Appendix

A. Theoretical Results

We first define the Lipschitz constant L_f again for a better readability. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a mapping function. Then, L_f is the minimum real number such that:

$$|f(x) - f(y)| \leq L_f \|x - y\|, \forall x, y \in \mathbb{R}^n. \tag{1}$$

Lemma 3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function and L_f be the Lipschitz constant of f. Then the Lipschitz constraint (1) is equivalent to*

$$\|\nabla_x f(x)\| \leq L_f, \forall x \in \mathbb{R}^n. \tag{2}$$

Proof. We first prove the sufficient condition.
($\Rightarrow$) From the definition of Lipschitz constraint (1), we know

$$|f(x) - f(y)| \leq L_f \|x - y\|. \tag{3}$$

Now, we consider the norm of directional derivative at x along with the direction of $(y - x)$:

$$\langle \nabla f(x), \frac{y - x}{\|y - x\|} \rangle = \lim_{y \to x} \frac{|f(y) - f(x)|}{\|x - y\|} \leq L_f, \tag{4}$$

where $\langle \cdot, \cdot \rangle$ is the inner product. Since the norm of gradient is the maximum norm of directional derivative, then

$$\|\nabla f(x)\| \leq L_f. \tag{5}$$

We then prove the necessary condition.
($\Leftarrow$) By the assumption, f is continuous and differentiable. Therefore, the conditions of *Gradient theorem* are satisfied, and thus we can only consider the line integral along the straight line from y to x:

$$\begin{aligned}
&|f(x) - f(y)| && \text{(6a)}\\
&= \Big| \int_y^x \nabla f(r) dr \Big| && \text{(6b)}\\
&= \Big| \int_0^1 \langle \nabla f(xt + y(1-t)), x - y \rangle dt \Big| && \text{(6c)}\\
&\leq \Big| \int_0^1 \|\nabla f(xt + y(1-t))\| \cdot \|x - y\| dt \Big| && \text{(6d)}\\
&\leq L_f \Big| \int_0^1 \|x - y\| dt \Big| && \text{(6e)}\\
&= L_f \|x - y\|. && \text{(6f)}
\end{aligned}$$

The theorem follows. □

Theorem 5. *Let $f_K : \mathbb{R}^n \to \mathbb{R}$ be a layer-wise 1-Lipschitz constrained network with K layers. Then the Lipschitz constant of the first k-layer network L_{f_k} is upper-bounded by $L_{f_{k-1}}$, i.e.,*

$$L_{f_k} \leq L_{f_{k-1}}, \forall k \in \{2 \cdots K\}. \tag{7}$$

Proof. Since all the layers including activation functions are all 1-Lipschitz constrained, *i.e.*,

$$\begin{gathered}
\|\mathbf{W}_k \cdot x - \mathbf{W}_k \cdot y\| \leq \|x - y\|, \forall x, y \in \mathbb{R}^{d_{k-1}} \\
L_{\phi_k} = 1.
\end{gathered} \tag{8}$$

We can infer the upper bound of feature distance at layer k by Eq.(8):

$$\begin{aligned}
&\|f_k(x) - f_k(y)\| \\
&= \|\phi_k(\mathbf{W}_k \cdot f_{k-1}(x) + \mathbf{b}_k) - \phi_k(\mathbf{W}_k \cdot f_{k-1}(y) + \mathbf{b}_k)\| \\
&\leq L_{\phi_k} \|(\mathbf{W}_k \cdot f_{k-1}(x) + \mathbf{b}_k) - (\mathbf{W}_k \cdot f_{k-1}(y) + \mathbf{b}_k)\| \\
&\leq L_{\phi_k} L_k \|f_{k-1}(x) - f_{k-1}(y)\| \\
&= \|f_{k-1}(x) - f_{k-1}(y)\|.
\end{aligned} \tag{9}$$

This result implies

$$\frac{\|f_k(x) - f_k(y)\|}{\|x - y\|} \leq \frac{\|f_{k-1}(x) - f_{k-1}(y)\|}{\|x - y\|}, \forall x, y \in \mathbb{R}^n. \tag{10}$$

The theorem follows. □

Theorem 6. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function which is modeled by a neural network, and all the activation functions of network f are piecewise linear. Then the normalized function $\hat{f}(x) = f(x) / \big(\|\nabla_x f(x)\| + \|f(x)\|\big)$ is 1-Lipschitz constrained,* i.e.*,*

$$\|\nabla_x \hat{f}(x)\| = \left\| \frac{\|\nabla f\|}{\|\nabla f\| + |f|} \right\|^2 \leq 1. \tag{11}$$

Proof. For simplicity, function arguments are ignored here.

By definition, the gradient norm of $\hat{f}(x)$ is:

$$\|\nabla \hat{f}\| = \left\| \nabla \left(\frac{f}{\|\nabla f\| + |f|} \right) \right\| \tag{12a}$$

$$= \left\| \frac{\nabla f \left(\|\nabla f\| + |f| \right) - f \nabla \left(\|\nabla f\| + |f| \right)}{\left(\|\nabla f\| + |f| \right)^2} \right\|. \tag{12b}$$

By simple chain rule, we know that:

$$\nabla \|\nabla f\| = \nabla^2 f \frac{\nabla f}{\|\nabla f\|}, \tag{13a}$$

$$\nabla |f| = \nabla f \frac{f}{|f|}. \tag{13b}$$

Since the network f contains only piecewise linear activation functions, the Hessian matrix $\nabla^2 f$ is a zero matrix. The Eq.(12b) can be simplified:

$$\|\nabla \hat{f}\| = \left\| \frac{\|\nabla f\|^2}{\left(\|\nabla f\| + |f| \right)^2} \right\| = \left\| \frac{\|\nabla f\|}{\|\nabla f\| + |f|} \right\|^2 \leq 1. \tag{14}$$

The theorem follows. □

B. Supplemental Experiments

Please note that the source codes are archived for the verification in supplementary materials.

B.1. Supplemental Ablation Study

Figure 1 compares the effectiveness of different activation functions in terms of IS and FID on CIFAR-10 dataset. The results show that the ReLU activation function achieves best IS and FID for different approaches. Moreover, the ReLU activation function with the proposed GN outperforms other state-of-the-art normalization and regularization approaches. It is worth noting that the original Softplus activation function achieves low IS and high FID for different approaches. However, by setting β to 20, the result can be significantly better since Softplus becomes similar to ReLU if β increases. Moreover, Figure 2 compares the effectiveness of different $\zeta(x)$ in terms of IS and FID on CIFAR-10 dataset. The results indicate that the variance of Inception Score and FID for GN_0 is large for different architectures and datasets. The proposed GN outperforms the alternatives, which is consistent to the experiments on STL-10 dataset.

B.2. Decision Boundary Visualization

We conduct an experiment similar to [11] for the visualization. The value surfaces of binary classification tasks are demonstrated in Figure 3. The results demonstrate that the value surface of vanilla GAN (Figure 3b) contains steep cliffs near to the decision boundary, which causes gradient explosion when the synthetic samples are located in this area. With the regularization or normalization applied to discriminator, the value surface becomes smooth in varying levels as shown in Figures 3(c)-(f).

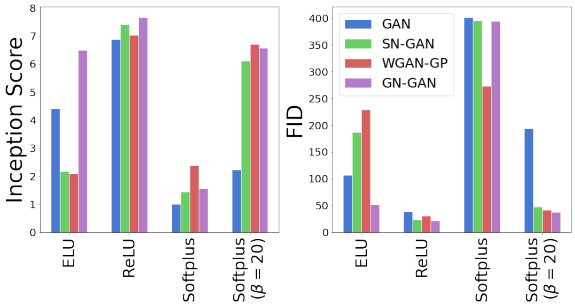


Figure 1: Comparison of activation functions including ELU [2], ReLU [10] and Softplus [3] on CIFAR-10.

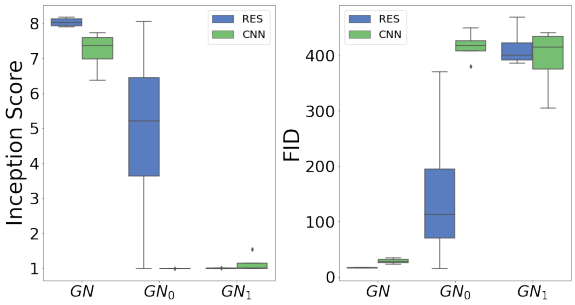


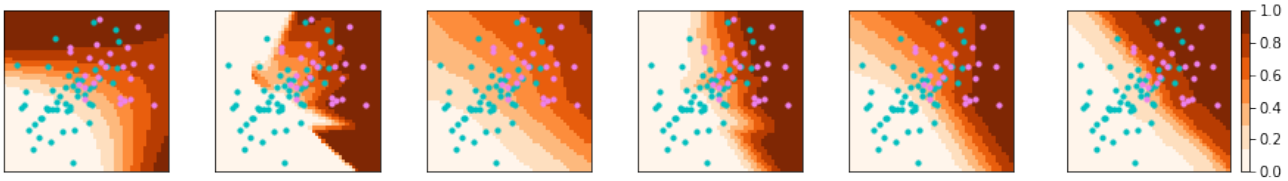
Figure 2: Comparison of variants of gradient normalization on CIFAR-10. The experiments include $\zeta(x) = |f(x)|$ (GN), $\zeta(x) = 1$ (GN_1) and $\zeta(x) = 0$ (GN_0).

B.3. Training Speed

Table 1 shows the training speed of different approaches with ResNet as the backbone network on CIFAR-10 dataset. All the training processes are performed on NVIDIA RTX 2080Ti five times, and we report the average results in terms of update iterations per second. The results show that different approaches require additional computation as compared to the Vanilla GAN. It is worth noting that although the training speed of the proposed GN is only compatible with 1-GP, the proposed GN outperforms the other approaches in terms of IS and FID. In other words, even with more computation, other approaches can not improve their results. On the other hand, the training process is offline, while the inference speed is the same for different approaches.

B.4. Loss Function Comparison

We further investigate the performance of the proposed GN with different loss functions. Notably, the Gradient Normalization makes the outputs of discriminators saturate



(a) Theoretical (b) Vanilla GAN (c) 0-GP (d) 1-GP (e) SN (f) GN

Figure 3: The theoretical and empirical value surfaces of discriminators which are parameterized by a 2-layer MLP with hidden size 512. Real samples are drawn from a 2D multivariate Gaussian and fixed for all discriminators, while the fake samples are sampled from the other 2D multivariate Gaussian infinitely. (a) The theoretically optimal discriminator $D^*(x) = p_r(x)/(p_r(x) + p_g(x))$. (f) Our gradient normalization.

Method	Generator (it/s)	Discriminator (it/s)
Vanilla	6.91	15.73
SN	6.52	14.41
1-GP	4.70	7.66
GN	3.68	6.48

Table 1: Training speed of generator and discriminator.

in range $[-1, 1]$, and thus the sigmoid at the end of discriminator can be eliminated when the non-saturating loss is used. Moreover, the hinge loss is equivalent to Wasserstein loss in the perspective of gradients when GN is used, *i.e.*,

$$\begin{aligned}\nabla\mathcal{L}_{hinge} =& \nabla\mathbb{E}_{x\sim p_g(x)}[\max(1+\hat{D}(x),0)]+ \\ & \nabla\mathbb{E}_{x\sim p_r(x)}[\max(1-\hat{D}(x),0)] \\ =& \nabla\mathbb{E}_{x\sim p_g(x)}[1+\hat{D}(x)]+ \\ & \nabla\mathbb{E}_{x\sim p_r(x)}[1-\hat{D}(x)] \\ =& \nabla\mathbb{E}_{x\sim p_g(x)}[\hat{D}(x)] - \nabla\mathbb{E}_{x\sim p_r(x)}[\hat{D}(x)] \\ =& \nabla\mathcal{L}_{wasserstein}\end{aligned} \tag{15}$$

Table 2 shows the evaluation results of different loss functions on CIFAR-10 in terms of Inception score and FID. Both ResNet and CNN architectures are reported. Since the Wasserstein loss is equivalent to hinge loss, the Wasserstein loss is not listed. The performance of GN-GANs is consistent with different loss functions.

C. Evaluation Details

Inception Score. For the Inception Score (IS), we divide 50k generated images into 10 partitions and calculate the average and the standard deviation of Inception Score over each partition. The final results are the average scores of different training sessions.

Frechet Inception Distance. The configurations of FID are described as follow. For the CIFAR-10 dataset, we use 50k generated samples vs. 50k training images and 10k generated samples vs. 10k test images. For the STL-10 dataset, we use 50k generated samples vs. 100k unlabeled images and 10k generated samples vs. 100k unlabeled images. For the CelebA-HQ, we use 30k generated samples vs. 30k training images. For the LSUN Church Outdoor, we use 50k generated samples vs. 126k training images. In the training process, models are trained on CIFAR-10 training set, STL-10 unlabeled images, CelebA-HQ training set and LSUN Church Outdoor training set.

Loss Function	IS↑	FID(train)↓	FID(test)↓
Standard CNN			
Hinge	7.67±.14	18.20±.12	22.24±.88
NS	7.78±.11	18.17±.61	22.36±.59
$\text{NS}_{-sigmoid}$	7.69±.16	18.93±.87	23.19±.86
ResNet			
Hinge	8.49±.11	11.13±.18	15.33±.16
NS	8.49±.11	10.97±.22	15.15±.29
$\text{NS}_{-sigmoid}$	8.49±.09	11.01±.26	15.14±.32

Table 2: Loss function comparison of GN-GANs on CIFAR-10. Note that the non-saturating loss without sigmoid at the last layer is denoted by $\text{NS}_{-sigmoid}$.

D. Experimental Details

Unconditional Image Generation on CIFAR-10 and STL-10. For the fair comparison, we use the ResNet architecture as well as the Standard CNN used in [9]. The last layer of ResNet, *i.e.*, global sum pooling, is replaced by the global average pooling. Moreover, all the weights of fully-connected layers and CNN layers are initialized by Kaiming Normal Initialization [4], and the biases are initialized to zero. We use Adam [7] as the optimizer with parameters $\alpha_G = 2 \times 10^{-4}$, $\alpha_D = 4 \times 10^{-4}$, $\beta_1 = 0$, $\beta_2 = 0.9$ and batch size $M = 64$. The learning rate linearly

decays to 0 through the training. The generator is updated once for every 5 discriminator update steps. All the training processes are stopped after the generator update 200k steps. For the data augmentation, the random horizontal flipping is applied for every method (including our method and re-implementation). The augmentation setting in Table 3 is used for Consistency Regularization [12]. For more qualitative results, please refer to Figures 4 and 5.

1.	RandomHorizontalFlipping(p=0.5)
2.	RandomPixelShifting(pixel=0.2×ImageSize)

Table 3: Augmentation for consistency regularization on CIFAR-10 and STL-10.

Conditional Image Generation on CIFAR-10. To show the results of conditional image generation on CIFAR-10 dataset, we compare the results of BigGAN [1], BigGAN with the Consistency Regularization (CR), BigGAN with the proposed GN, and BigGAN with the proposed GN and CR. Here, the discriminator in the conditional GAN is considered as a conditional function, *i.e.*, $D_y(x)$, instead of the multi-variable function, *i.e.*, D(x, y). Therefore, the Gradient Normalization can be formulated as follows:

$$\hat{D}_y(x) = \frac{D_y(x)}{\|\nabla_x D_y(x)\| + \|D_y(x)\|}, \tag{16}$$

where $D_y(x)$ is the discriminator conditional on y. Similarly, by Theorem 5, $\hat{D}_y(x)$ is a Lipschitz constrained network with respect to x.

Moreover, we take the official implementation of BigGAN [1] for the reference. We use Adam as the optimizer with parameters $\alpha_G = 1 \times 10^{-4}$, $\alpha_D = 2 \times 10^{-4}$, $\beta_1 = 0$, $\beta_2 = 0.999$ and the batch size as 50. The generator is updated once for every 4 discriminator update steps. All the training processes are stopped after the generator updates $125k$ steps. The real images are augmented by the random horizontal flipping. Following the previous setting [5, 8], we employ the moving averages on generator weights with a decay of 0.9999. The pipeline for CR is shown in Table 3. Table 4 shows the performance of different approaches in terms of IS, FID (train) and FID (test). The results indicate that BigGAN with the proposed GN is better than BigGAN with CR, while BigGAN with both GN and CR achieves the best performance. For more qualitative results, please refer to Figure 6.

Unconditional Image Generation on CelebA-HQ and LSUN Church Outdoor. We further evaluate the proposed Gradient Normalization on two high-resolution image datasets, i.e., CelebA-HQ and LSUN Church Outdoor. For the augmentation, the random horizontal flipping is adopted for both datasets. We use the architecture proposed by SN-GAN [9] for generating 256×256 images.

Table 4: Inception Score(IS) and FID of conditional image generation on CIFAR-10.

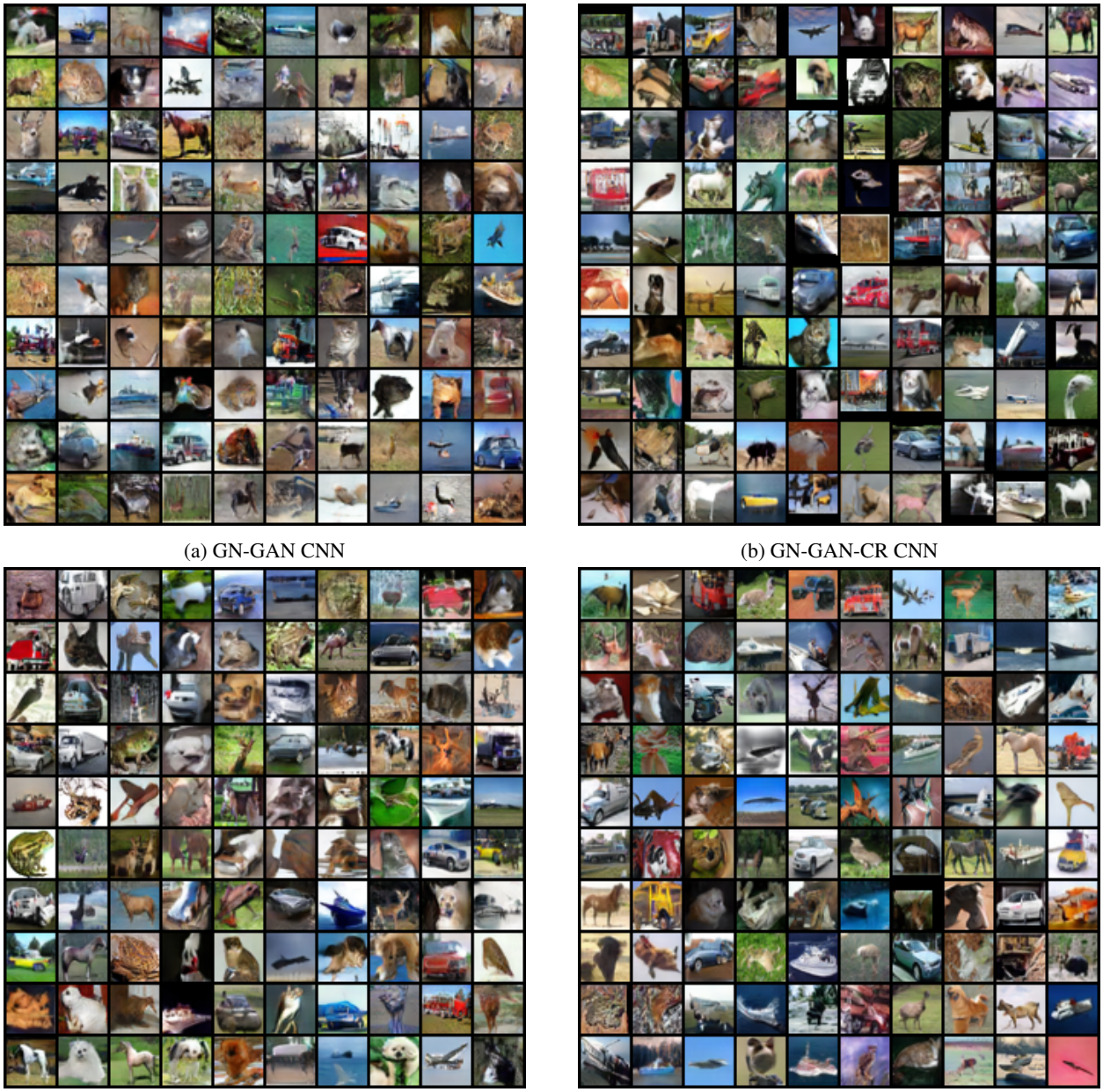
Method	IS↑	FID(train)↓	FID(test)↓
BigGAN [1]	9.22	-	14.73
BigGAN-CR [12]	-	-	11.48
GN-BigGAN	9.22±.13	5.87±.15	10.05±.23
GN-BigGAN-CR	**9.35±.14**	**4.86±.07**	**8.92±.15**

We use Adam again as the optimizer with parameters $\alpha_G = 2 \times 10^{-4}$, $\alpha_D = 2 \times 10^{-4}$, $\beta_1 = 0$, $\beta_2 = 0.9$ and batch size as 64. The generator is updated once for every 5 discriminator update steps. All the training processes are stopped after the generator update 100k steps. We employ the moving averages on generator weights with a decay of 0.9999. The Inception Score and FID are shown in Table 5. It is worth noting that the performance can be further improved with a better architecture. For more qualitative results, please refer to Figures 7 and 8.

Table 5: Inception Score and FID of unconditional image generation on CelebA-HQ and LSUN Church Outdoor. † represents that we provide SN-GAN implementation as the baseline.

Dataset	GN-GAN	SN-GAN
CelebA-HQ 128	**14.78**	25.95 (from [30])
CelebA-HQ 256	**7.67**	14.45†
LSUN Church 256	**5.41**	8.44†

Experiments on Progressive Growing Architecture. We further test the StyleGAN [6] with the proposed Gradient Normalization on CelebA-HQ 1024×1024. Note that the R1 regularization and Gradient Penalty are replaced with GN in our experiment. We use hinge loss as the objective function and Adam as the optimizer. The learning rates α_G and α_D are both set to 0.001 for resolutions of 8^2, 16^2, 32^2 and 64^2, and 0.0015 otherwise. For the other settings, we use the same parameters as StyleGAN. The FID of GN-StyleGAN is 8.65 which is calculated by 50k generated images vs. 30k training images. The generated samples are shown in Figures 9-12.



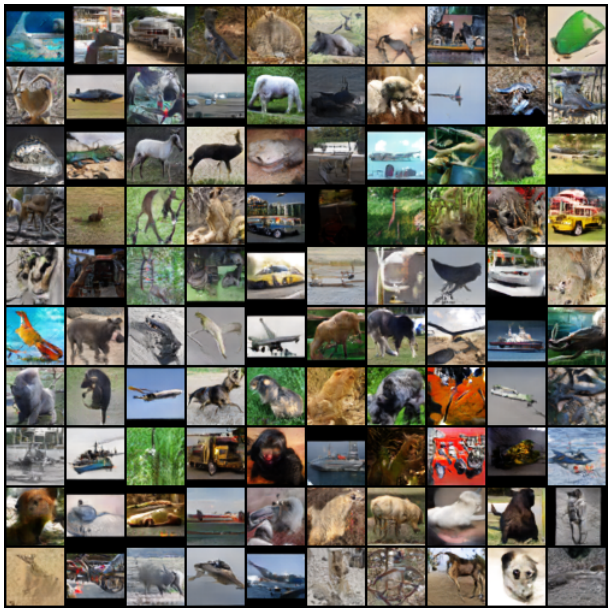
(a) GN-GAN CNN

(b) GN-GAN-CR CNN

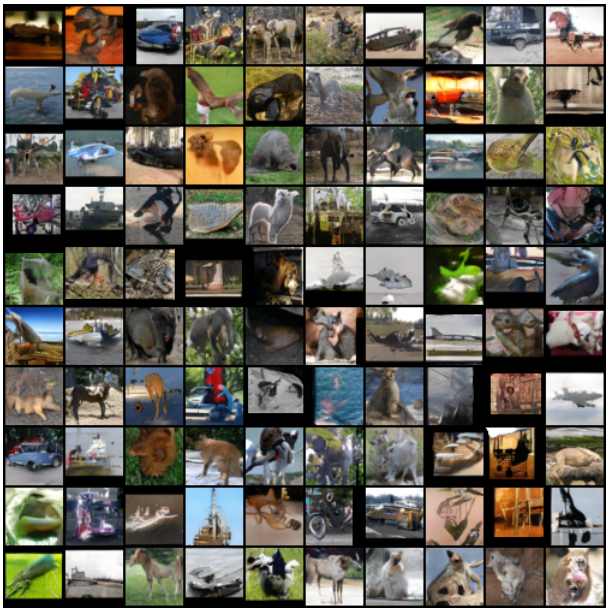
(c) GN-GAN ResNet

(d) GN-GAN-CR ResNet

Figure 4: Unconditional image generation on CIFAR-10.

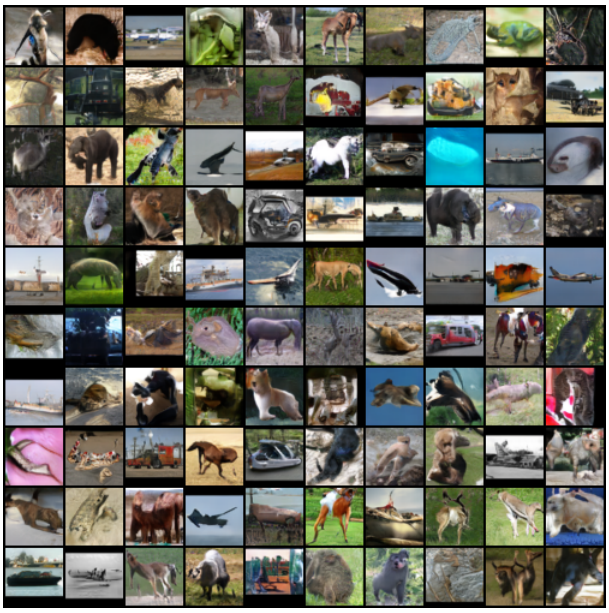


(a) GN-GAN CNN



(b) GN-GAN-CR CNN

(c) GN-GAN ResNet



(d) GN-GAN-CR ResNet

Figure 5: Unconditional image generation on STL-10.

Figure 6: Conditional image generation on CIFAR-10.

Figure 7: Unconditional image generation on CelebA-HQ 256×256.

Figure 8: Unconditional image generation on LSUN Church Outdoor 256×256.

Figure 9: GN-StyleGAN on CelebA-HQ 1024×1024.

Figure 10: GN-StyleGAN on CelebA-HQ 1024×1024.

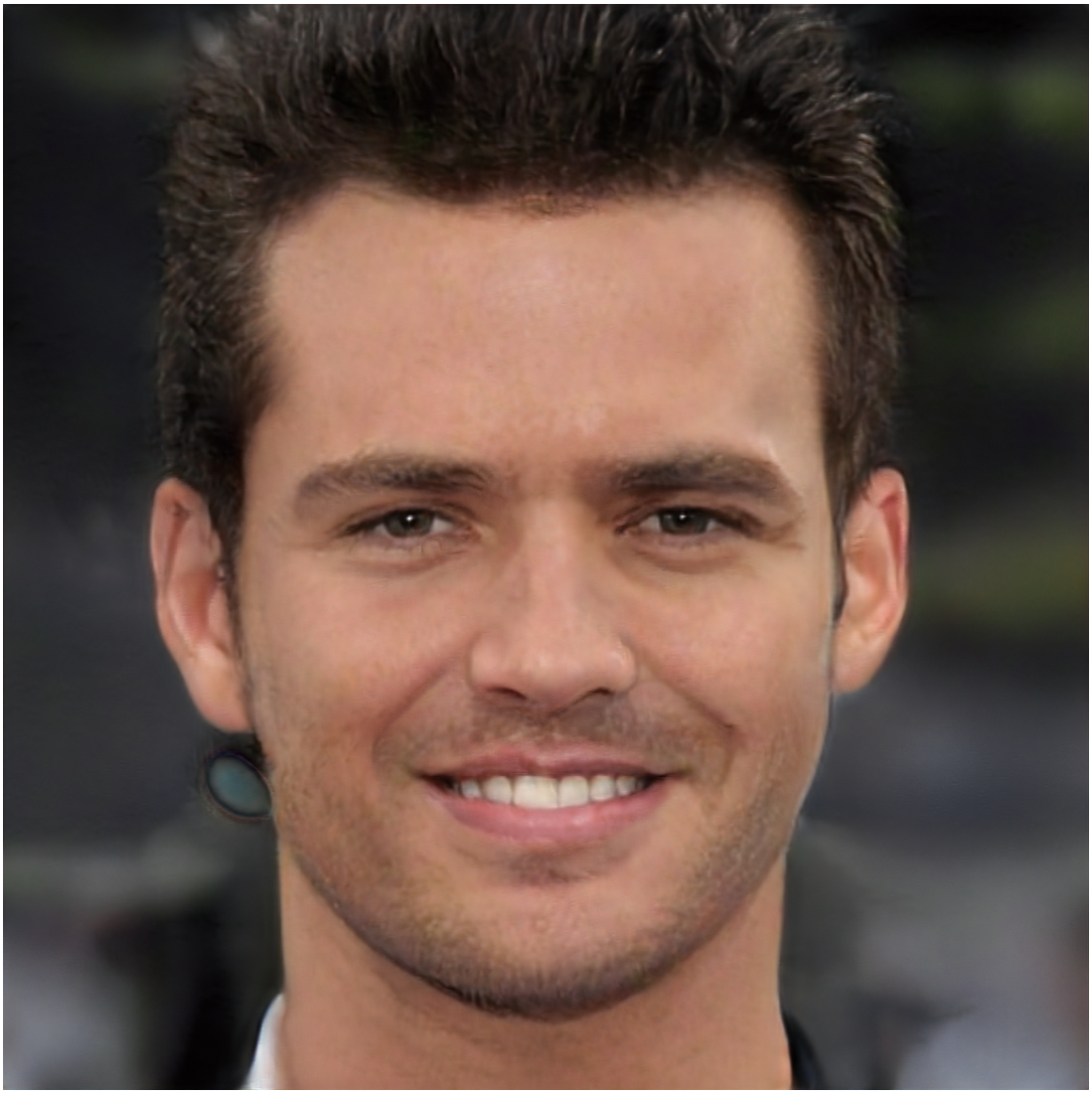


Figure 11: GN-StyleGAN on CelebA-HQ 1024×1024.

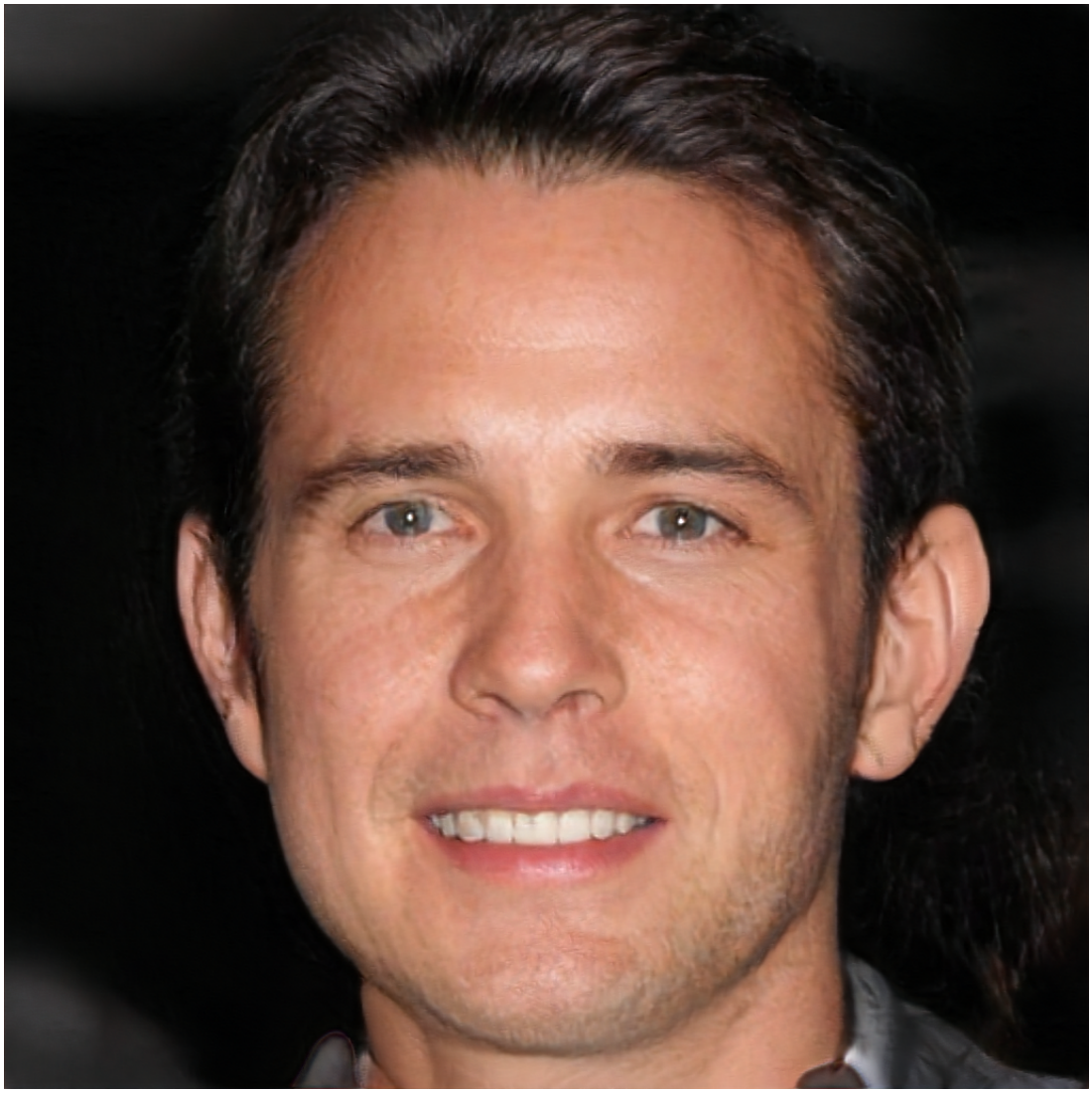


Figure 12: GN-StyleGAN on CelebA-HQ 1024×1024.

References

[1] Karen Simonyan Andrew Brock, Jeff Donahue. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2019. 4

[2] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations (ICLR)*, 2016. 2

[3] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 315–323, 2011. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. 3

[5] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2017. 4

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 3

[8] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Which training methods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, 2018. 4

[9] Takeru Miyato, T. Kataoka, Masanori Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018. 3, 4

[10] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, pages 807–814, 2010. 2

[11] Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. Improving generalization and stability of generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[12] H. Zhang, Zizhao Zhang, Augustus Odena, and H. Lee. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2020. 4