_{j \in I} p(\boldsymbol{x}|\boldsymbol{\theta}_j) \\
&= -\mathbb{E}_{P_{data}} \mathbb{E}_{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M \sim p_{\boldsymbol{\phi}}(\boldsymbol{\theta})} \log \frac{1}{M} \sum_{j=1}^M p(\boldsymbol{x}|\boldsymbol{\theta}_j) = \mathbb{H}_M
\end{aligned}$$

By law of large number $\log \frac{1}{M} \sum_{j=1}^M p(\boldsymbol{x}|\boldsymbol{\theta}_j)$ converges to $\log \int p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$ a.s. as $M \to \infty$ so $\mathbb{H}_M \to \mathbb{H}(P_{data}, P_{\text{model}})$. When $M = 1$, assume $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} -\mathbb{E}_{P_d(\boldsymbol{x})} \log[p(\boldsymbol{x}|\boldsymbol{\theta})]$ then

$$-\mathbb{E}_{P_d(\boldsymbol{x})} \log[p(\boldsymbol{x}|\boldsymbol{\theta})] \geq -\mathbb{E}_{P_d(\boldsymbol{x})} \log[p(\boldsymbol{x}|\boldsymbol{\theta}^*)]$$

Multiply both sides by $p_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ and integrate over $\boldsymbol{\theta}$, we have

$$-\mathbb{E}_{\boldsymbol{\theta} \sim p_{\boldsymbol{\phi}}(\boldsymbol{\theta})} \mathbb{E}_{p_d(\boldsymbol{x})} \log[p(\boldsymbol{x}|\boldsymbol{\theta})] \geq -\mathbb{E}_{p_d(\boldsymbol{x})} \log[p(\boldsymbol{x}|\boldsymbol{\theta}^*)]$$

The minimal is reached when implicit distribution degenerates to the point probability mass $p_{\boldsymbol{\phi}}(\boldsymbol{\theta}) = \delta_{\boldsymbol{\theta}^*}(\boldsymbol{\theta})$ where θ^* maximizes average log-likelihood over data. □

Proof of Theorem 1. Suppose P_{data} is defined on a discrete multi-modal space $\mathcal{X} = \bigcup_{i=1}^K U_i$. For $\boldsymbol{x}_i \sim P_{data}, i = 1, \cdots, N$, assume $\boldsymbol{x}_i \in U_{t_i}$; for $\boldsymbol{\theta}_j \sim P_{\boldsymbol{\phi}}(\boldsymbol{\theta}), j = 1, \cdots, M$, assume $\boldsymbol{\theta}_j \in U_{z_j}$, where t_i, z_j denote the mode label of true data and generated data center respectively. Then we have $n_k = \sum_{i=1}^N \mathbb{1}\{t_i = k\}$ and $m_k = \sum_{j=1}^M \mathbb{1}\{z_j = k\}$ for $k = 1, \cdots, K$.

$$\min_{\{m_1, \cdots, m_k\}} -\frac{1}{N} \sum_{i=1}^N \log \frac{1}{M} \sum_{j=1}^M \mathbb{E}[p(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}_j)] \Leftrightarrow \min_{\{m_1, \cdots, m_k\}} -\sum_{i=1}^N \log \sum_{j=1}^M \mathbb{E} \exp^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{\theta}_j\|^2}{2\sigma^2}} \tag{14}$$

Notice $\sum_{k=1}^K n_k = N$, $\sum_{k=1}^K m_k = M$. By definition of $\mathcal{X}$, if $\boldsymbol{x}, \boldsymbol{\theta} \in U_k$, $r(\boldsymbol{x}, \boldsymbol{\theta}) = \exp^{-\frac{\|\boldsymbol{x} - \boldsymbol{\theta}\|^2}{2\sigma^2}} \geq \exp^{-\frac{\epsilon_0^2}{2\sigma^2}}$ and if $\boldsymbol{x} \in U_i, \boldsymbol{\theta} \in U_j$ $i \neq j$, $r(\boldsymbol{x}, \boldsymbol{\theta}) = \exp^{-\frac{\|\boldsymbol{x} - \boldsymbol{\theta}\|^2}{2\sigma^2}} \leq \exp^{-\frac{c_0^2}{2\sigma^2}}$. With the definition of u and v in theorem 1, we have $u \geq \exp -\frac{\epsilon_0^2}{2\sigma^2} > -\frac{c_0^2}{2\sigma^2} \geq v$. Then we have objective (14) as a constrained optimization problem with Lagrange multiplier β

$$\begin{aligned}
\min_{\{m_1, \cdots, m_k\}} & -\sum_{i=1}^N \log \sum_{j=1}^M \mathbb{E} r(\boldsymbol{x}_i, \boldsymbol{\theta}_j) + \beta(\sum_{k=1}^K m_k - M) \\
&= -\sum_{i=1}^N \log(m_{t_i} u + (M - m_{t_i}) v) + \beta(\sum_{k=1}^K m_k - M) \\
&= -\sum_{k=1}^K n_k \log(m_k + Mv - m_k v) + \beta(\sum_{k=1}^K m_k - M)
\end{aligned}$$

Taking the gradient with respect to $(m_1, \cdots, m_k)$ and set to zero gives

$$\begin{aligned}&\frac{\partial}{\partial m_k} - \sum_{k=1}^{K} n_k \log(m_k + Mv - m_k v) + \beta(\sum_{k=1}^{K} m_k - M) \\ =&\frac{-n_k(u-v)}{m_k(u-v)+Mv} + \beta = 0, \quad \text{for } k = 1, \cdots, K\end{aligned}$$

Together with constraint $\sum_{k=1}^{K} m_k = M$, we have

$$\frac{m_k^*}{M} = \frac{n_k}{N} + (\frac{n_k}{N} - \frac{1}{K})\frac{Kv}{(u-v)} \tag{15}$$

The Hessian $H = \text{diag}(\frac{n_k(u-v)^2}{(m_k(u-v)+M)^2}) \succ 0$ shows convexity and $\frac{m_k^*}{M}$ is global minimum. Let the right-hand-side greater than 0, the condition for mode k not vanishing is $n_k > \frac{N}{K}\frac{1}{1+\frac{u-v}{Kv}}$ □

Proof of Corollary 1. Assume $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \sigma^2 I)$, $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}_\theta, \sigma^2 I)$. Let $\boldsymbol{z} = \frac{\boldsymbol{x}-\boldsymbol{\theta}}{\sqrt{2}\sigma}$ and $\boldsymbol{\mu} = \boldsymbol{\mu}_x - \boldsymbol{\mu}_\theta$ then $\boldsymbol{z} \sim \mathcal{N}(\frac{\boldsymbol{\mu}}{\sqrt{2}\sigma}, I)$. Let $\chi = \boldsymbol{z}^T\boldsymbol{z}$, then χ follows noncentral chi-squared distribution $\chi \sim \chi(N, \lambda)$ where N is the dimension of $\boldsymbol{z}$, $\lambda = \frac{\boldsymbol{\mu}^T\boldsymbol{\mu}}{2\sigma^2}$ is noncentrality parameter. By moment genrating function (MGF) of noncentral chi-squared distribution, we have

$$\begin{aligned}&\mathbb{E}e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\theta}\|^2}{2\sigma^2}} = \mathbb{E}e^{-\boldsymbol{z}^T\boldsymbol{z}} \\ =&\mathbb{E}e^{-\chi} = MGF_\chi(-1) \\ =&3^{-N/2}e^{-\lambda/3}\end{aligned} \tag{16}$$

For u, $\boldsymbol{\mu} = 0$, $\lambda = 0$ and for v, $\|\boldsymbol{\mu}\| = c\sigma$, $\lambda = \frac{c^2}{2}$. Plugging into (16), we have $u = 3^{-N/2}$, $v = 3^{-N/2}e^{-c^2/6}$, therefore $\frac{v}{u-v} = \frac{1}{e^{c^2/6}-1}$. □

B Algorithm for GAN-SI

Algorithm 1: Mini-batch training of GAN-SI

while *not converged* **do**
  Sample $\boldsymbol{x}_i \sim P_{data}(\boldsymbol{x})$ for $i \in \{1, \cdots, N\}$;
  Sample $\boldsymbol{z}_j \sim g(\boldsymbol{z})$ for $j \in \{1, \cdots, M\}$;
  Set $\boldsymbol{\theta}_j = T_{\boldsymbol{\phi}}(\boldsymbol{z}_j)$ for all j ;
  $g_{\boldsymbol{\gamma}} \leftarrow -\nabla_{\boldsymbol{\gamma}}\big[\frac{1}{N}\sum_{i=1}^{N}\log D_{\boldsymbol{\gamma}}(\boldsymbol{x}_i) + \frac{1}{M}\sum_{j=1}^{M}\log(1 - D_{\boldsymbol{\gamma}}(\boldsymbol{\theta}_j))\big]$;
  $g_{\boldsymbol{\phi}} \leftarrow -\nabla_{\boldsymbol{\phi}}\big[\frac{1}{M}\sum_{j=1}^{M}\log D_{\boldsymbol{\gamma}}(\boldsymbol{\theta}_j) - \lambda\frac{1}{N}\sum_{i=1}^{N}\log\frac{1}{M}\sum_{j=1}^{M}p(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}_j)\big]$;
  $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} - \eta g_{\boldsymbol{\gamma}}, \quad \boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \eta g_{\boldsymbol{\phi}}$
end
The first order optimization is used as Adam [37] in our experiments.

C Network architecture and samples for MNIST-3

The generator network is defined with parameter K_g to adjust network size

	Number of output	Kernel size	Stride	Padding
Input $z \sim N(0, I_{256})$	-			
Fully connected	4*4*64	-		
Transpose Convolution	64*K_g	4	1	VALID
Transpose Convolution	32*K_g	4	2	SAME
Transpose Convolution	8*K_g	4	1	SAME
Convolution	3	4	2	SAME

The discriminator network is defined with parameter K_d to adjust network size

	Number of output	Kernel size	Stride	Padding
Input is image batch with size 28*28*3	-			
Transpose Convolution	8*K_d	4	2	VALID
Transpose Convolution	16*K_d	4	2	SAME
Transpose Convolution	32*K_d	4	1	SAME
Flat+Fully connected	1	-		

For the network work size denoted as (S), (M), (L), the (K_g,K_d) pair is chosen as $(1, 0.5)$, $(1, 1)$, $(2, 1)$ respectively.

D Additional figures

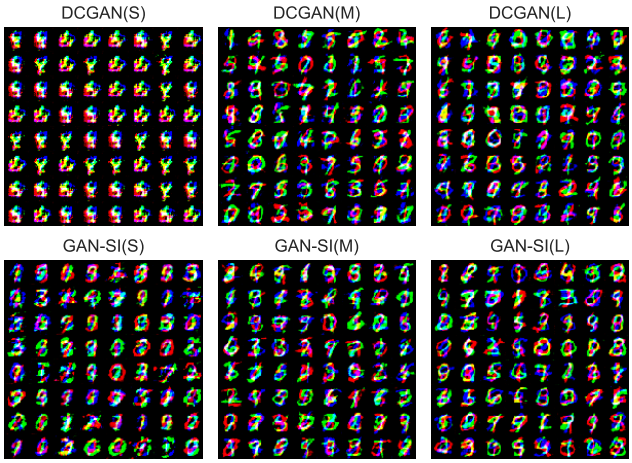


Figure 4: MNIST-3, highest inception score cases among 10 independent trials

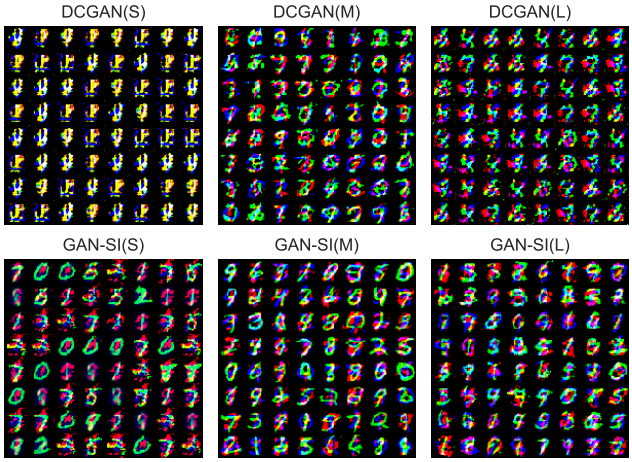
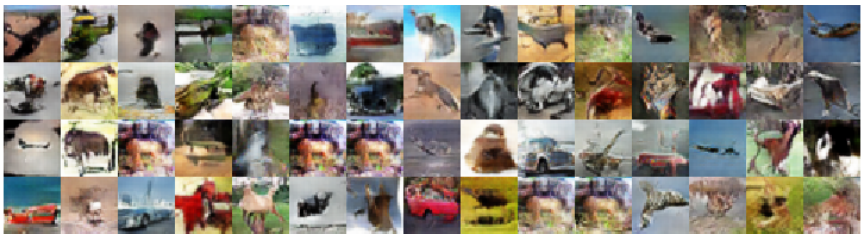
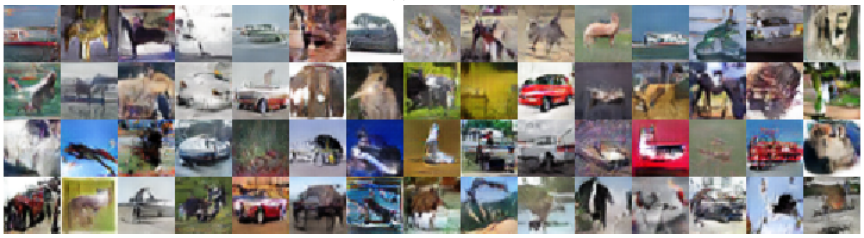


Figure 5: MNIST-3, lowest inception score cases among 10 independent trials



(a)DCGAN



(b)DCGAN-SI

Figure 6: Randomly generated images by DCGAN and DCGAN with semi-implicit regularizer.