A. Supplementary Results

A.1. Datasets

We evaluate our models across five different datasets: ImageNet [13], CelebA [36], CIFAR-10 [30], STL-10 [12], and CIFAR-100 [30]. In general, for preprocessing our images, we follow settings in [10, 43]. Specifically, for ImageNet, we use the 1.3M training images downsampled to size 128×128. For CelebA, we use the aligned version of the 200K images downsampled to size 128×128. For CIFAR-10 and CIFAR-100 we use all 50K training images, and for STL-10, we use all 100K unlabeled images downsampled to size 48×48.

A.2. Training Settings

For all models, we use Residual Network [21] backbones following [42]. For training the models on all datasets, we adopt the Adam optimizer [27] with a learning rate of 2×10^{-4} and batch size of 64, following [20, 42]. Specifically, for CIFAR-10, CIFAR-100 and STL-10, we follow settings in [43] by linearly decaying learning rate over 100K generator steps, each taken every 5 discriminator update steps. For ImageNet, we follow [42] by increasing the number of generator updates to 450K steps instead, but with no learning rate decay. For CelebA, we follow [10] by taking 100K generator steps, each taken after 2 discriminator updates and with no learning rate decay.

We emphasize that for *fairness* in our comparisons, we re-implemented all considered models using the same code base and framework, and trained all models under the *exact same training conditions* for each dataset.

A.3. Evaluation Settings

In our work, we use three different evaluation metrics: Fréchet Inception Distance (FID) [23] and Kernel Inception Distance (KID) [7] to evaluate generated image diversity, and Inception Score (IS) [54] to evaluate image quality.

Fréchet Inception Distance Firstly, FID is a popular metric measuring the diversity of generated images, which we adopt for ease of comparisons since it is widely used in the literature. Formally, FID computes the Wasserstein-2 Distance between features produced by a pre-trained Inception [56] network for input real and generated images, and is defined as:

$$d_{\text{FID}} = \mu_r - \mu_g{}_2^2 + (\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}) \tag{10}$$

where μ_r and Σ_r denotes the mean and covariance of feature vectors produced by forwarding real images through a pre-trained Inception [56] network, and μ_g and Σ_g similarly represents the equivalent for fake images. Intuitively, FID measures the diversity of the generated images, since the features of the generated images should ideally have a small distance with those of real images if they look similar on average. However, we note that FID can produce highly biased estimates [7], where using larger sample sizes can produce better scores, which can causes FID comparisons to be often mismatched [31] in practice. Thus, we emphasize for fairness in comparisons, we use the *exact* same number of real and fake images for computing FID.

Kernel Inception Distance KID is an alternative metric highly correlated with FID that also measures diversity of images, but produces unbiased estimates [7], which is useful for corroborating our findings on FID. Formally, KID measures the square of the Maximum Mean Discrepancy (MMD) [17] between two probability distributions in a metric space, and can be defined as:

$$\begin{aligned} d_{\text{KID}} &= \text{MMD}^2(X, Y) \\ &= \frac{1}{m(m-1)} \sum_{i \neq j}^{m} k(x_i, x_j) \\ &+ \frac{1}{n(n-1)} \sum_{i \neq j}^{n} k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j) \end{aligned} \tag{11}$$

for two random variables X and Y from different distributions, sample sizes m and n, and k is the polynomial kernel defined as:

$$k(x, y) = (\frac{1}{d} x^T y + 1)^3 \tag{12}$$

where d represents the dimensions of the samples. Intuitively, MMD measures the distance between distributions using a function from a class of witness functions such that if the true distance between the distributions is zero, the distance between the mean embeddings produced by this function will also be zero. Here, the polynomial kernel is cubic in order to measure the first three moments of the distributions (mean, variance, and skewness), and the embedding is defined on the feature space through the Inception network. Similar to FID, we use the same number of real and fake images for all models when computing KID.

Inception Score Finally, IS aims to measure the realism of generated images using the same Inception network, and can be formally defined as:

$$d_{IS} = \exp(\mathbb{E}_{x \sim p_g} \mathcal{D}_{\text{KL}}(p(y|x)||p(y))) \tag{13}$$

where a high score is achieved if the conditional class distribution $p(y|x)$ has low entropy and the marginal class distribution $p(y)$ has high entropy, causing a large KL divergence between the two distributions for some samples x from the generated image distribution p_g. Intuitively, a large score is

Metric	K	DCGAN	DCGAN + IM
# Modes	1/4	27.67 ± 0.47	$\mathbf{62.00 \pm 1.63}$
# Modes	1/2	610.00 ± 8.83	$\mathbf{716.67 \pm 1.25}$
$\mathcal{D}_{\text{KL}}(p\|\|q)$	1/4	5.44 ± 0.01	$\mathbf{4.68 \pm 0.01}$
$\mathcal{D}_{\text{KL}}(p\|\|q)$	1/2	1.98 ± 0.01	$\mathbf{1.64 \pm 0.01}$

Table 5: Number of modes (higher is better) recovered by the generator on the Stacked MNIST dataset, where the maximum value is 1000; and KL divergence $\mathcal{D}_{\text{KL}}(p||q)$ between the distribution of generated modes p and the uniform distribution q, where lower is better. '+ IM" refers to adding our proposed InfoMax-GAN objective.

produced if the Inception network gives a high probability to one class, indicating it looks realistically in one class. Thus, IS tends to correlate well with human assessment for quality of images [54].

Sample sizes For all FID scores reported in this paper, we compute them using 50K real samples and 10K fake samples across 3 random seeds to report the mean and standard deviation of the scores. As 50K real samples are much lesser than the 1.3M images in ImageNet, we randomly sample without replacement 50 images from each of the 1000 classes to compute the real image statistics, to avoid high bias in the results. We emphasize that for fairness in comparisons, we used the same number of real and fake samples when computing FID, since FID can produce highly bias estimates at different sample sizes [7]. In fact, we note that *lower FID scores* can indeed be obtained if we simply use larger sample sizes, particularly for larger datasets like ImageNet. However, our experiments show that in practice, the performance *margins* remain the same above our current configuration. For KID, we follow the same procedure for all datasets but use 50K real and fake samples instead. Finally, for IS, we use 50K fake samples.

We emphasize that all these evaluation settings are kept the same for all model evaluations for each dataset, in order to ensure fairness and accuracy in our comparisons.

A.4. Improved Mode Recovery

Following settings in [41], we re-implement the DCGAN [52] in [41] and evaluate its ability in recovering all 1000 modes of the Stacked MNIST dataset [41], composed by randomly stacking 3 grayscale MNIST [33] digits into an RGB image, resulting in 1000 possible modes. We use a pre-trained MNIST classifier to classify each color channel of a generated image, and the model is said to recover 1 mode if it generates at least 1 image for that mode. We similarly set $K \in \{\frac{1}{4}, \frac{1}{2}\}$, where K indicates the size of the discriminator relative to the generator. Intuitively, the smaller K is, the easier it is for the generator to fool the discriminator with just a few modes, resulting in less modes recovered. Furthermore, we compute the KL divergence $\mathcal{D}_{\text{KL}}(p||q)$ between the generated mode distribution p and optimal uniform distribution of the modes q. We see from Table 5 that our method helps to recover more modes for all K, with the recovered distribution having a consistently lower KL divergence with the ideal uniform distribution as a result.

A.5. Additional Ablation Studies

In this section, we analyze the impact of our framework design choices and their performance impact.

Relative Scale of Objective From Figure 7, we see in both our chosen hyperparameters of $\alpha = \beta = 0.2$ and the other extreme of $\alpha = \beta = 1.0$, the InfoMax objective loss decays very quickly relative to the GAN loss. In practice, we found that $\alpha = \beta = 0.2$ performs better, which could be attributed to the relative magnitude of the InfoMax objective loss at the start of the training. When $\alpha = \beta = 0.2$, the scales of the GAN and InfoMax objective losses are approximately equal initially. We highlight this is the same loss scaling principle applied in [11].

Position of feature maps While we have chosen the local and global features to be the penultimate and final features of the discriminator encoder respectively, we examine the effect of alternative designs. For clarity, we note there is only *one* global feature vector, which is the final feature output of the encoder. Correspondingly, our design can be called local-global, and other designs involving extracting intermediate local feature maps $C_{\psi,k}(x), 1 \leq k \leq n$ can be described as local-local. However, in practice, our original design of local-global is the only feasible option compared to local-local option, mainly due to the memory consumption: for any two feature maps of spatial size $M_1 \times M_1$ and $M_2 \times M_2$ respectively, we have the space complexity as $O(NM_1^2M_2^2R)$ for batch size N and RKHS R. Fixing the first feature map size, the local-local approach has space complexity growing quadratically on the second feature map size M_2, which is in turn dependent on the image resolution. On the other hand, the local-global approach effectively sets $M_2 = 1$, which dramatically reduces memory consumption.

In fact, in practice, we found the local-local approach cannot scale to datasets above 32×32 resolution as it would exceed 11GB for a single GPU. To still test this approach on the 32×32 resolution CIFAR-10 dataset, we reduce the memory consumption by randomly sampling only half of local spatial vectors from each feature map. Even so, the memory consumption is approximately 7 times of the local-global approach, making it highly memory intensive. In contrast, the local-global approach scales for even high

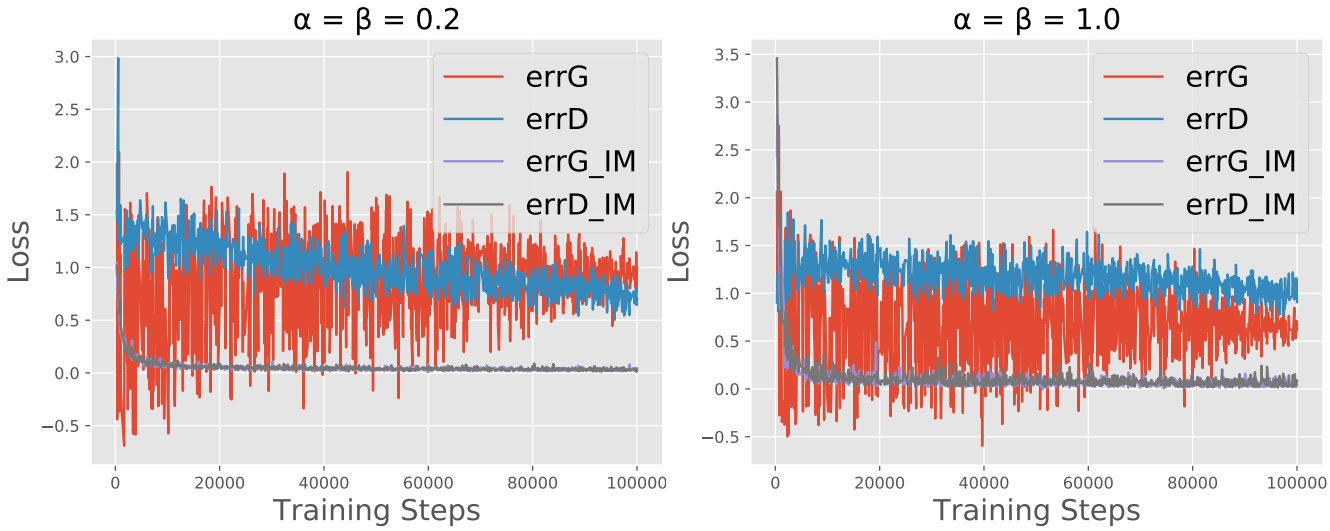


Figure 7: We show that the InfoMax objective loss decays very quickly regardless of the choice of scale for both α and β. `errD` and `errG` represents the GAN losses for the discriminator and generator respectively, and similarly, `errD_IM` and `errG_IM` represents the InfoMax objective losses for the discriminator and generator respectively.

resolution (e.g. 128×128) datasets and takes only a small portion of the memory size compared to the GAN models. Importantly, the local-local approach worsens FID by **3.1 points** from 17.14 ± 0.20 to 20.20 ± 0.05. Thus, this ablation study show that in practice, our current design is the most optimal for achieving both performance and memory consumption gains.

Effect of spectral normalizing critic Interestingly, using spectral normalisation for the InfoMax-GAN critic networks leads to FID improvements. On CIFAR-10, using spectral normalisation for these critic networks improved FID by **1.5 points** from 18.67 ± 0.25 to 17.14 ± 0.20. We conjecture this could be related to the Wasserstein Dependency Measure [48], a variant of mutual information which replaces the KL divergence term with Wasserstein distance, as measured using encoders from the class of 1-Lipschitz functions. However, in contrast to this work, our method enforces 1-Lipschitzness of the encoder using spectral normalization rather than gradient penalty. A theoretical treatment of this relationship is beyond the scope of this paper, which we leave as future work.

A.6. Generated Image Samples

In Figure 8, we show generated images at 128×128 resolution for CelebA. In general, we observe that images generated by InfoMax-GAN have less visual artifacts for both the background and facial attributes, with even attributes like spectacles and caps generated. In contrast, both SNGAN and SSGAN generated images tend to have more severe background artifacts, with certain prominent facial features like eyes and noses not well blended together. This blending problem is more commonly seen in SSGAN generated images, which may explain its worse FID performance compared to both SNGAN and InfoMax-GAN. We further provide image samples randomly generated for all datasets in Appendix A.6.

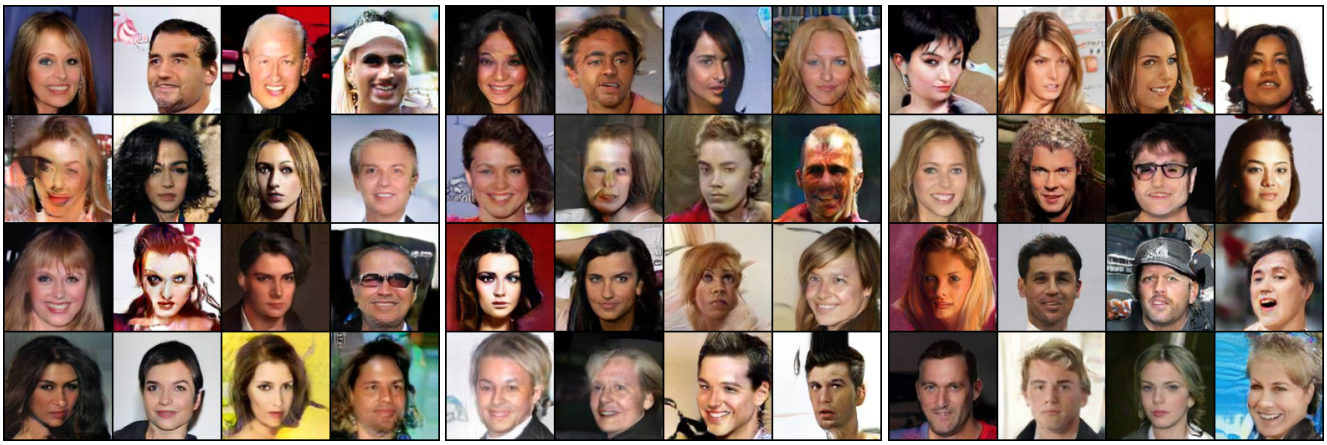
For further qualitative comparisons, we present randomly sampled, non-cherry picked images generated by SNGAN and InfoMax-GAN for all datasets in Figures 9, 10 and 11. We qualitatively observe that the images are more diverse and have sharper shapes after the use of an InfoMax objective.

B. InfoGAN Comparison

For clarity and disambiguity, Table 6 illustrates the differences in our work with InfoGAN. Our works have different focuses: InfoGAN focuses on learning disentanglements in image generation, while we focus on improving image synthesis as a whole.

C. Model Architectures

We detail the exact GAN architectures used for all datasets in Tables 7, 8, 9. We also detail the architectures for projecting the local and global features to a higher dimensional RKHS for solving the InfoNCE task in Table 10.



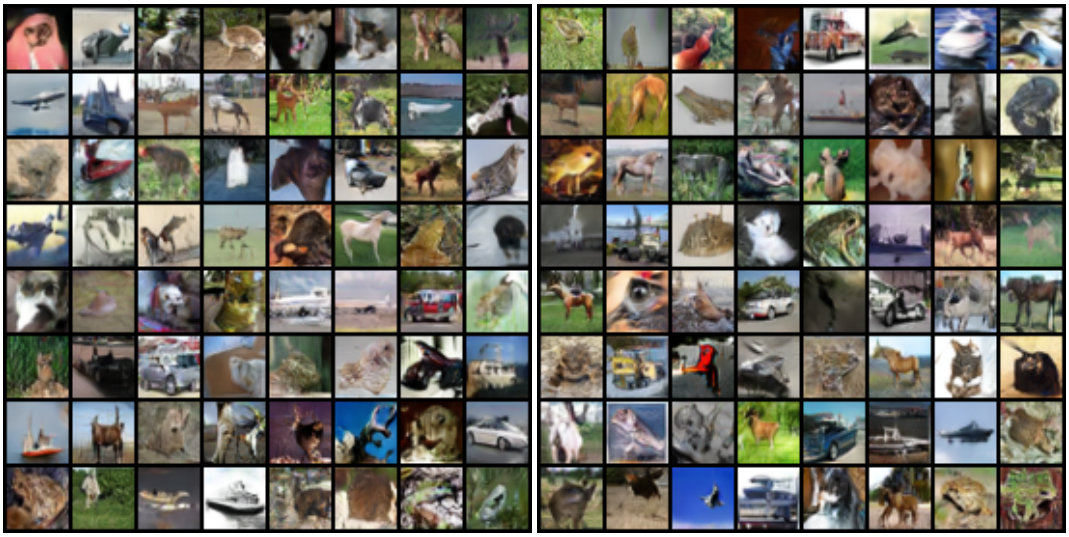
(a) (b) (c)

Figure 8: Generated CelebA images at 128×128 resolution for (a) SNGAN, (b) SSGAN, and (c) InfoMax-GAN. In general, we observe InfoMax-GAN generated images have less visual artifacts in both the background and the facial attributes. We note these images are randomly generated and non-cherry picked.

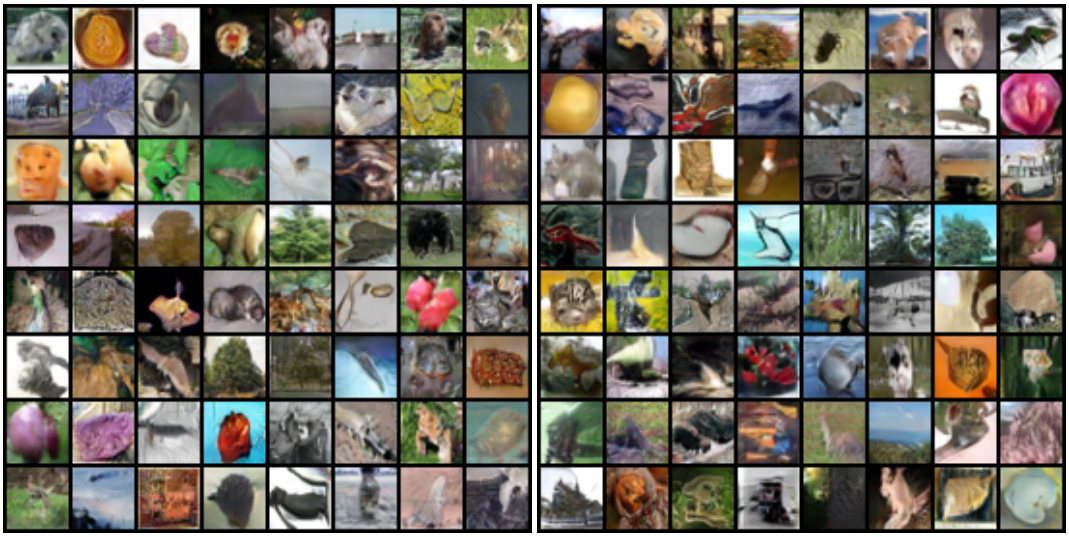
Work	**Target Outcome**	**MI Objective**	**MI Approximation Technique**
InfoGAN [11]	Disentangled representation learning by using an input encoding c to the generator to control its output.	$\mathcal{I}(c; G(z, c))$	Variational Information Maximization [4]
InfoMax-GAN (ours)	Improve image synthesis by reducing catastrophic forgetting of discriminator and mode collapse of generator.	$\mathcal{I}(C_\psi(X); E_\psi(X))$	InfoNCE [47] Task

Table 6: Comprehensive differences with InfoGAN. Our work mainly differs in the intended outcome, the objective to meet the outcome, and the approximation technique needed to solve the objective.

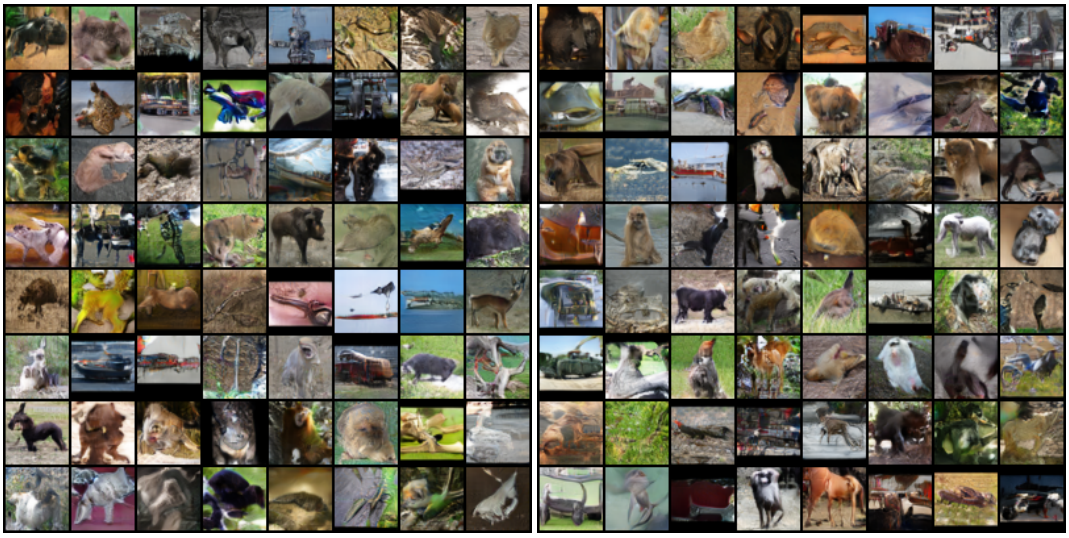
Figure 9: Randomly sampled and non-cherry picked images for SNGAN (left) and InfoMax-GAN (right) for CIFAR-10, CIFAR-100, and STL-10.



(a) CIFAR-10.



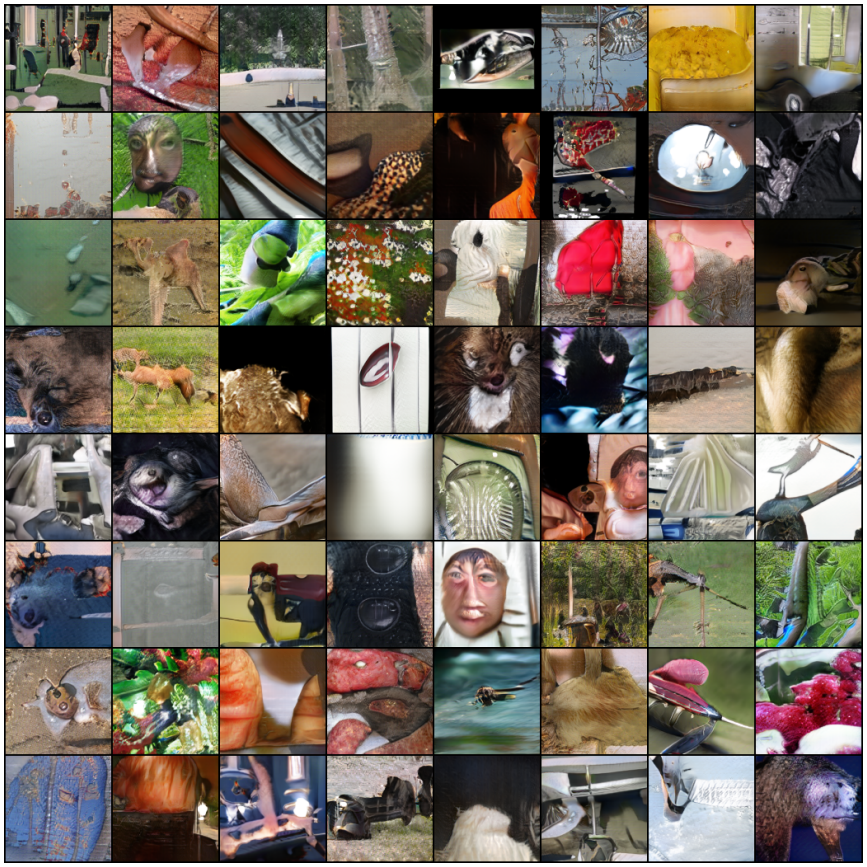
(b) CIFAR-100.



(c) STL-10.

Figure 10: Randomly sampled and non-cherry picked generated CelebA images for SNGAN (top) and InfoMax-GAN (bottom).

Figure 11: Randomly sampled and non-cherry picked generated ImageNet images for SNGAN (top) and InfoMax-GAN (bottom).



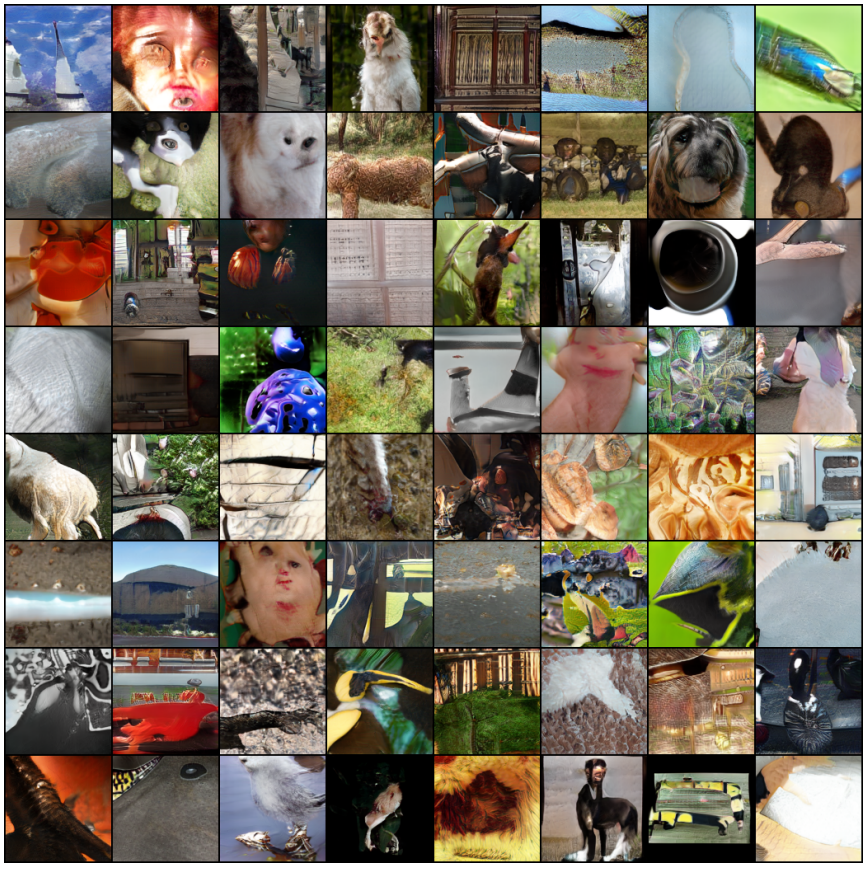


Table 7: Network architectures for the CIFAR-10 and CIFAR-100 datasets, which follows exact settings in [42].

(a) Generator

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, 1)$
Linear, $4 \times 4 \times 256$
ResBlock up 256
ResBlock up 256
ResBlock up 256
BN; ReLU; 3×3 conv, 3; Tanh

(b) Discriminator

RGB image $x \in \mathbb{R}^{32 \times 32 \times 3}$
ResBlock down 128
ResBlock down 128
ResBlock 128 $\rightarrow$ Local Features
ResBlock 128
ReLU
Global Sum Pooling $\rightarrow$ Global Features
Linear $\rightarrow$ 1

(c) Self-supervised Discriminator

RGB image $x \in \mathbb{R}^{32 \times 32 \times 3}$
ResBlock down 128
ResBlock down 128
ResBlock 128 $\rightarrow$ Local Features
ResBlock 128
ReLU
Global Sum Pooling $\rightarrow$ Global Features
Linear $\rightarrow$ 1; Linear $\rightarrow$ 4

Table 8: Network architectures for the STL-10 dataset, which follows exact settings in [42].

(a) Generator

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, 1)$
Linear, $6 \times 6 \times 512$
ResBlock up 256
ResBlock up 128
ResBlock up 64
BN; ReLU; 3×3 conv, 3; Tanh

(b) Discriminator

RGB image $x \in \mathbb{R}^{48 \times 48 \times 3}$
ResBlock down 64
ResBlock down 128
ResBlock down 256
ResBlock down 512 $\rightarrow$ Local Features
ResBlock 1024
ReLU
Global Sum Pooling $\rightarrow$ Global Features
Linear $\rightarrow$ 1

(c) Self-supervised Discriminator

RGB image $x \in \mathbb{R}^{48 \times 48 \times 3}$
ResBlock down 64
ResBlock down 128
ResBlock down 256
ResBlock down 512 $\rightarrow$ Local Features
ResBlock 1024
ReLU
Global Sum Pooling $\rightarrow$ Global Features
Linear $\rightarrow$ 1; Linear $\rightarrow$ 4

Table 9: Network architectures for the CelebA and ImageNet datasets. This follows the exact settings in the official SNGAN code [1].

(a) Generator

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, 1)$
Linear, $4 \times 4 \times 1024$
ResBlock up 1024
ResBlock up 512
ResBlock up 256
ResBlock up 128
ResBlock up 64
BN; ReLU; 3×3 conv, 3; Tanh

(b) Discriminator

RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$
ResBlock down 64
ResBlock down 128
ResBlock down 256
ResBlock down 512 $\rightarrow$ Local Features
ResBlock down 1024
ResBlock 1024
ReLU
Global Sum Pooling $\rightarrow$ Global Features
Linear $\rightarrow$ 1

(c) Self-supervised Discriminator

RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$
ResBlock down 64
ResBlock down 128
ResBlock down 256
ResBlock down 512 $\rightarrow$ Local Features
ResBlock down 1024
ResBlock 1024
ReLU
Global Sum Pooling $\rightarrow$ Global Features
Linear $\rightarrow$ 1; Linear $\rightarrow$ 4

Table 10: InfoNCE projection architectures, which follow what were proposed in [24]. In practice, we extract the local features and global features from the penultimate and final residual blocks of the discriminator respectively. This decides the corresponding values of feature depth K.

(a) Local features projection architecture.

1×1 Conv, K; 1×1 Conv, $R \rightarrow$ Shortcut
ReLU
1×1 Conv, R + Shortcut

(b) Global features projection architecture.

Linear $\rightarrow K$; Linear $\rightarrow R \rightarrow$ Shortcut
ReLU
1×1 Conv, R + Shortcut