ENTROPIC GANS MEET VAES: A STATISTICAL APPROACH TO COMPUTE SAMPLE LIKELIHOODS IN GANS

Anonymous authors
Paper under double-blind review

ABSTRACT

Building on the success of deep learning, two modern approaches to learn a probability model of the observed data are Generative Adversarial Networks (GANs) and Variational AutoEncoders (VAEs). VAEs consider an explicit probability model for the data and compute a generative distribution by maximizing a variational lower-bound on the log-likelihood function. GANs, however, compute a generative model by minimizing a distance between observed and generated probability distributions without considering an explicit model for the observed data. The lack of having explicit probability models in GANs prohibits computation of sample likelihoods in their frameworks and limits their use in statistical inference problems. In this work, we show that an optimal transport GAN with the entropy regularization can be viewed as a generative model that maximizes a lower-bound on sample likelihoods, an approach that VAEs are based on. In particular, our proof constructs an explicit probability model for GANs that can be used to compute likelihood statistics within GAN's framework. Our numerical results on several datasets demonstrate consistent trends with the proposed theory.

1 INTRODUCTION

Learning generative models is becoming an increasingly important problem in machine learning and statistics with a wide range of applications in self-driving cars (Santana & Hotz, 2016), robotics (Hirose et al., 2017), natural language processing (Lee & Tsao, 2018), domain-transfer (Tzeng et al., 2017), computational biology (Ghahramani et al., 2018), etc. Two modern approaches to deal with this problem are Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and Variational AutoEncoders (VAEs) (Makhzani et al., 2015; Rosca et al., 2017; Tolstikhin et al., 2017; Mescheder et al., 2017).

VAEs compute a generative model by maximizing a variational lower-bound on average sample likelihoods using an explicit probability distribution for the data. GANs, however, compute a generative model by minimizing a distance between observed and generated distributions without considering an explicit model for the data. Empirically, GANs have been shown to produce higher-quality generative samples than that of VAEs (Karras et al., 2017). However, since GANs do not consider an explicit probability model for the data, we are unable to compute sample likelihoods using their generative models. Computations of sample likelihoods and *posterior* distributions of latent variables are critical in several statistical inference. Inability in obtaining such statistics within GAN's framework severely limits their application in such statistical inference problems.

In this paper, we resolve these issues for a general formulation of GANs by providing a theoretically-justified approach to compute sample likelihoods using GAN's generative model. Our results can open new directions to use GANs in massive-data applications such as model selection, sample selection, hypothesis-testing, etc (see more details in Section 5). Below, we state our main results *informally* without going into technical conditions while precise statements of our results are presented in Section 2.

Let Y and $\hat{Y} := \mathbf{G}(X)$ represent observed (i.e. real) and generative (i.e. fake or synthetic) variables, respectively. X (i.e. the latent variable) is the randomness used as the input to the generator $\mathbf{G}(.)$.

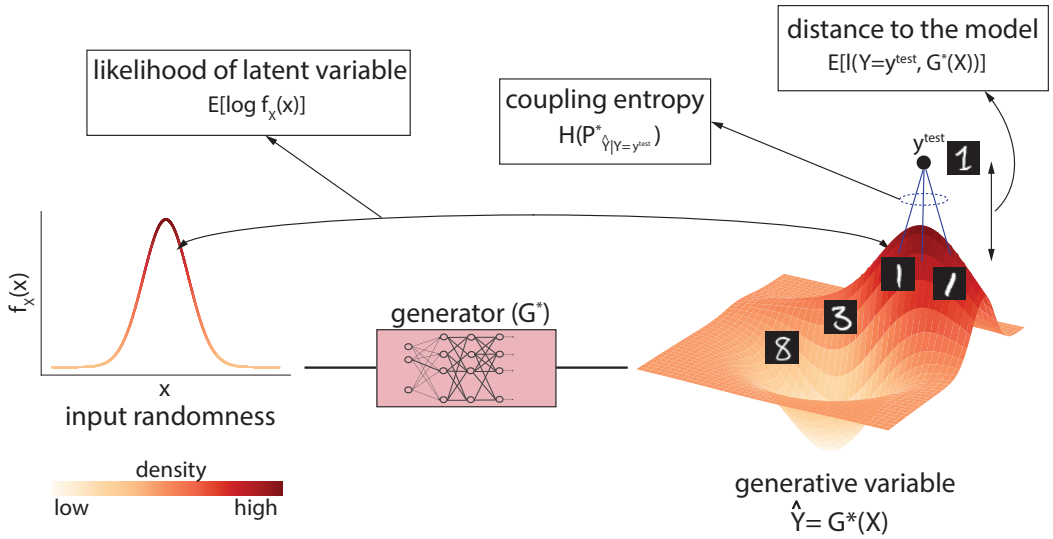


Figure 1: A statistical framework for GANs. By training a GAN architecture, we first compute optimal generator function $\mathbf{G}^*$ and optimal coupling between the real variable Y and the latent variable X. The sample likelihood of a test sample $\mathbf{y}^{\text{test}}$ can then be lower-bounded using a combination of three terms: (1) the expected distance of $\mathbf{y}^{\text{test}}$ to the distribution learnt by the generative model, (2) the entropy of the coupled latent variable given $\mathbf{y}^{\text{test}}$ and (3) the average likelihood of the coupled latent variable with $\mathbf{y}^{\text{test}}$.

Consider the following explicit probability model of the data given a latent sample $X = \mathbf{x}$:

$$f_{Y|X=\mathbf{x}}(\mathbf{y}) \propto \exp(-\ell(\mathbf{y}, \mathbf{G}(\mathbf{x}))), \tag{1.1}$$

where $\ell(.,.)$ is a loss function. Under this explicit probability model, we show that minimizing the objective of an *optimal transport* GAN (e.g. Wasserstein GAN Arjovsky et al. (2017)) with the cost function $\ell(.,.)$ and an entropy regularization (Cuturi, 2013; Seguy et al., 2017) maximizes a variational lower-bound on average sample likelihoods. I.e.

$$\text{average sample likelihoods} \;\geq\; -\,(\text{entropic GAN objective}) \;+\; \text{constants}. \tag{1.2}$$

If $\ell(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2$, the optimal transport (OT) GAN simplifies to WGAN (Arjovsky et al., 2017) while if $\ell(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$, the OT GAN simplifies to the quadratic GAN (or, W2GAN) (Feizi et al., 2017). The precise statement of this result can be found in Theorem 1. This result provides a statistical justification for GAN's optimization and puts it in par with VAEs whose goal is to maximize a lower bound on sample likelihoods. We note that the entropy regularization has been proposed primarily to improve computational aspects of GANs (Cuturi, 2013). Our results provide an additional statistical justification for this regularization term. Moreover, using GAN's training, we obtain a coupling between the observed variable Y and the latent variable X. This coupling provides the conditional distribution of the latent variable X given an observed sample $Y = \mathbf{y}$. The explicit model of equation 1.1 acts similar to the *decoder* in the VAE framework, while the coupling computed using GANs acts as an *encoder* in the VAE framework.

Another key question that we address here is how to estimate the likelihood of a new sample $\mathbf{y}^{\text{test}}$ given the generative model trained using GANs. For instance, if we train a GAN on *stop-sign* images, upon receiving a new image, one may wish to compute the likelihood of the new sample $\mathbf{y}^{\text{test}}$ coming from the trained generative model. In standard GAN formulations, the support of the generative distribution lies on the range of the optimal generator function. Thus, if the observed sample $\mathbf{y}^{\text{test}}$ does not lie on that range (which is very likely in practice), there is no way to assign a sensible likelihood score to that sample. Below, we show that using the explicit probability model equation 1.1 characterized for entropic GANs, we can *lower-bound* the likelihood of this sample $\mathbf{y}^{\text{test}}$ from the trained generative model. This is similar to the variational lower-bound on sample likelihoods used in VAEs. Our numerical results show that this lower-bound well-reflect the expected trends of the true sample likelihoods.

Let $\mathbf{G}^*$ and $\mathbb{P}^*_{Y,X}$ be the optimal generator function and the optimal coupling between real and latent variables, respectively. The optimal coupling $\mathbb{P}^*_{Y,X}$ can be computed efficiently for entropic GANs as we explain in Section 3. For other GAN architectures, one may approximate such optimal couplings as we explain in Section 4. Let $\mathbf{y}^{\text{test}}$ be a new observed sample. We can lower-bound the log likelihood of this sample as

$$\underbrace{\log f_Y(\mathbf{y}^{\text{test}})}_{\text{log likelihood}} \geq \underbrace{-\mathbb{E}_{\mathbb{P}^*_{X|Y=\mathbf{y}^{\text{test}}}}\left[\ell(\mathbf{y}^{\text{test}}, \mathbf{G}^*(\mathbf{x}))\right]}_{\text{distance to the generative model}} + \underbrace{H\left(\mathbb{P}^*_{X|Y=\mathbf{y}^{\text{test}}}\right)}_{\text{coupling entropy}} + \underbrace{\mathbb{E}_{\mathbb{P}^*_{X|Y=\mathbf{y}^{\text{test}}}}\left[-\frac{\|\mathbf{x}\|^2}{2}\right]}_{\text{likelihood of latent variable}}. \tag{1.3}$$

We present the precise statement of this result in Corollary 2). This result combines three components in order to approximate the likelihood of a sample given a trained generative model:

- The distance between $\mathbf{y}^{\text{test}}$ to the generative model: the larger this distance, the smaller is the likelihood of observing $\mathbf{y}^{\text{test}}$ from the generative model.
- The entropy of the coupled latent variable: larger entropy means larger the randomness of the coupled latent variable which translates to larger sample likelihood.
- The likelihood of the coupled latent variable: the larger the likelihood of the coupled latent samples, the larger the observed sample likelihood.

Figure 2a provides a pictorial illustration of these components.

In what follows, we explain the technical ingredients of our main results. In Section 3, we present computational methods for GANs and entropic GANs, while in Section 4, we provide numerical experiments on MNIST(LeCun et al., 1998) and CIFAR-10 (Krizhevsky, 2009) datasets.

2 Main Results

Let $Y \in \mathbb{R}^d$ represent the real-data random variable with a probability density function $f_Y(\mathbf{y})$. GAN's goal is to find a generator function $\mathbf{G} : \mathbb{R}^r \to \mathbb{R}^d$ such that $\hat{Y} := \mathbf{G}(X)$ has a similar distribution to Y. Let X be an r-dimensional random variable with a fixed probability density function $f_X(\mathbf{x})$. Here, we assume $f_X(.)$ is the density of a normal distribution. In practice, we observe m samples $\{\mathbf{y}_1, ..., \mathbf{y}_m\}$ from Y and generate m' samples from $\hat{Y}$, i.e., $\{\hat{\mathbf{y}}_1, ..., \hat{\mathbf{y}}_{m'}\}$ where $\hat{\mathbf{y}}_i = \mathbf{G}(\mathbf{x}_i)$ for $1 \leq i \leq m$. We represent these empirical distributions by $\mathbb{P}_Y$ and $\mathbb{P}_{\hat{Y}}$, respectively. Note that the number of generative samples m' can be arbitrarily large. Finally, we assume that the generator function $\mathbf{G}$ is injective which is often the case in practice since $\mathbf{G}$ maps from an r dimensional space to a d dimensional one where $r \ll d$.

GAN computes the optimal generator $\mathbf{G}^*$ by minimizing a distance between the observed distribution $\mathbb{P}_Y$ and the generative one $\mathbb{P}_{\hat{Y}}$. Common distance measures include *optimal transport* measures (e.g. Wasserstein GAN (Arjovsky et al., 2017), WGAN+Gradient Penalty (Gulrajani et al., 2017), GAN+Spectral Normalization (Miyato et al., 2018), WGAN+Truncated Gradient Penalty (Petzka et al., 2017), relaxed WGAN (Guo et al., 2017)), and *divergence* measures (e.g. the original GAN's formulation (Goodfellow et al., 2014), f-GAN (Nowozin et al., 2016)), etc.

In this paper, we focus on GANs based on optimal transport (OT) distance (Villani, 2008; Arjovsky et al., 2017) defined for a general loss function $\ell(.,.)$ as follows

$$W_\ell(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) := \min_{\mathbb{P}_{Y,\hat{Y}}} \mathbb{E}\left[\ell(Y, \hat{Y})\right]. \tag{2.1}$$

$\mathbb{P}_{Y,\hat{Y}}$ is the joint distribution whose marginal distributions are equal to $\mathbb{P}_Y$ and $\mathbb{P}_{\hat{Y}}$, respectively. If $\ell(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2$, this distance is called the first-order Wasserstein distance and is referred to by $W_1(.,.)$, while if $\ell(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$, this measure is referred to by $W_2^2(.,.)$ where W_2 is the second-order Wasserstein distance (Villani, 2008).

The optimal transport (OT) GAN is formulated using the following optimization (Arjovsky et al., 2017; Villani, 2008):

$$\min_{\mathbf{G} \in \mathcal{G}} W_\ell(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}), \tag{2.2}$$

where $\mathcal{G}$ is the set of generator functions. Examples of the OT GAN are WGAN (Arjovsky et al., 2017) corresponding to the first-order Wasserstein distance $W_1(.,.)$ and the quadratic GAN (or, the W2GAN) (Feizi et al., 2017) corresponding to the second-order Wasserstein distance $W_2(.,.)$.

Note that optimization equation 2.2 is a *min-min* optimization. The objective of this optimization is not smooth in $\mathbf{G}$ and it is often computationally expensive to obtain a solution (Sanjabi et al., 2018). One approach to improve computational aspects of this optimization is to add a regularization term to make its objective *strictly* convex (Cuturi, 2013; Seguy et al., 2017). A common strictly-convex regularization term is the negative Shannon entropy function defined as $H(\mathbb{P}_{Y,\hat{Y}}) := -\mathbb{E}\left[\log \mathbb{P}_{Y,\hat{Y}}\right]$. This leads to the following optimal transport GAN formulation with the entropy regularization, or for simplicity, the *entropic GAN* formulation:

$$\min_{\mathbf{G}\in\mathcal{G}} \min_{\mathbb{P}_{Y,\hat{Y}}} \mathbb{E}\left[\ell(Y,\hat{Y})\right] - \lambda H\left(\mathbb{P}_{Y,\hat{Y}}\right), \tag{2.3}$$

where λ is the regularization parameter.

There are two approaches to solve the optimization problem 2.3. The first approach uses an iterative method to solve the *min-min* formulation (Genevay et al., 2017). Another approach is to solve an equivelent *min-max* formulation by writing the dual of the inner minimization part (Seguy et al., 2017; Sanjabi et al., 2018). The latter is often referred to as a GAN formulation since the min-max optimization is over a set of generator functions and a set of discriminator functions. We will explain these computational approaches in more details in Section 3.

In the following, we present an explicit probability model for entropic GANs under which they can be viewed as maximizing a lower bound on average sample likelihoods.

Theorem 1 *Let the loss function be shift invariant, i.e.,* $\ell(\mathbf{y},\hat{\mathbf{y}}) = h(\mathbf{y}-\hat{\mathbf{y}})$. *Let*

$$f_{Y|X=\mathbf{x}}(\mathbf{y}) = C\exp(-\ell(\mathbf{y},\mathbf{G}(\mathbf{x}))/\lambda), \tag{2.4}$$

be an explicit probability model for Y *given* $X=\mathbf{x}$ *for a well-defined normalization*

$$C := \frac{1}{\int_{\mathbf{y}\in\mathbb{R}^d}\exp(-\ell(\mathbf{y},G(\mathbf{x}))/\lambda)}. \tag{2.5}$$

Then, we have

$$\underbrace{\mathbb{E}_{\mathbb{P}_Y}\left[\log f_Y(Y)\right]}_{\text{ave. sample likelihoods}} \geq -\frac{1}{\lambda}\underbrace{\left\{\mathbb{E}_{\mathbb{P}_{Y,\hat{Y}}}\left[\ell(Y,\hat{Y})\right] - \lambda H\left(\mathbb{P}_{Y,\hat{Y}}\right)\right\}}_{\text{entropic GAN objective}} + constants. \tag{2.6}$$

In words, the entropic GAN maximizes a lower bound on sample likelihoods according to the explicit probability model of equation 2.4.

The proof of this theorem is presented in Section A. This result has a similar flavor to that of VAEs (Makhzani et al., 2015; Rosca et al., 2017; Tolstikhin et al., 2017; Mescheder et al., 2017) where a generative model is computed by maximizing a lower bound on sample likelihoods.

Having a shift invariant loss function is critical for Theorem 1 as this makes the normalization term C independent from $\mathbf{G}$ and $\mathbf{x}$ (to see this, one can define $\mathbf{y}' := \mathbf{y} - \mathbf{G}(\mathbf{x})$ in equation A.4). The most standard OT GAN loss functions such as the L_2 for WGAN (Arjovsky et al., 2017) and the quadratic loss for W2GAN (Feizi et al., 2017) satisfy this property.

One can further simplify this result by considering specific loss functions. For example, we have the following result for the entropic GAN with the quadratic loss function.

Corollary 1 *Let* $\ell(\mathbf{y},\hat{\mathbf{y}}) = \|\mathbf{y}-\hat{\mathbf{y}}\|^2/2$. *Then,* $f_{Y|X=\mathbf{x}}(.)$ *of equation 2.4 corresponds to the multivariate Gaussian density function and* $C = \frac{1}{\sqrt{(2\pi\lambda)^d}}$. *In this case, the constant term in equation A.4 is equal to* $-\log(m) - d\log(2\pi\lambda)/2 - r/2 - \log(2\pi)/2$.

Let $\mathbf{G}^*$ and $\mathbb{P}^*_{Y,X}$ be optimal solutions of an entropic GAN optimization 2.3 (note that the optimal coupling can be computed efficiently for entropic GAN using equation 3.7). Let $\mathbf{y}^{\text{test}}$ be a newly

observed sample. An important question is what the likelihood of this sample is given the trained generative model. Using the explicit probability model of equation 2.4 and the result of Theorem 1, we can (approximately) compute sample likelihoods using the trained generative model. We explain this result in the following corollary.

Corollary 2 *Let $\mathbf{G}^*$ and $\mathbb{P}^*_{Y,\hat{Y}}$ (or, alternatively $\mathbb{P}^*_{Y,X}$) be optimal solutions of the entropic GAN equation 2.3. Let $\mathbf{y}^{test}$ be a new observed sample. We have*

$$\log f_Y(\mathbf{y}^{test}) \geq -\frac{1}{\lambda}\left\{\mathbb{E}_{\mathbb{P}^*_{X|Y=\mathbf{y}^{test}}}\left[\ell(\mathbf{y}^{test}, \mathbf{G}^*(\mathbf{x}))\right] - \lambda H\left(\mathbb{P}^*_{X|Y=\mathbf{y}^{test}}\right)\right\} + \mathbb{E}_{\mathbb{P}^*_{X|Y=\mathbf{y}^{test}}}\left[-\frac{\|\mathbf{x}\|^2}{2}\right] + constants. \tag{2.7}$$

The inequality becomes tight iff $D_{\mathrm{KL}}\left(\mathbb{P}^*_{X|Y=\mathbf{y}^{test}} \| f_{X|Y=\mathbf{y}^{test}}\right) = 0$ where $D_{\mathrm{KL}}(.\|.)$ is the Kullback-Leibler divergence between two distributions.

3 GAN's Dual Formulation

In this section, we discuss dual formulations for OT GAN equation 2.2 and entropic GAN equation 2.3 optimizations. These dual formulations are *min-max* optimizations over two function classes, namely the generator and the discriminator. Often local search methods such as alternating gradient descent (GD) are used to compute a solution for these min-max optimizations.

First, we discuss the dual formulation of OT GAN optimization equation 2.2. Using the duality of the inner minimization, which is a linear program, we can re-write optimization equation 2.2 as follows (Villani, 2008):

$$\min_{\mathbf{G}\in\mathcal{G}} \max_{\mathbf{D}_1,\mathbf{D}_2} \mathbb{E}\left[\mathbf{D}_1(Y)\right] - \mathbb{E}\left[\mathbf{D}_2(\mathbf{G}(X))\right], \tag{3.1}$$

where $\mathbf{D}_1(\mathbf{y}) - \mathbf{D}_2(\hat{\mathbf{y}}) \leq \ell(\mathbf{y}, \hat{\mathbf{y}})$ for all $(\mathbf{y}, \hat{\mathbf{y}})$. The maximization is over two sets of functions $\mathbf{D}_1$ and $\mathbf{D}_2$ which are coupled using the loss function. Using the Kantorovich duality Villani (2008), we can further simplify this optimization as follows:

$$\min_{\mathbf{G}\in\mathcal{G}} \max_{\mathbf{D}:\ell-\text{convex}} \mathbb{E}\left[\mathbf{D}(Y)\right] - \mathbb{E}\left[\mathbf{D}^{(\ell)}(\mathbf{G}(X))\right], \tag{3.2}$$

where $\mathbf{D}^{(\ell)}(\hat{Y}) := \inf_Y \ell(Y, \hat{Y}) + \mathbf{D}(Y)$ is the ℓ-conjugate function of $\mathbf{D}(.)$ and $\mathbf{D}$ is restricted to ℓ-convex functions (Villani, 2008). The above optimization provides a general formulation for OT GANs. If the loss function is $\|.\|_2$, then the optimal transport distance is referred to as the first order Wasserstein distance. In this case, the min-max optimization equation 3.2 simplifies to the following optimization (Arjovsky et al., 2017):

$$\min_{\mathbf{G}\in\mathcal{G}} \max_{\mathbf{D}:1\text{-Lip}} \mathbb{E}\left[\mathbf{D}(Y)\right] - \mathbb{E}\left[\mathbf{D}(\mathbf{G}(X))\right]. \tag{3.3}$$

This is often referred to as Wasserstein GAN, or simply WGAN (Arjovsky et al., 2017). If the loss function is quadratic, then the OT optimization is referred to as the quadratic GAN (or, W2GAN) (Feizi et al., 2017).

Similarly, the dual formulation of the entropic GAN equation 2.3 can be written as the following optimization (Cuturi, 2013; Seguy et al., 2017) [1]:

$$\min_{\mathbf{G}\in\mathcal{G}} \max_{\mathbf{D}_1,\mathbf{D}_2} \mathbb{E}\left[\mathbf{D}_1(Y)\right] - \mathbb{E}\left[\mathbf{D}_2(\mathbf{G}(X))\right] - \lambda\mathbb{E}_{\mathbb{P}_Y\times\mathbb{P}_{\hat{Y}}}\left[\exp\left(v(\mathbf{y}, \hat{\mathbf{y}})/\lambda\right)\right], \tag{3.4}$$

where

$$v(\mathbf{y}, \hat{\mathbf{y}}) := \mathbf{D}_1(\mathbf{y}) - \mathbf{D}_2(\hat{\mathbf{y}}) - \ell(\mathbf{y}, \hat{\mathbf{y}}). \tag{3.5}$$

Note that the hard constraint of optimization equation 3.1 is being replaced by a soft constraint in optimization equation 3.2. In this case, optimal primal variables $\mathbb{P}^*_{Y,\hat{Y}}$ can be computed according to the following lemma (Seguy et al., 2017):

[1] Note that optimization equation 3.4 is dual of optimization equation 2.3 when the terms $\lambda H(\mathbb{P}_Y) + \lambda H(\mathbb{P}_{\hat{Y}})$ have been added to its objective. Since we have assumed that $\mathbf{G}$ is injective, these terms are constants and thus can be ignored from the optimization objective without loss of generality.

Lemma 1 *Let $\mathbf{D}_1^*$ and $\mathbf{D}_2^*$ be the optimal discriminator functions for a given generator function $\mathbf{G}$ according to optimization equation 3.4. Let*

$$v^*(\mathbf{y}, \hat{\mathbf{y}}) := \mathbf{D}_1^*(\mathbf{y}) - \mathbf{D}_2^*(\hat{\mathbf{y}}) - \ell(\mathbf{y}, \hat{\mathbf{y}}). \tag{3.6}$$

Then,

$$\mathbb{P}^*_{Y,\hat{Y}}(\mathbf{y}, \hat{\mathbf{y}}) = \mathbb{P}_Y(\mathbf{y})\mathbb{P}_{\hat{Y}}(\hat{\mathbf{y}}) \exp\left(v^*\left(\mathbf{y}, \hat{\mathbf{y}}\right)/\lambda\right). \tag{3.7}$$

This lemma is important for our results since it provides an efficient way to compute the optimal coupling between real and generative variables (i.e. $P^*_{Y,\hat{Y}}$) using the optimal generator ($\mathbf{G}^*$) and discriminators ($\mathbf{D}_1^*$ and $\mathbf{D}_2^*$) of optimization equation 3.4. It is worth to note that without the entropy regularization term, computing the optimal coupling in OT GAN using the optimal generator and discriminator functions is not straightforward in general (unless in some special cases such as that of the W2GAN (Villani, 2008; Feizi et al., 2017)). This is another additional computational benefit of using entropic GAN. We use the algorithm presented in Sanjabi et al. (2018) to solve optimization equation 3.4.

4 Experimental Results

In this section, we supplement our theoretical results with experimental validations. One of the main objectives of our work is to provide a framework to compute sample likelihoods in GANs. Such likelihood statistics can then be used in several statistical inference applications that we discuss in Section 5. With a trained entropic WGAN, the likelihood of a test sample can be lower-bounded using Corollary 2. As shown in Lemma 1, WGAN with entropy regularization provides a closed-form solution to the conditional density of the latent variable. From equation 3.7, we have

$$\mathbb{P}^*_{\hat{Y}|Y=\mathbf{y}^{\text{test}}}(\hat{\mathbf{y}}) \propto \mathbb{P}_{\hat{Y}}(\hat{\mathbf{y}}) \exp\left(v^*\left(\mathbf{y}^{\text{test}}, \hat{\mathbf{y}}\right)/\lambda\right).$$

By change of variables (and under the assumption that the generator is injective), we have

$$\mathbb{P}^*_{X|Y=\mathbf{y}^{\text{test}}}(\mathbf{x}) \propto \mathbb{P}_X(\mathbf{x}) \exp\left(v^*\left(\mathbf{y}^{\text{test}}, G^*(\mathbf{x})\right)/\lambda\right). \tag{4.1}$$

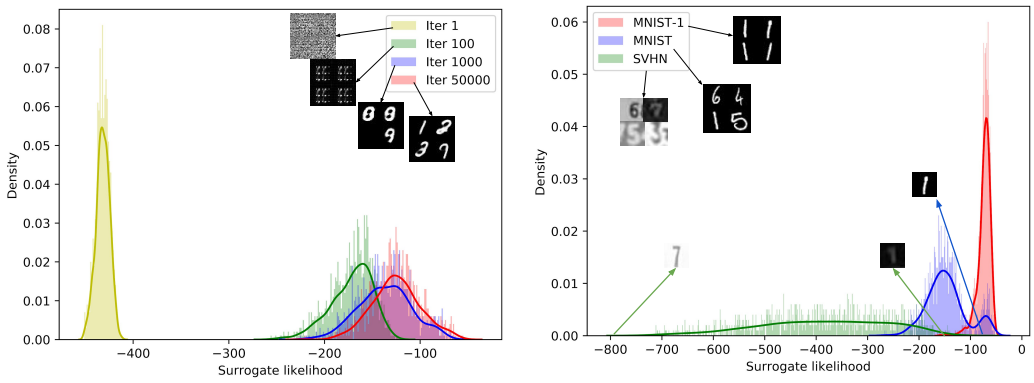
In order to compute our proposed likelihood surrogate of Corollary 2, we need to draw samples from the distribution $\mathbb{P}^*_{X|Y=\mathbf{y}^{\text{test}}}(\mathbf{x})$. One approach is to use a Markov chain Monte Carlo (MCMC) method to sample from this distribution. In our experiments, however, we found that MCMC demonstrates poor performance owing to the high dimensional nature of X. A similar issue with MCMC has been reported for VAEs in Kingma & Welling (2013). Thus, we use a different estimator to compute the likelihood surrogate which provides a better exploration of the latent space. We present our sampling procedure in Alg. 1.

Algorithm 1 Estimating sample likelihoods in GANs

1: Sample N points $\mathbf{x}_i \overset{i.i.d}{\sim} P_X(\mathbf{x})$
2: Compute $u_i := \mathbb{P}_X(\mathbf{x}_i) \exp\left(v^*\left(\mathbf{y}^{\text{test}}, G^*(\mathbf{x}_i)\right)/\lambda\right)$
3: Normalize to get probabilities $p_i = \frac{u_i}{\sum_{i=1}^N u_i}$
4: Compute $L = -\frac{1}{\lambda}\left[\sum_{i=1}^N p_i l(\mathbf{y}^{\text{test}}, G^*(\mathbf{x}_i)) + \lambda \sum_{i=1}^N p_i \log p_i\right] - \sum_{i=1}^N p_i \frac{\|\mathbf{x}_i\|^2}{2}$
5: Return L

4.1 Likelihood Evolution in GAN's Training

In experiments of this section, we study how sample likelihoods vary as GAN's training progresses. An entropic WGAN is first trained on MNIST dataset. Then, we randomly choose $n_s = 1,000$ samples from MNIST test-set to compute the surrogate likelihoods using Algorithm 1 at different training iterations. We expect sample likelihoods to increase over training iterations as the generative model improves. A proper surrogate likelihood function should capture this trend. Fig. 2a demonstrates the evolution of sample likelihood distributions at different training iterations of the entropic WGAN. At iteration 1, surrogate likelihood values are very low as GAN's generated images are merely random noise. The likelihood distribution shifts towards high values during the training and saturates beyond a point. Details of this experiment are presented in Appendix C.



(a) (b)

Figure 2: (a) Distributions of surrogate sample likelihoods at different iterations of entropic WGAN's training using MNIST dataset. (b) Distributions of surrogate sample likelihoods of MNIST, MNIST-1 and SVHM datasets using a GAN trained on MNIST-1.

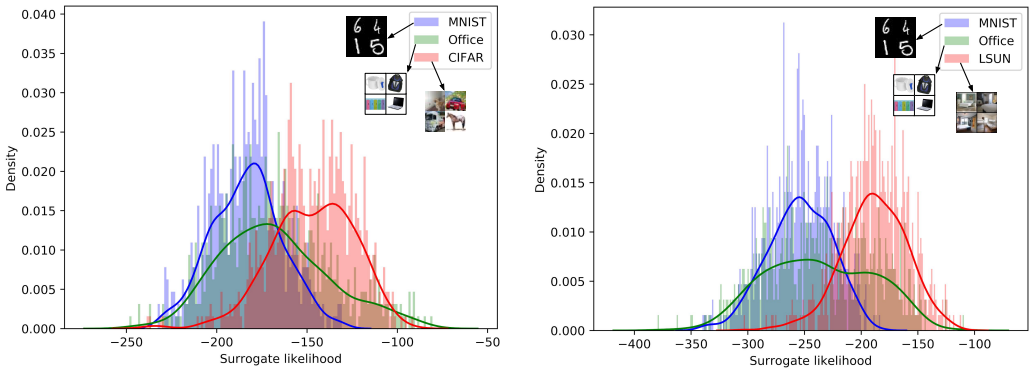
4.2 Likelihood Comparison Across different datasets

The previous experiment studies how our likelihood estimate improves over the iterations for samples from the same dataset. In this section, we perform experiments across different datasets. An entropic WGAN is first trained on a subset of samples from the MNIST dataset containing digit 1 (which we call the MNIST-1 dataset). With this trained model, likelihood estimates are computed for (1) 1,000 samples from the entire MNIST dataset, and (2) 1,000 samples from the Street View House Numbers (SVHN) dataset (Netzer et al., 2011) (Fig. 2b). We note that highest likelihood estimates are obtained for samples from MNIST-1 dataset, the same dataset used in GAN's training. Moreover, the likelihood distribution for the MNIST dataset is bimodal with one mode peaking inline with the MNIST-1 mode. Samples from this mode correspond to digit 1 in the MNIST dataset. The other mode, which is the dominant mode, contains the rest of the digits and has relatively low likelihood estimates. The SVHN dataset, on the other hand, has much lower likelihood estimates than that of MNIST as its distribution is significantly different than that of MNIST images. Furthermore, we observe that the likelihood distribution of SVHN samples has a large spread (variance). This is because samples of the SVHN dataset is more diverse with varying backgrounds and styles than samples from MNIST. We note that SVHN samples with high likelihood estimates correspond to images that are similar to MNIST digits, while samples with low scores are very different than MNIST samples. Details of this experiment are presented in Appendix C.

4.3 Approximate Likelihood Computation in Un-regularized GANs

Most standard GAN architectures do not have the entropy regularization. Likelihood lower bounds of Theorem 1 and Corollary 2, however, hold even for those GANs as long as we obtain the optimal coupling $\mathbb{P}^*_{Y,\hat{Y}}$ in addition to the optimal generator $\mathbf{G}^*$ from GAN's training. Computation of optimal coupling $\mathbb{P}^*_{Y,\hat{Y}}$ from the dual formulation of OT GAN can be done when the loss function is quadratic (Feizi et al., 2017). In this case, the gradient of the optimal discriminator provides the optimal coupling between Y and $\hat{Y}$ (Villani, 2008) (see Lemma. 2 in Appendix A).

For a general GAN architecture, however, the exact computation of optimal coupling $\mathbb{P}^*_{Y,\hat{Y}}$ may be difficult. One sensible approximation is to couple $Y = \mathbf{y}^{\text{test}}$ with a single latent sample $\tilde{\mathbf{x}}$ (we are assuming the conditional distribution $\mathbb{P}^*_{X|Y=\mathbf{y}^{\text{test}}}$ is an impulse function). To compute $\tilde{\mathbf{x}}$ corresponding to a $\mathbf{y}^{\text{test}}$, we sample k latent samples $\{\mathbf{x}'_i\}_{i=1}^k$ and select the $\mathbf{x}'_i$ whose $\mathbf{G}^*(\mathbf{x}'_i)$ is closest to $\mathbf{y}^{\text{test}}$. This heuristic takes into account both the likelihood of the latent variable as well as the distance between $\mathbf{y}^{\text{test}}$ and the model (similarly to equation 3.7). We can then use Corollary 2 to approximate sample likelihoods for various GAN architectures.



(a) (b)

Figure 3: (a) Sample likelihood estimates of MNIST, Office and CIFAR datasets using a GAN trained on the CIFAR dataset. (b) Sample likelihood estimates of MNIST, Office and LSUN datasets using a GAN trained on the LSUN dataset.

We use this approach to compute likelihood estimates for the CIFAR-10 (Krizhevsky, 2009) and LSUN-Bedrooms (Yu et al., 2015) datasets. For CIFAR-10, we train DCGAN while for LSUN, we train WGAN (details of these experiments can be found in Appendices **??** and **??**, respectively).

Fig. 3a demonstrates sample likelihood estimates of different datasets using a GAN trained on the CIFAR dataset. Likelihoods assigned to samples from MNIST and Office datasets are lower than that of the CIFAR dataset. Samples from the Office dataset are assigned to higher likelihood values than samples of the MNIST dataset. We note that the Office dataset is indeed more similar to the CIFAR dataset than MNIST. A similar experiment has been repeated for LSUN-Bedrooms (Yu et al., 2015) dataset. Similar performance trends have been observed in this experiment (Fig. 3b).

5 Conclusion

In this paper, we have provided a statistical framework to understand GANs. Our main result shows that the entropic GAN optimization can be viewed as maximization of a variational lower-bound on average log-likelihoods, an approach that VAEs are based upon. This result makes a connection between two most-popular generative models, namely GANs and VAEs. More importantly, our result constructs an explicit probability model for GANs that can be used to compute a lower-bound on sample likelihoods. Our experimental results on various datasets demonstrate that this likelihood surrogate can be a good approximation of the true likelihood function. Although in this paper we mainly focus on understanding the behavior of the sample likelihood surrogate in different datasets, the proposed statistical framework of GANs can be used in various applications. For example, our proposed likelihood surrogate can be used as a quantitative measure to evaluate the performance of different GAN architectures in various datasets, it can be used to quantify the domain shifts across different datasets, it can be used to select a proper generator class for a given dataset, it can be used to detect outlier samples, it can be used in statistical tests such as hypothesis testing, etc. We leave exploring these directions for future work.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pp. 2292–2300, 2013.

Soheil Feizi, Changho Suh, Fei Xia, and David Tse. Understanding GANs: the LQG setting. *arXiv preprint arXiv:1710.10793*, 2017.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Sinkhorn-autodiff: Tractable wasserstein learning of generative models. *arXiv preprint arXiv:1706.00292*, 2017.

Arsham Ghahramani, Fiona M Watt, and Nicholas M Luscombe. Generative adversarial networks uncover epidermal regulators and predict single cell perturbations. *bioRxiv*, pp. 262501, 2018.

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.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. *arXiv preprint arXiv:1704.00028*, 2017.

Xin Guo, Johnny Hong, Tianyi Lin, and Nan Yang. Relaxed wasserstein with applications to GANs. *arXiv preprint arXiv:1705.07164*, 2017.

Noriaki Hirose, Amir Sadeghian, Patrick Goebel, and Silvio Savarese. To go or not to go? a near unsupervised learning approach for robot navigation. *arXiv preprint arXiv:1709.05439*, 2017.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. URL http://arxiv.org/abs/1710.10196.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

Hung-yi Lee and Yu Tsao. Generative adversarial network and its applications to speech signal and natural language processing. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*, 2017.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 5, 2011.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pp. 271–279, 2016.

Henning Petzka, Asja Fischer, and Denis Lukovnicov. On the regularization of Wasserstein GANs. *arXiv preprint arXiv:1709.08894*, 2017.

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

Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017.

Maziar Sanjabi, Jimmy Ba, Meisam Razaviyayn, and Jason D Lee. Solving approximate Wasserstein GANs to stationarity. *arXiv preprint arXiv:1802.08249*, 2018.

Eder Santana and George Hotz. Learning a driving simulator. *arXiv preprint arXiv:1608.01230*, 2016.

Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation. *arXiv preprint arXiv:1711.02283*, 2017.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

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

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

APPENDIX A PROOFS

Using the Baye's rule, one can compute the log-likelihood of an observed sample $\mathbf{y}$ as follows:

$$\begin{aligned}\log f_Y(\mathbf{y}) &= \log f_{Y|X=\mathbf{x}}(\mathbf{y}) + \log f_X(\mathbf{x}) - \log f_{X|Y=\mathbf{y}}(\mathbf{x}) \\ &= \log C(\mathbf{x}) - \ell(\mathbf{y}, G(\mathbf{x})) - \log\sqrt{2\pi} - \frac{\|\mathbf{x}\|^2}{2} - \log f_{X|Y=\mathbf{y}}(\mathbf{x}),\end{aligned} \tag{A.1}$$

where the second step follows from equation 2.4.

Consider a joint density function $\mathbb{P}_{X,Y}$ such that its marginal distributions match $\mathbb{P}_X$ and $\mathbb{P}_Y$. Note that the equation equation A.1 is true for every $\mathbf{x}$. Thus, we can take the expectation of both sides with respect to a distribution $\mathbb{P}_{X|Y=\mathbf{y}}$. This leads to the following equation:

$$\begin{aligned}\log f_Y(\mathbf{y}) =& \mathbb{E}_{\mathbb{P}_{X|Y=\mathbf{y}}}\left[-\ell(\mathbf{y}, \mathbf{G}(\mathbf{x}))/\lambda + \log C(\mathbf{x}) - \frac{1}{2}\log 2\pi - \frac{\|\mathbf{x}\|^2}{2} - \log f_{X|Y=\mathbf{y}}(\mathbf{x})\right] \\ =& \mathbb{E}_{\mathbb{P}_{X|Y=\mathbf{y}}}\Bigg[-\ell(\mathbf{y}, \mathbf{G}(\mathbf{x}))/\lambda + \log C(\mathbf{x}) - \frac{1}{2}\log 2\pi - \frac{\|\mathbf{x}\|^2}{2} - \log f_{X|Y=\mathbf{y}}(\mathbf{x}) \\ &+ \log\left(\mathbb{P}_{X|Y=\mathbf{y}}(\mathbf{x})\right) - \log\left(\mathbb{P}_{X|Y=\mathbf{y}}(\mathbf{x})\right)\Bigg] \\ =& -\mathbb{E}_{\mathbb{P}_{X|Y=\mathbf{y}}}\left[\ell(\mathbf{y}, \mathbf{G}(\mathbf{x}))/\lambda\right] - \frac{1}{2}\log 2\pi + \mathbb{E}_{\mathbb{P}_{X|Y=\mathbf{y}}}\left[\log C(\mathbf{x}) - \frac{\|\mathbf{x}\|^2}{2}\right] \\ &+ \mathrm{KL}\left(\mathbb{P}_{X|Y=\mathbf{y}} \| f_{X|Y=\mathbf{y}}\right) + H\left(\mathbb{P}_{X|Y=\mathbf{y}}\right),\end{aligned} \tag{A.2}$$

where $H(.)$ is the Shannon-entropy function.

Next we take the expectation of both sides with respect to $\mathbb{P}_Y$:

$$\begin{aligned}\mathbb{E}\left[\log f_Y(Y)\right] =& -\frac{1}{\lambda}\mathbb{E}_{\mathbb{P}_{X,Y}}\left[\ell(\mathbf{y}, G(\mathbf{x}))\right] - \frac{1}{2}\log 2\pi + \mathbb{E}_{f_X}\left[\log C(\mathbf{x}) - \frac{\|\mathbf{x}\|^2}{2}\right] \\ &+ \mathbb{E}_{\mathbb{P}_Y}\left[\mathrm{KL}\left(\mathbb{P}_{X|Y=\mathbf{y}} \| f_{X|Y=\mathbf{y}}\right)\right] + H\left(\mathbb{P}_{X,Y}\right) - H\left(\mathbb{P}_Y\right).\end{aligned} \tag{A.3}$$

Here, we replaced the expectation over $\mathbb{P}_X$ with the expectation over f_X since one can generate an arbitrarily large number of samples from the generator. Since the KL divergence is always non-negative, we have

$$\underbrace{\mathbb{E}\left[\log f_Y(Y)\right]}_{\text{sample likelihood}} \geq -\frac{1}{\lambda}\underbrace{\left\{\mathbb{E}_{\mathbb{P}_{X,Y}}\left[\ell(\mathbf{y}, \mathbf{G}(\mathbf{x}))\right] - \lambda H\left(\mathbb{P}_{X,Y}\right)\right\}}_{\text{GAN objective with entropy regularizer}} + \mathbb{E}_{f_X}\left[\log C(\mathbf{x})\right] - \log(m) - \frac{r + \log 2\pi}{2} \tag{A.4}$$

This inequality is true for every $\mathbb{P}_{X,Y}$ satisfying the marginal conditions. Thus, similar to VAEs, we can pick $\mathbb{P}_{X,Y}$ to maximize the lower bound on average sample log-likelihoods. This leads to the entropic GAN optimization equation 2.3.

Lemma 2 *Let $\mathbb{P}_Y$ be absolutely continuous whose support contained in a convex set in $\mathbb{R}^d$. Let $\mathbf{D}^{opt}$ be the optimal discriminator for a given generator $\mathbf{G}$ in W2GAN. This solution is unique. Moreover, we have*

$$\nabla \mathbf{D}^{opt}(Y) \overset{dist}{=} \mathbf{G}(X), \tag{A.5}$$

where $\overset{dist}{=}$ means matching distributions.

Appendix B Sinkhorn loss

In practice, it has been observed that a slightly modified version of the entropic GAN demonstrates improved computational properties (Genevay et al., 2017; Sanjabi et al., 2018). We explain this modification in this section. Let

$$W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) := \min_{\mathbb{P}_{Y,\hat{Y}}} \mathbb{E}\left[\ell(Y,\hat{Y})\right] + \lambda D_{\mathrm{KL}}\left(\mathbb{P}_{Y,\hat{Y}}\right), \tag{B.1}$$

where $D_{\mathrm{KL}}(.\|.)$ is the KullbackLeibler divergence. Note that the object of this optimization differs from that of the entropic GAN optimization equation 2.3 by a constant term $\lambda H(\mathbb{P}_Y) + \lambda H(\mathbb{P}_{\hat{Y}})$. A sinkhorn distance function is then defined as (Genevay et al., 2017):

$$\bar{W}_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) := 2W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) - W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_Y) - W_{\ell,\lambda}(\mathbb{P}_{\hat{Y}}, \mathbb{P}_{\hat{Y}}). \tag{B.2}$$

$\bar{W}$ is called the Sinkhorn loss function. Reference Genevay et al. (2017) has shown that as $\lambda \to 0$, $\bar{W}_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}})$ approaches $W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}})$. For a general λ, we have the following upper and lower bounds:

Lemma 3 *For a given $\lambda > 0$, we have*

$$\bar{W}_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) \le 2W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) \le \bar{W}_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) + \lambda H(\mathbb{P}_Y) + \lambda H(\mathbb{P}_{\hat{Y}}). \tag{B.3}$$

Proof From the definition equation B.2, we have $W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) :\ge \bar{W}_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}})/2$. Moreover, since $W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_Y) \le H(\mathbb{P}_Y)$ (this can be seen by using an identity coupling as a feasible solution for optimization equation B.1) and similarly $W_{\ell,\lambda}(\mathbb{P}_{\hat{Y}}, \mathbb{P}_{\hat{Y}}) \le H(\mathbb{P}_{\hat{Y}})$, we have $W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) \le \bar{W}_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}})/2 + \lambda/2 H(\mathbb{P}_Y) + \lambda/2 H(\mathbb{P}_{\hat{Y}})$. ■

Since $H(\mathbb{P}_Y) + H(\mathbb{P}_{\hat{Y}})$ is constant in our setup, optimizing the GAN with the Sinkhorn loss is equivalent to optimizing the entropic GAN. So, our likelihood estimation framework can be used with models trained using SInkhorn loss as well. This is particularly important from a practical standpoint as training models with Sinkhorn loss tends to be more stable in practice.

Appendix C Training entropic GANs

In this section, we discuss how WGANs with entropic regularization is trained. As discussed in Section 3, the dual of the entropic GAN formulation can be written as

$$\min_{\mathbf{G}\in\mathcal{G}} \max_{\mathbf{D}_1,\mathbf{D}_2} \mathbb{E}\left[\mathbf{D}_1(Y)\right] - \mathbb{E}\left[\mathbf{D}_2(\mathbf{G}(X))\right] - \lambda \mathbb{E}_{\mathbb{P}_Y\times\mathbb{P}_{\hat{Y}}}\left[\exp\left(v(\mathbf{y},\hat{\mathbf{y}})/\lambda\right)\right],$$

where

$$v(\mathbf{y},\hat{\mathbf{y}}) := \mathbf{D}_1(\mathbf{y}) - \mathbf{D}_2(\hat{\mathbf{y}}) - \ell(\mathbf{y},\hat{\mathbf{y}}).$$

We can optimize this min-max problem using alternating optimization. A better approach would be to take into account the smoothness introduced in the problem due to the entropic regularizer, and solve the generator problem to stationarity using first-order methods. Please refer to Sanjabi et al. (2018) for more details. In all our experiments, we use Algorithm 1 of Sanjabi et al. (2018) to train our GAN model.

C.1 TRAINING DETAILS FOR MNIST DIGITS

MNIST dataset constains 28×28 grayscale images. As a pre-processing step, all images were resized in the range $[0, 1]$. The Discriminator and the Generator architectures used in our experiments are given in Tables. 1,2. Note that the dual formulation of GANs employ two discriminators - D_1 and D_2, and we use the same architecture for both. The hyperparameter details are given in Table 3. Some sample generations are shown in Fig. 4

C.2 TRAINING DETAILS FOR CIFAR DATASET

We trained a DCGAN model on CIFAR dataset using the discriminator and generator architecture used in Radford et al. (2015). The hyperparamer details are mentioned in Table. 4. Some sample generations are provided in Figure 6

Table 1: Generator architecture

Layer	Output size	Filters
Input	128	-
Fully connected	4.4.256	$128 \rightarrow 256$
Reshape	$256 \times 4 \times 4$	-
BatchNorm+ReLU	$256 \times 4 \times 4$	-
Deconv2d (5×5, str 2)	$128 \times 8 \times 8$	$256 \rightarrow 128$
BatchNorm+ReLU	$128 \times 8 \times 8$	-
Remove border row and col.	$128 \times 7 \times 7$	-
Deconv2d (5×5, str 2)	$64 \times 14 \times 14$	$128 \rightarrow 64$
BatchNorm+ReLU	$128 \times 8 \times 8$	-
Deconv2d (5×5, str 2)	$1 \times 28 \times 28$	$64 \rightarrow 1$
Sigmoid	$1 \times 28 \times 28$	-

Table 2: Discriminator architecture

Layer	Output size	Filters
Input	$1 \times 28 \times 28$	-
Conv2D(5×5, str 2)	$32 \times 14 \times 14$	$1 \rightarrow 32$
LeakyReLU(0.2)	$32 \times 14 \times 14$	-
Conv2D(5×5, str 2)	$64 \times 7 \times 7$	$32 \rightarrow 64$
LeakyReLU(0.2)	$64 \times 7 \times 7$	-
Conv2d (5×5, str 2)	$128 \times 4 \times 4$	$64 \rightarrow 128$
LeakyRelU(0.2)	$128 \times 4 \times 4$	-
Reshape	128.4.4	-
Fully connected	1	$2048 \rightarrow 1$

Table 3: Hyper-parameter details for MNIST experiment

Parameter	Config
λ	5
Generator learning rate	0.0002
Discriminator learning rate	0.0002
Batch size	100
Optimizer	Adam
Optimizer params	$\beta_1 = 0.5$, $\beta_2 = 0.9$
Number of critic iters / gen iter	5
Number of training iterations	10000

Figure 4: Samples generated by Entropic GAN trained on MNIST

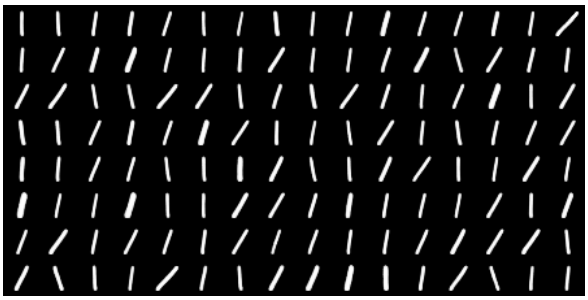


Figure 5: Samples generated by Entropic GAN trained on MNIST-1 dataset

C.3 Training details for LSUN-Bedrooms dataset

We trained a WGAN model on LSUN-Bedrooms dataset with DCGAN architectures for generator and discriminator networks (Arjovsky et al., 2017). The hyperparameter details are given in Table. 5, and some sample generations are provided in Fig. 7

Table 4: Hyper-parameter details for CIFAR-10 experiment

Parameter	Config
Generator learning rate	0.0002
Discriminator learning rate	0.0002
Batch size	64
Optimizer	Adam
Optimizer params	β_1 = 0.5, β_2 = 0.99
Number of training epochs	100

Table 5: Hyper-parameter details for LSUN-Bedrooms experiment

Parameter	Config
Generator learning rate	0.00005
Discriminator learning rate	0.00005
Clipping parameter c	0.01
Number of critic iters per gen iter	5
Batch size	64
Optimizer	RMSProp
Number of training iterations	70000

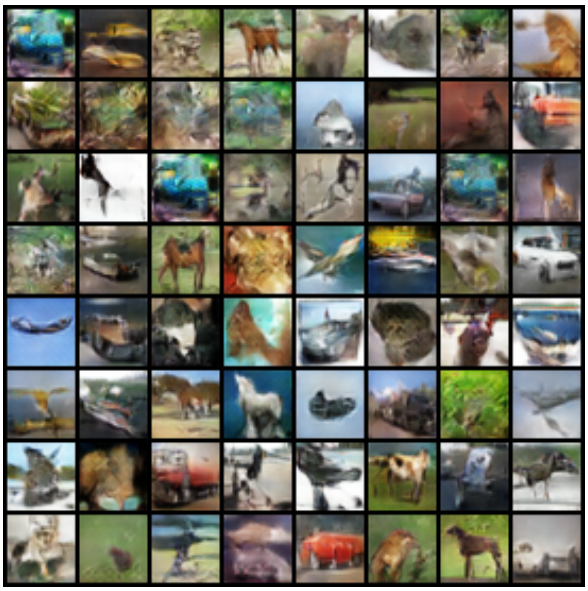


Figure 6: Samples generated by DCGAN model trained on CIFAR dataset

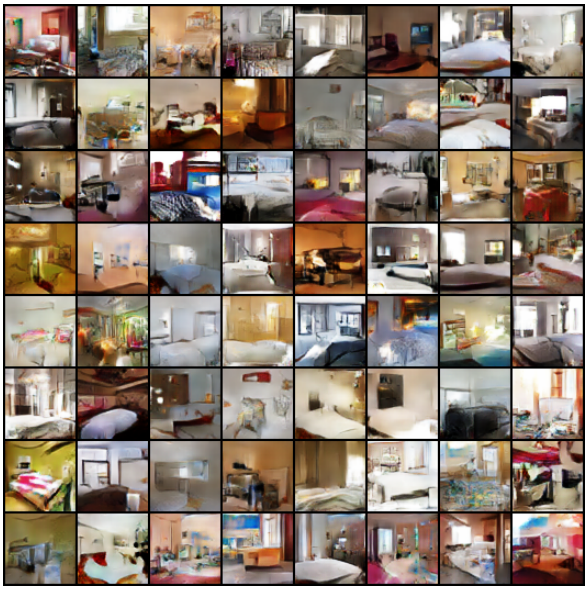


Figure 7: Samples generated by WGAN model trained on LSUN-Bedrooms dataset