Supplementary Materials
Entropic GANs meet VAEs: A Statistical Approach to Compute Sample Likelihoods in GANs

1. Proof of Theorem 1

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

$$
\begin{aligned}
\log f_Y(\mathbf{y}) &= \log f_{Y|X=\mathbf{x}}(\mathbf{y}) + \log f_X(\mathbf{x}) - \log f_{X|Y=\mathbf{y}}(\mathbf{x}) \qquad (1.1)\\
&= \log C - \ell(\mathbf{y}, G(\mathbf{x})) - \log\sqrt{2\pi}\\
&\quad - \frac{\|\mathbf{x}\|^2}{2} - \log f_{X|Y=\mathbf{y}}(\mathbf{x}),
\end{aligned}
$$

where the second step follows from Equation 2.4 (main paper).

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 1.1 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:

$$
\begin{aligned}
\log f_Y(\mathbf{y}) =& \mathbb{E}_{\mathbb{P}_{X|Y=\mathbf{y}}}\Big[-\ell(\mathbf{y}, \mathbf{G}(\mathbf{x}))/\lambda + \log C - \frac{1}{2}\log 2\pi \qquad (1.2)\\
& - \frac{\|\mathbf{x}\|^2}{2} - \log f_{X|Y=\mathbf{y}}(\mathbf{x})\Big] \qquad (1.3)\\
=& \mathbb{E}_{\mathbb{P}_{X|Y=\mathbf{y}}}\Bigg[-\ell(\mathbf{y}, \mathbf{G}(\mathbf{x}))/\lambda + \log C - \frac{1}{2}\log 2\pi\\
& - \frac{\|\mathbf{x}\|^2}{2} - \log f_{X|Y=\mathbf{y}}(\mathbf{x}) + \log\left(\mathbb{P}_{X|Y=\mathbf{y}}(\mathbf{x})\right)\\
& - \log\left(\mathbb{P}_{X|Y=\mathbf{y}}(\mathbf{x})\right)\Bigg]\\
=& -\mathbb{E}_{\mathbb{P}_{X|Y=\mathbf{y}}}\left[\ell(\mathbf{y}, \mathbf{G}(\mathbf{x}))/\lambda\right] - \frac{1}{2}\log 2\pi\\
& + \log C + \mathbb{E}_{\mathbb{P}_{X|Y=\mathbf{y}}}\left[-\frac{\|\mathbf{x}\|^2}{2}\right]\\
& + \text{KL}\left(\mathbb{P}_{X|Y=\mathbf{y}} || f_{X|Y=\mathbf{y}}\right) + H\left(\mathbb{P}_{X|Y=\mathbf{y}}\right), \qquad (1.4)
\end{aligned}
$$

where $H(.)$ is the Shannon-entropy function. Please note that Corrolary 2 follows from Equation (1.4).

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

$$
\begin{aligned}
\mathbb{E}\left[\log f_Y(Y)\right] =& -\frac{1}{\lambda}\mathbb{E}_{\mathbb{P}_{X,Y}}\left[\ell(\mathbf{y}, G(\mathbf{x}))\right] - \frac{1}{2}\log 2\pi\\
& + \log C + \mathbb{E}_{f_X}\left[-\frac{\|\mathbf{x}\|^2}{2}\right] \qquad (1.5)\\
& + \mathbb{E}_{\mathbb{P}_Y}\left[\text{KL}\left(\mathbb{P}_{X|Y=\mathbf{y}} || f_{X|Y=\mathbf{y}}\right)\right]\\
& + H\left(\mathbb{P}_{X,Y}\right) - H\left(\mathbb{P}_Y\right).
\end{aligned}
$$

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

$$
\begin{aligned}
\mathbb{E}\left[\log f_Y(Y)\right] \geq& -\frac{1}{\lambda}\left\{\mathbb{E}_{\mathbb{P}_{X,Y}}\left[\ell(\mathbf{y}, \mathbf{G}(\mathbf{x}))\right] - \lambda H\left(\mathbb{P}_{X,Y}\right)\right\}\\
& + \log C - \log(m) - \frac{r + \log 2\pi}{2} \qquad (1.6)
\end{aligned}
$$

Moreover, using the data processing inequality, we have $H(\mathbb{P}_{X,Y}) \geq H(\mathbb{P}_{\mathbf{G}(X),Y})$ (Cover & Thomas, 2012). Thus,

$$
\begin{aligned}
\underbrace{\mathbb{E}\left[\log f_Y(Y)\right]}_{\text{sample likelihood}} \geq& -\frac{1}{\lambda}\underbrace{\left\{\mathbb{E}_{\mathbb{P}_{X,Y}}\left[\ell(\mathbf{y}, \mathbf{G}(\mathbf{x}))\right] - \lambda H\left(\mathbb{P}_{Y,\hat{Y}}\right)\right\}}_{\text{GAN objective with entropy regularizer}}\\
& + \log C - \log(m) - \frac{r + \log 2\pi}{2} \qquad (1.7)
\end{aligned}
$$

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 2.3 (main paper).

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 $L$
```

2. Optimal Coupling for W2GAN

Optimal coupling $\mathbb{P}^*_{Y,\hat{Y}}$ for the W2GAN (quadratic GAN (Feizi et al., 2017)) can be computed using the gradient of the optimal discriminator (Villani, 2008) as follows.

Lemma 1 *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*

$$\hat{Y} \overset{dist}{=} Y - \nabla \mathbf{D}^{opt}(Y), \tag{2.1}$$

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

3. 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

$$W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) := \min_{\mathbb{P}_{Y,\hat{Y}}} \mathbb{E}\left[\ell(Y,\hat{Y})\right] + \lambda \mathrm{KL}\left(\mathbb{P}_{Y,\hat{Y}}\right), \tag{3.1}$$

where KL(.||.) is the KullbackLeibler divergence. Note that the objective of this optimization differs from that of the entropic GAN optimization 2.3 (main paper) 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):

$$\begin{aligned}\bar{W}_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) :=& 2W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) - W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_Y) \\ &- W_{\ell,\lambda}(\mathbb{P}_{\hat{Y}}, \mathbb{P}_{\hat{Y}}).\end{aligned} \tag{3.2}$$

$\bar{W}$ 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 2 *For a given $\lambda > 0$, we have*

$$\begin{aligned}\bar{W}_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) \leq 2W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) \leq \bar{W}_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) \\ + \lambda H(\mathbb{P}_Y) + \lambda H(\mathbb{P}_{\hat{Y}}).\end{aligned} \tag{3.3}$$

Proof From the definition (3.2), we have $W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) :\geq \bar{W}_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}})/2$. Moreover, since $W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_Y) \leq H(\mathbb{P}_Y)$ (this can be seen by using an identity coupling as a feasible solution for optimization (3.1)) and similarly $W_{\ell,\lambda}(\mathbb{P}_{\hat{Y}}, \mathbb{P}_{\hat{Y}}) \leq H(\mathbb{P}_{\hat{Y}})$, we have $W_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}}) \leq \bar{W}_{\ell,\lambda}(\mathbb{P}_Y, \mathbb{P}_{\hat{Y}})/2 + \lambda/2H(\mathbb{P}_Y) + \lambda/2H(\mathbb{P}_{\hat{Y}})$. ■

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.

4. Approximate Likelihood Computation in Un-regularized GANs

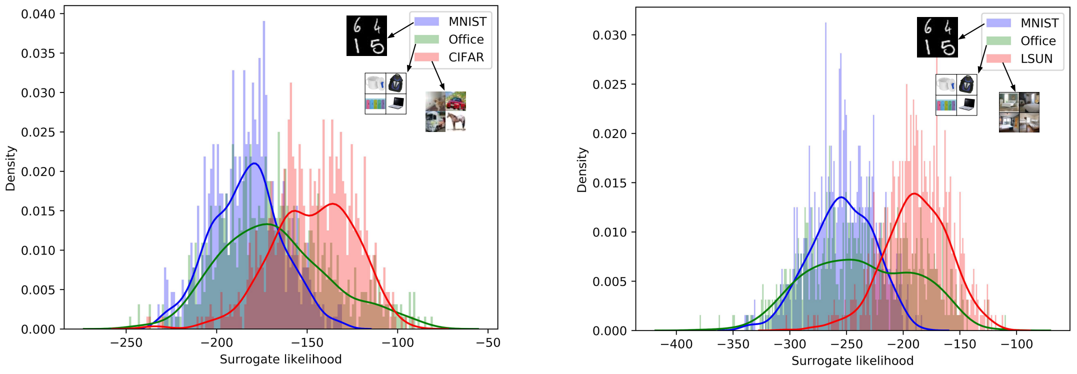
Most standard GAN architectures do not have the entropy regularization. Likelihood lower bounds of Theorem 1 and Corollary 2 hold even for those GANs as long as we obtain the optimal coupling $\mathbb{P}^*_{Y,\hat{Y}}$ in addition to the optimal generator $\mathbf{G}^*$ from GAN's training. Computation 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 Supplementary material).

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 Eq 3.7). We can then use Corollary 2 to approximate sample likelihoods for various GAN architectures.

We use this approach to compute likelihood estimates for CIFAR-10 (Krizhevsky, 2009) and LSUN-Bedrooms (Yu et al., 2015) datasets. For CIFAR-10, we train DCGAN while for LSUN, we train WGAN. Fig. 1a demonstrates sample likelihood estimates of different datasets using a GAN trained on CIFAR-10. Likelihoods assigned to samples from MNIST and Office datasets are lower than that of the CIFAR dataset. Samples from the Office dataset, however, are assigned to higher likelihood values than MNIST samples. 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. We observe similar performance trends in this experiment (Fig. 1b).

5. Training Entropic GANs

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



(a)

(b)

Figure 1: (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.

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

where

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

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.

5.1. GAN's Training on MNIST

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. 2

5.2. GAN's Training on CIFAR

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 4

5.3. GAN's Training on 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. 5

Table 1: Generator architecture

Layer	Output size	Filters
Input	128	-
Fully connected	4.4.256	128 → 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 → 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 → 64
BatchNorm+ReLU	$128 \times 8 \times 8$	-
Deconv2d (5×5, str 2)	$1 \times 28 \times 28$	64 → 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 → 32
LeakyReLU(0.2)	$32 \times 14 \times 14$	-
Conv2D(5×5, str 2)	$64 \times 7 \times 7$	32 → 64
LeakyReLU(0.2)	$64 \times 7 \times 7$	-
Conv2d (5×5, str 2)	$128 \times 4 \times 4$	64 → 128
LeakyRelU(0.2)	$128 \times 4 \times 4$	-
Reshape	128.4.4	-
Fully connected	1	2048 → 1

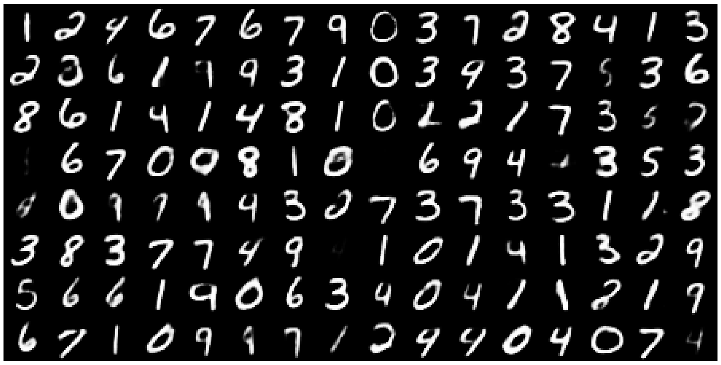


Figure 2: Samples generated by Entropic GAN trained on MNIST

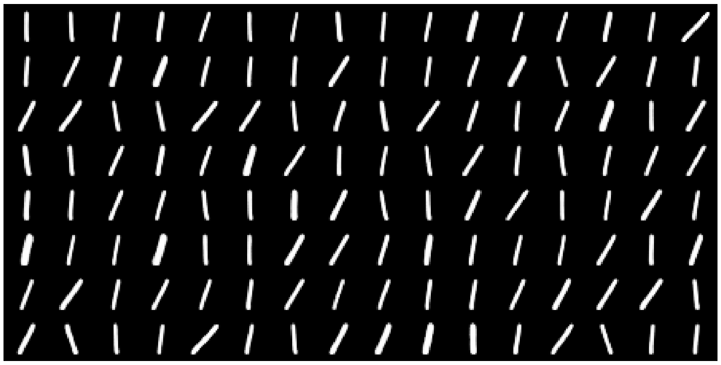


Figure 3: Samples generated by Entropic GAN trained on MNIST-1 dataset

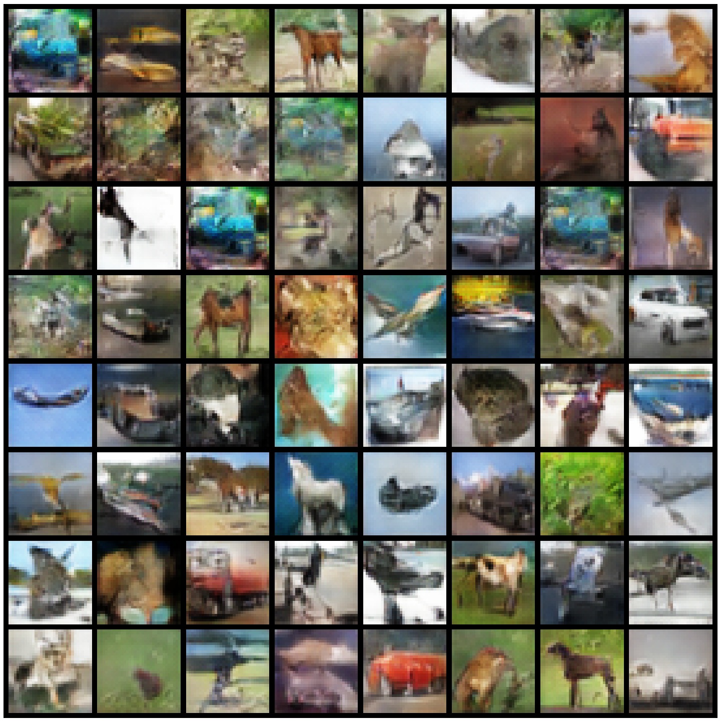


Figure 4: Samples generated by DCGAN model trained on CIFAR dataset

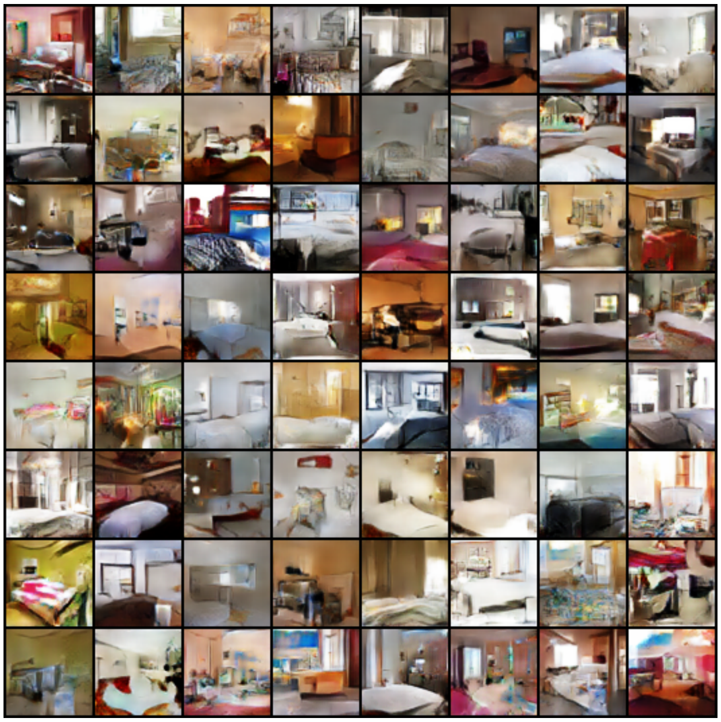


Figure 5: Samples generated by WGAN model trained on LSUN-Bedrooms dataset

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

Cover, T. M. and Thomas, J. A. *Elements of information theory*. John Wiley & Sons, 2012.

Feizi, S., Suh, C., Xia, F., and Tse, D. Understanding GANs: the LQG setting. *arXiv preprint arXiv:1710.10793*, 2017.

Genevay, A., Peyré, G., and Cuturi, M. Sinkhorn-autodiff: Tractable wasserstein learning of generative models. *arXiv preprint arXiv:1706.00292*, 2017.

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

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

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	β_1 = 0.5, β_2 = 0.9
Number of critic iters / gen iter	5
Number of training iterations	10000

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

Sanjabi, M., Ba, J., Razaviyayn, M., and Lee, J. D. Solving approximate Wasserstein GANs to stationarity. *Neural Information Processing Systems (NIPS)*, 2018.

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

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.