A PROOFS

Proof of Lemma 1. If $\boldsymbol{y} \sim p_{\boldsymbol{\theta}}(\boldsymbol{y})$ is equal to $\boldsymbol{x} \sim p_X(\boldsymbol{x})$ in distribution, then $p_X(\boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathbb{R}^V$. For (2) we have $\int p_X(\boldsymbol{x})\pi_{\boldsymbol{\phi}}(\boldsymbol{y} \,|\, \boldsymbol{x})\mathrm{d}\boldsymbol{y} = p_X(\boldsymbol{x}) \int \pi_{\boldsymbol{\phi}}(\boldsymbol{y} \,|\, \boldsymbol{x})\mathrm{d}\boldsymbol{y} = p_X(\boldsymbol{x})$ and

$$\begin{aligned}\int p_X(\boldsymbol{x})\pi_{\boldsymbol{\phi}}(\boldsymbol{y} \,|\, \boldsymbol{x})\mathrm{d}\boldsymbol{x} &= \int p_X(\boldsymbol{x}) \frac{e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} p_{\boldsymbol{\theta}}(\boldsymbol{y})}{\int e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} p_{\boldsymbol{\theta}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}} \mathrm{d}\boldsymbol{x} \\ &= p_{\boldsymbol{\theta}}(\boldsymbol{y}) \int \frac{e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} p_X(\boldsymbol{x})}{\int e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} p_{\boldsymbol{\theta}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}} \mathrm{d}\boldsymbol{x}.\end{aligned}$$

If $e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} = \mathbf{1}(\boldsymbol{x} = \boldsymbol{y})$, then we further have

$$\int \frac{e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} p_X(\boldsymbol{x})}{\int e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} p_{\boldsymbol{\theta}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}} \mathrm{d}\boldsymbol{x} = \int \frac{\mathbf{1}(\boldsymbol{x} = \boldsymbol{y}) p_X(\boldsymbol{x})}{\int \mathbf{1}(\boldsymbol{x} = \boldsymbol{y}) p_{\boldsymbol{\theta}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}} \mathrm{d}\boldsymbol{x} = \int \mathbf{1}(\boldsymbol{x} = \boldsymbol{y}) \frac{p_X(\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} = 1$$

and hence it is true that

$$\int p_X(\boldsymbol{x})\pi_{\boldsymbol{\phi}}(\boldsymbol{y} \,|\, \boldsymbol{x})\mathrm{d}\boldsymbol{x} = p_{\boldsymbol{\theta}}(\boldsymbol{y}).$$

Similarly, for (3) we have $\int p_{\boldsymbol{\theta}}(\boldsymbol{y})\pi_{\boldsymbol{\phi}}(\boldsymbol{x} \,|\, \boldsymbol{y})\mathrm{d}\boldsymbol{x} = p_{\boldsymbol{\theta}}(\boldsymbol{y})$ and can prove $\int p_{\boldsymbol{\theta}}(\boldsymbol{y})\pi_{\boldsymbol{\phi}}(\boldsymbol{x} \,|\, \boldsymbol{y})\mathrm{d}\boldsymbol{y} = p_X(\boldsymbol{x})$ given these two conditions. □

Proof of Lemma 2. Since $c(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ by definition, we have $\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \to \nu) \geq 0$ and $\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \leftarrow \nu) \geq 0$. When $\mu = \nu$, it is known that $\mathcal{W}(\mu, \nu) = 0$. If $\boldsymbol{y} \sim p_{\boldsymbol{\theta}}(\boldsymbol{y})$ is equal to $\boldsymbol{x} \sim p_X(\boldsymbol{x})$ in distribution, which means $p_X(\boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $p_X(\boldsymbol{y}) = p_{\boldsymbol{\theta}}(\boldsymbol{y})$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^V$ and $\mu = \nu$, then we have

$$\begin{aligned}\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \to \nu) &= \int\int c(\boldsymbol{x}, \boldsymbol{y}) p_X(\boldsymbol{x}) \pi_{\boldsymbol{\phi}}(\boldsymbol{y} \,|\, \boldsymbol{x}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= \int\int c(\boldsymbol{x}, \boldsymbol{y}) \frac{e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} p_X(\boldsymbol{x}) p_{\boldsymbol{\theta}}(\boldsymbol{y})}{\int e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} p_{\boldsymbol{\theta}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= \int\int c(\boldsymbol{y}, \boldsymbol{x}) \frac{e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}))} p_X(\boldsymbol{y}) p_{\boldsymbol{\theta}}(\boldsymbol{x})}{\int e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}))} p_{\boldsymbol{\theta}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= \int\int c(\boldsymbol{y}, \boldsymbol{x}) \frac{e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}))} p_{\boldsymbol{\theta}}(\boldsymbol{y}) p_X(\boldsymbol{x})}{\int e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}))} p_X(\boldsymbol{x})\mathrm{d}\boldsymbol{x}} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= \int\int c(\boldsymbol{x}, \boldsymbol{y}) p_{\boldsymbol{\theta}}(\boldsymbol{y}) \frac{e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} p_X(\boldsymbol{x})}{\int e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} p_X(\boldsymbol{x})\mathrm{d}\boldsymbol{x}} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= \int\int c(\boldsymbol{x}, \boldsymbol{y}) p_{\boldsymbol{\theta}}(\boldsymbol{y}) \pi_{\boldsymbol{\phi}}(\boldsymbol{x} \,|\, \boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= \mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \leftarrow \nu)\end{aligned}$$

and hence $C_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu, \nu) = \mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \to \nu) = \mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \leftarrow \nu) \geq 0 = W(\mu, \nu)$.

If $e^{-d(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{y}))} = \mathbf{1}(\boldsymbol{x} = \boldsymbol{y})$, since $c(\boldsymbol{x}, \boldsymbol{x}) = 0$ by definition, we have

$$\begin{aligned}\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \to \nu) &= \int\int c(\boldsymbol{x}, \boldsymbol{y}) \frac{\mathbf{1}(\boldsymbol{x} = \boldsymbol{y}) p_X(\boldsymbol{x}) p_{\boldsymbol{\theta}}(\boldsymbol{y})}{\int \mathbf{1}(\boldsymbol{x} = \boldsymbol{y}) p_{\boldsymbol{\theta}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= \int\int c(\boldsymbol{x}, \boldsymbol{y}) \frac{\mathbf{1}(\boldsymbol{x} = \boldsymbol{y}) p_X(\boldsymbol{x}) p_{\boldsymbol{\theta}}(\boldsymbol{y})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= \int\int c(\boldsymbol{x}, \boldsymbol{y}) \mathbf{1}(\boldsymbol{x} = \boldsymbol{y}) p_{\boldsymbol{\theta}}(\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= \int c(\boldsymbol{x}, \boldsymbol{x}) p_{\boldsymbol{\theta}}(\boldsymbol{x}) d\boldsymbol{x} \\ &= 0.\end{aligned}$$

□

Proof of Lemma 3. According to the strong law of large numbers, when $M \to \infty$, $\hat{\nu}_M(A) = \frac{1}{M}\sum_{j=1}^{M} \mathbf{1}(\boldsymbol{y}_j \in A)$ converges almost surely to

$$\frac{1}{M}\sum_{j=1}^{M} \mathbb{E}_{\boldsymbol{y}_j \sim p_{\boldsymbol{\theta}}(\boldsymbol{y})}[\mathbf{1}(\boldsymbol{y}_j \in A)] = \int_A p_{\boldsymbol{\theta}}(\boldsymbol{y})d\boldsymbol{y} = \nu(A)$$

and hence $C_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \to \hat{\nu}_M)$ converges to $C_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \to \nu)$. Therefore, $\mathbb{E}_{\boldsymbol{y}_{1:M} \overset{iid}{\sim} p_{\boldsymbol{\theta}}(\boldsymbol{y})}[\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \to \hat{\nu}_M)]$ converges to $\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \to \nu)$. Similarly, we can prove that as $N \to \infty$, $\mathbb{E}_{\boldsymbol{x}_{1:N} \overset{iid}{\sim} p_X(\boldsymbol{x})}[\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\hat{\mu}_N \leftarrow \nu)$ converges to $\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \leftarrow \nu)$. Therefore, $\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu, \nu, N, M)$ defined in (12) converges to $\frac{1}{2}\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \to \nu) + \frac{1}{2}\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \leftarrow \nu) = \mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu, \nu)$ as $N, M \to \infty$. □

Proof of Lemma 4.

$$\begin{aligned}
&\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu, \nu, N, M) = \tfrac{1}{2}\mathbb{E}_{\boldsymbol{y}_{1:M} \overset{iid}{\sim} p_{\boldsymbol{\theta}}(\boldsymbol{y})}[\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu \to \hat{\nu}_M)] + \tfrac{1}{2}\mathbb{E}_{\boldsymbol{x}_{1:N} \overset{iid}{\sim} p_X(\boldsymbol{x})}[\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\hat{\mu}_N \leftarrow \nu)] \\
&= \tfrac{1}{2}\mathbb{E}_{\boldsymbol{x} \sim p_X(\boldsymbol{x}),\, \boldsymbol{y}_{1:M} \overset{iid}{\sim} p_{\boldsymbol{\theta}}(\boldsymbol{y})}\left[\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{x} \to \hat{\nu}_M)\right] + \tfrac{1}{2}\mathbb{E}_{\boldsymbol{x}_{1:N} \overset{iid}{\sim} p_X(\boldsymbol{x}),\, \boldsymbol{y} \sim p_{\boldsymbol{\theta}}(\boldsymbol{y})}\left[\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\hat{\mu}_N \leftarrow \boldsymbol{y})\right] \\
&= \tfrac{1}{2}\mathbb{E}_{\boldsymbol{x} \sim \hat{p}_N(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{x}_{1:M} \overset{iid}{\sim} p_X(\boldsymbol{x}),\, \boldsymbol{y}_{1:M} \overset{iid}{\sim} p_{\boldsymbol{\theta}}(\boldsymbol{y})}\left[\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{x} \to \hat{\nu}_M)\right] \\
&\quad + \tfrac{1}{2}\mathbb{E}_{\boldsymbol{y} \sim \hat{p}_M(\boldsymbol{y})}\mathbb{E}_{\boldsymbol{x}_{1:N} \overset{iid}{\sim} p_X(\boldsymbol{x}),\, \boldsymbol{y}_{1:M} \overset{iid}{\sim} p_{\boldsymbol{\theta}}(\boldsymbol{y})}\left[\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\hat{\mu}_N \leftarrow \boldsymbol{y})\right] \\
&= \mathbb{E}_{\boldsymbol{x} \sim \hat{p}_N(\boldsymbol{x}),\, \boldsymbol{y} \sim \hat{p}_M(\boldsymbol{y})}\mathbb{E}_{\boldsymbol{x}_{1:N} \overset{iid}{\sim} p_X(\boldsymbol{x}),\, \boldsymbol{y}_{1:M} \overset{iid}{\sim} p_{\boldsymbol{\theta}}(\boldsymbol{y})}\left[\tfrac{1}{2}\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{x} \to \hat{\nu}_M) + \tfrac{1}{2}\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\hat{\mu}_N \leftarrow \boldsymbol{y})\right]. \quad (18)
\end{aligned}$$

Plugging (8) and (10) into the above equation concludes the proof.

□

Proof of Lemma 5. Solving the first expectation of (18), we have

$$\begin{aligned}
&\mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mu, \nu, N, M) \\
&= \mathbb{E}_{\boldsymbol{x}_{1:N} \overset{iid}{\sim} p_X(\boldsymbol{x}),\, \boldsymbol{y}_{1:M} \overset{iid}{\sim} p_{\boldsymbol{\theta}}(\boldsymbol{y})}\left[\tfrac{1}{2N}\textstyle\sum_{i=1}^{N} \mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{x}_i \to \hat{\nu}_M) + \tfrac{1}{2M}\sum_{j=1}^{M} \mathcal{C}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\hat{\mu}_N \leftarrow \boldsymbol{y}_j)\right].
\end{aligned}$$

Plugging (8) and (10) into the above equation concludes the proof.

□

B SUPPLEMENTARY EXPERIMENT RESULTS

B.1 DERIVATION OF THE UNIVARIATE NORMAL BASED TOY EXAMPLE DEFINED IN (17)

For the toy example specified in (17), exploiting the normal-normal conjugacy, we have an analytical conditional distribution for the forward navigator as

$$\begin{aligned}
\pi_\phi(y \,|\, x) &\propto e^{-\frac{(x-y)^2}{2e^\phi}}\mathcal{N}(y; 0, e^\theta) \\
&\propto \mathcal{N}(x; y, e^\phi)\mathcal{N}(y; 0, e^\theta) \\
&= \mathcal{N}\left(\frac{e^\theta}{e^\theta + e^\phi}x, \frac{e^\phi e^\theta}{e^\theta + e^\phi}\right),
\end{aligned}$$

and an analytical conditional distribution for the backward navigator as

$$\begin{aligned}
\pi_\phi(x \,|\, y) &\propto e^{-\frac{(x-y)^2}{2e^\phi}}\mathcal{N}(x; 0, 1) \\
&\propto \mathcal{N}(y; x, e^\phi)\mathcal{N}(x; 0, 1) \\
&= \mathcal{N}\left(\frac{y}{1 + e^\phi}, \frac{e^\phi}{1 + e^\phi}\right).
\end{aligned}$$

Plugging them into (2) and (3), respectively, and solving the expectations, we have

$$\begin{aligned}\mathcal{C}_{\phi,\theta}(\mu \to \nu) &= \mathbb{E}_{x\sim\mathcal{N}(0,1)}\left[\frac{e^\phi}{e^\theta+e^\phi}\left(e^\theta+\frac{e^\phi}{e^\theta+e^\phi}x^2\right)\right]\\ &= \frac{e^\phi}{e^\theta+e^\phi}\left(e^\theta+\frac{e^\phi}{e^\theta+e^\phi}\right),\end{aligned}$$

$$\begin{aligned}\mathcal{C}_{\phi,\theta}(\mu \leftarrow \nu) &= \mathbb{E}_{y\sim\mathcal{N}(0,e^\theta)}\left[\frac{e^\phi}{1+e^\phi}\left(1+\frac{e^\phi}{1+e^\phi}y^2\right)\right]\\ &= \frac{e^\phi}{1+e^\phi}\left(1+\frac{e^\phi}{1+e^\phi}e^\theta\right).\end{aligned}$$

B.2 MORE RESULTS ON 2D TOY DATASETS

We visualize the results on three additional 2D toy datasets here. Compared to the 8-Gaussian mixture dataset, the mode collapse issue of both GAN and WGAN-GP becomes more severe on the Swiss-Roll, Half-Moon, and 25-Gaussian datasets, while ACT consistently shows good and stable performance on all of them.

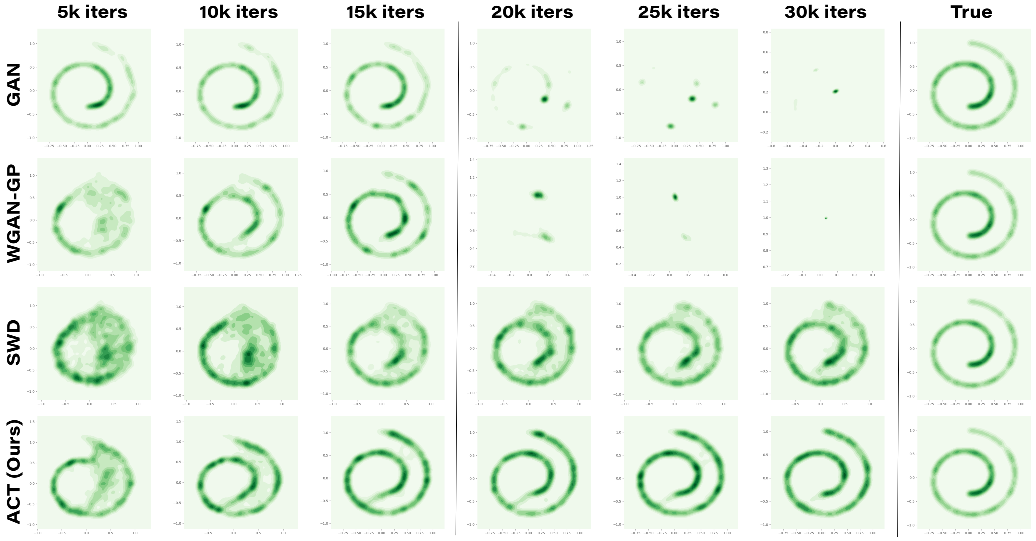

Figure 5: Analogous plot to Fig. 3 for the Swiss-Roll dataset.

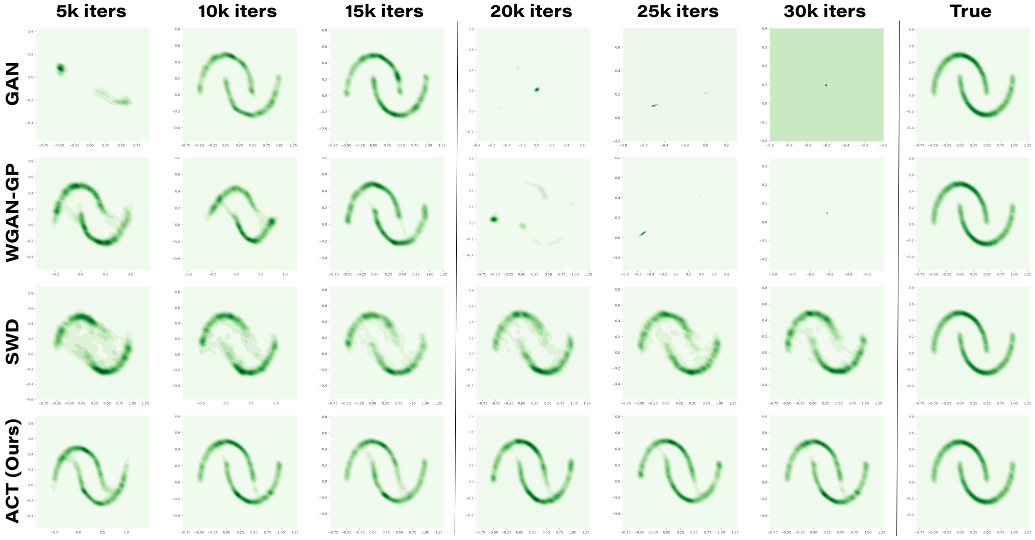

Figure 6: Analogous plot to Fig. 3 for the Half-Moon dataset.

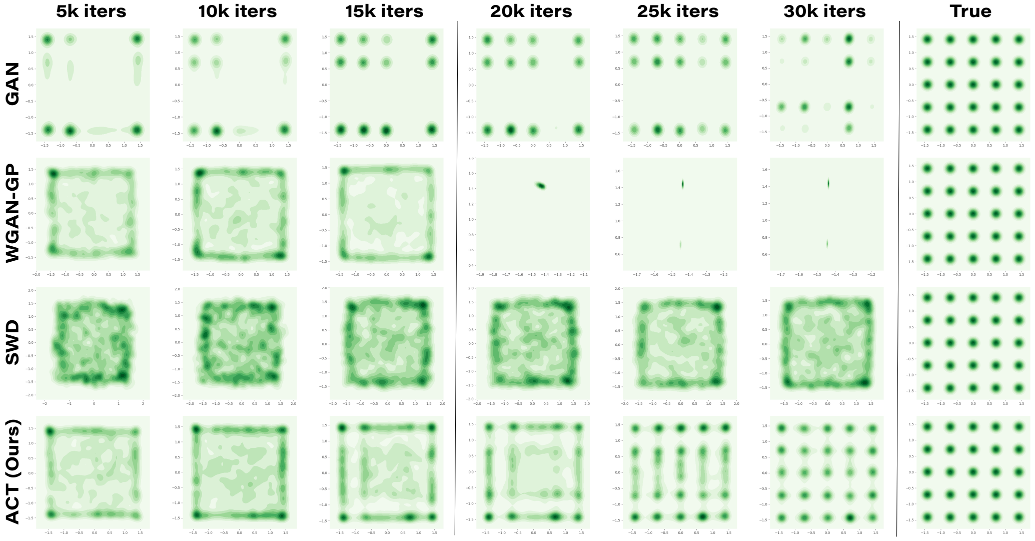

Figure 7: Analogous plot to Fig. 3 for the 25-Gaussian mixture dataset.

We also illustrate the data points and generated samples with empirical samples, shown in Fig. 8. The first column shows the generated samples (marked in blue) and the samples from data distribution (marked in red). To visualize how the feature extractor $\mathcal{T}_\phi$ used by both navigators works, we set its output dimension as 1 and plot the logits in the third and fifth columns and map the corresponding data (in the second column) and generated samples (in the fourth column) with the same color.

Similarly, we visualize the GAN's generated samples and logits produced by its discriminator in Fig. 9. We can observe that the discriminator maps the data to very close values. Specifically, in both the 8-Gaussian mixture and 25-Gaussian mixture cases, when the mode collapse occurs, the logits of the missed modes have similar value to the those in the other modes. This property results in GAN's mode collapse problem and it is commonly observed in GANs. Different from the GAN case, the navigator in our ACT model maps the data with non-saturating logits. We can observe in various multi-mode cases, different modes are assigned with different values by the navigator. This property helps ACT to well resist the mode collapse problem and stabilize the training.

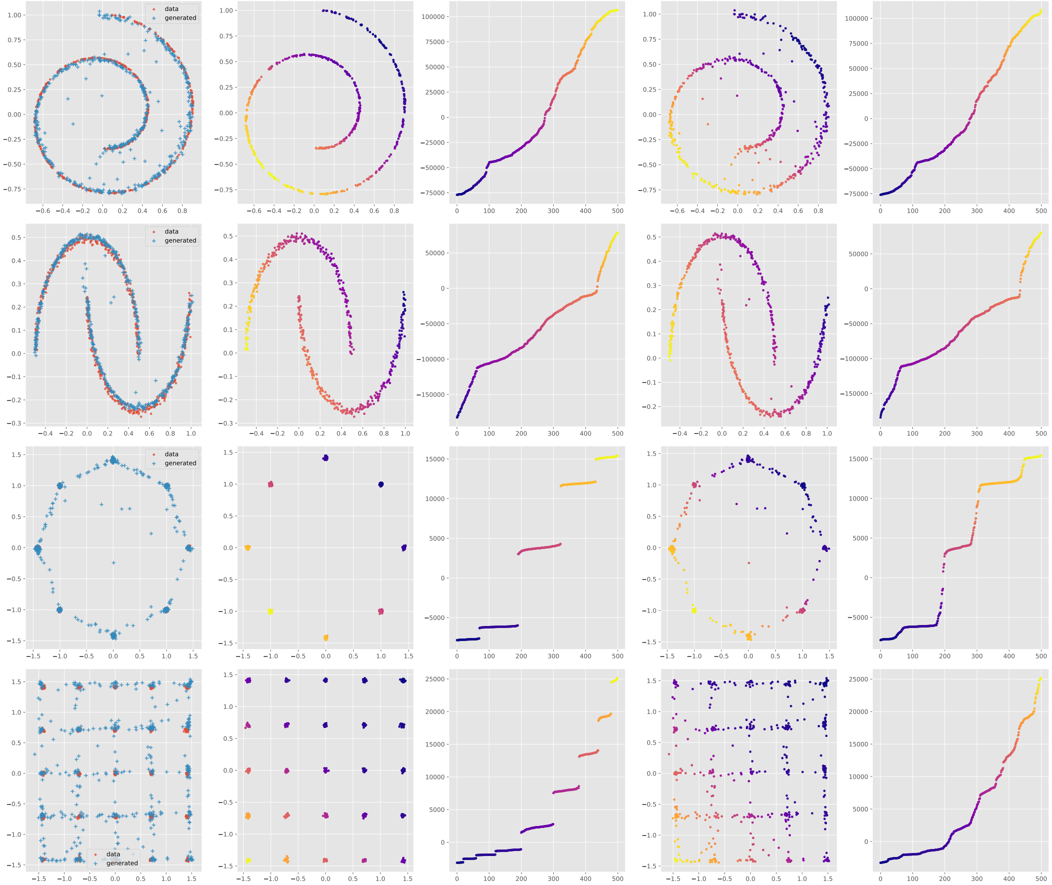

Figure 8: Visual results of *ACT* for generated samples (blue dots) compared to real samples (red dots) on Swiss Roll, Half Moons, 8-Gaussian mixture, and 25-Gaussian mixture. The second and third columns map the data points and their corresponding *navigator* logits by color; The fourth and fifth columns map the generated points and their corresponding *navigator* logits by color.

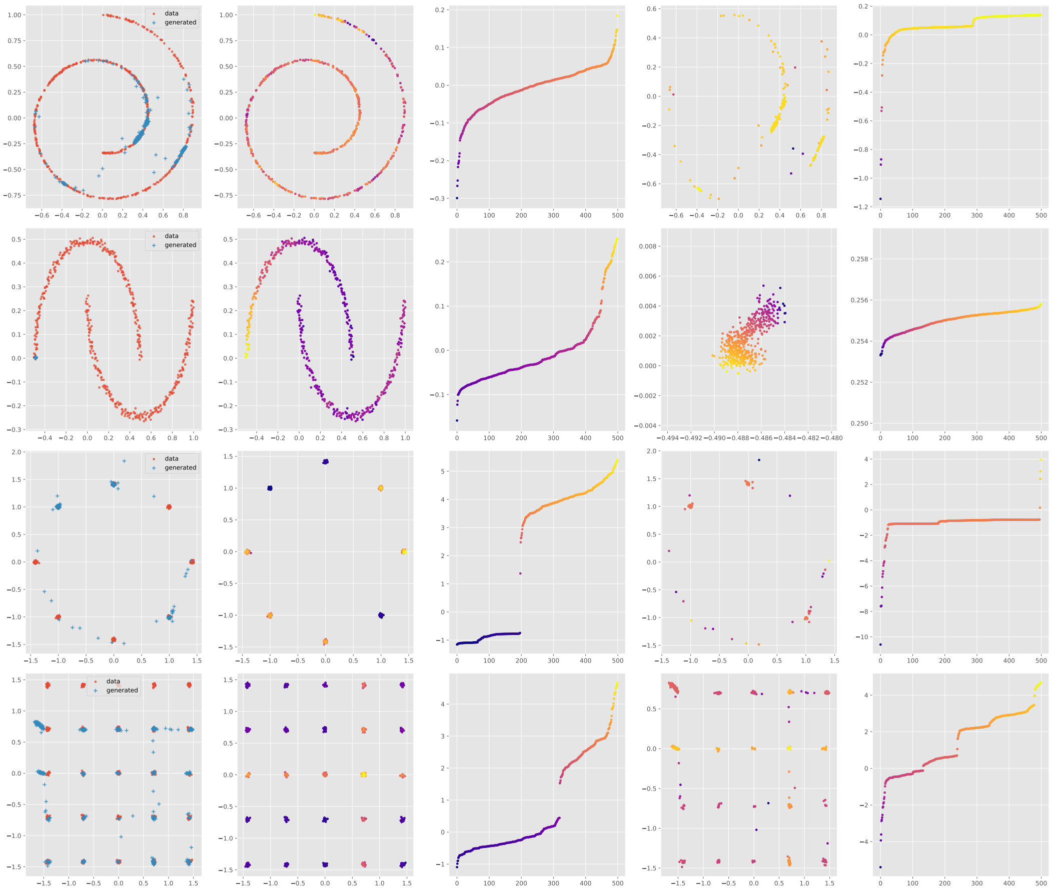

Figure 9: Visual results of *GAN* for generated samples (blue dots) compared to real samples (red dots) on Swiss Roll, Half Moons, 8-Gaussian mixture, and 25-Gaussian mixture. The second and the third columns map the data points and their corresponding *discriminator* logits by color; The fourth and fifth columns map the generated points and the corresponding *discriminator* logits by color.

B.3 RESULTS FOR ABLATION STUDY

Transport cost in pixel space *vs.* feature space We visualize the difference of using the transport cost in the pixel space and in the feature space here. In both Figs. 10 and 11, we test with MNSIT and CIFAR-10 data and with the $\mathcal{L}_2^2$ distance and cosine dissimilarity as the transport cost, respectively. For the MNIST dataset, due to its simple data structure, ACT can still be trained to generate meaningful digits, though some digits appear blurry. On the CIFAR-10, we can observe the model fails to generate any class of CIFAR images. As the dimensionality of the input space increases, using the distance in the pixel space as transport cost might lose the essential information for the transport and increases the training complexity of the navigator.

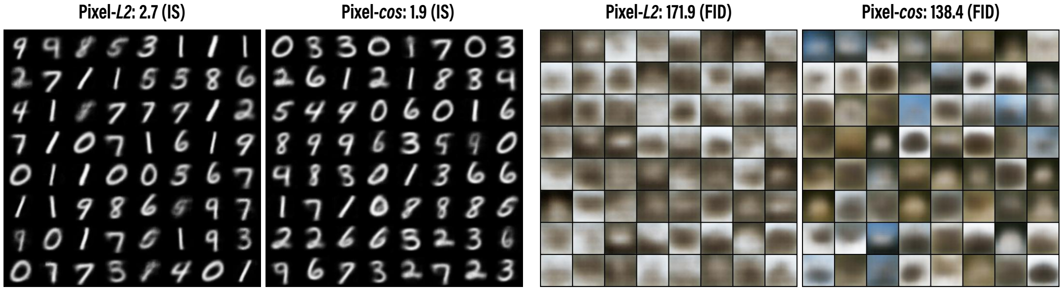

Figure 10: Visual results of generated samples on MNIST and CIFAR-10 using pixel-wise transport cost, with DCGAN (standard CNN) backbone.

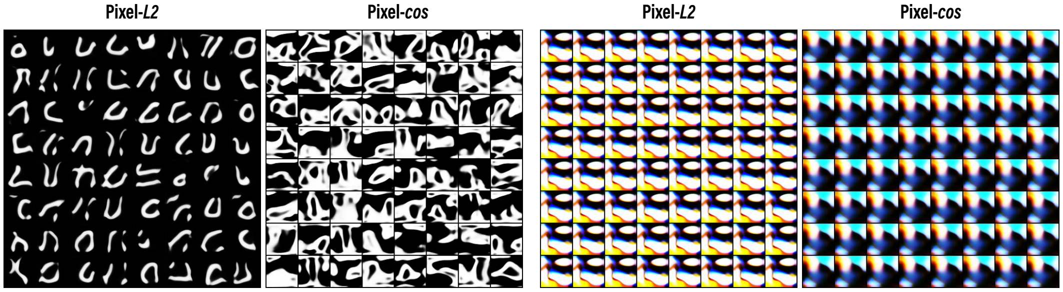

Figure 11: Visual results of generated samples on MNIST and CIFAR-10 using pixel-wise transport cost, with SNGAN (ResNet) backbone. The Inception and FID scores are not shown due to poor visual quality.

Training the critic with the discriminator loss of a vanilla GAN The quantitative and qualitative on MNIST, CIFAR-10, CelebA, and LSUN are shown in Fig. 12. We can observe the quality of generated samples, while clearly not as good as training the critic with the ACT divergence, can still catch up with some of the benchmarks in Table 1.

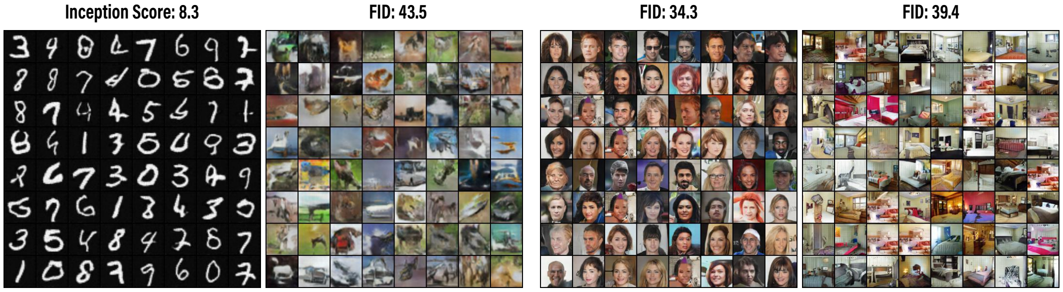

Figure 12: Visual results of using a standard cross-entropy discriminator loss in lieu of ACT divergence to train the critic of ACT.

B.4 MORE RESULTS ON IMAGE DATASETS

For the experiments on the image datasets, we provide more visual results in this part. Apart from the datasets described in the experiment part, we also test the capacity of single-channel image generation with the MNIST dataset. Considering the inception score and the FID score are designed for RGB natural images, we also calculate the inception score of the real testing sets for reference. The presented methods are all able to generate meaningful digits on MNIST. If we take a closer look at the digits, the digits generated with $\mathcal{L}_2$ cost is less natural than the one with cosine cost. Moreover, we show both unconditional and conditional generation results on CIFAR-10. For both unconditional and conditional generation, our proposed method achieves good quantitative and qualitative results.

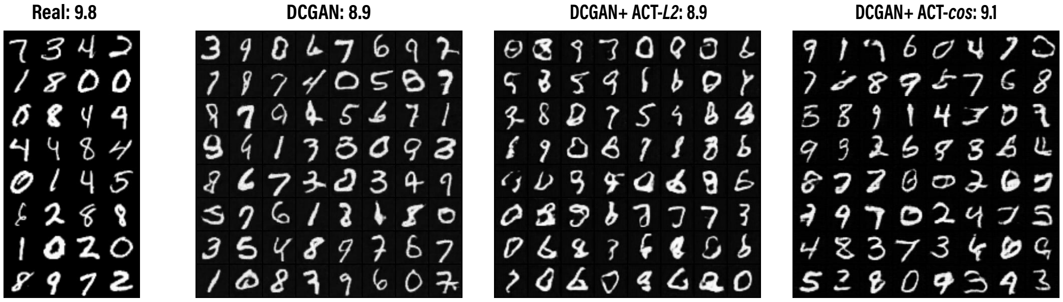

Figure 13: Unconditional generated samples and inception scores of MNIST, with DCGAN (standard CNN) backbone.

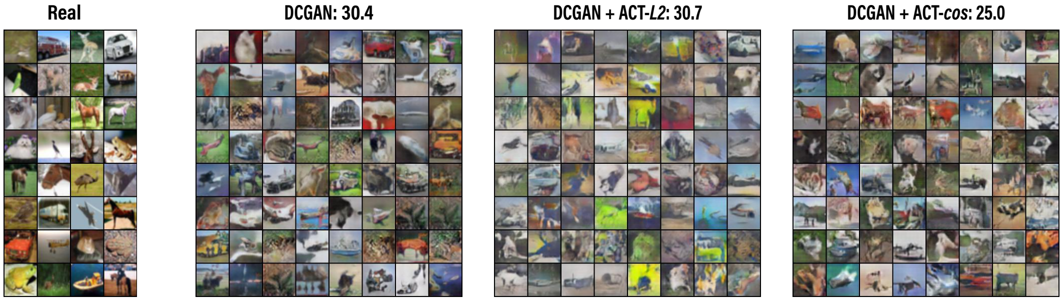

Figure 14: Unconditional generated samples and FIDs of CIFAR-10, with DCGAN (standard CNN) backbone.

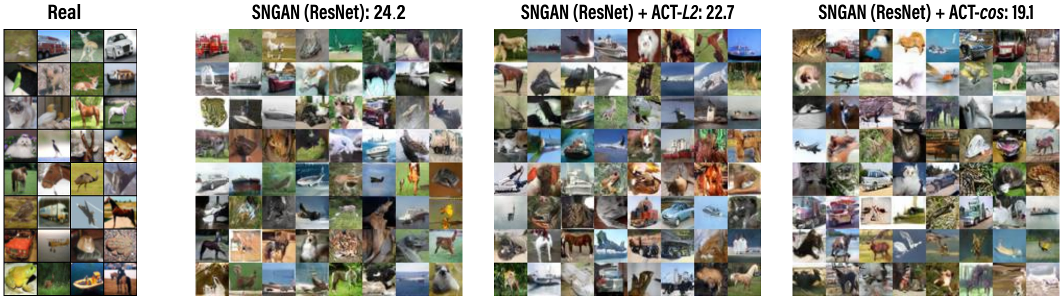
Real
SNGAN (ResNet): 24.2
SNGAN (ResNet) + ACT-*L2*: 22.7
SNGAN (ResNet) + ACT-*cos*: 19.1

Figure 15: Unconditional generated samples and FIDs of CIFAR-10, with SNGAN (ResNet) backbone.

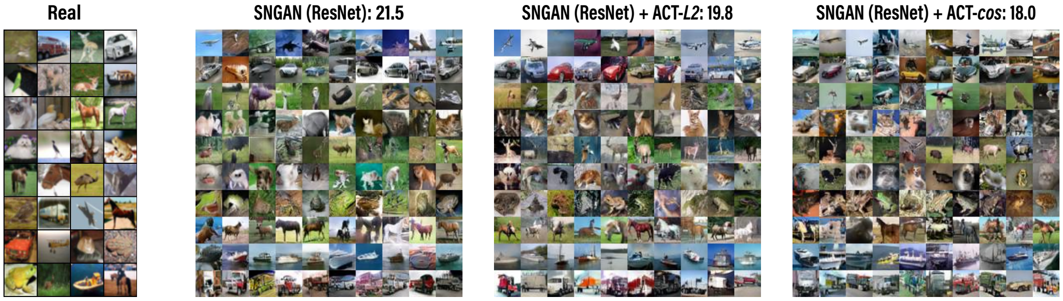
Real
SNGAN (ResNet): 21.5
SNGAN (ResNet) + ACT-*L2*: 19.8
SNGAN (ResNet) + ACT-*cos*: 18.0

Figure 16: Conditional generated samples and FIDs of CIFAR-10, with SNGAN (ResNet) backbone.

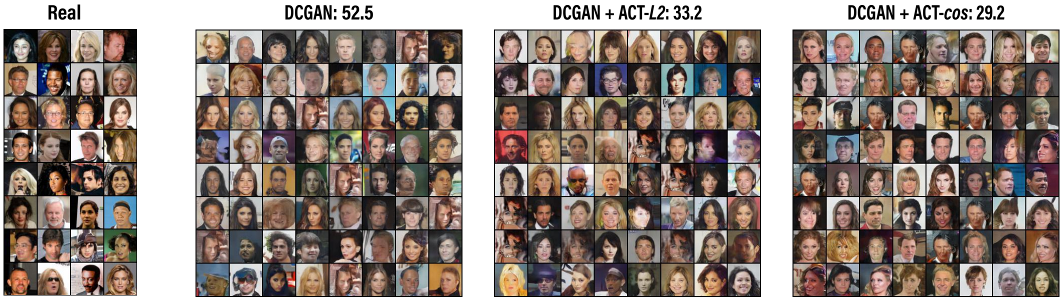

Figure 17: Generated samples and FIDs of CelebA, with DCGAN (standard CNN) backbone.

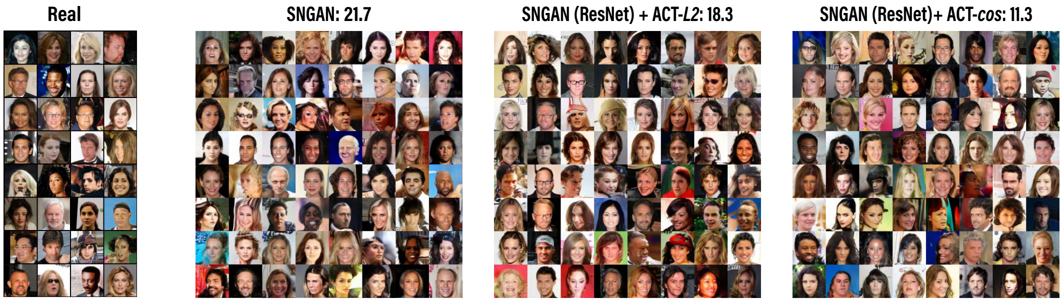

Figure 18: Generated samples and FIDs of CelebA, with SNGAN (ResNet) backbone.

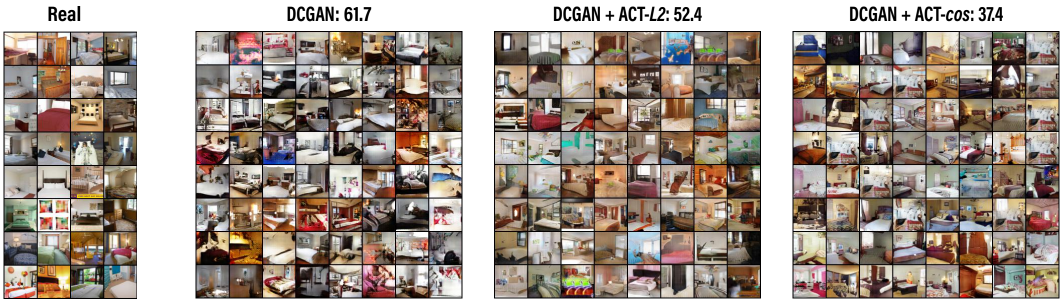

Figure 19: Generated samples and FIDs of LSUN, with DCGAN (standard CNN) backbone.

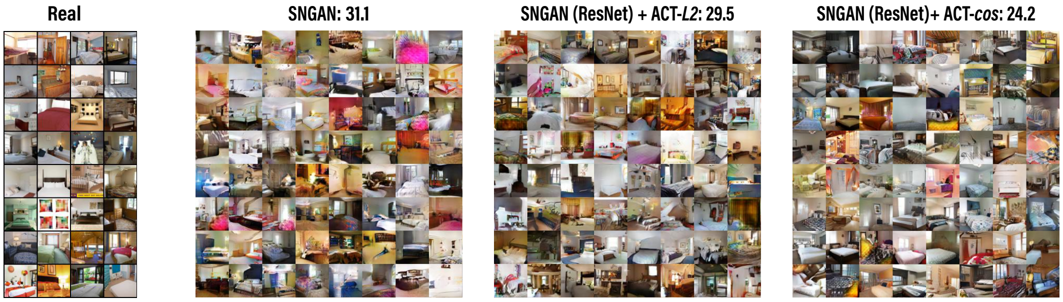

Figure 20: Generated samples and FIDs of LSUN, with SNGAN (ResNet) backbone.

B.5 EXPERIMENT DETAILS

Preparation of datasets We apply the commonly used training set of MNIST (50K images, 28×28 pixels) (Lecun et al., 1998), CIFAR-10 (50K images, 32×32 pixels) (Krizhevsky et al., 2009), CelebA (about 203K images, resized to 64×64 pixels) (Liu et al., 2015), and LSUN bedrooms (around 3 million images, resized to 64×64 pixels) (Yu et al., 2015). The images were scaled to range $[-1, 1]$. For MNIST, when calculate the inception score, we repeat the channel to convert each gray-scale image into a RGB format.

Network architecture and hyperparameters For the network architectures presented here, the slopes of all lReLU functions in the networks are set to 0.1 by default. For toy experiments, typically $10,000$ update steps are sufficient. However, our experiments show that the DGM optimized with the ACT divergence can be stably trained at least over $500,000$ steps (or possibly even more if allowed to running non-stop) regardless of whether the navigators are frozen or not after a certain number of iterations, where the GAN's discriminator usually diverges long before reaching that many iterations even if we do not freeze it after a certain number of iterations. For all image experiments, the output feature dimension of the navigators and that of the critic ($i.e.$, $\mathcal{T}_{\boldsymbol{\phi}}(\cdot), \mathcal{T}_{\boldsymbol{\eta}}(\cdot) \in \mathbb{R}^m$) are set to $m = 2048$. All models are able to be trained on a single GPU, such as NVidia GTX 1080-TI in our experiments, with $150,000$ generator updates (for CIFAR-10 we apply $50,000$ iterations).

To keep close to the configuration of the DCGAN and SNGAN experiments setting, we use the the Adam optimizer (Kingma and Ba, 2015) with learning rate $\alpha = 2 \times 10^{-4}$ and $\beta_1 = 0.5$, $\beta_2 = 0.99$ for the parameters of the generator, navigators, and critic. On the DCGAN backbone, we let all the modules update with the same frequency; while on the SNGAN backbone, the critic is updated once per 5 generator updating steps. The performance might be further improved with more careful fine-tuning. For example, the learning rate of the navigator parameter could be made smaller than that of the generator parameter. The true data minibatch size is fixed to $N = 64$ for all experiments. Moreover, with this batch-size we let the generated sample size M the same as minibatch size N for ACT computation and we have monitored the average time for each update step on a single NVidia GTX 1080-TI GPU: On CIFAR-10, each update step takes around 0.1s and 0.2s for DCGAN and SNGAN, respectively; For DCGAN and SNGAN backbone trained with ACT divergence each update step takes around 0.4s and 0.7s. On CelebA and LSUN, each update takes 0.6s and 0.7s for DCGAN and SNGAN, respectively; when trained with ACT, the elapsed time for each update increases to 3.3s and 3.6s, respectively.

Table 3: Network architecture for toy datasets (V indicates the dimensionality of data).

(a) Generator $G_{\boldsymbol{\theta}}$

$\boldsymbol{\epsilon} \in \mathbb{R}^{50} \sim \mathcal{N}(0,1)$
$50 \rightarrow 100$, dense, lReLU
$100 \rightarrow 50$, dense, lReLU
$50 \rightarrow V$, dense, linear

(b) Navigator $\mathcal{T}_{\boldsymbol{\phi}}$

$\boldsymbol{x} \in \mathbb{R}^V$
$2 \rightarrow 100$, dense, lReLU
$100 \rightarrow 50$, dense, lReLU
$50 \rightarrow 1$, dense, linear

Table 4: DCGAN architecture for CIFAR-10 datasets ($h = w = 4$).

(a) Generator $G_{\boldsymbol{\theta}}$

$\boldsymbol{\epsilon} \in \mathbb{R}^{128} \sim \mathcal{N}(0, 1)$
$128 \rightarrow 4 \times 4 \times 512$, dense, linear
4×4, stride=2 deconv. BN 256 ReLU
4×4, stride=2 deconv. BN 128 ReLU
4×4, stride=2 deconv. BN 64 ReLU
3×3, stride=1 conv. 3 Tanh

(b) Navigator $\mathcal{T}_{\boldsymbol{\phi}}$ / Critic $\mathcal{T}_{\boldsymbol{\eta}}$

$\boldsymbol{x} \in [-1, 1]^{32 \times 32 \times 3}$
3×3, stride=1 conv 64 lReLU 4×4, stride=2 conv 64 lReLU
3×3, stride=1 conv 128 lReLU 4×4, stride=2 conv 128 lReLU
3×3, stride=1 conv 256 lReLU 4×4, stride=2 conv 256 lReLU
3×3, stride=1 conv. 512 lReLU
$h \times w \times 512 \rightarrow m$, dense, linear

Table 5: DCGAN architecture on for CelebA and LSUN datasets ($h = w = 4$).

(a) Generator $G_{\boldsymbol{\theta}}$

$\boldsymbol{\epsilon} \in \mathbb{R}^{128} \sim \mathcal{N}(0, 1)$
$128 \rightarrow 4 \times 4 \times 1024$, dense, linear
4×4, stride=2 deconv. BN 512 ReLU
4×4, stride=2 deconv. BN 256 ReLU
4×4, stride=2 deconv. BN 128 ReLU
4×4, stride=2 deconv. BN 64 ReLU
3×3, stride=1 conv. 3 Tanh

(b) Navigator $\mathcal{T}_{\boldsymbol{\phi}}$ / Critic $\mathcal{T}_{\boldsymbol{\eta}}$

$\boldsymbol{x} \in [-1, 1]^{64 \times 64 \times 3}$
4×4, stride=2 conv 64 lReLU 4×4, stride=2 conv BN 128 lReLU
4×4, stride=2 conv BN 256 lReLU
3×3, stride=1 conv BN 512 lReLU
$h \times w \times 512 \rightarrow m$, dense, linear

Table 6: ResNet architecture on for CIFAR-10 datasets.

(a) Generator $G_{\boldsymbol{\theta}}$

$\boldsymbol{\epsilon} \in \mathbb{R}^{128} \sim \mathcal{N}(0, 1)$
$128 \rightarrow 4 \times 4 \times 256$, dense, linear
ResBlock up 256
ResBlock up 256
ResBlock up 256
BN, ReLU, 3×3 conv, 3 Tanh

(b) Navigator $\mathcal{T}_{\boldsymbol{\phi}}$ / Critic $\mathcal{T}_{\boldsymbol{\eta}}$

$\boldsymbol{x} \in [-1, 1]^{32 \times 32 \times 3}$
ResBlock down 128
ResBlock down 128
ResBlock 128
ResBlock 128
ReLU
Global sum pooling
$h = 128 \rightarrow m$, dense, linear

Table 7: ResNet architecture on for CelebA and LSUN datasets.

(a) Generator $G_{\boldsymbol{\theta}}$

$\boldsymbol{\epsilon} \in \mathbb{R}^{128} \sim \mathcal{N}(0, 1)$
$128 \rightarrow 4 \times 4 \times 1024$, dense, linear
ResBlock up 512
ResBlock up 256
ResBlock up 128
ResBlock up 64
BN, ReLU, 3×3 conv, 3 Tanh

(b) Navigator $\mathcal{T}_{\boldsymbol{\phi}}$ / Critic $\mathcal{T}_{\boldsymbol{\eta}}$

$\boldsymbol{x} \in [-1, 1]^{64 \times 64 \times 3}$
ResBlock down 128
ResBlock down 256
ResBlock down 512
ResBlock down 1024
ReLU Global sum pooling
$h = 1024 \rightarrow m$, dense, linear