The GAN is dead; long live the GAN!
A Modern Baseline GAN

Anonymous Author(s)
Affiliation
Address
email

Abstract

There is a widely-spread claim that GANs are difficult to train, and GAN architectures in the literature are littered with empirical tricks. We provide evidence against this claim and build a modern GAN baseline in a more principled manner. First, we derive a well-behaved regularized relativistic GAN loss that addresses issues of mode dropping and non-convergence that were previously tackled via a bag of ad-hoc tricks. We analyze our loss mathematically and prove that it admits local convergence guarantees, unlike most existing relativistic losses. Second, our new loss allows us to discard all ad-hoc tricks and replace outdated backbones used in common GANs with modern architectures. Using StyleGAN2 as an example, we present a roadmap of simplification and modernization that results in a new minimalist baseline—R3GAN. Despite being simple, our approach surpasses StyleGAN2 on FFHQ, ImageNet, CIFAR, and Stacked MNIST datasets, and compares favorably against state-of-the-art GANs and diffusion models.

1 Introduction

Generative adversarial networks (GANs; (11)) feature the ability to generate high quality images in a single forward pass. However, the original objective in Goodfellow *et al.* (11), is notoriously difficult to optimize due to its minimax nature. This leads to a fear that training might diverge at any point due to instability or lose diversity through mode collapse. While progress in GAN objectives has occurred (12; 20; 65; 43; 54), practically, the effects of brittle losses are still regularly felt, and this notoriety has had a lasting negative impact on GAN research.

A second issue—partly motivated by this instability—is that existing popular GAN backbones like StyleGAN (30; 31; 27; 28) use many poorly-understood empirical tricks with little theory. For instance, StyleGAN uses a gradient penalized non-saturating loss (43) to increase stability (affecting sample diversity), but then employs a minibatch standard deviation trick (25) to increase sample diversity. Without tricks, the StyleGAN backbone still resembles DCGAN (51) from 2015, yet it is still the common backbone of SOTA GANs such as GigaGAN (23) and StyleGAN-T (57). Advances in GANs have been conservative compared to other generative models such as diffusion models (18; 63; 26; 29), where modern computer vision techniques such as multi-headed self attention (69) and backbones such as preactivated ResNet (15), U-Net (53) and vision transformers (ViTs) (9) are the norm. Given outdated backbones, it is not surprising that there is a widely-spread belief that GANs do not scale in terms of quantitative metrics like Frechet Inception Distance (17).

We reconsider this situation: we show that by introducing a new regularized training loss, GANs gain improved training stability, which allows us to upgrade GANs with modern backbones. First, we propose a novel objective that augments the relativistic pairing GAN loss (RpGAN; (20)) with zero-centered gradient penalties (43; 54), improving stability (12; 54; 43). We show mathematically that gradient-penalized RpGAN enjoys the same guarantee of local convergence as regularized classic GANs, and that removing our regularization scheme induces non-convergence.

Once we have a well-behaved loss, none of the GAN tricks are necessary (25; 31), and we are free to engineer a modern SOTA backbone architecture. We strip StyleGAN of all its features,

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

identify those that are essential, then borrow new architecture designs from modern ConvNets and transformers (41; 75). Briefly, we find that proper ResNet design (15; 55), initialization (77), and resampling (30; 31; 28; 78) are important, along with grouped convolution (74; 5) and no normalization (31; 29; 12; 70; 3). This leads to a design that is simpler than StyleGAN and improves FID performance for the same network capacity (2.77 vs. 3.78 on FFHQ-256).

In summary, our work first argues mathematically that GANs need not be tricky to train and introduces a new regularized loss. Then, it empirically develops a simple GAN baseline that,without any tricks, compares favorably by FID to StyleGAN (30; 31; 28), other SOTA GANs (4; 37; 73), and diffusion models (18; 63; 68) across FFHQ, ImageNet, CIFAR, and Stacked MNIST datasets.

2 Serving Two Masters: Stability and Diversity with RpGAN $+R_1+R_2$

In defining a GAN objective, we tackle two challenges: stability and diversity. Some previous work deals with stability (30; 31; 28) and other previous work deals with mode collapse (20). We combine a stable method with a simple regularizer grounded by theory to overcome both.

2.1 Traditional GAN

A traditional GAN (11; 49) is formulated as a minimax game between a discriminator D_ψ and a generator G_θ. Given real data $x \sim p_\mathcal{D}$ and fake data $x \sim p_\theta$ produced by G_θ, the most general form of a GAN is given by:

$$\mathcal{L}(\theta,\psi) = \mathbb{E}_{z\sim p_z}[f(D_\psi(G_\theta(z)))] + \mathbb{E}_{x\sim p_\mathcal{D}}[f(-D_\psi(x))] \tag{1}$$

where G tries to minimize $\mathcal{L}$ while D tries to maximize it. The choice of f is flexible (42; 38). In particular, $f(t) = -\log(1+e^{-t})$ recovers the classic GAN by Goodfellow *et al.* (11). For the rest of this work, this will be our choice of f (49).

It has been shown that Equation 1 has convex properties when p_θ can be optimized directly (11; 65). However, in practical implementations, the empirical GAN loss typically shifts fake samples beyond the decision boundary set by D, as opposed to directly updating the density function p_θ. This deviation leads to a significantly more challenging problem, characterized by susceptibility to two prevalent failure scenarios: mode collapse/dropping[1] and non-convergence.

2.2 Relativistic f-GAN.

We employ a slightly different minimax game named relativistic pairing GAN (RpGAN) by Jolicoeur-Martineau *et al.* (20) to address mode dropping. The general RpGAN is defined as:

$$\mathcal{L}(\theta,\psi) = \mathbb{E}_{\substack{z\sim p_z \\ x\sim p_\mathcal{D}}}[f(D_\psi(G_\theta(z)) - D_\psi(x))] \tag{2}$$

Although Eq.2 differs only slightly from Eq.1, evaluating the critic difference has a fundamental impact on the landscape of $\mathcal{L}$. Since Eq.1 merely requires D to separate real and fake data, in the scenario where all real and fake data can be separated by a single decision boundary, the empirical GAN loss encourages G to simply move all fake samples barely past this single boundary—this degenerate solution is what we observe as mode collapse/dropping. Sun *et al.* (65) characterize such degenerate solutions as bad local minima in the landscape of $\mathcal{L}$, and show that Eq.1 has *exponentially many* bad local minima. The culprit is the existence of a single decision boundary that naturally arises when real and fake data are considered in isolation. RpGAN introduces a simple solution by coupling real and fake data, *i.e.* a fake sample is critiqued by its realness *relative to* a real sample, which effectively maintains a decision boundary in the neighborhood of *each* real sample and hence forbids mode dropping. Sun *et al.* (65) show that the landscape of Eq.2 contains no local minima that correspond to mode dropping solutions, and that every basin is a global minimum.

2.3 Training Dynamics of RpGAN

Although the landscape result (65) of RpGAN allows us to address mode dropping, the training dynamics of RpGAN have yet to be studied. The ultimate goal of Eq. 2 is to find the equilibrium (θ^*,ψ^*) such that $p_{\theta^*} = p_\mathcal{D}$ and D_{ψ^*} is constant everywhere on $p_\mathcal{D}$. Sun *et al.* (65) show that θ^* is globally reachable along a non-increasing trajectory in the landscape of Eq.2 under reasonable assumptions. However, the existence of such a trajectory does not necessarily mean that gradient descent will find it. Jolicoeur-Martineau *et al.* show empirically that unregularized RpGAN does not perform well (20).

Proposition I. (Informal) *Unregularized RpGAN does not always converge using gradient descent.*

[1] While mode collapse and mode dropping are technically distinct issues, they are used interchangeably in this context to describe the common problem where supp p_θ does not fully cover supp $p_\mathcal{D}$.

We confirm this proposition with a proof in Appendix H. We show analytically that RpGAN does not converge for certain types of $p_{\mathcal{D}}$, such as ones that approach a delta distribution. Thus, further regularization is necessary to fill in the missing piece of a well-behaved loss.

Zero-centered gradient penalties. To tackle RpGAN non-convergence, we explore gradient penalties as the solution since it is proven that zero-centered gradient penalties (0-GP) facilitate convergent training for classic GANs (43). The two most commonly-used 0-GPs are R_1 and R_2:

$$\begin{aligned} R_1(\psi) &= \frac{\gamma}{2}\mathbb{E}_{x\sim p_{\mathcal{D}}}\left[\|\nabla_x D_\psi\|^2\right] \\ R_2(\theta,\psi) &= \frac{\gamma}{2}\mathbb{E}_{x\sim p_{\theta}}\left[\|\nabla_x D_\psi\|^2\right] \end{aligned} \tag{3}$$

R_1 penalizes the gradient norm of D on real data, and R_2 on fake data. Analysis on the training dynamics of GANs has thus far focused on local convergence (47; 44; 43), *i.e.* whether the training at least converges when (θ,ψ) are in a neighborhood of (θ^*,ψ^*). In such a scenario, the convergence behavior can be analyzed (47; 44; 43) by examining the spectrum of the Jacobian of the gradient vector field $(-\nabla_\theta\mathcal{L},\nabla_\psi\mathcal{L})$ at (θ^*,ψ^*). The key insight here is that when G already produces the true distribution, we want $\nabla_x D=0$, so that G is not pushed away from its optimal state, and thus the training does not oscillate. R_1 and R_2 impose such a constraint when $p_\theta=p_{\mathcal{D}}$. This also explains why earlier attempts at gradient penalties, such as the one-centered gradient penalty (1-GP) in WGAN-GP (12), fail to achieve convergent training (43) as they still encourage D to have a non-zero slope when G has reached optimality.

Since the same insight also applies to RpGAN, we extend our previous analysis and show that:

Proposition II. (Informal) *RpGAN with R_1 or R_2 regularization is locally convergent subject to similar assumptions as in* Mescheder *et al.* (43).

In Appendix I, our proof similarly analyzes the eigenvalues of the Jacobian of the regularized RpGAN gradient vector field at (θ^*,ψ^*). We show that all eigenvalues have a negative real part; thus, regularized RpGAN is convergent in a neighborhood of (θ^*,ψ^*) for small enough learning rates (43).

Discussion. Another line of work (54) links R_1 and R_2 to instance noise (62) as its analytical approximation. Roth et al. (54) showed that for the classic GAN (11) by Goodfellow *et al.*, R_1 approximates convolving $p_{\mathcal{D}}$ with the density function of $\mathcal{N}(0,\gamma I)$, up to additional weighting and a Laplacian error term. R_2 likewise approximates convolving p_θ with $\mathcal{N}(0,\gamma I)$ up to similar error terms. The Laplacian error terms from R_1, R_2 cancel when D_ψ approaches D_{ψ^*}. We do not extend Roth *et al.*'s proof (54) to RpGAN; however, this approach might provide complimentary insights to our work, which follows the strategy of Mescheder *et al.* (43).

We demonstrate our loss in Appendix A where we focus on practical considerations such as global convergence. Building on Roth et al. (54), we apply both R_1 and R_2 to improve global stability.

3 A Roadmap to a New Baseline — R3GAN

The well-behaved RpGAN + R_1 + R_2 loss alleviates GAN optimization problems, and lets us proceed to build a minimalist baseline—R3GAN—with recent network backbone advances in mind (41; 75). Rather than simply state the new approach, we will draw out a roadmap from the StyleGAN2 baseline (27). This model (Config A; identical to (27)) consists of a VGG-like (60) backbone for G, a ResNet D, a few techniques that facilitate style-based generation, and many tricks that serve as patches to the weak backbone. Then, we remove all non-essential features of StyleGAN2 (Config B), apply our loss function (Config C), and gradually modernize the network backbone (Config D-E).

We evaluate each configuration on FFHQ 256×256 (30). Network capacity is kept roughly the same for all configurations—both G and D have about 25M trainable parameters. Each configuration is trained until D sees 5M real images. We inherit training hyperparameters (optimizer settings, batch size, EMA decay length, *etc.*) from Config A unless otherwise specified. We tune the training hyperparameters for our final model and show the converged result in Sec. 4.

Minimum Baseline (Config B). We strip away all StyleGAN2 features, retaining only the raw network backbone and basic image generation capability. The features fall into three categories:

- Style-based generation: mapping network (30), style injection (30), weight modulation/demodulation (31), noise injection (30).
- Image manipulation enhancements: mixing regularization (30), path length regularization (31).
- Tricks: z normalization (25), minibatch stddev (25), equalized learning rate (25), lazy regularization (31).

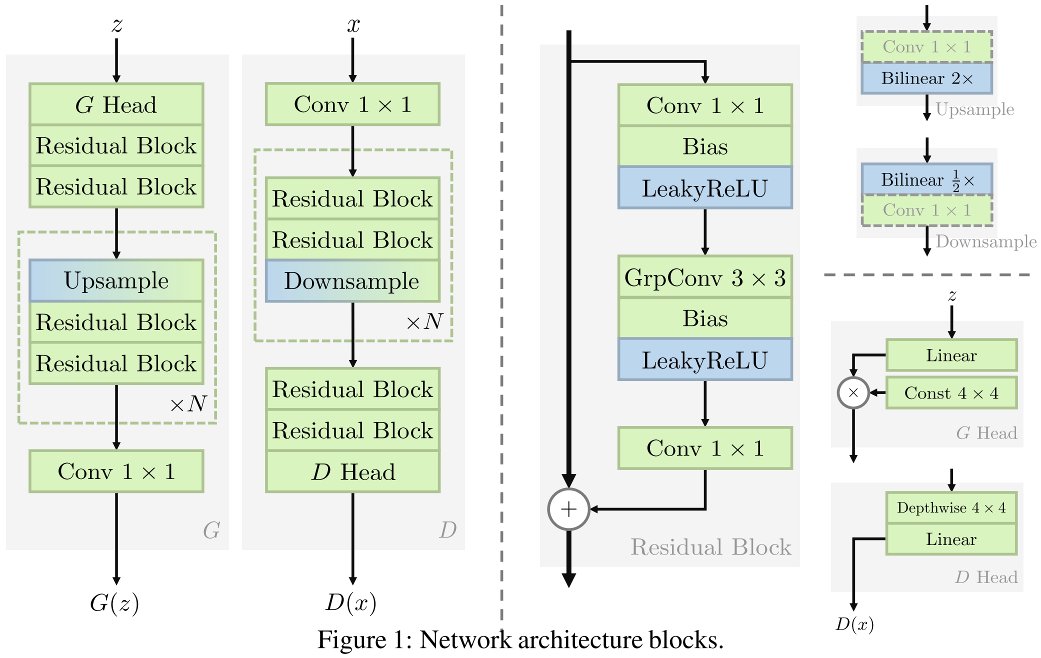

Figure 1: Network architecture blocks.

Following (58; 57), we reduce the dimension of z to 64. The absence of equalized learning rate necessitates a lower learning rate, reduced from 2.5×10^{-3} to 5×10^{-5}. Despite a higher FID of 12.46 than Config-A, this simplified baseline produces reasonable sample quality and stable training. We compare this with DCGAN (51), an early attempt at image generation. Key differences include:

a) Convergent training objective with R_1 regularization.
b) Smaller learning rate, avoiding momentum optimizer (Adam $\beta_1 = 0$).
c) No normalization layer in G or D.
d) Proper resampling via bilinear interpolation instead of strided (transposed) convolution.
e) Leaky ReLU in both G and D, no tanh in the output layer of G.
f) 4×4 constant input for G, output skips for G, ResNet D.

We discuss our findings about these principles in Appendix B and establish that a) through e) are critical to the success of StyleGAN2, and apply them to all subsequent configurations.

Well-behaved loss function (Config C). We use the loss function proposed in Section 2 and this reduces FID to 11.65. We hypothesize that the network backbone in Config B is the limiting factor.

General network modernization (Config D). First, we apply the 1-3-1 bottleneck ResNet architecture (14; 15) to both G and D. This is the direct ancestor of all modern vision backbones (41; 75). We also incorporate principles discovered in Config B and various modernization efforts from ConvNeXt (41). We categorize the roadmap of ConvNeXt as follows:

i. Consistently beneficial: i.1) increased width with depthwise conv., i.2) inverted bottleneck, i.3) fewer activation functions, and i.4) separate resampling layers

ii. Negligible performance gain: ii.1) large kernel depthwise conv. with fewer channels, ii.2) swap ReLU with GELU, ii.3) fewer normalization layers, and ii.4) swap batch norm. with layer norm.

iii. Irrelevant to our setting: iii.1) improved training recipe, iii.2) stage ratio, and iii.3) “patchify” stem

We aim to apply i) to our model, specifically i.3 and i.4 for the classic ResNet, while reserving i.1 and i.2 for Config E. Many aspects of ii) were introduced merely to mimic vision transformers (40; 9) without yielding significant improvements (41). ii.3 and ii.4 are inapplicable due to our avoidance of normalization layers following principle c). ii.2 contradicts our finding that GELU deteriorates GAN performance, thus we use leaky ReLU per principle e). Liu *et al*. emphasize large conv kernels (ii.1) (41), but this results in slightly worse performance compared to wider 3×3 conv layers, so we do not adopt this ConvNeXt design choice. We discuss the architecture details in Appendix C.

Model	FID↓
StyleGAN2	3.78
StyleGAN3-T	4.81
StyleGAN3-R	3.92
LDM	4.98
ADM (DDIM)	8.41
ADM (DPM-Solver)	8.40
Diffusion Autoencoder	5.81
Ours—Config E	2.77
With ImageNet feature leakage (36):	
PolyINR* (61)	2.72
StyleGAN-XL* (58)	2.19
StyleSAN-XL* (66)	1.36

Table 1: FFHQ-256. * denotes models that leak ImageNet features.

Bottleneck modernization (Config E). Now that we have settled on the overall architecture, we investigate how the residual block can be modernized, specifically i.1) and i.2). First, we explore i.1 and replace the 3×3 convolution in the residual block with a grouped convolution. We set the group size to 16 rather than 1 (*i.e.* depthwise convolution as in ConvNeXt) as depthwise convolution is highly inefficient on GPUs and is not much faster than using a larger group size. With grouped convolution, we can reduce the bottleneck compression ratio to two given the same model size. This increases the width of the bottleneck to $1.5\times$ as wide as Config A. Finally, we notice that the compute cost of grouped convolution is negligible compared to 1×1 convolution, and so we seek to enhance the capacity of grouped convolution. We apply i.2), which inverts the bottleneck width and the stem width, and which doubles the width of grouped convolutions without any increase in model size. Figure 1 depicts our final design, which reflects modern CNN architectures.

4 Experiments

We evaluate our model on FFHQ (256×256) (30) for high resolution unimodal image synthesis, and high diversity generation on CIFAR-10 (34), and ImageNet (32×32) (6). We compare our model with various baselines, in Table 1, 2, and 3.

We leave a detailed discussion of our results in Appendix E. Our model surpasses StyleGAN2 and StyleGAN3 by a large margin across datasets despite its simplicity. Unless with ImageNet feature leakage (56; 36) or certain regularization (79) that has been shown to overfit (76) on FFHQ 256×256, no GAN comes close to R3GAN in terms of FID. Our model also beats diffusion models despite having a considerably smaller model size and that it generates samples in one step.

Model	FID↓
BigGAN (4)	14.73
TransGAN (69)	9.26
ViTGAN (37)	6.66
DDGAN (73)	3.75
Diffusion StyleGAN2	3.19
StyleGAN2 + ADA	2.42
StyleGAN3-R + ADA	10.83
DDPM	3.21
DDIM	4.67
VE (26)	3.11
VP (26)	2.48
Ours—Config E	1.97
With ImageNet feature leakage (36):	
StyleGAN-XL* (58)	1.85

Table 2: CIFAR-10.

Model	FID↓
Unconditional	
DDPM++ (32)	8.42
VDM (33)	7.41
Conditional	
MSGAN (24)	12.3
ADM (7; 48)	3.60
DDPM-IP (48)	2.87
Ours—Config E	1.27
With ImageNet feature leakage (36):	
StyleGAN-XL* (58)	1.10

Table 3: ImageNet-32.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Koushik Biswas, Sandeep Kumar, Shilpak Banerjee, and Ashish Kumar Pandey. Smu: smooth activation function for deep networks using smoothing maximum technique. *arXiv preprint arXiv:2111.04682*, 2021.

[3] Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning*, pages 1059–1071. PMLR, 2021.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[6] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[8] Adji B Dieng, Francisco JR Ruiz, David M Blei, and Michalis K Titsias. Prescribed generative adversarial networks. *arXiv preprint arXiv:1910.04302*, 2019.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] Gauthier Gidel, Reyhane Askari Hemmat, Mohammad Pezeshki, Rémi Le Priol, Gabriel Huang, Simon Lacoste-Julien, and Ioannis Mitliagkas. Negative momentum for improved game dynamics. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1802–1811. PMLR, 2019.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

[16] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[20] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

[21] Alexia Jolicoeur-Martineau and Ioannis Mitliagkas. Gradient penalty from a maximum margin perspective. *arXiv preprint arXiv:1910.06922*, 2019.

[22] Alexia Jolicoeur-Martineau, Rémi Piché-Taillefer, Rémi Tachet des Combes, and Ioannis Mitliagkas. Adversarial score matching and improved sampling for image generation. *arXiv preprint arXiv:2009.05475*, 2020.

[23] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023.

[24] Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7799–7808, 2020.

[25] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[26] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

[27] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.

[28] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.

[29] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. *arXiv preprint arXiv:2312.02696*, 2023.

[30] 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*, pages 4401–4410, 2019.

[31] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[32] Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. *arXiv preprint arXiv:2106.05527*, 2021.

[33] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[34] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[35] Rithesh Kumar, Sherjil Ozair, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019.

[36] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fréchet inception distance. *arXiv preprint arXiv:2203.06026*, 2022.

[37] Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. *arXiv preprint arXiv:2107.04589*, 2021.

[38] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[39] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[41] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[42] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[43] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[44] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. *Advances in neural information processing systems*, 30, 2017.

[45] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *International Conference on Learning Representations*, 2016.

[46] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

[47] Vaishnavh Nagarajan and J Zico Kolter. Gradient descent gan optimization is locally stable. *Advances in neural information processing systems*, 30, 2017.

[48] Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. *arXiv preprint arXiv:2301.11706*, 2023.

[49] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

[50] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[51] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[52] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[53] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[54] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. *Advances in neural information processing systems*, 30, 2017.

[55] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[56] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34:17480–17492, 2021.

[57] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. In *International conference on machine learning*, pages 30105–30118. PMLR, 2023.

[58] Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

[59] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[60] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[61] Rajhans Singh, Ankita Shukla, and Pavan Turaga. Polynomial implicit neural representations for large diverse datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2041–2051, 2023.

[62] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

[63] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[64] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems*, 30, 2017.

[65] Ruoyu Sun, Tiantian Fang, and Alexander Schwing. Towards a better global loss landscape of gans. *Advances in Neural Information Processing Systems*, 33:10186–10198, 2020.

[66] Yuhta Takida, Masaaki Imaizumi, Takashi Shibuya, Chieh-Hsin Lai, Toshimitsu Uesaka, Naoki Murata, and Yuki Mitsufuji. SAN: Inducing metrizability of GAN with discriminative normalized linear layer. In *The Twelfth International Conference on Learning Representations*, 2024.

[67] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[68] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.

[69] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[70] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[71] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[72] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. Vaebm: A symbiosis between variational autoencoders and energy-based models. *arXiv preprint arXiv:2010.00654*, 2020.

[73] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

[74] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[75] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022.

[76] Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11304–11314, 2022.

[77] Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*, 2019.

[78] Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019.

[79] Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11033–11041, 2021.

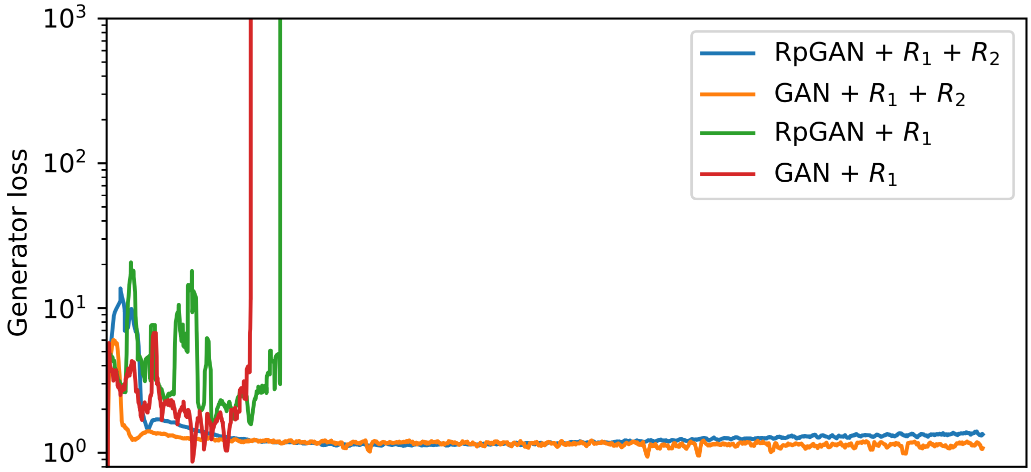

Figure 2: Generator G loss for different objectives over training. Regardless of which objective is used, training diverges with only R_1 and succeeded with both R_1 and R_2. Convergence failure with only R_1 was noted by Lee et al. (37).

Appendices

A A Practical Demonstration of Our Loss.

We experiment with how well-behaved our loss is on StackedMNIST (39) which consists of 1000 uniformly-distributed modes. The network is a small ResNet (15) for G and D without any normalization layers (19; 71; 1; 67). Through the use of a pretrained MNIST classifier, we can explicitly measure how many modes of $p_\mathcal{D}$ are recovered by p_θ. Furthermore, we can estimate the reverse KL divergence between the fake and real samples $D_{\text{KL}}(p_\theta \,\|\, p_\mathcal{D})$ via the KL divergence between the categorical distribution of p_θ and the true uniform distribution.

A conventional GAN loss with R_1, as used by Mescheder et al. (43) and the StyleGAN series (30; 31; 28), diverges quickly (Fig. 2). Next, while theoretically sufficient for local convergence, RpGAN with only R_1 regularization is also unstable and diverges quickly[2]. In each case, the gradient of D on fake samples explodes when training diverges. With both R_1 and R_2, training becomes stable for both the classic GAN and RpGAN. Now stable, we can see that the classic GAN suffers from mode dropping, whereas RpGAN achieves full mode coverage (Table 4) and reduces D_{KL} from 0.9270 to 0.0781. As a point of contrast, StyleGAN (30; 31; 27; 28) uses the minibatch standard deviation trick to reduce mode dropping, improving mode coverage from 857 to 881 on StackedMNIST and with barely any improvement on D_{KL} (25).

Loss	# modes↑	D_{KL}↓
RpGAN $+R_1+R_2$	**1000**	**0.0781**
GAN $+R_1+R_2$	693	0.9270
RpGAN $+R_1$	Fail	Fail
GAN $+R_1$	Fail	Fail

Table 4: StackedMNIST (39) result for each loss function. The maximum possible mode coverage is 1000. “Fail” indicates that training diverged early on.

R_1 alone is not sufficient for globally-convergent training. While a theoretical analysis of this is difficult, our small demonstration still provides insights into the assumptions of our convergence proof. In particular, the assumption that (θ,ψ) are sufficiently close to (θ^*,ψ^*) is highly unlikely early in training. In this scenario, if D is sufficiently powerful, regularizing D solely on real data is not likely to have much effect on D's behavior on fake data and so training can fail due to an ill-behaved D on fake data.

[2] Varying γ from 0.1 to 100 does not stabilize training.

Thus, the practical solution is to regularize D on both real and fake data. The benefit of doing so can be viewed from the insight of Roth *et al*. (54): that applying R_1 and R_2 in conjunction smooths both $p_{\mathcal{D}}$ and p_θ which makes learning easier than only smoothing $p_{\mathcal{D}}$. We also find empirically that with both R_1 and R_2 in place, D tends to satisfy $\mathbb{E}_{x\sim p_{\mathcal{D}}}\left[\|\nabla_x D\|^2\right] \approx \mathbb{E}_{x\sim p_\theta}\left[\|\nabla_x D\|^2\right]$ even early in the training. Jolicoeur-Martineau *et al*. (21) show that in this case D becomes a maximum margin classifier—but if only one regularization term is applied, this does not hold.

B Experimental Findings from Config B.

Violating a), b), or c) often leads to training failures. Gidel *et al*. (10) show that *negative* momentum can improve GAN training dynamics. Since optimal negative momentum is another challenging hyperparameter, we do not use any momentum to avoid worsening GAN training dynamics. Studies (31; 29) suggest normalization layers harm generative models. Batch normalization (19) often cripples training due to dependencies across multiple samples, and is incompatible with R_1, R_2, or RpGAN that assume independent handling of each sample. Weaker data-independent normalizations (31; 29) might help; we leave this for future work. Early GANs may succeed despite violating a) and c), possibly constituting a full-rank solution (43) to Eq. 1.

Violations of d) or e) do not significantly impair training stability but negatively affect sample quality. Improper transposed convolution can cause checkerboard artifacts, unresolved even with subpixel convolution (59) or carefully tuned transposed convolution unless a low-pass filter is applied. Interpolation methods avoid this issue, varying from nearest neighbor (25) to Kaiser filters (28). We use bilinear interpolation for simplicity. For activation functions, smooth approximations of (leaky) ReLU, such as Swish (52), GELU (16), and SMU (2), worsen FID. PReLU (13) marginally improves FID but increases VRAM usage, so we use leaky ReLU.

All subsequent configurations adhere to a) through e). Violation of f) is acceptable as it pertains to the network backbone of StyleGAN2 (31), modernized in Config D and E.

C Network Architecture Details of Config D

Given i.3, i.4, and principles c), d), and e), we can replace the StyleGAN2 backbone with a modernized ResNet. We use a fully symmetric design for G and D with 25M parameters each, comparable to Config-A. The architecture is minimalist: each resolution stage has one transition layer and two residual blocks. The transition layer consists of bilinear resampling and an optional 1×1 conv for changing spatial size and feature map channels. The residual block includes five operations: Conv$1\times1 \rightarrow$ Leaky ReLU $\rightarrow$ Conv$3\times3 \rightarrow$ Leaky ReLU $\rightarrow$ Conv1×1, with the final Conv1×1 having no bias term. For the 4×4 resolution stage, the transition layer is replaced by a basis layer for G and a classifier head for D. The basis layer, similar to StyleGAN (30; 31), uses 4×4 learnable feature maps modulated by z via a linear layer. The classifier head uses a global 4×4 depthwise conv. to remove spatial extent, followed by a linear layer to produce D's output. We maintain the width ratio for each resolution stage as in Config A, making the stem width $3\times$ as wide due to the efficient 1×1 conv. The 3×3 conv in the residual block has a compression ratio of 4, following (14; 15), making the bottleneck width $0.75\times$ as wide as Config A.

To avoid variance explosion due to the lack of normalization, we employ fix-up initialization (77) for our modernized networks. Specifically, we zero-initialize the last convolutional layer in each residual block and scale down the initialization of the other two convolutional layers in the block by $L^{-0.25}$, where L is the number of residual blocks. We avoid other fix-up tricks, such as excessive bias terms and a learnable multiplier.

D Roadmap Insights

As per Table 5, Config A (vanilla StyleGAN2) achieves an FID of 7.52 using the official implementation on FFHQ-256. Config B with all tricks removed achieves an FID of 12.46—performance drops as expected. Config C, with a well-behaved loss, achieves an FID of 11.65. But, now training is sufficiently stable to improve the architecture.

Config D, which improves G and D based on the classic ResNet and ConvNeXt findings, achieves an FID of 9.95. The output skips of the StyleGAN2 generator are no longer useful given our new architecture; including them produces a worse FID of 10.17. Karras *et al*. find that the benefit of output skips is mostly related to gradient magnitude dynamics (28), and this has been addressed by our ResNet architecture. For StyleGAN2, Karras *et al*. conclude that a ResNet architecture is harmful to G (31), but this

	Configuration	FID↓	G #params	D #params
A	StyleGAN2	7.516	24.767M	24.001M
B	Stripped StyleGAN2	12.46	18.890M	23.996M
	- z normalization			
	- Minibatch stddev			
	- Equalized learning rate			
	- Mapping network			
	- Style injection			
	- Weight mod / demod			
	- Noise injection			
	- Mixing regularization			
	- Path length regularization			
	- Lazy regularization			
C	Well-behaved Loss			
	+ RpGAN loss	11.77	18.890M	23.996M
	+ R_2 gradient penalty	11.65		
D	ConvNeXt-ify pt. 1			
	+ ResNet-ify G & D	10.17	23.400M	23.282M
	- Output skips	9.950	23.378M	
E	ConvNeXt-ify pt. 2			
	+ ResNeXt-ify G & D	7.507	23.188M	23.091M
	+ Inverted bottleneck	7.045	23.058M	23.010M

Table 5: Model configuration performance and size.

is not true in our case as their ResNet implementation is considerably different from ours: 1) Karras *et al*. use one 3-3 residual block for each resolution stage, while we have a separate transition layer and two 1-3-1 residual blocks; 2) i.3) and i.4) are violated as they do not have a linear residual block (55) and the transition layer is placed on the skip branch of the residual block rather than the stem; 3) the essential principle of ResNet (14)—identity mapping (15)—is violated as Karras *et al*. divide the output of the residual block by $\sqrt{2}$ to avoid variance explosion due to the absence of a proper initialization scheme.

For Config E, we conduct two experiments that ablate i.1 (increased width with depthwise conv.) and i.2 (an inverted bottleneck). We add GroupedConv and reduce the bottleneck compression ratio to two given the same model size. Each bottleneck is now 1.5× the width of Config A, and the FID drops to 7.51, surpassing the performance of StyleGAN2. By inverting the stem and the bottleneck dimensions to enhance the capacity of GroupedConv, our final model achieves an FID of 7.05, exceeding StyleGAN2.

E Experiments Details

E.1 Mode recovery — StackedMNIST (45)

We repeat the earlier experiment in 1000-mode convergence on StackedMNIST (unconditional generation), but this time with our updated architecture and with comparisons to SOTA GANs and likelihood-based methods (Tab. 6, Fig. 5). One advantage brought up of likelihood-based models such as diffusion over GANs is that they achieve mode coverage (7). We find that most GANs struggle to find all modes. But, PresGAN (8), DDGAN (73) and our approach are successful. Further, our method outperforms all other tested GAN models in term of KL divergence.

Model	# modes↑	D_{KL}↓
DCGAN (51)	99	3.40
VEEGAN (64)	150	2.95
WGAN-GP (12)	959	0.73
PacGAN (39)	992	0.28
StyleGAN2 (31)	940	0.42
PresGAN (8)	**1000**	0.12
Adv. DSM (22)	**1000**	1.49
VAEBM (72)	**1000**	0.087
DDGAN (73)	**1000**	0.071
MEG (35)	**1000**	0.031
Ours—Config E	**1000**	**0.029**

Table 6: StackedMNIST 1000-mode coverage.

E.2 FID — FFHQ-256 (30) (Optimized)

We train Config E model until convergence and with optimized hyperparameters and training schedule on FFHQ at 256 × 256 (unconditional generation) (Tab. 1, Figs. 3 and 6). The hyperparameters and schedule are listed in Appendix J. We outperform existing StyleGAN methods, plus four more recent diffusion-based methods. This particular dataset experimental setting is so common that many methods (not listed here) use the bCR (79) trick—this has only been shown to improve performance on FFHQ-256 (not even at different resolutions of FFHQ) (79; 76). We use no such tricks in our method.

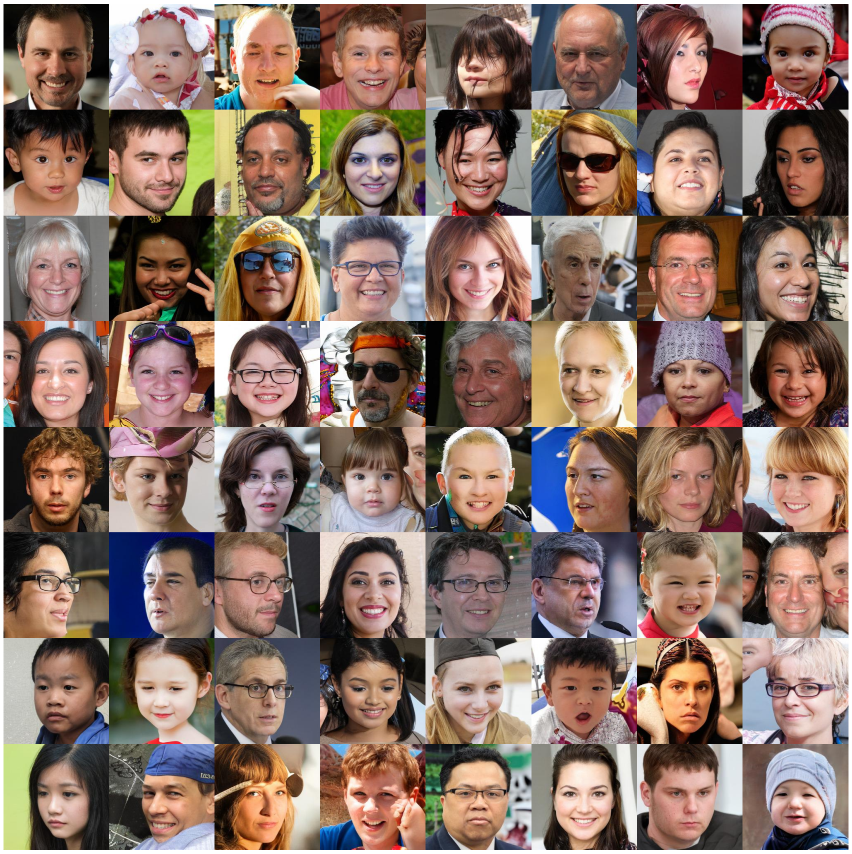

Figure 3: Qualitative examples of sample generation from our Config E on FFHQ-256.

E.3 FID — CIFAR-10 (34)

We train Config E model until convergence and with optimized hyperparameters and training schedule on CIFAR-10 (conditional generation) (Tab. 2, Fig. 7). Our method outperforms many other GANs by FID even though the model has relatively small capacity. For instance, StyleGAN-XL (58) has 18 M parameters in the generator and 125 M parameters in the discriminator, while our model has a 40 M parameters between the generator and discriminator combined (Fig. 4). Compared to diffusion models like LDM or ADM, GAN inference is significantly cheaper as it requires only one network function evaluation compared to the tens or hundreds of network function evaluations for diffusion models without distillation.

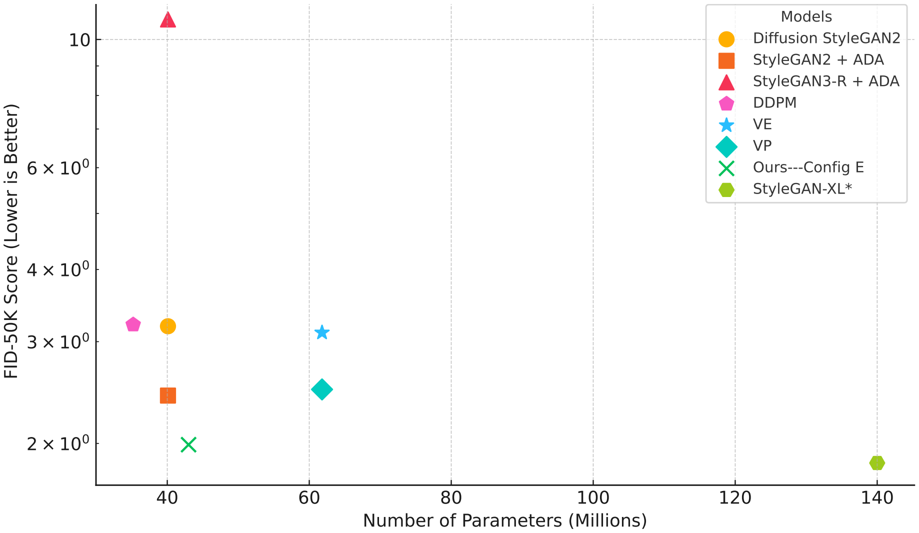

Figure 4: Number of parameters (millions) vs. FID-50K (log scale) on CIFAR-10. Lower is better.

Many state-of-the-art GANs are derived from Projected GAN (56), including StyleGAN-XL (58) and the concurrent work of StyleSAN-XL (66). These methods use a pre-trained ImageNet classifier in the discriminator. Prior work has shown that a pre-trained ImageNet discriminator can leak ImageNet features into the model (36), causing the model to perform better when evaluating on FID since it relies on a pre-trained ImageNet classifier for the loss. But, this does not improve results in perceptual studies (36). Our model produces its low FID without any ImageNet pre-training.

E.4 FID — ImageNet-32 (6)

We train Config E model until convergence and with optimized hyperparameters and training schedule on ImageNet-32 (conditional generation). We compare against recent GAN models and recent diffusion models in Table 3. We adjust the number of parameters in the generator of our model to match StyleGAN-XL (58)'s generator (84 million parameters). Specifically, we make the model significantly wider to match. Our method achieves comparable FID despite using a 60% smaller discriminator (Tab. 3) and despite not using a pre-trained ImageNet classifier.

F Discussion and Limitations

We have shown that a simplication of GANs is possible for image generation tasks, built upon a more stable RpGAN$+R_1+R_2$ objective with mathematically-demonstrated convergence properties that still provides diverse output. This stability is what lets us re-engineer a modern network architecture without the tricks of previous methods, producing the R3GAN model with competitive FID on the common datasets of Stacked-MNIST, FFHQ, CIFAR-10, and ImageNet-32 as an empirical demonstration of the mathematical benefits.

The focus of our work is to elucidate the essential components of a minimum GAN for image generation. As such, we prioritize simplicity over functionality—we do not claim to beat the performance of every existing model on every dataset or task; merely to provide a new simple baseline that converges easily. While this makes our model an ideal backbone for future GANs, it also means that it is not suitable to apply our model directly to downstream applications such as image editing or controllable generation, as our model lacks dedicated features for easy image inversion or disentangled image synthesis. For instance, we remove style injection functionality from StyleGAN even though this has a clear use. We also omitted common techniques that have been shown in previous literature to improve FID considerably. Examples include some form of adaptive normalization modulated by the latent code (7; 26; 30; 76; 50), and using multiheaded self attention at lower resolution stages (7; 26; 29). We aim to explore these techniques in a subsequent study.

Further, our work is limited in its evaluation of the scalability of R3GAN models. While they show promising results on 32×32 ImageNet, we are yet to verify the scalability on higher resolution ImageNet data or large-scale text to image generation tasks.

Finally, as a method that can improve the quality of generative models, it would be amiss not to mention that generative models—especially of people—can cause direct harm (e.g., through personalized deep fakes) and societal harm through the spread of disinformation (e.g., fake influencers).

G Local convergence

Following (43), GAN training can be formulated as a dynamical system where the update operator is given by $F_h(\theta,\psi)=(\theta,\psi)+hv(\theta,\psi)$. h is the learning rate and v denotes the gradient vector field:

$$v(\theta,\psi)=\begin{pmatrix}-\nabla_\theta\mathcal{L}(\theta,\psi)\\ \nabla_\psi\mathcal{L}(\theta,\psi)\end{pmatrix} \tag{4}$$

Mescheder et al. (44) showed that local convergence near (θ^*,ψ^*) can be analyzed by examining the spectrum of the Jacobian $\mathbf{J}_{F_h}$ at the equilibrium: if the Jacobian has eigenvalues with absolute value bigger than 1, then training does not converge. On the other hand, if all eigenvalues have absolute value smaller than 1, then training will converge to (θ^*,ψ^*) at a linear rate. If all eigenvalues have absolute value equal to 1, the convergence behavior is undetermined.

Given some calculations (43), we can show that the eigenvalues of the Jacobian of the update operator $\lambda_{\mathbf{J}_{F_h}}$ can be determined by $\lambda_{\mathbf{J}_v}$:

$$\lambda_{\mathbf{J}_{F_h}}=1+h\lambda_{\mathbf{J}_v}\ . \tag{5}$$

That is, given small enough h (43), the training dynamics can instead be examined using $\lambda_{\mathbf{J}_v}$, *i.e.*, the eigenvalues of the Jacobian of the gradient vector field. If all $\lambda_{\mathbf{J}_v}$ have a negative real part, the training will locally converge to (θ^*,ψ^*) at a linear rate. On the other hand, if some $\lambda_{\mathbf{J}_v}$ have a positive real part, the training is not convergent. If all $\lambda_{\mathbf{J}_v}$ have a zero real part, the convergence behavior is inconclusive.

H DiracRpGAN: A demonstration of non-convergence

Summary. To obtain DiracRpGAN, we apply Eq. 2 to the DiracGAN (43) problem setting. After simplification, DiracRpGAN and DiracGAN are different only by a constant. They have the same gradient vector field, therefore all proofs are identical to Mescheder *et al.* (43).

Definition B.1. *The DiracRpGAN consists of a (univariate) generator distribution $p_\theta=\delta_\theta$ and a linear discriminator $D_\psi(x)=\psi\cdot x$. The true data distribution $p_\mathcal{D}$ is given by a Dirac distribution concentrated at 0.*

In this setup, the RpGAN training objective is given by:

$$\mathcal{L}(\theta,\psi)=f(\psi\theta)\ . \tag{6}$$

We can now show analytically that DiracRpGAN does not converge without regularzation.

Lemma B.2. *The unique equilibrium point of the training objective in Eq. 6 is given by $\theta=\psi=0$. Moreover, the Jacobian of the gradient vector field at the equilibrium point has the two eigenvalues $\pm f'(0)i$ which are both on the imaginary axis.*

The gradient vector field v of Eq. 6 is given by:

$$v(\theta,\psi)=\begin{pmatrix}-\nabla_\theta\mathcal{L}(\theta,\psi)\\ \nabla_\psi\mathcal{L}(\theta,\psi)\end{pmatrix}=\begin{pmatrix}-\psi f'(\psi\theta)\\ \theta f'(\psi\theta)\end{pmatrix} \tag{7}$$

and the Jacobian of v:

$$\mathbf{J}_v=\begin{pmatrix}-\psi^2 f''(\psi\theta) & -f'(\psi\theta)-\psi\theta f''(\psi\theta)\\ f'(\psi\theta)+\psi\theta f''(\psi\theta) & \theta^2 f''(\psi\theta)\end{pmatrix} \tag{8}$$

Evaluating $\mathbf{J}_v$ at the equilibrium point $\theta=\psi=0$ gives us:

$$\mathbf{J}_v\Big|_{(0,0)}=\begin{pmatrix}0 & -f'(0)\\ f'(0) & 0\end{pmatrix} \tag{9}$$

Therefore, the eigenvalues of $\mathbf{J}_v$ are $\lambda_{1/2}=\pm f'(0)i$, both of which have a real part of 0. Thus, the convergence of DiracRpGAN is inconclusive and further analysis is required.

Lemma B.3. *The integral curves of the gradient vector field $v(\theta,\psi)$ do not converge to the equilibrium point. More specifically, every integral curve $(\theta(t),\psi(t))$ of the gradient vector field $v(\theta,\psi)$ satisfies $\theta(t)^2+\psi(t)^2=const$ for all $t\in[0,\infty)$.*

Let $R(\theta,\psi)=\frac{1}{2}(\theta^2+\psi^2)$, then:

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}R(\theta(t),\psi(t))\\ &=-\theta(t)\psi(t)f'(\theta(t)\psi(t))+\psi(t)\theta(t)f'(\theta(t)\psi(t))\\ &=0\ .\end{aligned} \tag{10}$$

We see that the distance between (θ,ψ) and the equilibrium point $(0,0)$ stays constant. Therefore, training runs in circles and never converges.

Next, we investigate the convergence behavior of DiracRpGAN with regularization. For DiracRpGAN, both R_1 and R_2 can be reduced to the following form:

$$R(\psi) = \frac{\gamma}{2}\psi^2 \tag{11}$$

Lemma B.4. *The eigenvalues of the Jacobian of the gradient vector field for the gradient-regularized DiracRpGAN at the equilibrium point are given by*

$$\lambda_{1/2} = -\frac{\gamma}{2} \pm \sqrt{\frac{\gamma^2}{4} - f'(0)} \tag{12}$$

In particular, for $\gamma > 0$ all eigenvalues have a negative real part. Hence, gradient descent is locally convergent for small enough learning rates.

With regularization, the gradient vector field becomes

$$\tilde{v}(\theta,\psi) = \begin{pmatrix} -\psi f'(\psi\theta) \\ \theta f'(\psi\theta) - \gamma\psi \end{pmatrix} \tag{13}$$

the Jacobian of $\tilde{v}$ is then given by

$$\mathbf{J}_{\tilde{v}} = \begin{pmatrix} -\psi^2 f''(\psi\theta) & -f'(\psi\theta) - \psi\theta f''(\psi\theta) \\ f'(\psi\theta) + \psi\theta f''(\psi\theta) & \theta^2 f''(\psi\theta) - \gamma \end{pmatrix} \tag{14}$$

evaluating the Jacobian at $\theta = \psi = 0$ yields

$$\mathbf{J}_{\tilde{v}}\Big|_{(0,0)} = \begin{pmatrix} 0 & -f'(0) \\ f'(0) & -\gamma \end{pmatrix} \tag{15}$$

given some calculations, we arrive at Eq.12.

I General Convergence Results

Summary. The proofs are largely the same as Mescheder *et al.* (43). We use the same proving techniques, and only slightly modify the assumptions and proof details to adapt Mescheder *et al.*'s effort to RpGAN. Like in (43), our proofs do not rely on unrealistic assumptions such as $\operatorname{supp} p_{\mathcal{D}} = \operatorname{supp} p_\theta$.

I.1 Assumptions

We closely follow (43) but modify the assumptions wherever necessary to tailor the proofs for RpGAN. Like in (43), we also consider the realizable case where there exists θ such that G_θ produces the true data distribution.

Assumption I. *We have $p_{\theta^*} = p_{\mathcal{D}}$, and $D_{\psi^*} = C$ in some local neighborhood of* $\operatorname{supp} p_{\mathcal{D}}$*, where C is some arbitrary constant.*

Since RpGAN is defined on critic difference rather than raw logits, we no longer require D_{ψ^*} to produce 0 on $\operatorname{supp} p_{\mathcal{D}}$, instead any constant C would suffice.

Assumption II. *We have $f'(0) \neq 0$ and $f''(0) < 0$.*

This assumption is the same as in (43). The choice $f(t) = -\log(1 + e^{-t})$ adopted in the main text satisfies this assumption.

As discussed in (43), there generally is not a single equilibrium point (θ^*, ψ^*), but a submanifold of equivalent equilibria corresponding to different parameterizations of the same function. It is therefore necessary to represent the equilibrium as *reparameterization manifolds* $\mathcal{M}_G$ and $\mathcal{M}_D$. We modify the reparameterization h as follows:

$$h(\psi) = \mathbb{E}_{\substack{x \sim p_{\mathcal{D}} \\ y \sim p_{\mathcal{D}}}}\left[|D_\psi(x) - D_\psi(y)|^2 + \|\nabla_x D_\psi(x)\|^2\right] \tag{16}$$

to account for the fact that D_{ψ^*} is now allowed to have any constant value on $\operatorname{supp} p_{\mathcal{D}}$. The *reparameterization manifolds* are then given by:

$$\mathcal{M}_G = \{\theta \mid p_\theta = p_{\mathcal{D}}\} \tag{17}$$

$$\mathcal{M}_D = \{\psi \mid h(\psi) = 0\} \tag{18}$$

We assume the same regularity properties as in (43) for $\mathcal{M}_G$ and $\mathcal{M}_D$ near the equilibrium. To state these assumptions, we need:

$$g(\theta)=\mathbb{E}_{x\sim p_\theta}[\nabla_\psi D_\psi|_{\psi=\psi^*}] \tag{19}$$

which leads to:

Assumption III. *There are ϵ-balls $B_\epsilon(\theta^*)$ and $B_\epsilon(\psi^*)$ around θ^* and ψ^* so that $\mathcal{M}_G\cap B_\epsilon(\theta^*)$ and $\mathcal{M}_D\cap B_\epsilon(\psi^*)$ define $\mathcal{C}^1$-manifolds. Moreover, the following holds*:

(i) *if $v\in\mathbb{R}^n$ is not in $\mathcal{T}_{\psi^*}\mathcal{M}_D$, then $\partial_v^2 h(\psi^*)\neq 0$.*

(ii) *if $w\in\mathbb{R}^m$ is not in $\mathcal{T}_{\theta^*}\mathcal{M}_G$, then $\partial_w g(\theta^*)\neq 0$.*

These two conditions have exactly the same meanings as in (43): the first condition indicates the geometry of $\mathcal{M}_D$ can be locally described by the second derivative of h. The second condition implies that D is strong enough that it can detect any deviation from the equilibrium generator distribution. This is the only assumption we have about the expressiveness of D.

I.2 Convergence

We can now show the general convergence result for gradient penalized RpGAN, consider the gradient vector field with either R_1 or R_2 regularization:

$$\tilde{v}_i(\theta,\psi)=\begin{pmatrix}-\nabla_\theta\mathcal{L}(\theta,\psi)\\ \nabla_\psi\mathcal{L}(\theta,\psi)-\nabla_\psi R_i(\theta,\psi)\end{pmatrix} \tag{20}$$

note that the convergence result can also be trivially extended to the case where both R_1 and R_2 are applied. We omit the proof for this case as it is redundant once the convergence with either regularization is proven.

Theorem. *Assume Assumption I, II and III hold for (θ^*,ψ^*). For small enough learning rates, gradient descent for $\tilde{v}_1$ and $\tilde{v}_2$ are both convergent to $\mathcal{M}_G\times\mathcal{M}_D$ in a neighborhood of (θ^*,ψ^*). Moreover, the rate of convergence is at least linear.*

We extend the convergence proof by Mescheder *et al.* (43) to our setting. We first prove lemmas necessary to our main proof.

Lemma C.2.1. *Assume $J\in\mathbb{R}^{(n+m)\times(n+m)}$ is of the following form*:

$$J=\begin{pmatrix}0 & -B^\top\\ B & -Q\end{pmatrix} \tag{21}$$

where $Q\in\mathbb{R}^{m\times m}$ is a symmetric positive definite matrix and $B\in\mathbb{R}^{m\times n}$ has full column rank. Then all eigenvalues λ of J satisfy $\Re(\lambda)<0$.

Proof. See Mescheder *et al.* (43), Theorem A.7.

Lemma C.2.2. *The gradient of $\mathcal{L}(\theta,\psi)$ w.r.t. θ and ψ are given by*:

$$\begin{aligned}\nabla_\theta\mathcal{L}(\theta,\psi)=&\mathbb{E}_{\substack{z\sim p_z\\ x\sim p_\mathcal{D}}}[f'(D_\psi(G_\theta(z))-D_\psi(x))\\ &[\nabla_\theta G_\theta(z)]^\top\nabla_x D_\psi(G_\theta(z))] \quad (22)\\ \nabla_\psi\mathcal{L}(\theta,\psi)=&\mathbb{E}_{\substack{z\sim p_z\\ x\sim p_\mathcal{D}}}[f'(D_\psi(G_\theta(z))-D_\psi(x))\\ &(\nabla_\psi D_\psi(G_\theta(z))-\nabla_\psi D_\psi(x))] \quad (23)\end{aligned}$$

Proof. This is just the chain rule.

Lemma C.2.3. *Assume that (θ^*,ψ^*) satisfies Assumption I. The Jacobian of the gradient vector field $v(\theta,\psi)$ at (θ^*,ψ^*) is then*

$$\mathbf{J}_v\Big|_{(\theta^*,\psi^*)}=\begin{pmatrix}0 & -K_{DG}^\top\\ K_{DG} & K_{DD}\end{pmatrix} \tag{24}$$

the terms K_{DD} and K_{DG} are given by

$$\begin{aligned}K_{DD}=&f''(0)\mathbb{E}_{\substack{x\sim p_\mathcal{D}\\ y\sim p_\mathcal{D}}}[(\nabla_\psi D_{\psi^*}(x)-\nabla_\psi D_{\psi^*}(y))\\ &(\nabla_\psi D_{\psi^*}(x)-\nabla_\psi D_{\psi^*}(y))^\top] \quad (25)\\ K_{DG}=&f'(0)\nabla_\theta\mathbb{E}_{x\sim p_\theta}[\nabla_\psi D_{\psi^*}(x)]\,|_{\theta=\theta^*} \quad (26)\end{aligned}$$

Proof. Note that

$$\mathbf{J}_v\bigg|_{(\theta^*,\psi^*)} = \begin{pmatrix} -\nabla_\theta^2 \mathcal{L}(\theta^*,\psi^*) & -\nabla_{\theta,\psi}^2 \mathcal{L}(\theta^*,\psi^*) \\ \nabla_{\theta,\psi}^2 \mathcal{L}(\theta^*,\psi^*) & \nabla_\psi^2 \mathcal{L}(\theta^*,\psi^*) \end{pmatrix} \tag{27}$$

By Assumption I, $D_{\psi^*} = C$ in some neighborhood of $\mathrm{supp} p_{\mathcal{D}}$. Therefore we also have $\nabla_x D_{\psi^*} = 0$ and $\nabla_x^2 D_{\psi^*} = 0$ for $x \in \mathrm{supp} p_{\mathcal{D}}$. Using these two conditions, we see that $\nabla_\theta^2 \mathcal{L}(\theta^*,\psi^*) = 0$.

To see Eq.25 and Eq.26, simply take the derivatives of Eq.23 and evaluate at (θ^*,ψ^*).

Lemma C.2.4. *The gradient $\nabla_\psi R_i(\theta,\psi)$ of the regularization terms R_i, $i \in \{1,2\}$, w.r.t. ψ are*

$$\nabla_\psi R_1(\theta,\psi) = \gamma \mathbb{E}_{x\sim p_{\mathcal{D}}}[\nabla_{\psi,x} D_\psi \nabla_x D_\psi] \tag{28}$$

$$\nabla_\psi R_2(\theta,\psi) = \gamma \mathbb{E}_{x\sim p_\theta}[\nabla_{\psi,x} D_\psi \nabla_x D_\psi] \tag{29}$$

Proof. See Mescheder *et al.* (43), Lemma D.3.

Lemma C.2.5. *The second derivatives $\nabla_\psi^2 R_i(\theta^*,\psi^*)$ of the regularization terms R_i, $i \in \{1,2\}$, w.r.t. ψ at (θ^*,ψ^*) are both given by*

$$L_{DD} = \gamma \mathbb{E}_{x\sim p_{\mathcal{D}}}[AA^\top] \tag{30}$$

where $A = \nabla_{\psi,x} D_{\psi^*}$. Moreover, both regularization terms satisfy $\nabla_{\theta,\psi} R_i(\theta^*,\psi^*) = 0$.

Proof. See Mescheder *et al.* (43), Lemma D.4.

Given Lemma C.2.3, Lemma C.2.5 and Eq.20, we can now show that the Jacobian of the regularized gradient field at the equilibrium point is given by

$$\mathbf{J}_{\tilde{v}}\bigg|_{(\theta^*,\psi^*)} = \begin{pmatrix} 0 & -K_{DG}^\top \\ K_{DG} & M_{DD} \end{pmatrix} \tag{31}$$

where $M_{DD} = K_{DD} - L_{DD}$. To prove our main theorem, we need to examine $\mathbf{J}_{\tilde{v}}$ when restricting it to the space orthogonal to $\mathcal{T}_{(\theta^*,\psi^*)}\mathcal{M}_G \times \mathcal{M}_D$.

Lemma C.2.6. *Assume Assumptions II and III hold. If $v \neq 0$ is not in $\mathcal{T}_{\psi^*}\mathcal{M}_D$, then $v^\top M_{DD} v < 0$.*

Proof. By Lemma C.2.3 and Lemma C.2.5, we have

$$v^\top K_{DD} v = f''(0) \mathbb{E}_{\substack{x\sim p_{\mathcal{D}} \\ y\sim p_{\mathcal{D}}}}[((\nabla_\psi D_{\psi^*}(x) - \nabla_\psi D_{\psi^*}(y))^\top v)^2] \tag{32}$$

$$v^\top L_{DD} v = \gamma \mathbb{E}_{x\sim p_{\mathcal{D}}}\left[\|Av\|^2\right] \tag{33}$$

By Assumption II, we have $f''(0) < 0$. Therefore $v^\top M_{DD} v \leq 0$. Suppose $v^\top M_{DD} v = 0$, this implies

$$(\nabla_\psi D_{\psi^*}(x) - \nabla_\psi D_{\psi^*}(y))^\top v = 0 \quad \text{and} \quad Av = 0 \tag{34}$$

for all $(x,y) \in \mathrm{supp}\ p_{\mathcal{D}} \times \mathrm{supp}\ p_{\mathcal{D}}$. Recall the definition of $h(\psi)$ from Eq.16. Using the fact that $D_{\psi^*} = C$ and $\nabla_x D_{\psi^*} = 0$ for $x \in \mathrm{supp}\ p_{\mathcal{D}}$, we see that the Hessian of $h(\psi)$ at ψ^* is

$$\begin{aligned} \nabla_\psi^2 h(\psi^*) = 2\mathbb{E}_{\substack{x\sim p_{\mathcal{D}} \\ y\sim p_{\mathcal{D}}}}[&(\nabla_\psi D_{\psi^*}(x) - \nabla_\psi D_{\psi^*}(y)) \\ &(\nabla_\psi D_{\psi^*}(x) - \nabla_\psi D_{\psi^*}(y))^\top + AA^\top] \end{aligned} \tag{35}$$

The second directional derivative $\partial_v^2 h(\psi)$ is therefore

$$\begin{aligned} &\partial_v^2 h(\psi) \\ &= 2\mathbb{E}_{\substack{x\sim p_{\mathcal{D}} \\ y\sim p_{\mathcal{D}}}}\left[\left|(\nabla_\psi D_{\psi^*}(x) - \nabla_\psi D_{\psi^*}(y))^\top v\right|^2 + \|Av\|^2\right] \\ &= 0 \end{aligned} \tag{36}$$

By Assumption III, this can only hold if $v \in \mathcal{T}_{\psi^*}\mathcal{M}_D$.

Lemma C.2.7. *Assume Assumption III holds. If $w \neq 0$ is not in $\mathcal{T}_{\theta^*}\mathcal{M}_G$, then $K_{DG} w \neq 0$.*

Proof. See Mescheder *et al.* (43), Lemma D.6.

Proof for the main theorem. Given previous lemmas, by choosing local coordinates $\theta(\alpha,\gamma_G)$ and $\psi(\beta,\gamma_D)$ for $\mathcal{M}_G$ and $\mathcal{M}_D$ such that $\theta^* = 0$, $\psi^* = 0$ as well as

$$\mathcal{M}_G = \mathcal{T}_{\theta^*}\mathcal{M}_G = \{0\}^k \times \mathbb{R}^{n-k} \tag{37}$$

$$\mathcal{M}_D = \mathcal{T}_{\psi^*}\mathcal{M}_D = \{0\}^l \times \mathbb{R}^{m-l} \tag{38}$$

our proof is *exactly* the same as Mescheder *et al.* (43), Theorem 4.1.

J Hyperparameters, training configurations, and compute

We implement our models on top of the official StyleGAN3 code base. While the loss function and the models are implemented from scratch, we reuse support code from the existing implementation whenever possible. This includes exponential moving average (EMA) of generator weights (25), non-leaky data augmentation (27), and metric evaluation (28).

Training schedule. To speed up the convergence early in training, we specify a cosine schedule for the following hyperparameters before they reach their target values:

- Learning rate
- γ for R_1 and R_2 regularization
- Adam β_2
- EMA half-life
- Augmentation probability

We call this early training stage the burn-in phase. Burn-in length and schedule for each hyperparameter are listed in Table 7 for each experiment. A schedule for the EMA half-life can already be found in Karras *et al.* (27), albeit they use a linear schedule. A lower initial Adam β_2 is crucial to the initial large learning rate as it allows the optimizer to adapt to the gradient magnitude change much quicker. We use a large initial γ to account for that early in training: p_θ and $p_\mathcal{D}$ are far apart and a large γ smooths both distributions more aggressively which makes learning easier. Augmentation is not necessary until D starts to overfit later on; thus, we set the initial augmentation probability to 0.

Dataset augmentation. We apply horizontal flips and non-leaky augmentation (27) to all datasets where augmentation is enabled. Following (27), we include pixel blitting, geometric transformations, and color transforms in the augmentation pipeline. We additionally include cutout augmentation which works particularly well with our model, although it does not seem to have much effect on StyleGAN2. We also find it beneficial to apply color transforms less often and thus set their probability multiplier to 0.5 while retaining the multiplier 1 for other types of augmentations. As previously mentioned, we apply a fixed cosine schedule to the augmentation probability rather than adjusting it adaptively as in (27). We did not observe any performance degradation with this simplification.

Network capacity. We keep the capacity distribution for each resolution the same as in (27; 28). We place two residual blocks per resolution which makes our model roughly $3\times$ as deep, $1.5\times \sim 3\times$ as wide as StyleGAN2 while maintaining the same model size on CIFAR-10 and FFHQ. For the ImageNet model, we double the number of channels which results in roughly $4\times$ as many parameters as the default StyleGAN2 configuration.

Mixed precision training. We apply mixed precision training as in (27; 28) where all parameters are stored in FP32, but cast to lower precision along with the activation maps for the 4 highest resolutions. We notice that using FP16 as the low precision format cripples the training of our model. However, we see no problem when using BFloat16 instead.

Class conditioning. For class conditional models, we follow the same conditioning scheme as in (27). For G, the conditional latent code z' is the concatenation of z and the embedding of the class label c, specifically $z' = \text{concat}(z, \text{embed}(c))$. For D, we use a projection discriminator (46) which evaluates the dot product of the class embedding and the feature vector $D'(x)$ produced by the last layer of D, concretely $D(x) = \text{embed}(c) \cdot D'(x)^\top$. We do not employ any normalization-based conditioning such as AdaIN (30), AdaGN (7; 26), AdaBN (4) or AdaLN (50) for simplicity, even though they improve FID considerably.

Stacked MNIST. We base this model off of the CIFAR-10 model but without class conditioning. We disable all data augmentation and shorten the burn-in phase considerably. We use a constant learning rate and did not observe any benefit of using a lower learning rate later in the training.

Compute resources. We train the Stacked MNIST and CIFAR-10 models on an $8\times$ NVIDIA L40 node. Training took 7 hours for Stacked MNIST and 4 days for CIFAR-10. The FFHQ model was trained on an $8\times$ NVIDIA A6000 f0r roughly 3 weeks. The ImageNet model was trained on NVIDIA A100/H100 clusters and training took one day on 32 H100s (about 5000 H100 hours).

Hyperparameter	Stacked MNIST	CIFAR-10	FFHQ	ImageNet
Resolution	32×32	32×32	256×256	32×32
Class conditional	-	✓	-	✓
Number of GPUs	8	8	8	32
Duration (Mimg)	10	200	150	700
Burn-in (Mimg)	2	20	20	200
Minibatch size	512	512	256	4096
Learning rate	2×10^{-4}	$2\times10^{-4}\rightarrow5\times10^{-5}$	$2\times10^{-4}\rightarrow5\times10^{-5}$	$2\times10^{-4}\rightarrow5\times10^{-5}$
γ for R_1 and R_2	$1\rightarrow0.1$	$0.05\rightarrow0.005$	$500\rightarrow50$	$0.5\rightarrow0.05$
Adam β_2	$0.9\rightarrow0.99$	$0.9\rightarrow0.99$	$0.9\rightarrow0.99$	$0.9\rightarrow0.99$
EMA half-life (Mimg)	$0\rightarrow0.5$	$0\rightarrow5$	$0\rightarrow0.5$	$0\rightarrow50$
Channels per resolution	768-768-768-768	768-768-768-768	96-192-384-768-768-768-768	1536-1536-1536-1536
ResBlocks per resolution	2-2-2-2	2-2-2-2	2-2-2-2-2-2-2	2-2-2-2
Groups per resolution	96-96-96-96	96-96-96-96	12-24-48-96-96-96-96	96-96-96-96
G params	20.73M	20.78M	23.06M	82.91M
D params	20.68M	21.28M	23.01M	86.55M
Dataset x-flips	-	✓	✓	✓
Augment probability	-	$0\rightarrow0.55$	$0\rightarrow0.15$	$0\rightarrow0.5$

Table 7: Hyperparameters for each experiment.

K Qualitative Results

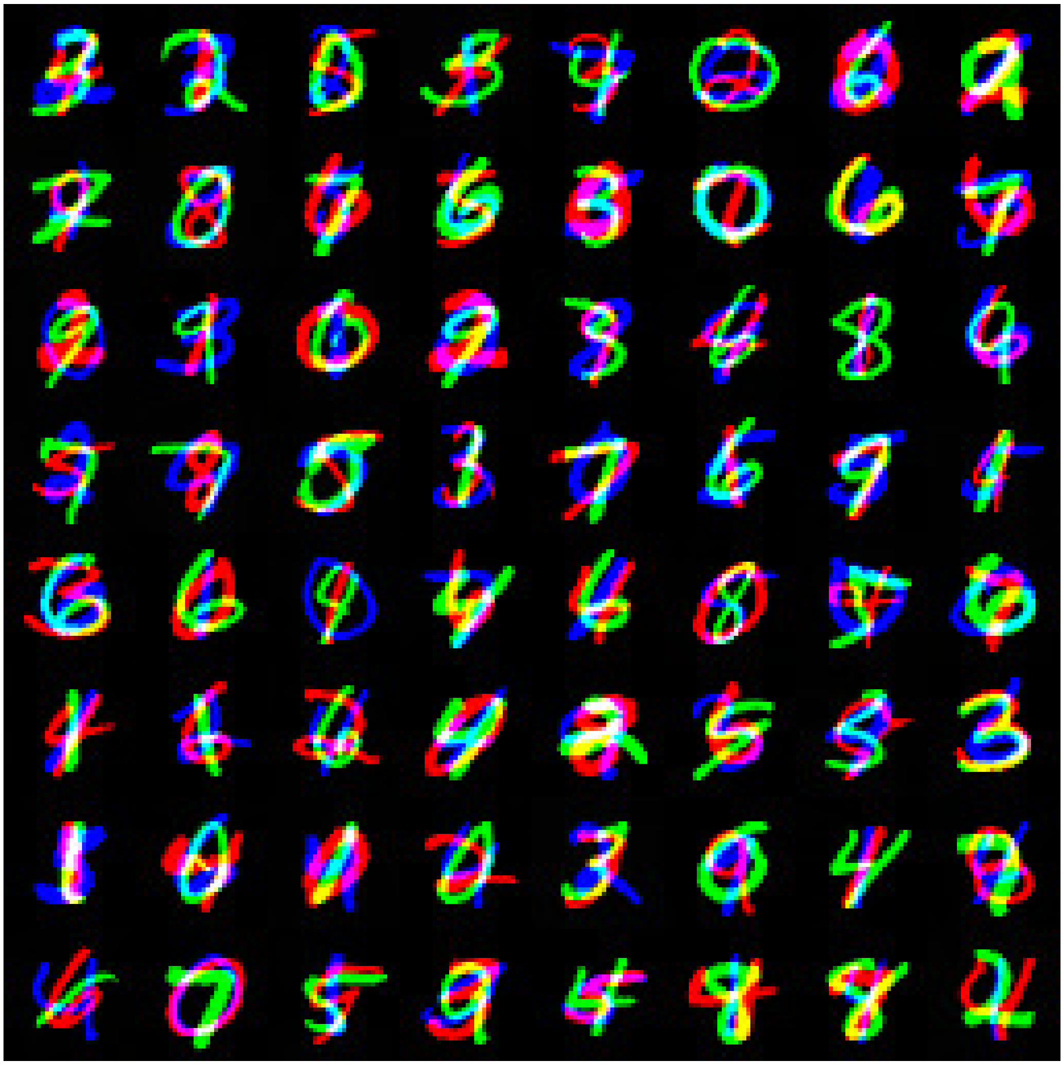

Figure 5: Qualitative examples of sample generation from our Config E on Stacked-MNIST.

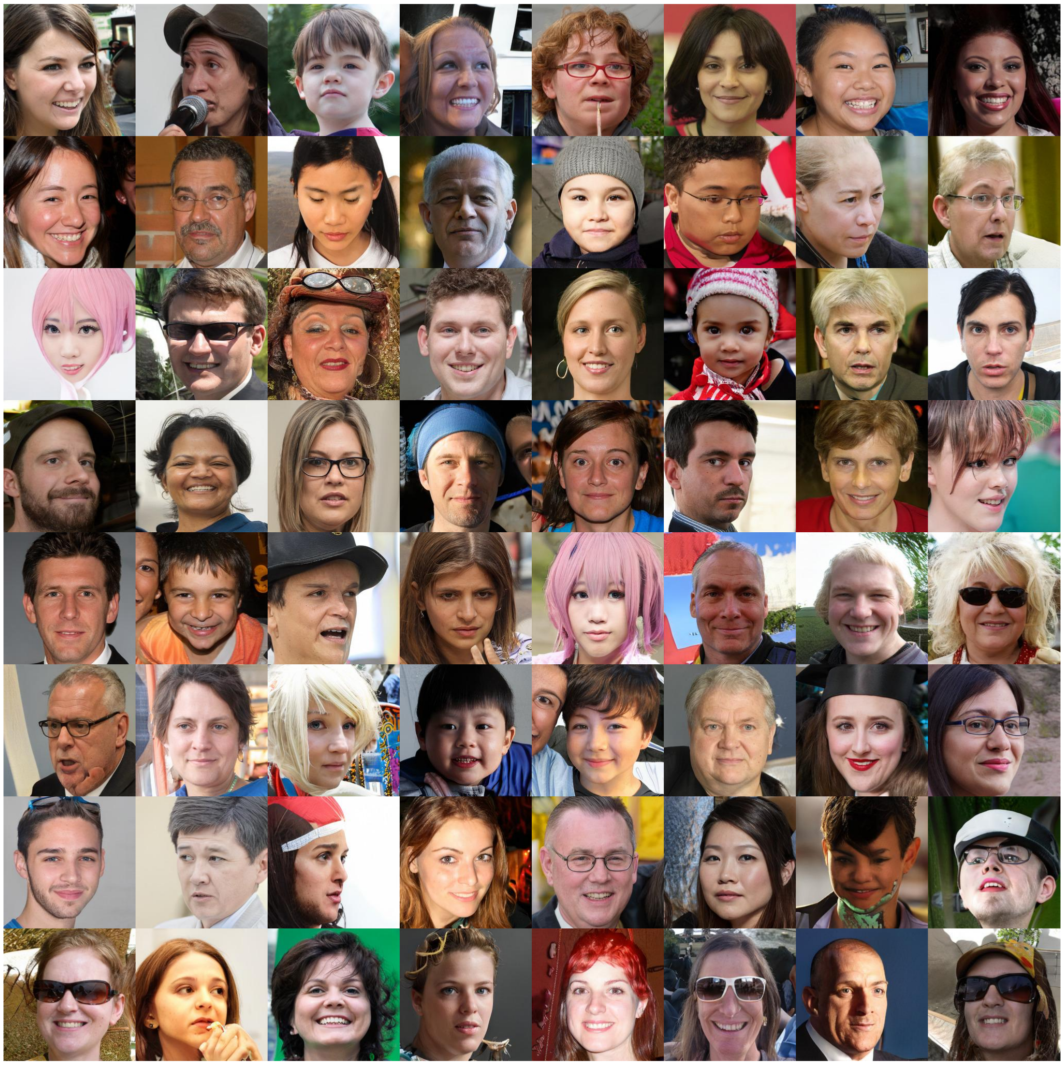

Figure 6: More qualitative examples of sample generation from our Config E on FFHQ-256.

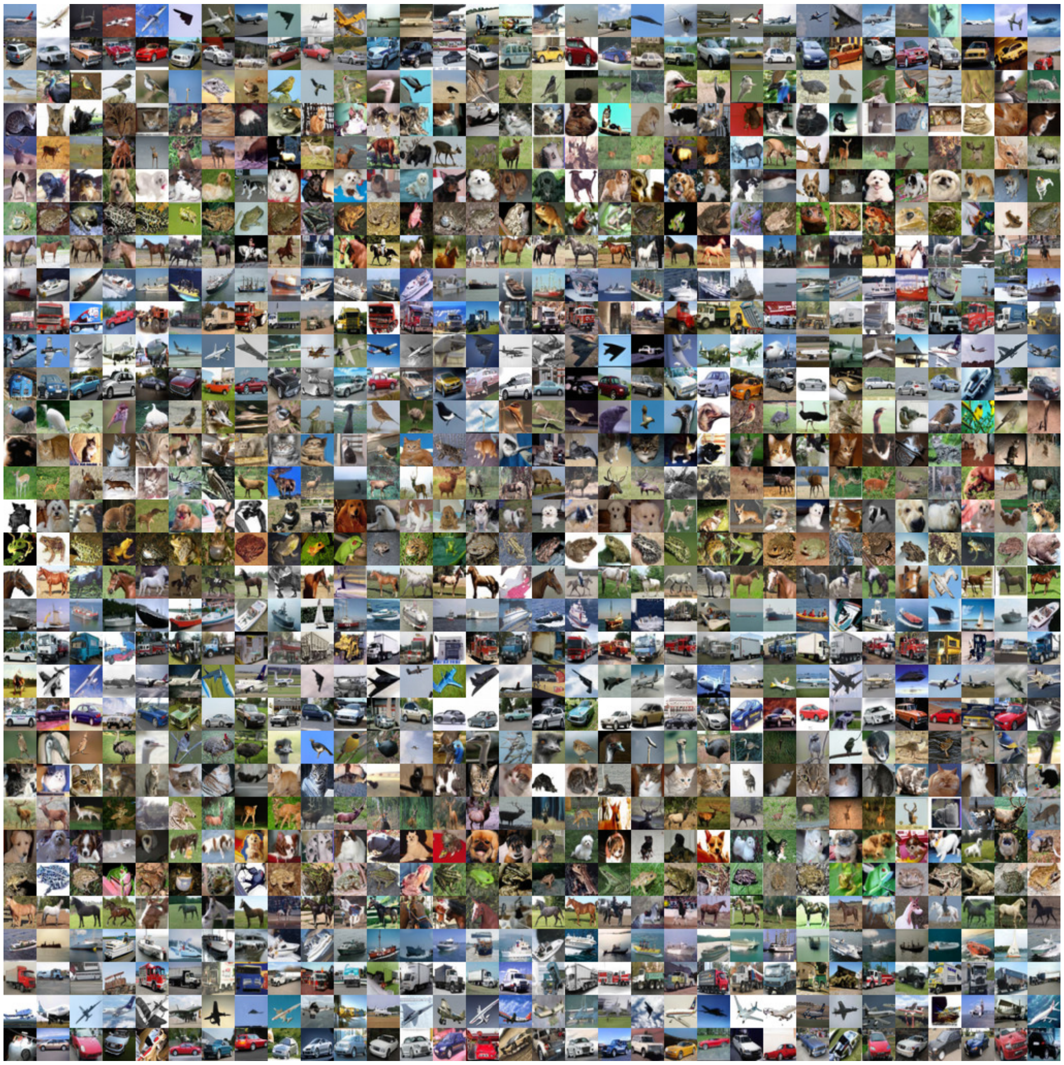

Figure 7: Qualitative examples of sample generation from our Config E on CIFAR-10.

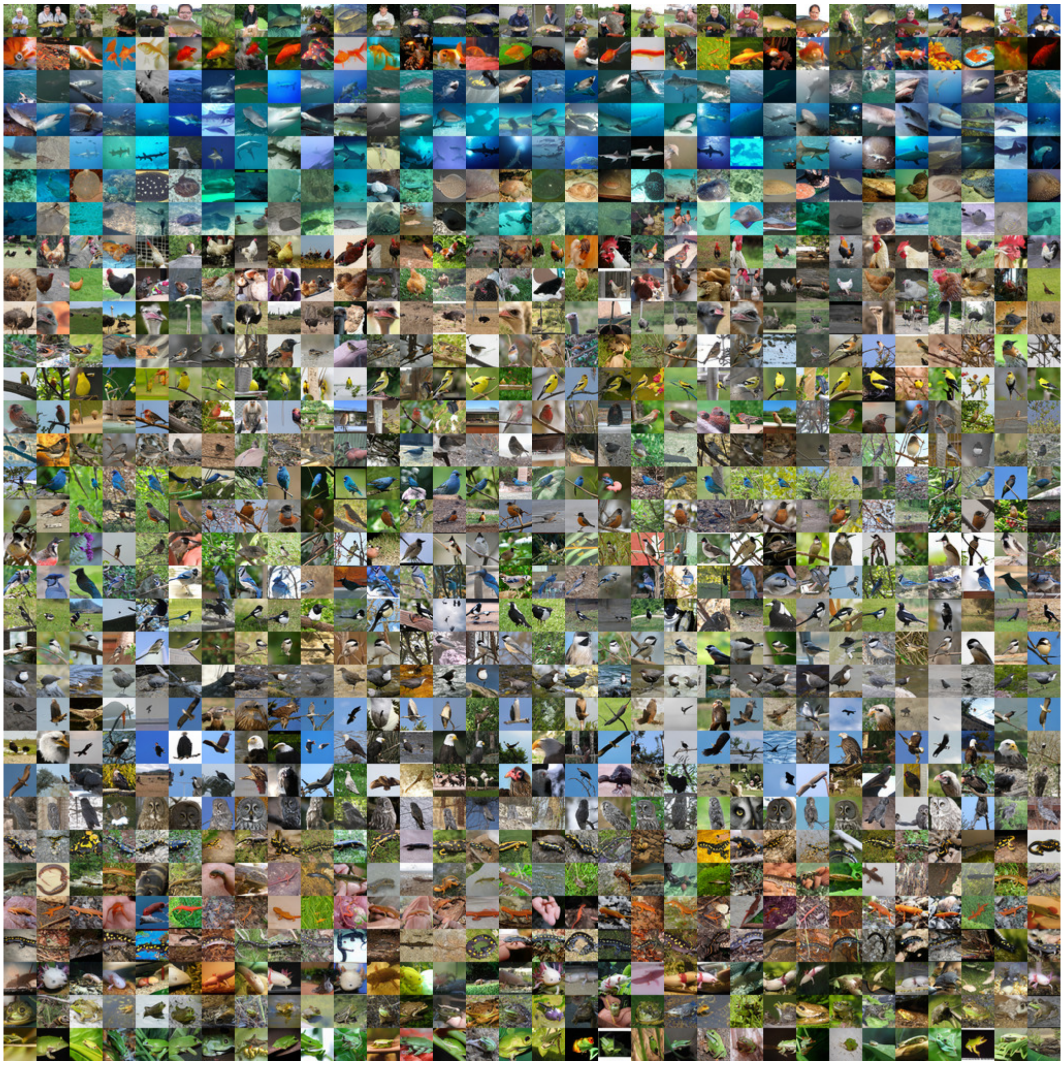

Figure 8: Qualitative examples of sample generation from our Config E on ImageNet-32.