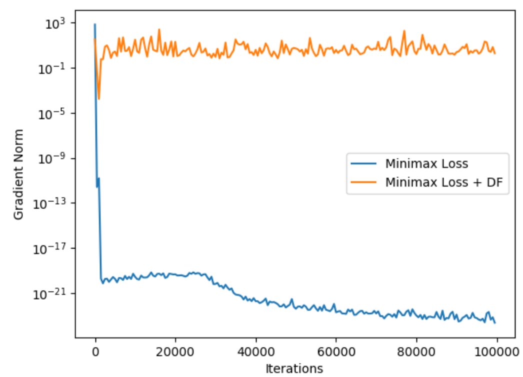
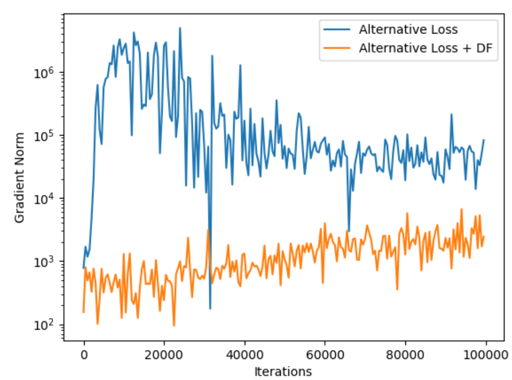
Supplementary Material for:
On Stabilizing Generative Adversarial Training with Noise

1. Influence on the Generator Gradient Norm

We compare the norm of the generator gradient with and without DF for a GAN trained with the original minimax objective and a GAN trained with the alternative generator objective $\max_G \log(D(z))$ in Figure 1. The models were trained on CIFAR-10. We can observe the vanishing gradient phenomenon in Figure 1a when no distribution filtering is applied. With our proposed method the gradient norms are stable. In the case of the alternative loss in Figure 1b we can observe that the gradient norms are orders of magnitude higher when no distribution filtering is applied. This results in highly unstable weight updates due to the overconfident discriminator.



(a) GAN with minimax loss



(b) GAN with alternative loss

Figure 1: We show the norm of the generator gradient over the course of training for a GAN using the original minimax objective in (a) and a GAN using the alternative objective $\max_G \log(D(z))$ in (b).

2. Experiments on synthetic data

We performed experiments with a standard GAN and a DFGAN using Gaussian noise on synthetic 2-D data. The generator and discriminator architectures are both MLPs consisting of three fully-connected layers with a hidden-layer size of 512. We use ReLU activations and batch-normalization ([1]) in all but the first discriminator layer and the output layers. The Adam optimzer ([2]) was used with a learning rate of 10^{-4} and we trained for 20K iterations. The results are shown in Figure 2. We can observe how the matching of both clean and filtered distribution leads to a better fit in the case of DFGAN.

3. Implementation Details

Noise Generator. The noise-generator architecture in all our experiments is equivalent to the generator architecture with the number of filters reduced by a factor of eight. The output of the noise-generator has a `tanh` activation scaled by a factor of two to allow more noise if necessary. We also experimented with a linear activation but didn't find a significant difference in performance.

GAN+GP. For the comparisons to the GAN regularizer proposed by [4] we used the same settings as used in their work in experiments with DCGAN.

SNGAN+DF. We used the standard GAN loss (same as DCGAN) in all our experiments with models using spectral normalization. When combining SNGAN with DF we batch-normalized the noisy inputs to the discriminator.

4. Qualitative Results for Experiments

We provide qualitative results for some of the ablation experiments in Figure 3 and for the robustness experiments in Figure 4. As we can see in Figure 4, none of the tested settings led to degenerate solutions in the case of DFGAN while the other methods would show failure cases in some settings.

5. Application to Progressive GAN

To test our method on a state-of-the-art GAN we applied our training method to the progressive GAN model. We used the DCGAN loss, trained for a total of 6M images and did not use label conditioning. We used fixed Gaussian noise for the distribution filtering. On CIFAR-10 progressiveGAN without DF achieved a FID of 29.4. Adding DF improved the performance to 26.8. Note that the original WGAN-GP loss in the same setup only achieved a FID of 29.8.

We also trained progressive-GAN+DF on higher resolution 256×256 images of LSUN bedrooms. See Figure 5 for results.

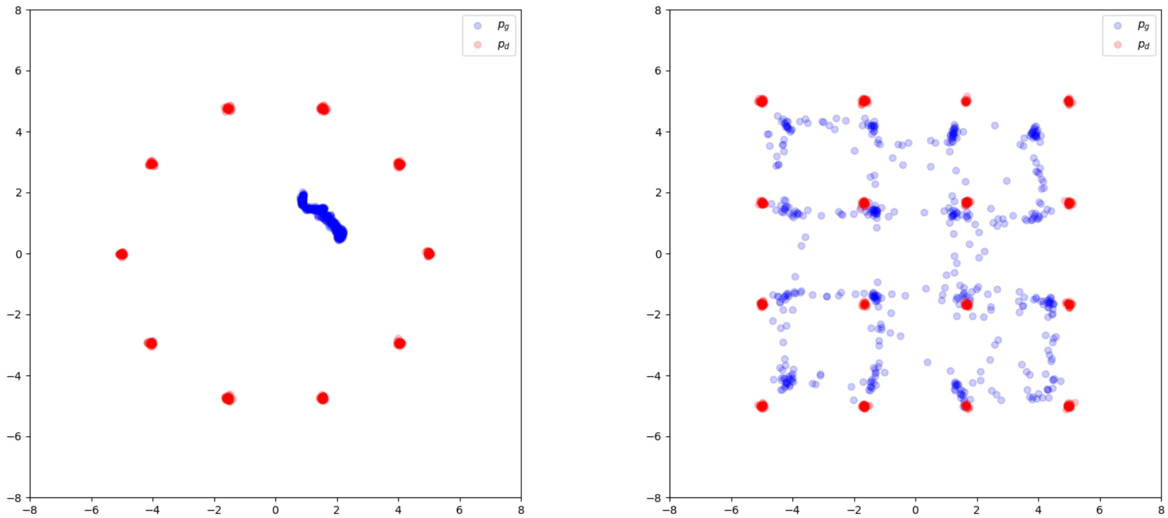
References

[1] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 448–456. JMLR. org, 2015. 1

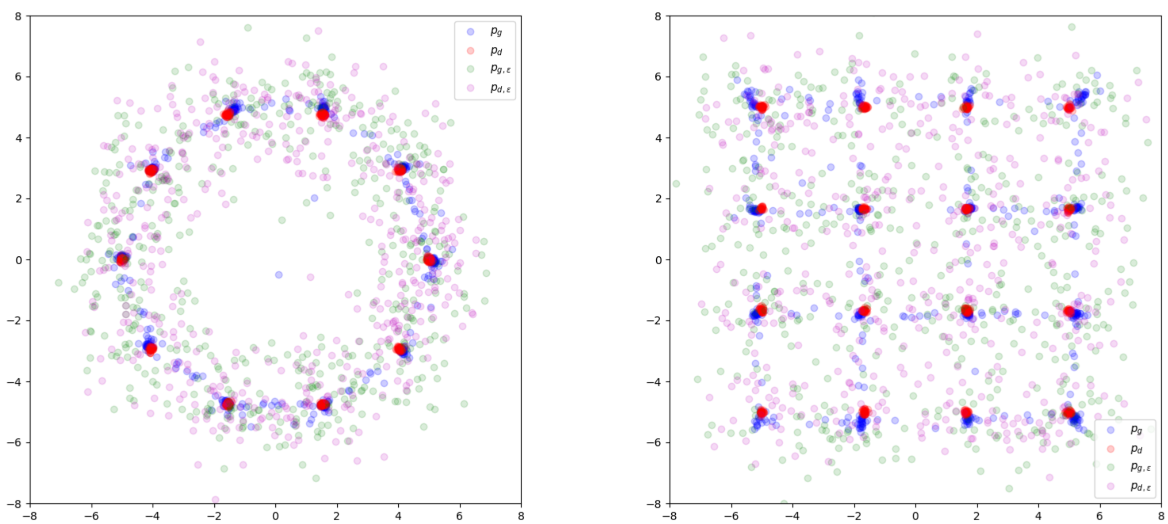
[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[3] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 5

[4] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems*, pages 2015–2025, 2017. 1, 5

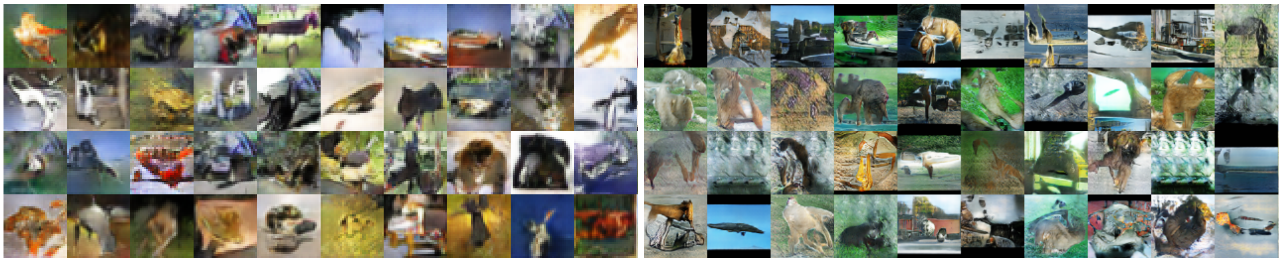


(a) Standard GAN

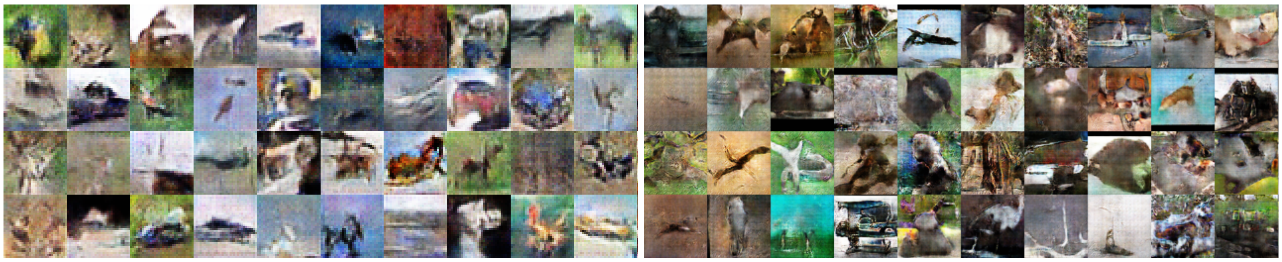


(b) DFGAN with Gaussian noise

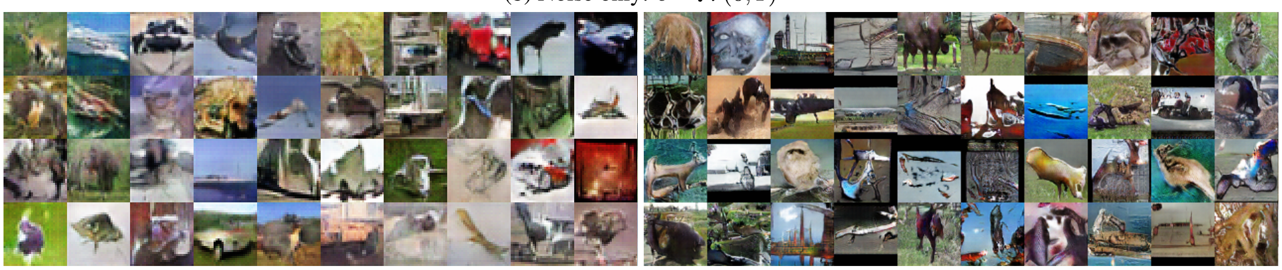
Figure 2: We performed experiments on synthetic 2D data with a standard GAN (*top*) and a DFGAN (*bottom*). The ground truth data is shown in *red* and the model generated data is shown in *blue*. For DFGAN we also show samples from the blurred data distribution $p_{d,\epsilon}$ in *green* and the blurred model distribution $p_{g,\epsilon}$ in *purple*.



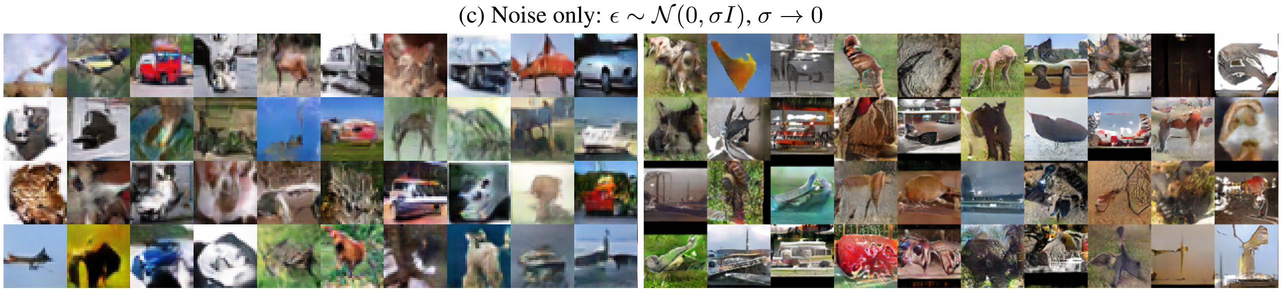
(a) Standard GAN



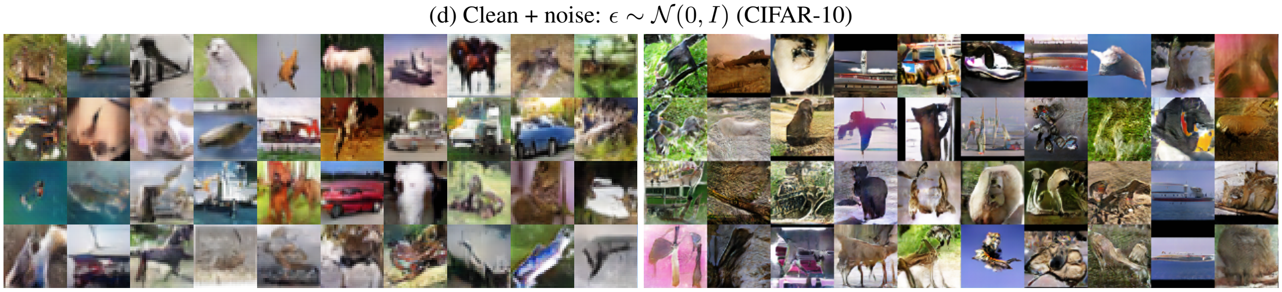
(b) Noise only: $\epsilon \sim \mathcal{N}(0, I)$



(c) Noise only: $\epsilon \sim \mathcal{N}(0, \sigma I)$, $\sigma \to 0$

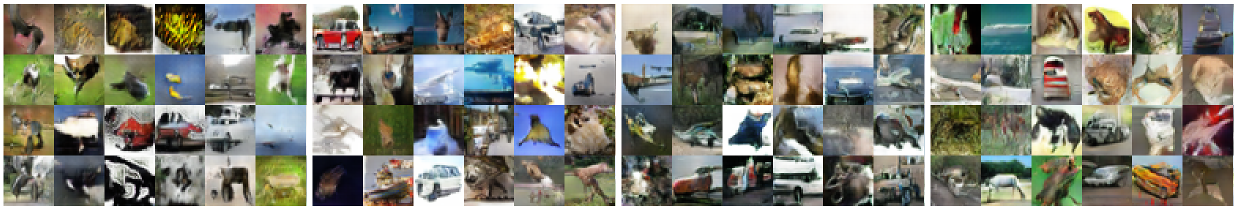


(d) Clean + noise: $\epsilon \sim \mathcal{N}(0, I)$ (CIFAR-10)

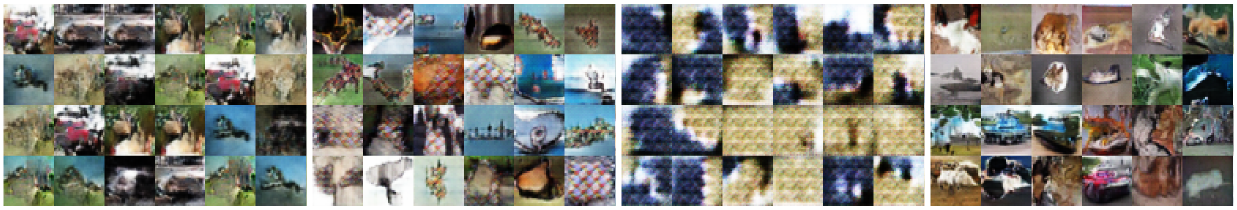


(e) DFGAN ($\lambda = 1$)

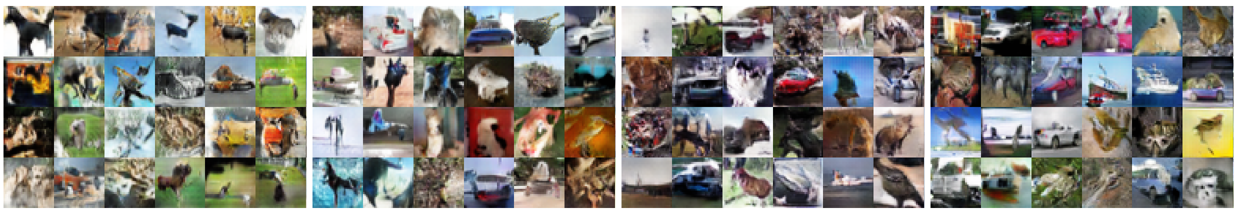
Figure 3: We show random reconstructions for some of the ablation experiments listed in Table 2 of the paper. The left column shows results on CIFAR-10 and the right column shows results on STL-10.



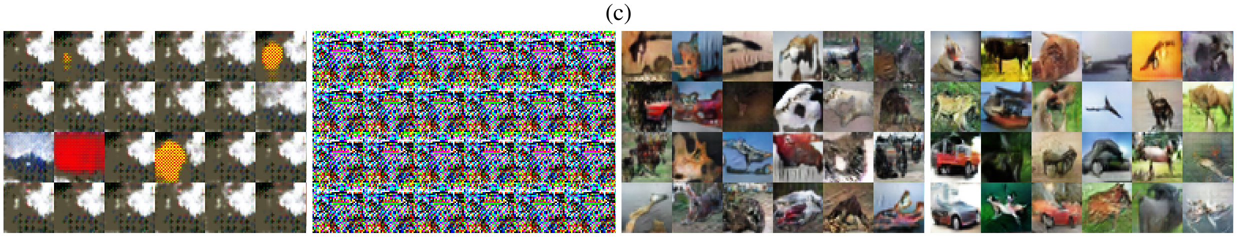
(a)



(b)



(c)



(d)

(e)

(f)

Figure 4: We show random reconstructions for the robustness experiments (see Table 4). We compare a standard GAN (*1st column*), a GAN with gradient penalty by [4] (*2nd column*), a GAN with spectral normalization by [3] (*3rd column*) and a GAN with our proposed method (*4th column*).

Figure 5: Results of progressive-GAN+DF trained on LSUN bedrooms. We used the DCGAN loss and Gaussian noise for DF in this experiment.