Calibrating Energy-based Generative Adversarial Networks

Zihang Dai[1], **Amjad Almahairi**[2]*, **Philip Bachman**[3], **Eduard Hovy**[1] **& Aaron Courville**[2]
[1] Language Technologies Institute, Carnegie Mellon University.
[2] MILA, Université de Montréal.
[3] Maluuba Research.

Abstract

In this paper we propose equipping Generative Adversarial Networks with the ability to produce direct energy estimates for samples. Specifically, we develop a flexible adversarial training framework, and prove this framework not only ensures the generator converges to the true data distribution, but also enables the discriminator to retain the density information at the global optimum. We derive the analytic form of the induced solution, and analyze its properties. In order to make the proposed framework trainable in practice, we introduce two effective approximation techniques. Empirically, the experiment results closely match our theoretical analysis, verifying that the discriminator is able to recover the energy of data distribution.

1 Introduction

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) represent an important milestone on the path towards more effective generative models. GANs cast generative model training as a minimax game between a generative network (*generator*), which maps a random vector into the data space, and a discriminative network (*discriminator*), whose objective is to distinguish generated samples from real samples. Multiple researchers Radford et al. (2015); Salimans et al. (2016); Zhao et al. (2016) have shown that the adversarial interaction with the discriminator can result in a generator that produces compelling samples. The empirical successes of the GAN framework were also supported by the theoretical analysis of Goodfellow et al., who showed that, under certain conditions, the distribution produced by the generator converges to the true data distribution, while the discriminator converges to a degenerate uniform solution.

While GANs have excelled as compelling sample generators, their use as general purpose probabilistic generative models has been limited by the difficulty in using them to provide density estimates or even unnormalized energy values for sample evaluation.

It is tempting to consider the GAN discriminator as a candidate for providing this sort of scoring function. Conceptually, it is a trainable sample evaluation mechanism that – owing to GAN training paradigm – could be closely calibrated to the distribution modeled by the generator. If the discriminator could retain fine-grained information of the relative quality of samples, measured for instance by probability density or unnormalized energy, it could be used as an evaluation metric. Such data-driven evaluators would be highly desirable for problems where it is difficult to define evaluation criteria that correlate well with human judgment. Indeed, the real-valued discriminator of the recently introduced energy-based GANs Zhao et al. (2016) might seem like an ideal candidate energy function. Unfortunately, as we will show, the degenerate fate of the GAN discriminator at the optimum equally afflicts the energy-based GAN of Zhao et al..

In this paper we consider the questions: (i) does there exists an adversarial framework that induces a non-degenerate discriminator, and (ii) if so, what form will the resulting discriminator take? We introduce a novel adversarial learning formulation, which leads to a non-degenerate discriminator while ensuring the generator distribution matches the data distribution at the global optimum. We derive a general analytic form of the optimal discriminator, and discuss its properties and their

[*] Part of this work was completed while author was at Maluuba Research

relationship to the specific form of the training objective. We also discuss the connection between the proposed formulation and existing alternatives such as the approach of Kim & Bengio (2016). Finally, for a specific instantiation of the general formulation, we investigate two approximation techniques to optimize the training objective, and verify our results empirically.

2 Related Work

Following a similar motivation, the field of Inverse Reinforcement Learning (IRL) (Ng & Russell, 2000) has been exploring ways to recover the "intrinsic" reward function (analogous to the discriminator) from observed expert trajectories (real samples). Taking this idea one step further, apprenticeship learning or imitation learning (Abbeel & Ng, 2004; Ziebart et al., 2008) aims at learning a policy (analogous to the generator) using the reward signals recovered by IRL. Notably, Ho & Ermon draw a connection between imitation learning and GAN by showing that the GAN formulation can be derived by imposing a specific regularization on the reward function. Also, under a special case of their formulation, Ho & Ermon provide a duality-based interpretation of the problem, which inspires our theoretical analysis. However, as the focus of (Ho & Ermon, 2016) is only on the policy, the authors explicitly propose to bypass the intermediate IRL step, and thus provide no analysis of the learned reward function.

The GAN models most closely related to our proposed framework are energy-based GAN models of Zhao et al. (2016) and Kim & Bengio (2016). In the next section, We show how one can derive both of these approaches from different assumptions regarding regularization of the generative model.

3 Alternative Formulation of Adversarial Training

3.1 Background

Before presenting the proposed formulation, we first state some basic assumptions required by the analysis, and introduce notations used throughout the paper.

Following the original work on GANs (Goodfellow et al., 2014), our analysis focuses on the non-parametric case, where all models are assumed to have infinite capacities. While many of the non-parametric intuitions can directly transfer to the parametric case, we will point out cases where this transfer fails. We assume a finite data space throughout the analysis, to avoid technical machinery out of the scope of this paper. Our results, however, can be extended to continuous data spaces, and our experiments are indeed performed on continuous data.

Let $\mathcal{X}$ be the data space under consideration, and $\mathcal{P} = \{p \mid p(x) \geq 0, \forall x \in \mathcal{X}, \sum_{x \in \mathcal{X}} p(x) = 1\}$ be the set of all proper distributions defined on $\mathcal{X}$. Then, $p_{\text{data}} \in \mathcal{P} : \mathcal{X} \mapsto \mathbb{R}$ and $p_{\text{gen}} \in \mathcal{P} : \mathcal{X} \mapsto \mathbb{R}$ will denote the true data distribution and the generator distribution. $\mathbb{E}_{x \sim p} f(x)$ denotes the expectation of the quantity $f(x)$ w.r.t. x drawn from p. Finally, the term "discriminator" will refer to any structure that provides training signals to the generator based on some measure of difference between the generator distribution and the real data distribution, which which includes but is not limited to f-divergence.

3.2 Proposed Formulation

In order to understand the motivation of the proposed approach, it is helpful to analyze the optimization dynamics near convergence in GANs first.

When the generator distribution matches the data distribution, the training signal (gradient) w.r.t. the discriminator vanishes. At this point, assume the discriminator still retains density information, and views some samples as more real and others as less. This discriminator will produce a training signal (gradient) w.r.t. the generator, pushing the generator to generate samples that appear more real to the discriminator. Critically, this training signal is the sole driver of the generator's training. Hence, the generator distribution will diverge from the data distribution. In other words, as long as the discriminator retains relative density information, the generator distribution cannot stably match the data distribution. Thus, in order to keep the generator stationary as the data distribution, the discriminator must assign flat (exactly the same) density to all samples at the optimal.

From the analysis above, the fundamental difficulty is that the generator only receives a single training signal (gradient) from the discriminator, which it has to follow. To keep the generator stationary, this single training signal (gradient) must vanish, which requires a degenerate discriminator. In this work, we propose to tackle this single training signal constraint directly. Specifically, we introduce a novel adversarial learning formulation which incorporates an additional training signal to the generator, such that this additional signal can

- balance (cancel out) the discriminator signal at the optimum, so that the generator can stay stationary even if the discriminator assigns non-flat density to samples
- cooperate with the discriminator signal to make sure the generator converges to the data distribution, and the discriminator retains the *correct* relative density information

The proposed formulation can be written as the following minimax training objective,

$$\max_{c} \min_{p_{\text{gen}} \in \mathcal{P}} \mathop{\mathbb{E}}_{x \sim p_{\text{gen}}} \big[c(x)\big] - \mathop{\mathbb{E}}_{x \sim p_{\text{data}}} \big[c(x)\big] + K(p_{\text{gen}}), \tag{1}$$

where $c(x) : \mathcal{X} \mapsto \mathbb{R}$ is the discriminator that assigns each data point an unbounded scalar cost, and $K(p_{\text{gen}}) : \mathcal{P} \mapsto \mathbb{R}$ is some (functionally) differentiable, convex function of p_{gen}. Compared to the original GAN, despite the similar minimax surface form, the proposed fomulation has two crucial distinctions.

Firstly, while the GAN discriminator tries to distinguish "fake" samples from real ones using binary classification, the proposed discriminator achieves that by assigning lower cost to real samples and higher cost to "fake" one. This distinction can be seen from the first two terms of Eqn. (1), where the discriminator $c(x)$ is trained to widen the expected cost gap between "fake" and real samples, while the generator is adversarially trained to minimize it. In addition to the different adversarial mechanism, a calibrating term $K(p_{\text{gen}})$ is introduced to provide a countervailing source of training signal for p_{gen} as we motivated above. For now, the form of $K(p_{\text{gen}})$ has not been specified. But as we will see later, its choice will directly decide the form of the optimal discriminator $c^*(x)$.

With the specific optimization objective, we next provide theoretical characterization of both the generator and the discriminator at the global optimum.

Define $L(p_{\text{gen}}, c) = \mathbb{E}_{x \sim p_{\text{gen}}} \big[c(x)\big] - \mathbb{E}_{x \sim p_{\text{data}}} \big[c(x)\big] + K(p_{\text{gen}})$, then $L(p_{\text{gen}}, c)$ is the Lagrange dual function of the following optimization problem

$$\begin{aligned} \min_{p_{\text{gen}} \in \mathcal{P}} \quad & K(p_{\text{gen}}) \\ \text{s.t.} \quad & p_{\text{gen}}(x) - p_{\text{data}}(x) = 0, \forall x \in \mathcal{X} \end{aligned} \tag{2}$$

where $c(x), \forall x$ appears in $L(p_{\text{gen}}, c)$ as the dual variables introduced for the equality constraints. This duality relationship has been observed previously in (Ho & Ermon, 2016, equation (7)) under the adversarial imitation learning setting. However, in their case, the focus was fully on the generator side (induced policy), and no analysis was provided for the discriminator (reward function).

In order to characterize c^*, we first expand the set constraint on p_{gen} into explicit equality and inequality constraints:

$$\begin{aligned} \min_{p_{\text{gen}}} \quad & K(p_{\text{gen}}) \\ \text{s.t.} \quad & p_{\text{gen}}(x) - p_{\text{data}}(x) = 0, \forall x \\ & -p_{\text{gen}}(x) \leq 0, \forall x \\ & \sum_{x \in \mathcal{X}} p_{\text{gen}}(x) - 1 = 0. \end{aligned} \tag{3}$$

Notice that $K(p_{\text{gen}})$ is a convex function of $p_{\text{gen}}(x)$ by definition, and both the equality and inequality constraints are affine functions of $p_{\text{gen}}(x)$. Thus, problem (2) is a convex optimization problem. What's more, since (i) dom_K is open, and (ii) there exists a feasible solution $p_{\text{gen}} = p_{\text{data}}$ to (3), by the refined Slater's condition (Boyd & Vandenberghe, 2004, page 226), we can further verify that strong duality holds for (3). With strong duality, a typical approach to characterizing the optimal solution is to apply the Karush-Kuhn-Tucker (KKT) conditions, which gives rise to this theorem:

Proposition 3.1. *By the KKT conditions of the convex problem (3), at the global optimum, the optimal generator distribution p^*_{gen} matches the true data distribution p_{data}, and the optimal discriminator $c^*(x)$ has the following form:*

$$\begin{aligned} c^*(x) &= -\frac{\partial K(p_{gen})}{\partial p_{gen}(x)}\bigg|_{p_{gen}=p_{data}} - \lambda^* + \mu^*(x), \forall x \in \mathcal{X}, \\ \textit{where} \quad & \mu^*(x) = \begin{cases} 0, & p_{data}(x) > 0 \\ u_x, & p_{data}(x) = 0 \end{cases}, \\ & \lambda^* \in \mathbb{R}, \textit{ is an under-determined real number independent of } x, \\ & u_x \in \mathbb{R}_+, \textit{ is an under-determined non-negative real number.} \end{aligned} \tag{4}$$

The detailed proof of proposition 3.1 is provided in appendix A.1. From (4), we can see the exact form of the optimal discriminator depends on the term $K(p_{\text{gen}})$, or more specifically its gradient. But, before we instantiate $K(p_{\text{gen}})$ with specific choices and show the corresponding forms of $c^*(x)$, we first discuss some general properties of $c^*(x)$ that do not depend on the choice of K.

Weak Support Discriminator. As part of the optimal discriminator function, the term $\mu^*(x)$ plays the role of support discriminator. That is, it tries to distinguish the support of the data distribution, i.e. $\text{SUPP}(p_{\text{data}}) = \{x \in \mathcal{X} \mid p_{\text{data}}(x) > 0\}$, from its complement set with zero-probability, i.e. $\text{SUPP}(p_{\text{data}})^{\complement} = \{x \in \mathcal{X} \mid p_{\text{data}}(x) = 0\}$. Specifically, for any $x \in \text{SUPP}(p_{\text{data}})$ and $x' \in \text{SUPP}(p_{\text{data}})^{\complement}$, it is guaranteed that $\mu^*(x) \leq \mu^*(x')$. However, because $\mu^*(\cdot)$ is under-determined, there is nothing preventing the inequality from degenerating into an equality. Therefore, we name it the *weak* support discriminator. But, in all cases, $\mu^*(\cdot)$ assigns zero cost to all data points within the support. As a result, it does not possess any fine-grained density information inside of the data support. It is worth pointing out that, in the parametric case, because of the smoothness and the generalization properties of the parametric model, the learned discriminator may generalize beyond the data support.

Global Bias. In (4), the term λ^* is a scalar value shared for all x. As a result, it does not affect the relative cost among data points, and only serves as a global bias for the discriminator function.

Having discussed general properties, we now consider some specific cases of the convex function K, and analyze the resulting optimal discriminator $c^*(x)$ in detail.

1. First, let us consider the case where K is the negative entropy of the generator distribution, i.e. $K(p_{\text{gen}}) = -H(p_{\text{gen}})$. Taking the derivative of the negative entropy w.r.t. $p_{\text{gen}}(x)$, we have

   $$c^*_{\text{ent}}(x) = -\log p_{\text{data}}(x) - 1 - \lambda^* + \mu^*(x), \forall x \in \mathcal{X}, \tag{5}$$

   where $\mu^*(x)$ and λ^* have the same definitions as in (4).

   Up to a constant, this form of $c^*_{\text{ent}}(x)$ is exactly the energy function of the data distribution $p_{\text{data}}(x)$. This elegant result has deep connections to several existing formulations, which include max-entropy imitation learning (Ziebart et al., 2008) and the directed-generator-trained energy-based model (Kim & Bengio, 2016). The core difference is that these previous formulations are originally derived from maximum-likelihood estimation, and thus the minimax optimization is only implicit. In contrast, with an explicit minimax formulation we can develop a better understanding of the induced solution. For example, the global bias λ^* suggests that there exists more than one stable equilibrium the optimal discriminator can actually reach. Further, $\mu^*(x)$ can be understood as a support discriminator that poses extra cost on generator samples which fall in zero-probability regions of data space.

2. When $K(p_{\text{gen}}) = \frac{1}{2}\sum_{x\in\mathcal{X}} p_{\text{gen}}(x)^2 = \frac{1}{2}\|p_{\text{gen}}\|_2^2$, which can be understood as posing ℓ_2 regularization on p_{gen}, we have $\frac{\partial K(p_{\text{gen}})}{\partial p_{\text{gen}}(x)}\big|_{p_{\text{gen}}=p_{\text{data}}} = p_{\text{data}}(x)$, and it follows

   $$c^*_{\ell_2}(x) = -p_{\text{data}}(x) - \lambda^* + \mu^*(x), \forall x \in \mathcal{X}, \tag{6}$$

   with $\mu^*(x), \lambda^*$ similarly defined as in (4).

   Surprisingly, the result suggests that the optimal discriminator $c^*_{\ell_2}(x)$ directly recovers the negative probability $-p_{\text{data}}(x)$, shifted by a constant. Thus, similar to the entropy solution (5), it fully retains the relative density information of data points within the support.

However, because of the under-determined term $\mu^*(x)$, we cannot recover the distribution density p_{data} exactly from either $c^*_{\ell_2}$ or c^*_{ent} if the data support is finite. Whether this ambiguity can be resolved is beyond the scope of this paper, but poses an interesting research problem.

3. Finally, let's consider consider a degenerate case, where $K(p_{\text{gen}})$ is a constant. That is, we dont provide any additional training signal for pgen at all. With $K(p_{\text{gen}}) = \text{const}$, we simply have

$$c^*_{\text{cst}}(x) = \lambda^* + \mu^*(x), \forall x \in \mathcal{X}, \tag{7}$$

whose discriminative power is fully controlled by the weak support discriminator $\mu^*(x)$. Thus, it follows that $c^*_{\text{cst}}(x)$ won't be able to discriminate data points within the support of p_{data}, and its power to distinguish data from $\text{SUPP}(p_{\text{data}})$ and $\text{SUPP}(p_{\text{data}})^{\complement}$ is weak. This closely matches the intuitive argument in the beginning of this section.

Note that when $K(p_{\text{gen}})$ is a constant, the objective function (1) simplifies to:

$$\max_{c} \min_{p_{\text{gen}} \in \mathcal{P}} \mathop{\mathbb{E}}_{x \sim p_{\text{gen}}} \big[c(x)\big] - \mathop{\mathbb{E}}_{x \sim p_{\text{data}}} \big[c(x)\big], \tag{8}$$

which is very similar to the EBGAN objective (Zhao et al., 2016, equation (2) and (4)). As we show in appendix A.2, compared to the objective in (8), the EBGAN objective puts extra constraints on the allowed discriminator function. In spite of that, the EBGAN objective suffers from the single-training-signal problem and does not guarantee that the discriminator will recover the real energy function (see appendix A.2 for detailed analysis).

As we finish the theoretical analysis of the proposed formulation, we want to point out that simply adding the same term $K(p_{\text{gen}})$ to the original GAN formulation will not lead to both a generator that matches the data distribution, and a discriminator that retains the density information (see appendix A.3 for detailed analysis).

4 Parametric Instantiation with Entropy Approximation

While the discussion in previous sections focused on the non-parametric case, in practice we are limited to a finite amount of data, and the actual problem involves high dimensional continuous spaces. Thus, we resort to parametric representations for both the generator and the discriminator. In order to train the generator using standard back-propagation, we do not parametrize the generator distribution directly. Instead, we parametrize a directed generator network that transforms random noise $z \sim p_z(z)$ to samples from a continuous data space $\mathbb{R}^n$. Consequently, we don't have analytical access to the generator distribution, which is defined implicitly by the generator network's noise→data mapping. However, the regularization term $K(p_{\text{gen}})$ in the training objective (1) requires the generator distribution. Faced with this problem, we focus on the max-entropy formulation, and exploit two different approximations of the regularization term $K(p_{\text{gen}}) = -H(p_{\text{gen}})$.

4.1 Nearest-Neighbor Entropy Gradient Approximation

The first proposed solution is built upon an intuitive interpretation of the entropy gradient. Firstly, since we construct p_{gen} by applying a deterministic, differentiable transform g_θ to samples z from a fixed distribution p_z, we can write the gradient of $H(p_{\text{gen}})$ with respect to the generator parameters θ as follows:

$$-\nabla_\theta H(p_{\text{gen}}) = \mathbb{E}_{z \sim p_z}\left[\nabla_\theta \log p_{\text{gen}}(g_\theta(z))\right] = \mathbb{E}_{z \sim p_z}\left[\frac{\partial g_\theta(z)}{\partial \theta} \frac{\partial \log p_{\text{gen}}(g_\theta(z))}{\partial g_\theta(z)}\right], \tag{9}$$

where the first equality relies on the "reparametrization trick". Equation 9 implies that, if we can compute the gradient of the generator log-density $\log p_{\text{gen}}(x)$ w.r.t. any $x = g_\theta(z)$, then we can directly construct the Monte-Carlo estimation of the entropy gradient $\nabla_\theta H(p_{\text{gen}})$ using samples from the generator.

Intuitively, for any generated data $x = g_\theta(z)$, the term $\frac{\partial \log p_{\text{gen}}(x)}{\partial x}$ essentially describes the direction of *local change* in the sample space that will increase the log-density. Motivated by this intuition, we propose to form a local Gaussian approximation p^i_{gen} of p_{gen} around each point x_i in a batch of samples $\{x_1, ..., x_n\}$ from the generator, and then compute the gradient $\frac{\partial \log p_{\text{gen}}(x_i)}{\partial x_i}$ based on the

Gaussian approximation. Specifically, each local Gaussian approximation p_{gen}^{i} is formed by finding the k nearest neighbors of x_i in the batch $\{x_1, ..., x_n\}$, and then placing an isotropic Gaussian distribution at their mean (i.e. maximimum likelihood). Based on the isotropic Gaussian approximation, the resulting gradient has the following form

$$\frac{\partial \log p_{\text{gen}}(x_i)}{\partial x_i} \approx \mu_i - x_i, \quad \text{where } \mu_i = \frac{1}{k} \sum_{x' \in \text{KNN}(x_i)} x' \text{ is the mean of the Gaussian} \tag{10}$$

Finally, note the scale of this gradient approximation may not be reliable. To fix this problem, we normalize the approximated gradient into unit norm, and use a single hyper-parameter to model the scale for all x, leading to the following entropy gradient approximation

$$-\nabla_\theta H(p_{\text{gen}}) \approx \alpha \frac{1}{k} \sum_{x_i = g_\theta(z_i)} \frac{\mu_i - x_i}{\|\mu_i - x_i\|_2} \tag{11}$$

where α is the hyper-parameter and μ_i is defined as in equation (10).

An obvious weakness of this approximation is that it relies on Euclidean distance to find the k nearest neighbors. However, Euclidean distance is usually not the proper metric to use when the effective dimension is very high. As the problem is highly challenging, we leave it for future work.

4.2 Variational Lower bound on the Entropy

Another approach we consider relies on defining and maximizing a variational lower bound on the entropy $H(p_{\text{gen}}(x))$ of the generator distribution. We can define the joint distribution over observed data and the noise variables as $p_{\text{gen}}(x, z) = p_{\text{gen}}(x \mid z)p_{\text{gen}}(z)$, where simply $p_{\text{gen}}(z) = p_z(z)$ is a fixed prior. Using the joint, we can also define the marginal $p_{\text{gen}}(x)$ and the posterior $p_{\text{gen}}(z \mid x)$. We can also write the mutual information between the observed data and noise variables as:

$$\begin{aligned} I(p_{\text{gen}}(x); p_{\text{gen}}(z)) &= H(p_{\text{gen}}(x)) - H(p_{\text{gen}}(x \mid z)) \\ &= H(p_{\text{gen}}(z)) - H(p_{\text{gen}}(z \mid x)), \end{aligned} \tag{12}$$

where $H(p_{\text{gen}}(. \mid .))$ denotes the conditional entropy. By reorganizing terms in this definition, we can write the entropy $H(p_{\text{gen}}(x))$ as:

$$H(p_{\text{gen}}(x)) = H(p_{\text{gen}}(z)) - H(p_{\text{gen}}(z \mid x)) + H(p_{\text{gen}}(x \mid z)) \tag{13}$$

We can think of $p_{\text{gen}}(x \mid z)$ as a peaked Gaussian with a fixed, diagonal covariance, and hence its conditional entropy is constant and can be dropped. Furthermore, $H(p_{\text{gen}}(z))$ is also assumed to be fixed a priori. Hence, we can maximize $H(p_{\text{gen}}(x))$ by minimizing the conditional entropy:

$$H(p_{\text{gen}}(z \mid x)) = \mathbb{E}_{x \sim p_{\text{gen}}(x)} \left[\mathbb{E}_{z \sim p_{\text{gen}}(z|x)} \left[-\log p_{\text{gen}}(z \mid x) \right] \right] \tag{14}$$

Optimizing this term is still problematic, because (i) we do not have access to the posterior $p_{\text{gen}}(z \mid x)$, and (ii) we cannot sample from it. Therefore, we instead minimize a variational upper bound defined by an approximate posterior $q_{\text{gen}}(z \mid x)$:

$$\begin{aligned} H(p_{\text{gen}}(z \mid x)) &= \mathbb{E}_{x \sim p_{\text{gen}}(x)} \left[\mathbb{E}_{z \sim p_{\text{gen}}(z|x)} \left[-\log q_{\text{gen}}(z \mid x) \right] - \text{KL}(p_{\text{gen}}(z \mid x) \| q_{\text{gen}}(z \mid x)) \right] \\ &\le \mathbb{E}_{x \sim p_{\text{gen}}(x)} \left[\mathbb{E}_{z \sim p_{\text{gen}}(z|x)} \left[-\log q_{\text{gen}}(z \mid x) \right] \right] \\ &= \mathcal{U}(q_{\text{gen}}). \end{aligned} \tag{15}$$

We can also rewrite the variational upper bound as:

$$\mathcal{U}(q_{\text{gen}}) = \mathbb{E}_{x,z \sim p_{\text{gen}}(x,z)} \left[-\log q_{\text{gen}}(z \mid x) \right] = \mathbb{E}_{z \sim p_{\text{gen}}(z)} \left[\mathbb{E}_{x \sim p_{\text{gen}}(x|z)} \left[-\log q_{\text{gen}}(z \mid x) \right] \right], \tag{16}$$

which can be optimized efficiently with standard back-propagation and Monte Carlo integration of the relevant expectations based on independent samples drawn from the joint $p_{\text{gen}}(x, z)$. By minimizing this upper bound on the conditional entropy $H(p_{\text{gen}}(z \mid x))$, we are effectively maximizing a variational lower bound on the entropy $H(p_{\text{gen}}(x))$.

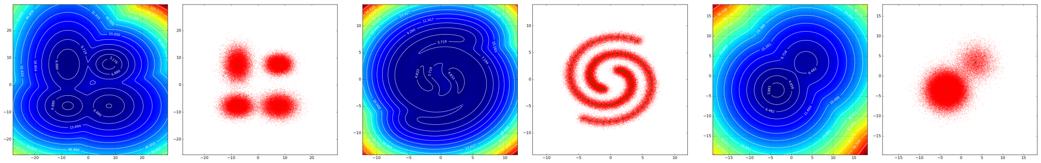


Figure 1: True energy functions and samples from synthetic distributions. Green dots in the sample plots indicate the mean of each Gaussian component.

5 Experiments

In this section, we verify our theoretical results empirically on several synthetic and real datasets. In particular, we evaluate whether the discriminator obtained from the entropy-regularized adversarial training can capture the density information (in the form of energy), while making sure the generator distribution matches the data distribution. For convenience, we refer to the obtained models as EGAN-Ent. Our experimental setting follows closely recommendations from Radford et al. (2015), except in Sec. 5.1 where we use fully-connected models (see appendix B.1 for details). [1]

5.1 Synthetic low-dimensional data

First, we consider three synthetic datasets in 2-dimensional space, which are drawn from the following distributions: (i) Mixture of 4 Gaussians with equal mixture weights, (ii) Mixture of 200 Gaussians arranged as two spirals (100 components each spiral), and (iii) Mixture of 2 Gaussians with highly biased mixture weights, $P(c_1) = 0.9, P(c_2) = 0.1$. We visualize the ground-truth energy of these distributions along with 100K training samples in Figure 1. Since the data lies in 2-dimensional space, we can easily visualize both the learned generator (by drawing samples) and the discriminator for direct comparison and evaluation. We evaluate here our EGAN-Ent model using both approximations: the nearest-neighbor based approximation (EGAN-Ent-NN) and the variational-inference based approximation (EGAN-Ent-VI), and compare them with two baselines: the original GAN and the energy based GAN with no regularization (EGAN-Const).

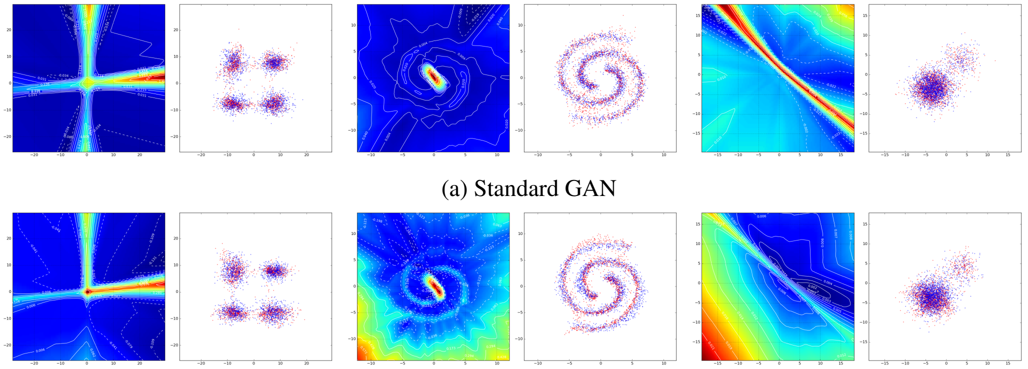
Experiment results are summarized in Figure 2 for baseline models, and Figure 3 for the proposed models. As we can see, all four models can generate perfect samples. However, for the discriminator, both GAN and EGAN-Const lead to degenerate solution, assigning flat energy inside the empirical data support. In comparison, EGAN-Ent-VI and EGAN-Ent-NN clearly capture the density information, though to different degrees. Specifically, on the equally weighted Gaussian mixture and the two-spiral mixture datasets, EGAN-Ent-NN tends to give more accurate and fine-grained solutions compared to EGAN-Ent-VI. However, on the biased weighted Gaussian mixture dataset, EGAN-Ent-VI actually fails to captures the correct mixture weights of the two modes, incorrectly assigning lower energy to the mode with lower probability (smaller weight). In contrast, EGAN-Ent-NN perfectly captures the bias in mixture weight, and obtains a contour very close to the ground truth.

To better quantify these differences, we present detailed comparison based on KL divergence in appendix B.2. What's more, the performance difference between EGAN-Ent-VI and EGAN-Ent-NN on biased Gaussian mixture reveals the limitations of the variational inference based approximation, i.e. providing inaccurate gradients. Due to space consideratiosn, we refer interested readers to the appendix B.3 for a detailed discussion.

5.2 Ranking NIST digits

In this experiment, we verify that the results in synthetic datasets can translate into data with higher dimensions. While visualizing the learned energy function is not feasible in high-dimensional space, we can verify whether the learned energy function learns relative densities by inspecting the ranking of samples according to their assigned energies. We train on 28×28 images of a single handwritten

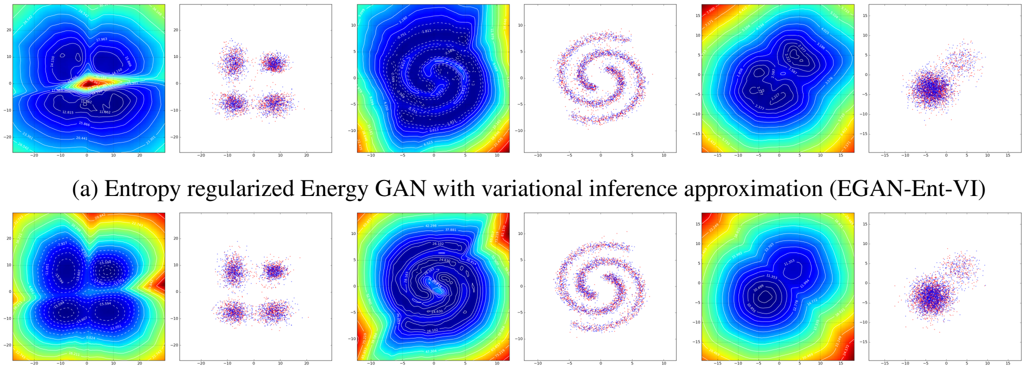
[1] For more details, please refer to https://github.com/zihangdai/cegan_iclr2017.



(a) Standard GAN

(b) Energy GAN without regularization (EGAN-Const)

Figure 2: Learned energies and samples from baseline models whose discriminator cannot retain density information at the optimal. In the sample plots, blue dots indicate generated samples, and red dots indicate real ones.



(a) Entropy regularized Energy GAN with variational inference approximation (EGAN-Ent-VI)

(b) Entropy regularized Energy GAN with nearest neighbor approximation (EGAN-Ent-NN)

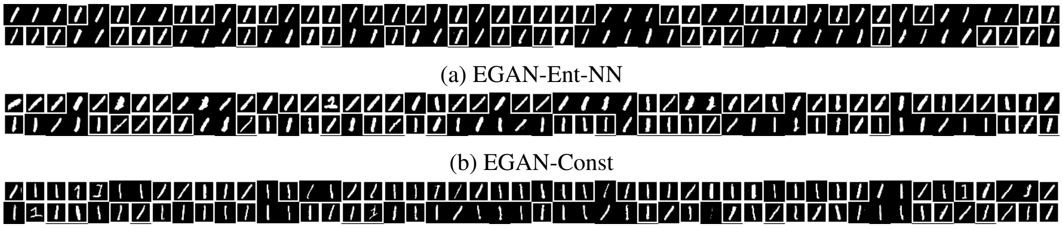
Figure 3: Learned energies and samples from proposed models whose discriminator can retain density information at the optimal. Blue dots are generated samples, and red dots are real ones.

digit from the NIST dataset. [2] We compare the ability of EGAN-Ent-NN with both EGAN-Const and GAN on ranking a set of 1,000 images, half of which are generated samples and the rest are real test images. Figures 4 and 5 show the top-100 and bottom-100 ranked images respectively for each model, after training them on digit 1. We also show in Figure 7 the mean of all training samples, so we can get a sense of what is the most common style (highest density) of digit 1 in NIST. We can notice that all of the top-ranked images by EGAN-Ent-NN look similar to the mean sample. In addition, the lowest-ranked images are clearly different from the mean image, with either high (clockwise or counter-clockwise) rotation degrees from the mean, or an extreme thickness level. We do not see such clear distinction in other models. We provide in the appendix B.4 the ranking of the full set of images.

5.3 Sample quality on natural image datasets

In this last set of experiments, we evaluate the visual quality of samples generated by our model in two datasets of natural images, namely CIFAR-10 and CelebA. We employ here the variational-based approximation for entropy regularization, which can scale well to high-dimensional data. Figure 6 shows samples generated by EGAN-Ent-VI. We can see that despite the noisy gradients provided by the variational approximation, our model is able to generate high-quality samples.

[2] https://www.nist.gov/srd/nist-special-database-19, which is an extended version of MNIST with an average of over 74K examples per digit.

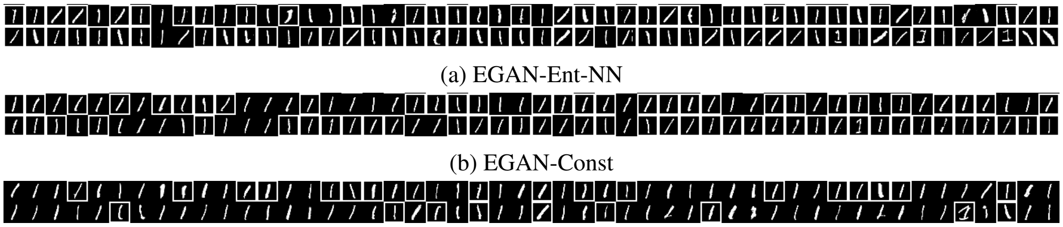


(a) EGAN-Ent-NN

(b) EGAN-Const

(c) GAN

Figure 4: 100 highest-ranked images out of 1000 generated and reals (bounding box) samples.



(a) EGAN-Ent-NN

(b) EGAN-Const

(c) GAN

Figure 5: 100 lowest-ranked images out of 1000 generated and reals (bounding box) samples.

We futher validate the quality of our model's samples on CIFAR-10 using the *Inception score* proposed by (Salimans et al., 2016) [3]. Table 1 shows the scores of our EGAN-Ent-VI, the best GAN model from Salimans et al. (2016) which uses only unlabeled data, and an EGAN-Const model which has the same architecture as our model. We notice that even without employing suggested techniques in Salimans et al. (2016), energy-based models perform quite similarly to the GAN model. Furthermore, the fact that our model scores higher than EGAN-Const highlights the importance of entropy regularization in obtaining good quality samples.

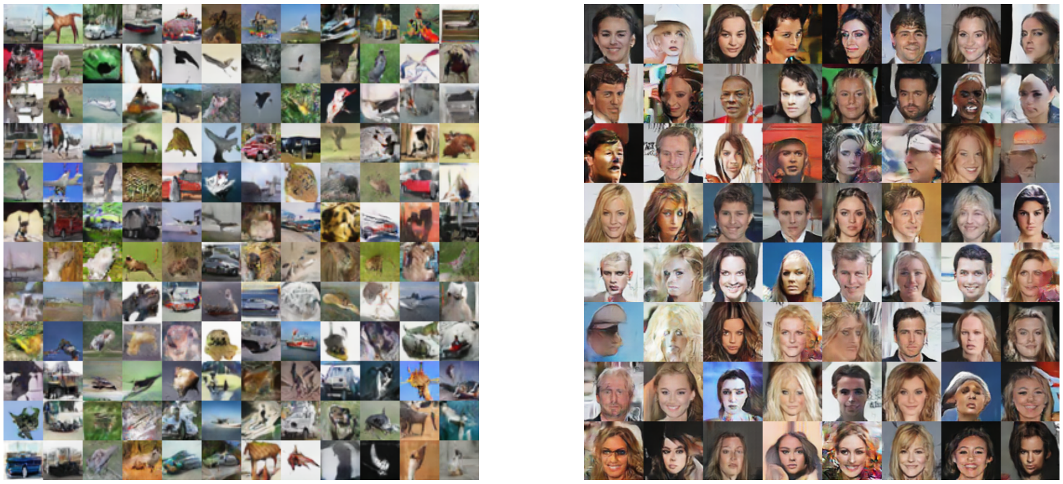
6 Conclusion

In this paper we have addressed a fundamental limitation in adversarial learning approaches, which is their inability of providing sensible energy estimates for samples. We proposed a novel adversarial learning formulation which results in a discriminator function that recovers the true data energy. We provided a rigorous characterization of the learned discriminator in the non-parametric setting, and proposed two methods for instantiating it in the typical parametric setting. Our experimental results verify our theoretical analysis about the discriminator properties, and show that we can also obtain samples of state-of-the-art quality.

7 Acknowledgements

We would like to thank the developers of Theano (Theano Development Team, 2016) for developing such a powerful tool for scientific computing. Amjad Almahairi was supported by funding from Maluuba Research.

[3] Using the evaluation script released in `https://github.com/openai/improved-gan/`



(a) CIFAR-10

(b) CelebA

Figure 6: Samples generated from our model.

Model	Our model	Improved GAN†	EGAN-Const
Score ± std.	7.07 ± .10	6.86 ± .06	6.7447 ± 0.09

Table 1: Inception scores on CIFAR-10. † As reported in Salimans et al. (2016) without using labeled data.

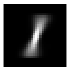


Figure 7: mean digit

References

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1. ACM, 2004.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *arXiv preprint arXiv:1606.03476*, 2016.

Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016.

A. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Icml*, pp. 663–670, 2000.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *arXiv preprint arXiv:1606.00709*, 2016.

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

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, pp. 1433–1438, 2008.

A Supplementary materials for Section 3

A.1 Optimal discriminator form under the proposed formulation

Proof of proposition 3.1. Refining the Lagrange $L(p_{\text{gen}}, c)$ by introducing additional dual variables for the probability constraints (the second and third), the new Lagrange function has the form

$$L(p_{\text{gen}}, c, \mu, \lambda) = K(p_{\text{gen}}) + \sum_{x \in \mathcal{X}} c(x) \Big(p_{\text{gen}}(x) - p_{\text{data}}(x) \Big) - \sum_{x \in \mathcal{X}} \mu(x) p_{\text{gen}}(x) + \lambda (\sum_{x \in \mathcal{X}} p_{\text{gen}}(x) - 1) \tag{17}$$

where $c(x) \in \mathbb{R}, \forall x$, $\mu(x) \in \mathbb{R}_+, \forall x$, and $\lambda \in \mathbb{R}$ are the dual variables. The KKT conditions for the optimal primal and dual variables are as follows

$$\begin{aligned} \frac{\partial K(p_{\text{gen}})}{\partial p_{\text{gen}}(x)} \Big|_{p_{\text{gen}} = p_{\text{data}}} + c^*(x) - \mu^*(x) + \lambda^* = 0, &\quad \forall x &\qquad \text{(stationarity)} \\ \mu^*(x) p^*_{\text{gen}}(x) = 0, &\quad \forall x &\qquad \text{(complement slackness)} \\ \mu^*(x) \geq 0, &\quad \forall x &\qquad \text{(dual feasibility)} \\ p^*_{\text{gen}}(x) \geq 0, \quad p^*_{\text{gen}}(x) = p_{\text{data}}(x), &\quad \forall x &\qquad \text{(primal feasibility)} \\ \sum_{x \in \mathcal{X}} p_{\text{gen}}(x) = 1 & &\qquad \text{(primal feasibility)} \end{aligned} \tag{18}$$

Rearranging the conditions above, we get $p^*_{\text{gen}}(x) = p_{\text{data}}(x), \forall x \in \mathcal{X}$ as well as equation (4), which concludes the proof. □

A.2 Optimal conditions of EBGAN

In (Zhao et al., 2016), the training objectives of the generator and the discriminator cannot be written as a single minimax optimization problem since the margin structure is only applied to the objective of the discriminator. In addition, the discriminator is designed to produce the mean squared reconstruction error of an auto-encoder structure. This restricted the range of the discriminator output to be non-negative, which is equivalent to posing a set constraint on the discriminator under the non-parametric setting.

Thus, to characterize the optimal generator and discriminator, we adapt the same analyzing logic used in the proof sketch of the original GAN (Goodfellow et al., 2014). Specifically, given a specific generator distribution p_{gen}, the optimal discriminator function given the generator distribution $c^*(x; p_{\text{gen}})$ can be derived by examining the objective of the discriminator. Then, the conditional optimal discriminator function is substituted into the training objective of p_{gen}, simplifying the "adversarial" training as a minimizing problem only w.r.t. p_{gen}, which can be well analyzed.

Firstly, given any generator distribution p_{gen}, the EBGAN training objective for the discriminator can be written as the following form

$$\begin{aligned} c^*(x; p_{\text{gen}}) &= \operatorname*{arg\,max}_{c \in \mathcal{C}} \quad -\mathop{\mathbb{E}}_{p_{\text{gen}}} \max(0, m - c(x)) - \mathop{\mathbb{E}}_{p_{\text{data}}} c(x) \\ &= \operatorname*{arg\,max}_{c \in \mathcal{C}} \quad \mathop{\mathbb{E}}_{p_{\text{gen}}} \min(0, c(x) - m) - \mathop{\mathbb{E}}_{p_{\text{data}}} c(x) \end{aligned} \tag{19}$$

where $\mathcal{C} = \{c : c(x) \geq 0, \forall x \in \mathcal{X}\}$ is the set of allowed non-negative discriminator functions. Note this set constraint comes from the fact the mean squared reconstruction error as discussed above.

Since the problem (19) is independent w.r.t. each x, the optimal solution can be easily derived as

$$c^*(x; p_{\text{gen}}) = \begin{cases} 0, & p_{\text{gen}}(x) < p_{\text{data}}(x) \\ m, & p_{\text{gen}}(x) > p_{\text{data}}(x) \\ \alpha_x, & p_{\text{gen}}(x) = p_{\text{data}}(x) > 0 \\ \beta_x, & p_{\text{gen}}(x) = p_{\text{data}}(x) = 0 \end{cases} \tag{20}$$

where $\alpha_x \in [0, m]$ is an under-determined number, a $\beta_x \in [0, \infty)$ is another under-determined non-negative real number, and the subscripts in m, α_x, β_x reflect that fact that these under-determined values can be distinct for different x.

This way, the overall training objective can be cast into a minimization problem w.r.t. p_{gen},

$$\begin{aligned} p^*_{\text{gen}} &= \arg\min_{p_{\text{gen}} \in \mathcal{P}} \mathop{\mathbb{E}}_{x \sim p_{\text{gen}}} c^*(x; p_{\text{gen}}) - \mathop{\mathbb{E}}_{x \sim p_{\text{data}}} c^*(x; p_{\text{gen}}) \\ &= \arg\min_{p_{\text{gen}} \in \mathcal{P}} \sum_{x \in \mathcal{X}} \Big[p_{\text{gen}}(x) - p_{\text{data}}(x) \Big] c^*(x; p_{\text{gen}}) \end{aligned} \tag{21}$$

where the second term of the first line is implicitly defined as the problem is an adversarial game between p_{gen} and c.

Proposition A.1. *The global optimal of the EBGAN training objective is achieved if and only if $p_{gen} = p_{data}$. At that point, $c^*(x)$ is fully under-determined.*

Proof. The proof is established by showing contradiction.

Firstly, assume the optimal $p^*_{\text{gen}} \neq p_{\text{data}}$. Thus, there must exist a non-equal set $\mathcal{X}_{\neq} = \{x \mid p_{\text{data}}(x) \neq p^*_{\text{gen}}(x)\}$, which can be further splitted into two subsets, the greater-than set $\mathcal{X}_{>} = \{x \mid p^*_{\text{gen}}(x) > p_{\text{data}}(x)\}$, and the less-than set $\mathcal{X}_{<} = \{x \mid p^*_{\text{gen}}(x) < p_{\text{data}}(x)\}$. Similarly, we define the equal set $\mathcal{X}_{=} = \{x : p^*_{\text{gen}}(x) = p_{\text{data}}(x)\}$. Obviously, $\mathcal{X}_{>} \bigcup \mathcal{X}_{<} \bigcup \mathcal{X}_{=} = \mathcal{X}$.

Let $L(p_{\text{gen}}) = \sum_{x \in \mathcal{X}} \Big[p_{\text{gen}}(x) - p_{\text{data}}(x) \Big] c^*(x; p_{\text{gen}})$, substituting the results from equation (20) into (21), the $L(p_{\text{gen}})^*$ can be written as

$$\begin{aligned} L(p^*_{\text{gen}}) &= \sum_{x \in \mathcal{X}_{<} \bigcup \mathcal{X}_{<} \bigcup \mathcal{X}_{=}} \big[p^*_{\text{gen}}(x) - p_{\text{data}}(x) \big] c^*(x; p^*_{\text{gen}}) \\ &= \sum_{x \in \mathcal{X}_{<}} \big[p^*_{\text{gen}}(x) - p_{\text{data}}(x) \big] c^*(x; p^*_{\text{gen}}) + \sum_{x \in \mathcal{X}_{>}} \big[p^*_{\text{gen}}(x) - p_{\text{data}}(x) \big] c^*(x; p^*_{\text{gen}}) \\ &= m \sum_{x \in \mathcal{X}_{>}} p^*_{\text{gen}}(x) - p_{\text{data}}(x) \\ &> 0 \end{aligned} \tag{22}$$

However, when $p'_{\text{gen}} = p_{\text{data}}$, we have

$$L(p'_{\text{gen}}) = 0 < L(p^*_{\text{gen}}) \tag{23}$$

which contradicts the optimal (miminum) assumption of p^*_{gen}. Hence, the contradiction concludes that at the global optimal, $p^*_{\text{gen}} = p_{\text{data}}$. By equation (20), it directly follows that $c^*(x; p^*_{\text{gen}}) = \alpha_x$, which completes the proof. □

A.3 Analysis of adding additional training signal to GAN formulation

To show that simply adding the same training signal to GAN will not lead to the same result, it is more convenient to directly work with the formulation of f-GAN (Nowozin et al., 2016, equation (6)) family, which include the original GAN formulation as a special case.

Specifically, the general f-GAN formulation takes the following form

$$\max_{c} \min_{p_{\text{gen}} \in \mathcal{P}} \mathop{\mathbb{E}}_{x \sim p_{\text{gen}}} \Big[f^\star(c(x)) \Big] - \mathop{\mathbb{E}}_{x \sim p_{\text{data}}} \Big[c(x) \Big], \tag{24}$$

where the $f^\star(\cdot)$ denotes the convex conjugate (Boyd & Vandenberghe, 2004) of the f-divergence function. The optimal condition of the discriminator can be found by taking the variation w.r.t. c, which gives the optimal discriminator

$$c^*(x) = f'(\frac{p_{\text{data}}(x)}{p_{\text{gen}}(x)}) \tag{25}$$

where $f'(\cdot)$ is the first-order derivative of $f(\cdot)$. Note that, even when we add an extra term $L(p_{\text{gen}})$ to equation (24), since the term $K(p_{\text{gen}})$ is a constant w.r.t. the discriminator, it does not change the result given by equation (25) about the optimal discriminator. As a consequence, for the optimal discriminator to retain the density information, it effectively means $p_{\text{gen}} \neq p_{\text{data}}$. Hence, there will be a contradiction if both $c^*(x)$ retains the density information, and the generator matches the data distribution.

Intuitively, this problem roots in the fact that f-divergence is quite "rigid" in the sense that given the $p_{\text{gen}}(x)$ it only allows one fixed point for the discriminator. In comparison, the divergence used in our proposed formulation, which is the expected cost gap, is much more flexible. By the expected cost gap itself, i.e. without the $K(p_{\text{gen}})$ term, the optimal discriminator is actually under-determined.

B Supplementary Materials for Section 5

B.1 Experiment setting

Here, we specify the neural architectures used for experiements presented in Section 5.

Firstly, for the Egan-Ent-VI model, we parameterize the approximate posterior distribution $q_{\text{gen}}(z \mid x)$ with a diagonal Gaussian distribution, whose mean and covariance matrix are the output of a trainable inference network, i.e.

$$\begin{aligned} q_{\text{gen}}(z \mid x) &= \mathcal{N}(\mu, \mathbf{I}\sigma^2) \\ \mu, \log\sigma &= f^{\text{infer}}(x) \end{aligned} \tag{26}$$

where f^{infer} denotes the inference network, and $\mathbf{I}$ is the identity matrix. Note that the Inference Network only appears in the Egan-Ent-VI model.

For experiments with the synthetic datasets, the following fully-connected feed forward neural networks are employed

- Generator: `FC(4,128)-BN-ReLU-FC(128,128)-BN-ReLU-FC(128,2)`
- Discriminator: `FC(2,128)-ReLU-FC(128,128)-ReLU-FC(128,1)`
- Inference Net: `FC(2,128)-ReLU-FC(128,128)-ReLU-FC(128,4*2)`

where `FC` and `BN` denote fully-connected layer and batch normalization layer respectively. Note that since the input noise to the generator has dimension `4`, the Inference Net output has dimension `4*2`, where the first 4 elements correspond the inferred mean, and the last 4 elements correspond to the inferred diagonal covariance matrix in log scale.

For the handwritten digit experiment, we closely follow the DCGAN (Radford et al., 2015) architecture with the following configuration

- Generator: `FC(10,512*7*7)-BN-ReLU-DC(512,256;4c2s)-BN-ReLU -DC(256,128;4c2s)-BN-ReLU-DC(128,1;3c1s)-Sigmoid`
- Discriminator: `CV(1,64;3c1s)-BN-LRec-CV(64,128;4c2s)-BN-LRec -CV(128,256;4c2s)-BN-LRec-FC(256*7*7,1)`
- Inference Net: `CV(1,64;3c1s)-BN-LRec-CV(64,128;4c2s)-BN-LRec -CV(128,256;4c2s)-BN-LRec-FC(256*7*7,10*2)`

Here, `LRec` is the leaky rectified non-linearity recommended by Radford et al. (2015). In addition, `CV(128,256,4c2s)` denotes a convolutional layer with 128 input channels, 256 output channels, and kernel size 4 with stride 2. Similarly, `DC(256,128,4c2s)` denotes a corresponding transposed convolutional operation. Compared to the original DCGAN architecture, the discriminator under our formulation does not have the last sigmoid layer which squashes a scalar value into a probability in [0, 1].

For celebA experiment with 64 × 64 color images, we use the following architecture

- Generator: `FC(10,512*4*4)-BN-ReLU-DC(512,256;4c2s)-BN-ReLU-DC(256,128;4c2s) -BN-ReLU-DC(256,128;4c2s)-BN-ReLU-DC(128,3;4c2s)-Tanh`
- Discriminator: `CV(3,64;4c2s)-BN-LRec-CV(64,128;4c2s)-BN-LRec-CV(128,256;4c2s) -BN-LRec-CV(256,256;4c2s)-BN-LRec-FC(256*4*4,1)`
- Inference Net: `CV(3,64;4c2s)-BN-LRec-CV(64,128;4c2s)-BN-LRec-CV(128,256;4c2s) -BN-LRec-CV(256,256;4c2s)-BN-LRec-FC(256*4*4,10*2)`

For Cifar10 experiment, where the image size is 32×32, similar architecture is used

- Generator: `FC(10,512*4*4)-BN-ReLU-DC(512,256;4c2s)-BN-ReLU-DC(256,128;3c1s) -BN-ReLU-DC(256,128;4c2s)-BN-ReLU-DC(128,3;4c2s)-Tanh`
- Discriminator: `CV(3,64;3c1s)-BN-LRec-CV(64,128;4c2s)-BN-LRec-CV(128,256;4c2s) -BN-LRec-CV(256,256;4c2s)-BN-LRec-FC(256*4*4,1)`
- Inference Net: `CV(3,64;3c1s)-BN-LRec-CV(64,128;4c2s)-BN-LRec-CV(128,256;4c2s) -BN-LRec-CV(256,256;4c2s)-BN-LRec-FC(256*4*4,10*2)`

Given the chosen architectures, we follow Radford et al. (2015) and use Adam as the optimization algorithm. For more detailed hyper-parameters, please refer to the code.

B.2 Quantitative comparison of different models

	Gaussian Mixture: $\mathrm{KL}(p_{\text{data}} \Vert p_{\text{emp}}) = 0.0291$, $\mathrm{KL}(p_{\text{emp}} \Vert p_{\text{data}}) = 0.0159$									
KL Divergence	$p_{\text{gen}} \Vert p_{\text{emp}}$	$p_{\text{emp}} \Vert p_{\text{gen}}$	$p_{\text{gen}} \Vert p_{\text{data}}$	$p_{\text{data}} \Vert p_{\text{gen}}$	$p_{\text{disc}} \Vert p_{\text{emp}}$	$p_{\text{emp}} \Vert p_{\text{disc}}$	$p_{\text{disc}} \Vert p_{\text{data}}$	$p_{\text{data}} \Vert p_{\text{disc}}$	$p_{\text{gen}} \Vert p_{\text{disc}}$	$p_{\text{disc}} \Vert p_{\text{gen}}$
GAN	0.3034	0.5024	0.2498	0.4807	6.7587	2.0648	6.2020	2.0553	2.4596	7.0895
EGAN-Const	0.2711	0.4888	0.2239	0.4735	6.7916	2.1243	6.2159	2.1149	2.5062	7.0553
EGAN-Ent-VI	0.1422	0.1367	0.0896	0.1214	0.8866	0.6532	0.7215	0.6442	0.7711	1.0638
EGAN-Ent-NN	**0.1131**	**0.1006**	**0.0621**	**0.0862**	**0.0993**	**0.1356**	**0.0901**	**0.1187**	**0.1905**	**0.1208**
	Biased Gaussian Mixture: $\mathrm{KL}(p_{\text{data}} \Vert p_{\text{emp}}) = 0.0273$, $\mathrm{KL}(p_{\text{emp}} \Vert p_{\text{data}}) = 0.0144$									
KL Divergence	$p_{\text{gen}} \Vert p_{\text{emp}}$	$p_{\text{emp}} \Vert p_{\text{gen}}$	$p_{\text{gen}} \Vert p_{\text{data}}$	$p_{\text{data}} \Vert p_{\text{gen}}$	$p_{\text{disc}} \Vert p_{\text{emp}}$	$p_{\text{emp}} \Vert p_{\text{disc}}$	$p_{\text{disc}} \Vert p_{\text{data}}$	$p_{\text{data}} \Vert p_{\text{disc}}$	$p_{\text{gen}} \Vert p_{\text{disc}}$	$p_{\text{disc}} \Vert p_{\text{gen}}$
GAN	0.0788	0.0705	0.0413	0.0547	7.1539	2.5230	6.4927	2.5018	2.5205	7.1140
EGAN-Const	0.1545	0.1649	0.1211	0.1519	7.1568	2.5269	6.4969	2.5057	2.5860	7.1995
EGAN-Ent-VI	**0.0576**	0.0668	**0.0303**	0.0518	3.9151	1.3574	2.9894	1.3365	1.4052	4.0632
EGAN-Ent-NN	0.0784	**0.0574**	0.0334	**0.0422**	**0.8505**	**0.3480**	**0.5199**	**0.3299**	**0.3250**	**0.7835**
	Two-spiral Gaussian Mixture: $\mathrm{KL}(p_{\text{data}} \Vert p_{\text{emp}}) = 0.3892$, $\mathrm{KL}(p_{\text{emp}} \Vert p_{\text{data}}) = 1.2349$									
KL Divergence	$p_{\text{gen}} \Vert p_{\text{emp}}$	$p_{\text{emp}} \Vert p_{\text{gen}}$	$p_{\text{gen}} \Vert p_{\text{data}}$	$p_{\text{data}} \Vert p_{\text{gen}}$	$p_{\text{disc}} \Vert p_{\text{emp}}$	$p_{\text{emp}} \Vert p_{\text{disc}}$	$p_{\text{disc}} \Vert p_{\text{data}}$	$p_{\text{data}} \Vert p_{\text{disc}}$	$p_{\text{gen}} \Vert p_{\text{disc}}$	$p_{\text{disc}} \Vert p_{\text{gen}}$
GAN	0.5297	0.2701	0.3758	0.7240	6.3507	1.7180	4.3818	1.0866	1.6519	5.7694
EGAN-Const	0.7473	1.0325	0.7152	1.6703	5.9930	1.5732	3.9749	0.9703	1.8380	6.0471
EGAN-Ent-VI	0.2014	0.1260	**0.4283**	**0.8399**	1.1099	0.3508	**0.3061**	**0.4037**	0.4324	0.9917
EGAN-Ent-NN	**0.1246**	**0.1147**	0.4475	1.2435	**0.1036**	**0.0857**	0.4086	0.7917	**0.1365**	**0.1686**

Table 2: Pairwise KL divergence between distributions. Bold face indicate the lowest divergence within group.

In order to quantify the quality of recovered distributions, we compute the pairwise KL divergence of the following four distributions:

- The real data distribution with analytic form, denoted as p_{data}
- The empirical data distribution approximated from the 100K training data, denoted as p_{emp}
- The generator distribution approximated from 100K generated data, denoted as p_{gen}
- The discriminator distribution re-normalized from the learned energy, denoted as p_{disc}

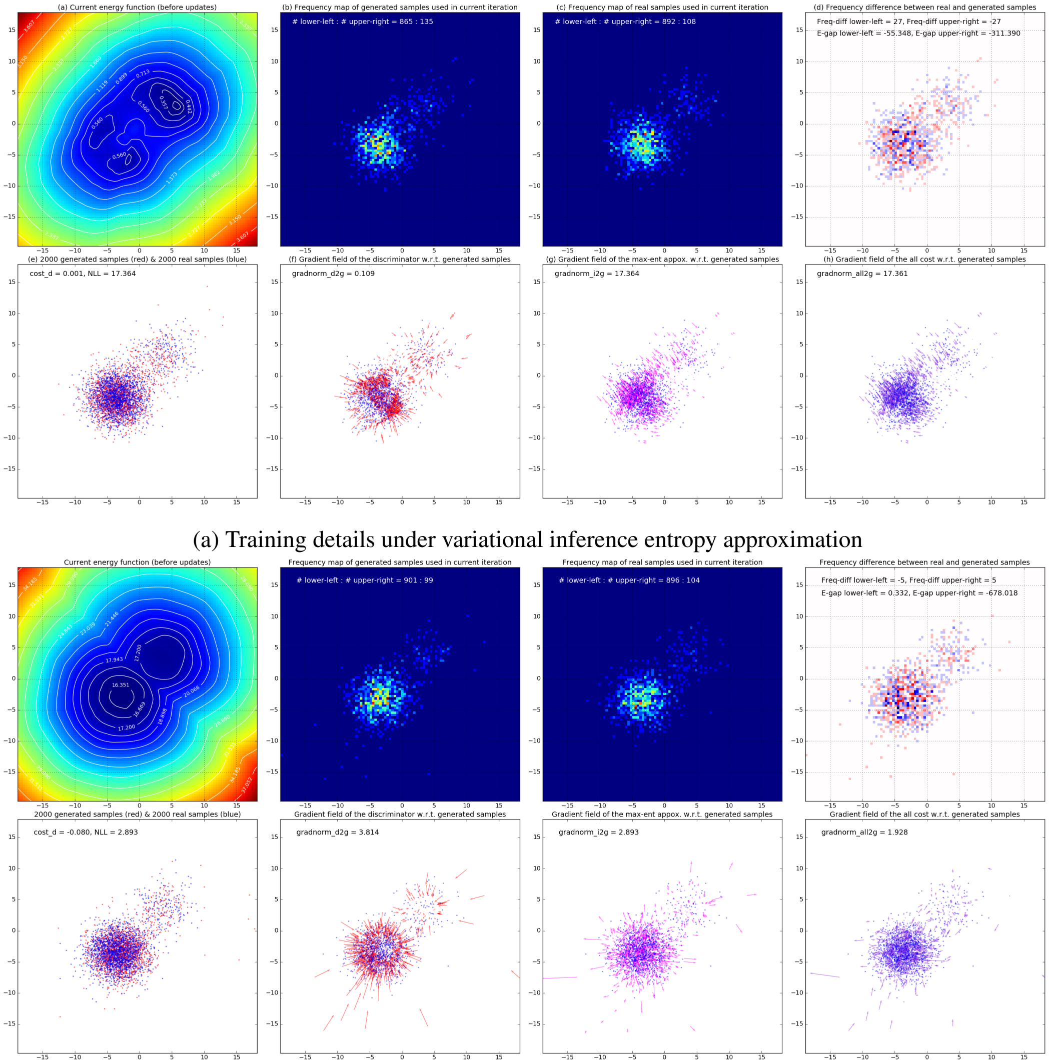
Since the synthetic datasets are two dimensional, we approximate both the empirical data distribution and the generator distribution using the simple histogram estimation. Specifically, we divide the canvas into a 100-by-100 grid, and assign each sample into its nearest grid cell based on euclidean distance. Then, we normalize the number of samples in each cell into a proper distribution. When recovering the discriminator distribution from the learned energy, we assume that $\mu^*(x) = 0$ (i.e. infinite data support), and discretize the distribution into the same grid cells

$$p_{\text{disc}}(x) = \frac{\exp(-c^*(x))}{\sum_{x' \in \text{Grid}} \exp(-c^*(x'))}, \forall x \in \text{Grid}$$

Based on these approximation, Table 2 summarizes the results. For all measures related to the discriminator distribution, EGAN-Ent-VI and EGAN-Ent-NN significantly outperform the other two baseline models, which matches our visual assessment in Figure 2 and 3. Meanwhile, the generator distributions learned from our proposed framework also achieve relatively lower divergence to both the empirical data distribution and the true data distribution.

B.3 Comparison of the entropy (gradient) approximation methods

In order to understand the performance difference between EGAN-Ent-VI and EGAN-Ent-NN, we analyze the quality of the entropy gradient approximation during training. To do that, we visualize some detailed training information in Figure 8.



(a) Training details under variational inference entropy approximation

(b) Training details under nearest neighbor entropy approximation

Figure 8: For convenience, we will use Fig. (i,j) to refer to the subplot in row i, column j. Fig. (1,1): current energy plot. Fig. (1,2): frequency map of generated samples in the current batch. Fig. (1,3): frequency map of real samples in the current batch. Fig-(1,4): frequency difference between real and generated samples. Fig. (2,1) comparison between more generated from current model and real sample. Fig. (2,2): the discriminator gradient w.r.t. each training sample. Fig. (2,3): the entropy gradient w.r.t. each training samples. Fig. (2,4): all gradient (discriminator + entropy) w.r.t. each training sample.

As we can see in figure 8a, the viarational entropy gradient approximation w.r.t. samples is not accurate:

- It is inaccurate in terms of gradient direction. Ideally, the direction of the entropy gradient should be pointing from the center of its closest mode towards the surroundings, with the direction orthogonal to the implicit contour in Fig. (1,2). However, the direction of gradients in the Fig. (2,3) does not match this.

- It is inaccurate in magnitude. As we can see, the entropy approximation gradient (Fig. (2,3)) has much larger norm than the discriminator gradient (Fig. (2,2)). As a result, the total gradient (Fig. (2,4)) is fully dominated by the entropy approximation gradient. Thus, it usually takes much longer for the generator to learn to generate rare samples, and the training also proceeds much slower compared to the nearest neighbor based approximation.

In comparison, the nearest neighbor based gradient approximation is much more accurate as shown in 8b. As a result, it leads to more accurate energy contour, as well as faster training. What's more, from Figure 8b Fig. (2,4), we can see the entropy gradient does have the cancel-out effect on the discriminator gradient, which again matches our theory.

B.4 Ranking NIST Digits

Figure 9 shows the ranking of all 1000 generated and real images (from the test set) for three models: EGAN-Ent-NN, EGAN-Const, and GAN. We can clearly notice that in EGAN-Ent-NN the top-ranked digits look very similar to the mean digit. From the upper-left corner to the lower-right corner, the transition trend is: the rotation degree increases, and the digits become increasingly thick or thin compared to the mean. In addition, samples in the last few rows do diverge away from the mean image: either highly diagonal to the right or left, or have different shape: very thin or thick, or typewriter script. Other models are not able to achieve a similar clear distinction for high versus low probability images. Finally, we consistently observe the same trend in modeling other digits, which are not shown in this paper due to space constraint.

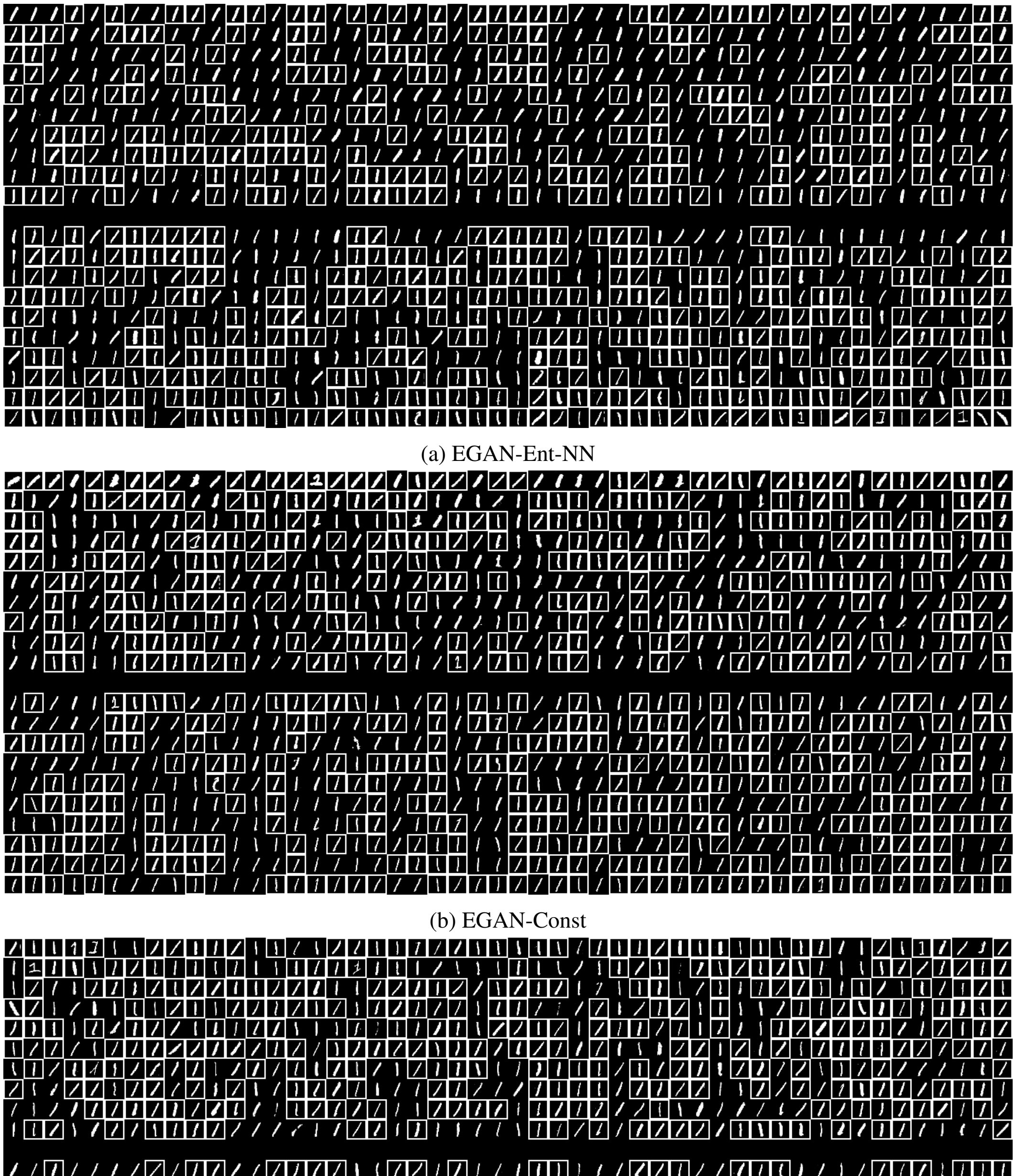
B.5 Classifier performance as a proxy measure

As mentioned in Section 5, evaluating the proposed formulation quantitatively on high-dimensional data is extremely challenging. Here, in order to provide more quantitative intuitions on the learned discriminator at convergence, we adopt a proxy measure. Specifically, we take the last-layer activation of the converged discriminator network as **fixed** pretrained feature, and build a linear classifier upon it. Hypothetically, if the discriminator does not degenerate, the extracted last-layer feature should maintain more information about the data points, especially compared to features from degenerated discriminators. Following this idea, we first train EGAN-Ent-NN, EGAN-Const, and GAN on the MNIST till convergence, and then extract the last-layer activation from their discriminator networks as fixed feature input. Based on fixed feature, a randomly initialized linear classifier is trained to do classification on MNIST. Based on 10 runs (with different initialization) of each of the three models, the test classification performance is summarized in Table 3. For comparison purpose, we also include a baseline where the input features are extracted from a discriminator network with random weights.

Test error (%)	EGAN-Ent-NN	EGAN-Const	GAN	Random
Min	**1.160**	1.280	1.220	3.260
Mean	**1.190**	1.338	1.259	3.409
Std.	0.024	0.044	0.032	0.124

Table 3: Test performance of linear classifiers based on last-layer discriminator features.

Based on the proxy measure, EGAN-Ent-NN seems to maintain more information of data, which suggests that the discriminator from our proposed formulation is more informative. Despite the positive result, it is important to point out that maintaining information about categories does not necessarily mean maintaining information about the energy (density). Thus, this proxy measure should be understood cautiously.



(a) EGAN-Ent-NN

(b) EGAN-Const

(c) GAN

Figure 9: 1000 generated and test images (bounding box) ranked according their assigned energies.