Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models

Aditya Grover, Manik Dhar, Stefano Ermon
Computer Science Department
Stanford University
{adityag, dmanik, ermon}@cs.stanford.edu

Abstract

Adversarial learning of probabilistic models has recently emerged as a promising alternative to maximum likelihood. Implicit models such as generative adversarial networks (GAN) often generate better samples compared to explicit models trained by maximum likelihood. Yet, GANs sidestep the characterization of an explicit density which makes quantitative evaluations challenging. To bridge this gap, we propose Flow-GANs, a generative adversarial network for which we can perform *exact* likelihood evaluation, thus supporting both adversarial and maximum likelihood training. When trained adversarially, Flow-GANs generate high-quality samples but attain extremely poor log-likelihood scores, inferior even to a mixture model memorizing the training data; the opposite is true when trained by maximum likelihood. Results on MNIST and CIFAR-10 demonstrate that hybrid training can attain high held-out likelihoods while retaining visual fidelity in the generated samples.

1 Introduction

Highly expressive parametric models have enjoyed great success in supervised learning, where learning objectives and evaluation metrics are typically well-specified and easy to compute. On the other hand, the learning objective for unsupervised settings is less clear. At a fundamental level, the idea is to learn a generative model that minimizes some notion of divergence with respect to the data distribution. Minimizing the Kullback-Liebler divergence between the data distribution and the model, for instance, is equivalent to performing maximum likelihood estimation (MLE) on the observed data. Maximum likelihood estimators are asymptotically statistically efficient, and serve as natural objectives for learning *prescribed generative models* (Mohamed and Lakshminarayanan 2016).

In contrast, an alternate principle that has recently attracted much attention is based on adversarial learning, where the objective is to generate data indistinguishable from the training data. Adversarially learned models such as generative adversarial networks (GAN; (Goodfellow et al. 2014)) can sidestep specifying an explicit density for any data point and belong to the class of *implicit generative models* (Diggle and Gratton 1984).

The lack of characterization of an explicit density in GANs is however problematic for two reasons. Several application areas of deep generative models rely on density estimates; for instance, count based exploration strategies based on density estimation using generative models have recently achieved state-of-the-art performance on challenging reinforcement learning environments (Ostrovski et al. 2017). Secondly, it makes the quantitative evaluation of the generalization performance of such models challenging. The typical evaluation criteria based on ad-hoc sample quality metrics (Salimans et al. 2016; Che et al. 2017) do not address this issue since it is possible to generate good samples by memorizing the training data, or missing important modes of the distribution, or both (Theis, Oord, and Bethge 2016). Alternatively, density estimates based on approximate inference techniques such as annealed importance sampling (AIS; (Neal 2001; Wu et al. 2017)) and non-parameteric methods such as kernel density estimation (KDE; (Parzen 1962; Goodfellow et al. 2014)) are computationally slow and crucially rely on assumptions of a Gaussian observation model for the likelihood that could lead to misleading estimates as we shall demonstrate in this paper.

To sidestep the above issues, we propose Flow-GANs, a generative adversarial network with a normalizing flow generator. A Flow-GAN generator transforms a prior noise density into a model density through a sequence of invertible transformations. By using an invertible generator, Flow-GANs allow us to tractably evaluate *exact* likelihoods using the change-of-variables formula and perform *exact* posterior inference over the latent variables while still permitting efficient ancestral sampling, desirable properties of any probabilistic model that a typical GAN would not provide.

Using a Flow-GAN, we perform a principled quantitative comparison of maximum likelihood and adversarial learning on benchmark datasets viz. MNIST and CIFAR-10. While adversarial learning outperforms MLE on sample quality metrics as expected based on strong evidence in prior work, the log-likelihood estimates of adversarial learning are orders of magnitude worse than those of MLE. The difference is so stark that a simple Gaussian mixture model baseline outperforms adversarially learned models on *both* sample quality and held-out likelihoods. Our quantitative analysis reveals that the poor likelihoods of adversarial learning can be explained as a result of an ill-conditioned Jacobian matrix for the generator function suggesting a mode collapse, rather than overfitting to the training dataset.

To resolve the dichotomy of perceptually good-looking samples at the expense of held-out likelihoods in the case of adversarial learning (and vice versa in the case of MLE), we propose a hybrid objective that bridges implicit and prescribed learning by augmenting the adversarial training objective with an additional term corresponding to the log-likelihood of the observed data. While the hybrid objective achieves the intended effect of smoothly trading-off the two goals in the case of CIFAR-10, it has a regularizing effect on MNIST where it outperforms MLE and adversarial learning on both held-out likelihoods and sample quality metrics.

Overall, this paper makes the following contributions:

1. We propose Flow-GANs, a generative adversarial network with an invertible generator that can perform efficient ancestral sampling and exact likelihood evaluation.
2. We propose a hybrid learning objective for Flow-GANs that attains good log-likelihoods and generates high-quality samples on MNIST and CIFAR-10 datasets.
3. We demonstrate the limitations of AIS and KDE for log-likelihood evaluation and ranking of implicit models.
4. We analyze the singular value distribution for the Jacobian of the generator function to explain the low log-likelihoods observed due to adversarial learning.

2 Preliminaries

We begin with a review of maximum likelihood estimation and adversarial learning in the context of generative models. For ease of presentation, all distributions are w.r.t. any arbitrary $\mathbf{x} \in \mathbb{R}^d$, unless otherwise specified. We use upper-case to denote probability distributions and assume they all admit absolutely continuous densities (denoted by the corresponding lower-case notation) on a reference measure $\mathrm{d}\mathbf{x}$.

Consider the following setting for learning generative models. Given some data $X = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^m$ sampled i.i.d. from an unknown probability density p_{data}, we are interested in learning a probability density p_θ where θ denotes the parameters of a model. Given a parametric family of models $\mathcal{M}$, the typical approach to learn $\theta \in \mathcal{M}$ is to minimize a notion of divergence between P_{data} and P_θ. The choice of divergence and the optimization procedure dictate learning, leading to the following two objectives.

2.1 Maximum likelihood estimation

In maximum likelihood estimation (MLE), we minimize the Kullback-Liebler (KL) divergence between the data distribution and the model distribution. Formally, the learning objective can be expressed as:

$$\min_{\theta \in \mathcal{M}} KL(P_{\mathrm{data}}, P_\theta) = \mathbb{E}_{\mathbf{x}\sim P_{\mathrm{data}}}\left[\log \frac{p_{\mathrm{data}}(\mathbf{x})}{p_\theta(\mathbf{x})}\right]$$

Since p_{data} is independent of θ, the above optimization problem can be equivalently expressed as:

$$\max_{\theta \in \mathcal{M}} \mathbb{E}_{\mathbf{x}\sim P_{\mathrm{data}}}[\log p_\theta(\mathbf{x})] \qquad (1)$$

Hence, evaluating the learning objective for MLE in Eq. (1) requires the ability to evaluate the model density $p_\theta(\mathbf{x})$. Models that provide an explicit characterization of the likelihood function are referred to as prescribed generative models (Mohamed and Lakshminarayanan 2016).

2.2 Adversarial learning

A generative model can be learned to optimize divergence notions beyond the KL divergence. A large family of divergences can be conveniently expressed as:

$$\max_{\phi \in \mathcal{F}} \mathbb{E}_{\mathbf{x}\sim P_\theta}[h_\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim P_{\mathrm{data}}}\left[h'_\phi(\mathbf{x})\right] \qquad (2)$$

where $\mathcal{F}$ denotes a set of parameters, h_ϕ and h'_ϕ are appropriate real-valued functions parameterized by ϕ. Different choices of $\mathcal{F}$, h_ϕ and h'_ϕ can lead to a variety of f-divergences such as Jenson-Shannon divergence and integral probability metrics such as the Wasserstein distance. For instance, the GAN objective proposed by Goodfellow et al. (2014) can also be cast in the form of Eq. (2) below:

$$\max_{\phi \in \mathcal{F}} \mathbb{E}_{\mathbf{x}\sim P_\theta}[\log(1 - D_\phi(\mathbf{x}))] + \mathbb{E}_{\mathbf{x}\sim P_{\mathrm{data}}}[D_\phi(\mathbf{x})] \qquad (3)$$

where ϕ denotes the parameters of a neural network function D_ϕ. We refer the reader to (Nowozin, Cseke, and Tomioka 2016; Mescheder, Nowozin, and Geiger 2017b) for further details on other possible choices of divergences. Importantly, a Monte Carlo estimate of the objective in Eq. (2) requires only samples from the model. Hence, any model that allows tractable sampling can be used to evaluate the following minimax objective:

$$\min_{\theta \in \mathcal{M}} \max_{\phi \in \mathcal{F}} \mathbb{E}_{\mathbf{x}\sim P_\theta}[h_\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim P_{\mathrm{data}}}\left[h'_\phi(\mathbf{x})\right]. \qquad (4)$$

As a result, even differentiable *implicit models* which do not provide a characterization of the model likelihood[1] but allow tractable sampling can be learned adversarially by optimizing minimax objectives of the form given in Eq. (4).

2.3 Adversarial learning of latent variable models

From a statistical perspective, maximum likelihood estimators are statistically efficient asymptotically (under some conditions) and hence minimizing the KL divergence is a natural objective for many prescribed models (Huber 1967). However, not all models allow for a well-defined, tractable, and easy-to-optimize likelihood.

For example, exact likelihood evaluation and sampling are tractable in directed, fully observed models such as Bayesian networks and autoregressive models (Larochelle and Murray 2011; Oord, Kalchbrenner, and Kavukcuoglu 2016). Hence, they are usually trained by maximum likelihood. Undirected models, on the other hand, provide only unnormalized likelihoods and are sampled from using expensive Markov chains. Hence, they are usually learned by approximating the likelihood using methods such as contrastive divergence (Carreira-Perpinan and Hinton 2005) and pseudolikelihood (Besag 1977). The likelihood is generally intractable to compute in latent variable models (even directed

[1] This could be either due to computational intractability in evaluating likelihoods or because the likelihood is ill-defined.

ones) as it requires marginalization. These models are typically learned by optimizing a stochastic lower bound to the log-likelihood using variational Bayes approaches (Kingma and Welling 2014).

Directed latent variable models allow for efficient ancestral sampling and hence these models can also be trained using other divergences, *e.g.*, adversarially (Mescheder, Nowozin, and Geiger 2017a; Mao et al. 2017; Song, Zhao, and Ermon 2017). A popular class of latent variable models learned adversarially consist of generative adversarial networks (GAN; (Goodfellow et al. 2014)). GANs comprise of a pair of generator and discriminator networks. The generator $G_\theta : \mathbb{R}^k \to \mathbb{R}^d$ is a deterministic function differentiable with respect to the parameters θ. The function takes as input a source of randomness $\mathbf{z} \in \mathbb{R}^k$ sampled from a tractable prior density $p(\mathbf{z})$ and transforms it to a sample $G_\theta(\mathbf{z})$ through a forward pass. Evaluating likelihoods assigned by a GAN is challenging because the model density p_θ is specified only implicitly using the prior density $p(\mathbf{z})$ and the generator function G_θ. In fact, the likelihood for any data point is ill-defined (with respect to the Lesbegue measure over $\mathbb{R}^n$) if the prior distribution over $\mathbf{z}$ is defined over a support smaller than the support of the data distribution.

GANs are typically learned adversarially with the help of a discriminator network. The discriminator $D_\phi : \mathbb{R}^d \to \mathbb{R}$ is another real-valued function that is differentiable with respect to a set of parameters ϕ. Given the discriminator function, we can express the functions h and h' in Eq. (4) as compositions of D_ϕ with divergence-specific functions. For instance, the Wasserstein GAN (WGAN; (Arjovsky, Chintala, and Bottou 2017)) optimizes the following objective:

$$\min_\theta \max_{\phi \in \mathcal{F}} \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [D_\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}} [D_\phi(G_\theta(\mathbf{z}))] \quad (5)$$

where $\mathcal{F}$ is defined such that D_ϕ is 1-Lipschitz. Empirically, GANs generate excellent samples of natural images (Radford, Metz, and Chintala 2015), audio signals (Pascual, Bonafonte, and Serrà 2017), and of behaviors in imitation learning (Ho and Ermon 2016; Li, Song, and Ermon 2017).

3 Flow Generative Adversarial Networks

As discussed above, generative adversarial networks can tractably generate high-quality samples but have intractable or ill-defined likelihoods. Monte Carlo techniques such as AIS and non-parameteric density estimation methods such as KDE get around this by assuming a Gaussian observation model $p_\theta(\mathbf{x}|\mathbf{z})$ for the generator.[2] This assumption alone is not sufficient for quantitative evaluation since the marginal likelihood of the observed data, $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z}) \mathrm{d}\mathbf{z}$ in this case would be intractable as it requires integrating over all the latent factors of variation. This would then require approximate inference (*e.g.*, Monte Carlo or variational methods) which in itself is a computational challenge for high-dimensional distributions. To circumvent these issues, we propose flow generative adversarial networks (Flow-GAN).

A Flow-GAN consists of a pair of generator-discriminator networks with the generator specified as a normalizing flow model (Dinh, Krueger, and Bengio 2014). A normalizing flow model specifies a parametric transformation from a prior density $p(\mathbf{z}) : \mathbb{R}^d \to \mathbb{R}_0^+$ to another density over the same space, $p_\theta(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}_0^+$ where $\mathbb{R}_0^+$ is the set of non-negative reals. The generator transformation $G_\theta : \mathbb{R}^d \to \mathbb{R}^d$ is invertible, such that there exists an inverse function $f_\theta = G_\theta^{-1}$. Using the change-of-variables formula and letting $\mathbf{z} = f_\theta(\mathbf{x})$, we have:

$$p_\theta(\mathbf{x}) = p(\mathbf{z}) \left| \det \frac{\partial f_\theta(\mathbf{x})}{\partial \mathbf{x}} \right| \quad (6)$$

where $\frac{\partial f_\theta(\mathbf{x})}{\partial \mathbf{x}}$ denotes the Jacobian of f_θ at $\mathbf{x}$. The above formula can be applied recursively over compositions of many invertible transformations to produce a complex final density. Hence, we can evaluate and optimize for the log-likelihood assigned by the model to a data point as long as the prior density is tractable and the determinant of the Jacobian of f_θ evaluated at $\mathbf{x}$ can be efficiently computed.

Evaluating the likelihood assigned by a Flow-GAN model in Eq. (6) requires overcoming two major challenges. First, requiring the generator function G_θ to be reversible imposes a constraint on the dimensionality of the latent variable $\mathbf{z}$ to match that of the data $\mathbf{x}$. Thereafter, we require the transformations between the various layers of the generator to be invertible such that their overall composition results in an invertible G_θ. Secondly, the Jacobian of high-dimensional distributions can however be computationally expensive to compute. If the transformations are designed such that the Jacobian is an upper or lower triangular matrix, then the determinant can be easily evaluated as the product of its diagonal entries. We consider two such family of transformations.

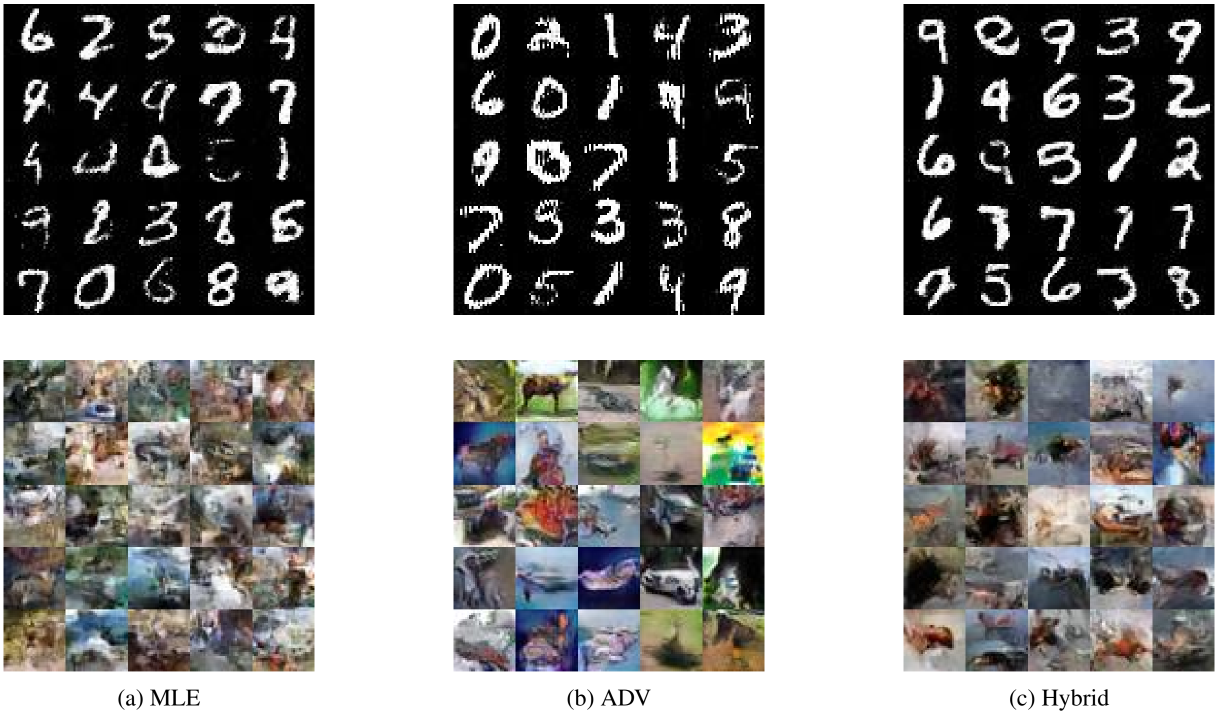
1. *Volume preserving transformations*. Here, the Jacobian of the transformations have a unit determinant. For example, the NICE model consists of several layers performing a location transformation (Dinh, Krueger, and Bengio 2014). The top layer is a diagonal scaling matrix with non-zero log determinant.
2. *Non-volume preserving transformations*. The determinant of the Jacobian of the transformations is not necessarily unity. For example, in Real-NVP, layers performs both location and scale transformations (Dinh, Sohl-Dickstein, and Bengio 2017).

For brevity, we direct the reader to Dinh, Krueger, and Bengio (2014) and Dinh, Sohl-Dickstein, and Bengio (2017) for the specifications of NICE and Real-NVP respectively. Crucially, both volume preserving and non-volume preserving transformations are invertible such that the determinant of the Jacobian can be computed tractably.

3.1 Learning objectives

In a Flow-GAN, the likelihood is well-defined and computationally tractable for exact evaluation of even expressive volume preserving and non-volume preserving transformations. Hence, a Flow-GAN can be trained via maximum likelihood estimation using Eq. (1) in which case the discriminator is redundant. Additionally, we can perform ancestral

[2] The true observation model for a GAN is a Dirac delta distribution, *i.e.*, $p_\theta(\mathbf{x}|\mathbf{z})$ is infinite when $\mathbf{x} = G_\theta(\mathbf{z})$ and zero otherwise.



(a) MLE

(b) ADV

(c) Hybrid

Figure 1: Samples generated by Flow-GAN models with different objectives for MNIST (**top**) and CIFAR-10 (**bottom**).

sampling just like a regular GAN whereby we sample a random vector $\mathbf{z} \sim P_z$ and transform it to a model generated sample via $G_\theta = f_\theta^{-1}$. This makes it possible to learn a Flow-GAN using an adversarial learning objective (for example, the WGAN objective in Eq. (5)).

A natural question to ask is why should one use adversarial learning given that MLE is statistically efficient asymptotically (under some conditions). Besides difficulties that could arise due to optimization (in both MLE and adversarial learning), the optimality of MLE holds only when there is no model misspecification for the generator *i.e.*, the true data distribution P_{data} is a member of the parametric family of distributions under consideration (White 1982). This is generally not the case for high-dimensional distributions, and hence the choice of the learning objective becomes largely an empirical question. Unlike other models, a Flow-GAN allows both maximum likelihood and adversarial learning, and hence we can investigate this question experimentally.

3.2 Evaluation metrics and experimental setup

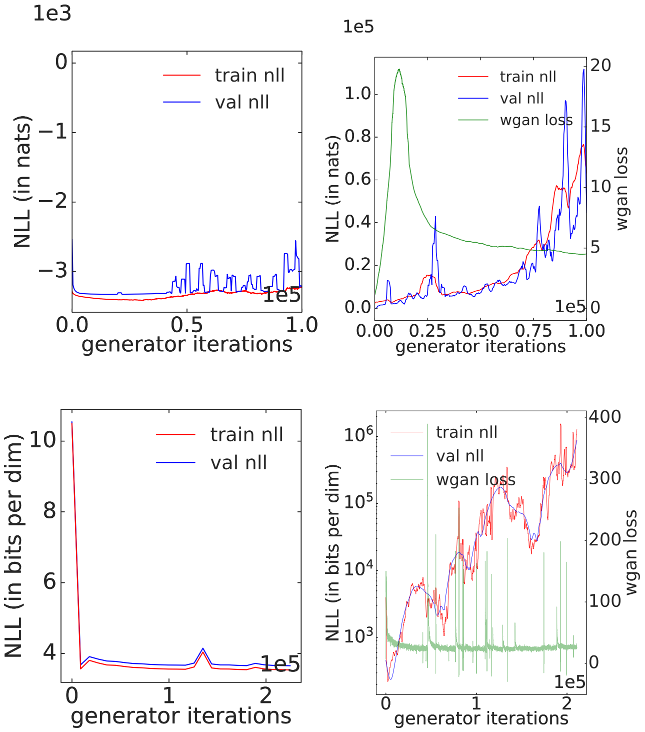
Our criteria for evaluation is based on held-out log-likelihoods and sample quality metrics. We focus on natural images since they allow visual inspection as well as quantification using recently proposed metrics. A "good" generative model should generalize to images outside the training data and assign high log-likelihoods to held-out data. The Inception and MODE scores are standard quantitative measures of the quality of generated samples of natural images for labelled datasets (Salimans et al. 2016; Che et al. 2017). The Inception scores are computed as:

$$\exp\left(\mathbb{E}_{\mathbf{x}\in P_\theta}[KL(p(y|\mathbf{x})\|p(y))]\right)$$

where $\mathbf{x}$ is a sample generated by the model, $p(y|\mathbf{x})$ is the softmax probability for the labels y assigned by a pretrained classifier for $\mathbf{x}$, and $p(y)$ is the overall distribution of labels in the generated samples (as predicted by the pretrained classifier). The intuition is that the conditional distribution $p(y|\mathbf{x})$ should have low entropy for good looking images while the marginal distribution $p(y)$ has high entropy to ensure sample diversity. Hence, a generative model can perform well on this metric if the KL divergence between the two distributions (and consequently, the Inception score for the generated samples) is large. The MODE score given below modifies the Inception score to take into account the distribution of labels in the training data, $p^*(y)$:

$$\exp\left(\mathbb{E}_{\mathbf{x}\in P_\theta}[KL(p(y|\mathbf{x})\|p^*(y)] - KL(p^*(y)\|p(y))\right).$$

We compare learning of Flow-GANs using MLE and adversarial learning (ADV) for the MNIST dataset of handwritten digits (LeCun, Cortes, and Burges 2010) and the CIFAR-10 dataset of natural images (Krizhevsky and Hinton 2009). The normalizing flow generator architectures are chosen to be NICE (Dinh, Krueger, and Bengio 2014) and Real-NVP (Dinh, Sohl-Dickstein, and Bengio 2017) for MNIST and CIFAR-10 respectively. We fix the Wasserstein distance as the choice of the divergence being optimized by ADV (see Eq. (5)) with the Lipschitz constraint over the critic imposed by penalizing the norm of the gradient with respect to the input (Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017). The discriminator is based on the DCGAN architecture (Radford, Metz, and Chintala 2015). The above choices are among the current state-of-the-art in maximum likelihood estimation and adversarial learning and greatly stabilize GAN training. Further experimental setup details are provided in Appendix A. The code for reproducing the results is available at `https://github.com/ermongroup/flow-gan`.



(a) MLE

(b) ADV

Figure 2: Learning curves for negative log-likelihood (NLL) evaluation on MNIST (**top**, in nats) and CIFAR (**bottom**, in bits/dim). Lower NLLs are better.

3.3 Evaluation results

Log-likelihood. The log-likelihood learning curves for Flow-GAN models learned using MLE and ADV are shown in Figure 2a and Figure 2b respectively. Following convention, we report the negative log-likelihoods (NLL) in nats for MNIST and bits/dimension for CIFAR-10.

MLE. In Figure 2a, we see that normalizing flow models attain low validation NLLs (blue curves) after few gradient updates as expected because it is explicitly optimizing for the MLE objective in Eq. (1). Continued training however could lead to overfitting as the train NLLs (red curves) begin to diverge from the validation NLLs.

ADV. Surprisingly, ADV models show a consistent *increase* in validation NLLs as training progresses as shown in Figure 2b (for CIFAR-10, the estimates are reported on a log scale!). Based on the learning curves, we can disregard overfitting as an explanation since the increase in NLLs is observed even on the training data. The training and validation NLLs closely track each other suggesting that ADV models are not simply memorizing the training data.

Comparing the left vs. right panels in Figure 2, we see that the log-likelihoods attained by ADV are orders of magnitude worse than those attained by MLE after sufficient training. Finally, we note that the WGAN loss (green curves) does not correlate well with NLL estimates. While the WGAN loss stabilizes after few iterations of training, the NLLs continue to increase. This observation is in contrast to prior work showing the loss to be strongly correlated with sample quality metrics (Arjovsky, Chintala, and Bottou 2017).

Sample quality. Samples generated from MLE and ADV-based models with the best MODE/Inception are shown in Figure 1a and Figure 1b respectively. ADV models significantly outperform MLE with respect to the final MODE/Inception scores achieved. Visual inspection of samples confirms the observations made on the based of the sample quality metrics. Curves monitoring the sample quality metrics at every training iteration are given in Appendix B.

3.4 Gaussian mixture models

The above experiments suggest that ADV can produce excellent samples but assigns low likelihoods to the observed data. However, a direct comparison of ADV with the log-likelihoods of MLE is unfair since the latter is explicitly optimizing for the desired objective. To highlight that generating good samples at the expense of low likelihoods is *not* a challenging goal, we propose a simple baseline. We compare the adversarially learned Flow-GAN models that achieves the highest MODE/Inception score with a Gaussian Mixture Model consisting of m isotropic Gaussians with equal weights centered at each of the m training points as the baseline Gaussian Mixture Model (GMM). The bandwidth hyperparameter, σ, is the same for each of the mixture components and optimized for the lowest validation NLL by doing a line search in $(0, 1]$. We show results for CIFAR-10 in Figure 3. Our observations below hold for MNIST as well; results deferred to Appendix C.

We overload the y-axis in Figure 3 to report both NLLs and sample quality metrics. The horizontal maroon and cyan dashed lines denote the best attainable MODE/Inception scores and corresponding validation NLLs respectively attained by the adversarially learned Flow-GAN model. The GMM can clearly attain better sample quality metrics since it is explicitly overfitting to the training data for low values of the bandwidth parameter (any σ for which the red curve is above the maroon dashed line). Surprisingly, the simple GMM also outperforms the adversarially learned model with respect to NLLs attained for several values of the bandwidth parameter (any σ for which the blue curve is below the cyan dashed line). Bandwidth parameters for which GMM models outperform the adversarially learned model on both log-likelihoods and sample quality metrics are highlighted using the green shaded area. We show samples from the GMM in the appendix. Hence, *a trivial baseline that is memorizing the training data can generate high quality samples and better held-out log-likelihoods*, suggesting that the log-likelihoods attained by adversarial training are very poor.

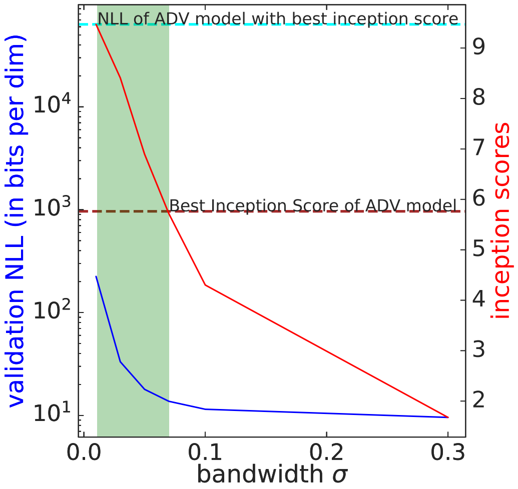


Figure 3: Gaussian Mixture Models outperform adversarially learned models on both held-out log-likelihoods and sampling metrics on CIFAR-10 (**green shaded region**).

4 Hybrid learning of Flow-GANs

In the previous section, we observed that adversarially learning Flow-GANs models attain poor held-out log-likelihoods. This makes it challenging to use such models for applications requiring density estimation. On the other hand, Flow-GANs learned using MLE are "mode covering" but do not generate high quality samples. With a Flow-GAN, it is possible to trade-off the two goals by combining the learning objectives corresponding to both these inductive principles. Without loss of generality, let $V(G_\theta, D_\phi)$ denote the minimax objective of any GAN model (such as WGAN). The hybrid objective of a Flow-GAN can be expressed as:

$$\min_{\theta} \max_{\phi} V(G_\theta, D_\phi) - \lambda \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [\log p_\theta(\mathbf{x})] \qquad (7)$$

where $\lambda \geq 0$ is a hyperparameter for the algorithm. By varying λ, we can interpolate between plain adversarial training ($\lambda = 0$) and MLE (very high λ).

We summarize the results from MLE, ADV, and Hybrid for log-likelihood and sample quality evaluation in Table 1 and Table 2 for MNIST and CIFAR-10 respectively. The tables report the test log-likelihoods corresponding to the best validated MLE and ADV models and the highest MODE/Inception scores observed during training. The samples generated by models with the best MODE/Inception scores for each objective are shown in Figure 1c.

While the results on CIFAR-10 are along expected lines, the hybrid objective interestingly outperforms MLE and ADV on both test log-likelihoods and sample quality metrics in the case of MNIST. One potential explanation for this is that the ADV objective can regularize MLE to generalize to the test set and in turn, the MLE objective can stabilize the optimization of the ADV objective. Hence, the hybrid objective in Eq. (7) can smoothly balance the two objectives using the tunable hyperparameter λ, and in some cases such as MNIST, the performance on both tasks could improve as a result of the hybrid objective.

Table 1: Best MODE scores and test negative log-likelihood estimates for Flow-GAN models on MNIST.

Objective	MODE Score	Test NLL (in nats)
MLE	7.42	−3334.56
ADV	9.24	−1604.09
Hybrid ($\lambda = 0.1$)	**9.37**	**−3342.95**

Table 2: Best Inception scores and test negative log-likelihood estimates for Flow-GAN models on CIFAR-10.

Objective	Inception Score	Test NLL (in bits/dim)
MLE	2.92	**3.54**
ADV	**5.76**	8.53
Hybrid ($\lambda = 1$)	3.90	4.21

5 Interpreting the results

Our findings are in contrast with prior work which report much better log-likelihoods for adversarially learned models with a standard generator architecture based on annealed importance sampling (AIS; (Wu et al. 2017)) and kernel density estimation (KDE; (Goodfellow et al. 2014)). These methods rely on approximate inference techniques for log-likelihood evaluation and make assumptions about a Gaussian observation model which does not hold for GANs. Since Flow-GANs allow us to compute *exact* log-likelihoods, we can evaluate the quality of approximation made by AIS and KDE for density estimation of invertible generators. For a detailed description of the methods, we refer the reader to prior work (Neal 2001; Parzen 1962).

We consider the MNIST dataset where these methods have been previously applied to by Wu et al. (2017) and Goodfellow et al. (2014) respectively. Since both AIS and KDE inherently rely on the samples generated, we evaluate these methods for the MLE, ADV, and Hybrid Flow-GAN model checkpoints corresponding to the best MODE scores observed during training. In Table 3, we observe that both AIS and KDE produce estimates of log-likelihood that are far from the ground truth, accessible through the exact Flow-GAN log-likelihoods. Even worse, the ranking of log-likelihood estimates for AIS (ADV>Hybrid>MLE) and KDE (Hybrid>MLE>ADV) do not obey the relative rankings of the Flow-GAN estimates (MLE>Hybrid>ADV).

5.1 Explaining log-likelihood trends

In order to explain the variation in log-likelihoods attained by various Flow-GAN learning objectives, we investigate the distribution of the magnitudes of singular values for the Jacobian matrix of several generator functions, G_θ for MNIST in Figure 4 evaluated at 64 noise vectors $\mathbf{z}$ randomly sampled from the prior density $p(\mathbf{z})$. The x-axis of the figure shows the singular value magnitudes on a log scale and for each singular value s, we show the corresponding cumulative distribution function value on the y-axis which signifies the fraction of singular values less than s. The results on CIFAR-10 in Appendix D show a similar trend.

The Jacobian is a good first-order approximation of the generator function locally. In Figure 4, we observe that the

Table 3: Comparison of inference techniques for negative log-likelihood estimation of Flow-GAN models on MNIST.

Objective	Flow-GAN NLL	AIS	KDE
MLE	-3287.69	-2584.40	-167.10
ADV	26350.30	-2916.10	-3.03
Hybrid	-3121.53	-2703.03	-205.69

singular value distribution for the Jacobian of an invertible generator learned using MLE (orange curves) is concentrated in a narrow range, and hence the Jacobian matrix is well-conditioned and easy to invert. In the case of invertible generators learned using ADV with Wasserstein distance (green curves) however, the spread of singular values is very wide, and hence the Jacobian matrix is ill-conditioned.

The average log determinant of the Jacobian matrices for MLE, ADV, and Hybrid models are -4170.34, -15588.34, and -5184.40 respectively which translates to the trend ADV<Hybrid<MLE. This indicates that the ADV models are trying to squish a sphere of unit volume centered at a latent vector $\mathbf{z}$ to a very small volume in the observed space $\mathbf{x}$. Tiny perturbations of training as well as held-out datapoints can hence manifest as poor log-likelihoods. In spite of not being limited in the representational capacity to cover the entire space of the data distribution (the dimensions of $\mathbf{z}$ (*i.e.*, k) and $\mathbf{x}$ (*i.e.*, d) match for invertible generators), ADV prefers to learn a distribution over a smaller support.

The Hybrid learning objective (blue curves), however, is able to correct for this behavior, and the distribution of singular value magnitudes matches closely to that of MLE. We also considered variations involving the standard DCGAN architectures with $k = d$ minimizing the Wasserstein distance (red curves) and Jenson-Shannon divergence (purple curves). The relative shift in distribution of singular value magnitudes to lower values is apparent even in these cases.

6 Discussion

Any model which allows for efficient likelihood evaluation and sampling can be trained using maximum likelihood and adversarial learning. This line of reasoning has been explored to some extent in prior work that combine the objectives of prescribed latent variable models such as VAEs (maximizing an evidence lower bound on the data) with adversarial learning (Larsen et al. 2015; Mescheder, Nowozin, and Geiger 2017a; Srivastava et al. 2017). However, the benefits of such procedures do not come for “free” since we still need some form of approximate inference to get a handle on the log-likelihoods. This could be expensive, for instance combining a VAE with a GAN introduces an additional inference network that increases the overall model complexity.

Our approach sidesteps the additional complexity due to approximate inference by considering a normalizing flow model. The trade-off made by a normalizing flow model is that the generator function needs to be invertible while other generative models such as VAEs have no such requirement. On the positive side, we can tractably evaluate exact log-likelihoods assigned by the model for any data point. Normalizing flow models have been previously used in the context of maximum likelihood estimation of fully observed and latent variable models (Dinh, Krueger, and Bengio 2014; Rezende and Mohamed 2015; Kingma, Salimans, and Welling 2016; Dinh, Sohl-Dickstein, and Bengio 2017).

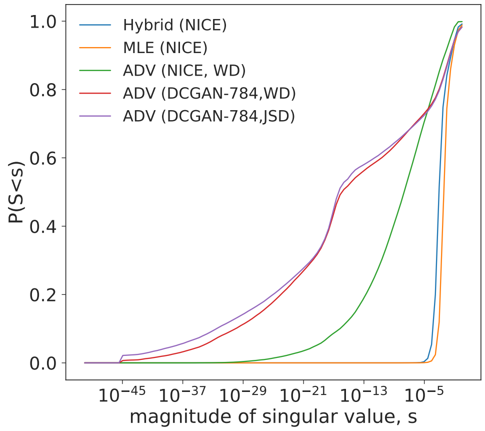


Figure 4: CDF of the singular values magnitudes for the Jacobian of the generator functions trained on MNIST.

The low dimensional support of the distributions learned by adversarial learning often manifests as lack of sample diversity and is referred to as mode collapse. In prior work, mode collapse is detected based on visual inspection or heuristic techniques (Goodfellow 2016; Arora and Zhang 2017). Techniques for avoiding mode collapse explicitly focus on stabilizing GAN training such as (Metz et al. 2016; Che et al. 2017; Mescheder, Nowozin, and Geiger 2017b) rather than quantitative methods based on likelihoods.

7 Conclusion

As an attempt to more quantitatively evaluate generative models, we introduced Flow-GAN. It is a generative adversarial network which allows for tractable likelihood evaluation, exactly like in a flow model. Since it can be trained both adversarially (like a GAN) and in terms of MLE (like a flow model), we can quantitatively evaluate the trade-offs involved. We observe that adversarial learning assigns very low-likelihoods to both training and validation data while generating superior quality samples. To put this observation in perspective, we demonstrate how a naive Gaussian mixture model can outperform adversarially learned models on both log-likelihood estimates and sample quality metrics. Quantitative evaluation methods based on AIS and KDE fail to detect this behavior and can be poor approximations of the true log-likelihood (at least for the models we considered).

Analyzing the Jacobian of the generator provides insights into the contrast between maximum likelihood estimation and adversarial learning. The latter have a tendency to learn distributions of low support, which can lead to low likelihoods. To correct for this behavior, we proposed a hybrid objective function which involves loss terms corresponding to both MLE and adversarial learning. The use of such models in applications requiring both density estimation and sample generation is an exciting direction for future work.

Acknowledgements

We are thankful to Ben Poole and Daniel Levy for helpful discussions. This research was supported by a Microsoft Research PhD fellowship in machine learning for the first author, NSF grants #1651565, #1522054, #1733686, a Future of Life Institute grant, and Intel.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein GAN. In *International Conference on Machine Learning*.

Arora, S., and Zhang, Y. 2017. Do GANs actually learn the distribution? An empirical study. *arXiv preprint arXiv:1706.08224*.

Besag, J. 1977. Efficiency of pseudolikelihood estimation for simple Gaussian fields. *Biometrika* 616–618.

Carreira-Perpinan, M. A., and Hinton, G. E. 2005. On contrastive divergence learning. In *Artificial Intelligence and Statistics*.

Che, T.; Li, Y.; Jacob, A. P.; Bengio, Y.; and Li, W. 2017. Mode regularized generative adversarial networks. In *International Conference on Learning Representations*.

Diggle, P. J., and Gratton, R. J. 1984. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society.* 193–227.

Dinh, L.; Krueger, D.; and Bengio, Y. 2014. NICE: Nonlinear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2017. Density estimation using Real NVP. In *International Conference on Learning Representations*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*.

Goodfellow, I. 2016. NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*.

Ho, J., and Ermon, S. 2016. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*.

Huber, P. J. 1967. The behavior of maximum likelihood estimates under nonstandard conditions. In *fifth Berkeley symposium on mathematical statistics and probability*.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations*.

Kingma, D. P.; Salimans, T.; and Welling, M. 2016. Improving variational inference with inverse autoregressive flow. In *International Conference on Learning Representations*.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Technical Report*.

Larochelle, H., and Murray, I. 2011. The neural autoregressive distribution estimator. In *Artificial Intelligence and Statistics*.

Larsen, A. B. L.; Sønderby, S. K.; Larochelle, H.; and Winther, O. 2015. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*.

LeCun, Y.; Cortes, C.; and Burges, C. J. 2010. MNIST handwritten digit database. *http://yann. lecun. com/exdb/mnist*.

Li, Y.; Song, J.; and Ermon, S. 2017. InfoGAIL: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*.

Mao, X.; Li, Q.; Xie, H.; Lau, R. Y.; Wang, Z.; and Smolley, S. P. 2017. Least squares generative adversarial networks. In *International Conference on Computer Vision*.

Mescheder, L.; Nowozin, S.; and Geiger, A. 2017a. Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine Learning*.

Mescheder, L.; Nowozin, S.; and Geiger, A. 2017b. The numerics of GANs. In *Advances in Neural Information Processing Systems*.

Metz, L.; Poole, B.; Pfau, D.; and Sohl-Dickstein, J. 2016. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*.

Mohamed, S., and Lakshminarayanan, B. 2016. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*.

Neal, R. M. 2001. Annealed importance sampling. *Statistics and Computing* 11(2):125–139.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*.

Oord, A. v. d.; Kalchbrenner, N.; and Kavukcuoglu, K. 2016. Pixel recurrent neural networks. In *International Conference on Machine Learning*.

Ostrovski, G.; Bellemare, M. G.; Oord, A. v. d.; and Munos, R. 2017. Count-based exploration with neural density models. In *International Conference on Machine Learning*.

Parzen, E. 1962. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics* 33(3):1065–1076.

Pascual, S.; Bonafonte, A.; and Serrà, J. 2017. SEGAN: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Rezende, D. J., and Mohamed, S. 2015. Variational inference with normalizing flows. In *International Conference on Machine Learning*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*.

Song, J.; Zhao, S.; and Ermon, S. 2017. A-NICE-MC: Adversarial training for MCMC. In *Advances in Neural Information Processing Systems*.

Srivastava, A.; Valkov, L.; Russell, C.; Gutmann, M.; and Sutton, C. 2017. VEEGAN: Reducing mode collapse in GANs using implicit variational learning. In *Advances in Neural Information Processing Systems*.

Theis, L.; Oord, A. v. d.; and Bethge, M. 2016. A note on the evaluation of generative models. In *International Conference on Learning Representations*.

Uria, B.; Murray, I.; and Larochelle, H. 2013. RNADE: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems*.

White, H. 1982. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society* 1–25.

Wu, Y.; Burda, Y.; Salakhutdinov, R.; and Grosse, R. 2017. On the quantitative analysis of decoder-based generative models. In *International Conference on Learning Representations*.

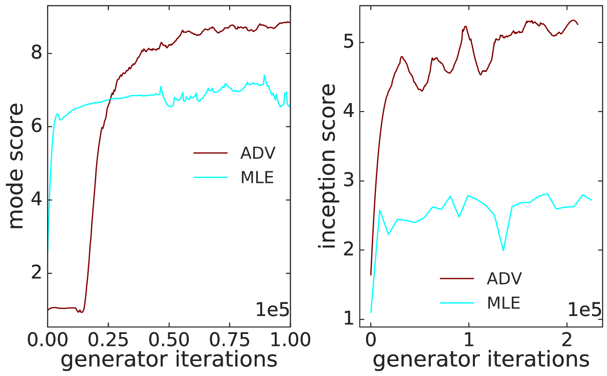
Appendices

A Experimental setup details

Datasets. The MNIST dataset contains $50,000$ train, $10,000$ validation, and $10,000$ test images of dimensions 28×28 (LeCun, Cortes, and Burges 2010). The CIFAR-10 dataset contains $50,000$ train and $10,000$ test images of dimensions $32 \times 32 \times 3$ by default (Krizhevsky and Hinton 2009). We held out a random subset of $5,000$ training set images as validation set.

Since we are modeling densities for discrete datasets (pixels can take a finite set of values ranging from 1 to 255), the model can assign arbitrarily high log-likelihoods to these discrete points. Following Uria, Murray, and Larochelle (2013), we dequantize the data by adding uniform noise between 0 and 1 to every pixel. Finally, we scale the pixels to lie in the range $[0, 1]$.

Model priors and hyperparameters. The Flow-GAN architectures trained on MNIST and CIFAR-10 used a logistic and an isotropic prior density respectively consistent with prior work (Dinh, Krueger, and Bengio 2014; Dinh, Sohl-Dickstein, and Bengio 2017). Hyperparameter details for learning all the Flow-GAN models are included in the README of the code repository: `https://github.com/ermongroup/flow-gan`



(a) MNIST (MODE)

(b) CIFAR-10 (Inception)

Figure 5: Sample quality curves during training.

B Sample quality

The progression of sample quality metrics for MLE and ADV objectives during training is shown in Figures 5 (a) and (b) for MNIST and CIFAR-10 respectively. Higher scores are reflective of better sample quality. ADV (maroon curves) significantly outperform MLE (cyan curves) with respect to the final MODE/Inception scores achieved.

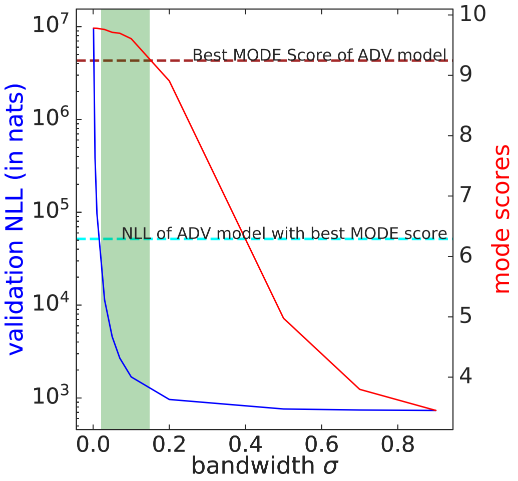


Figure 6: Gaussian Mixture Models outperform adversarially learned models on both held-out log-likelihoods and sampling metrics on MNIST (**green shaded region**).

C Gaussian mixture models

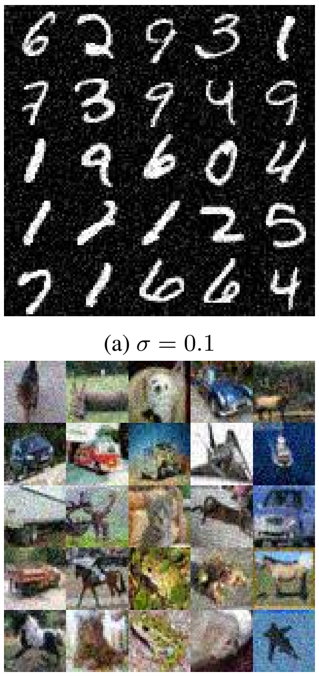
The comparison of GMMs with Flow-GANs trained using adversarial learning is shown in Figure 6. Similar to the observations made for CIFAR-10, the simple GMM outperforms the adversarially learned model with respect to NLLs and sample quality metrics for any bandwidth parameter within the green shaded area.

The samples obtained from the GMM are shown in Figure 7. Since the baseline is fitting Gaussian densities around every training point, the samples obtained for relatively small bandwidths are of high quality. Yet, even the held-out likelihoods for these bandwidths are better than those of ADV models with the best MODE/Inception scores.

D Explaining log-likelihood trends

The CDF of singular value magnitudes for the CIFAR-10 dataset in Figure 8 again suggests that the Jacobian matrix for the generator function is ill-conditioned for the ADV models (green, red, purple curves) since the distributions have a large spread. Using a hybrid objective (blue curves) can correct for this behavior with the distribution of singular values much more concentrated similar to MLE (orange curves).

The log determinant of the Jacobian for the MLE, ADV, and Hybrid models are -12818.84, -21848.09, -14729.51 respectively reflecting the trend ADV<Hybrid<MLE, providing further empirical evidence to suggest that adversarial training shows a strong preference for learning distributions with smaller support.



(a) $\sigma = 0.1$

(b) $\sigma = 0.07$

Figure 7: Samples from the Gaussian Mixture Model baseline for MNIST (**top**) and CIFAR-10 (**bottom**) with better MODE/Inception scores than ADV models.

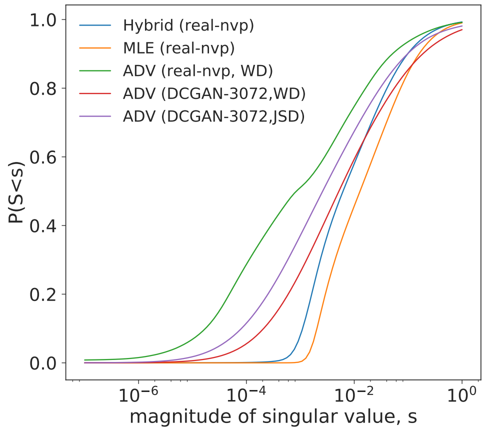


Figure 8: CDF of singular values magnitudes for the Jacobian of the generator function on CIFAR-10.