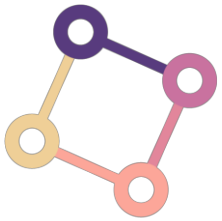
Training Generative Adversarial Networks with Limited Data

Arxiv 20.06.11

박성현



DAVIAN

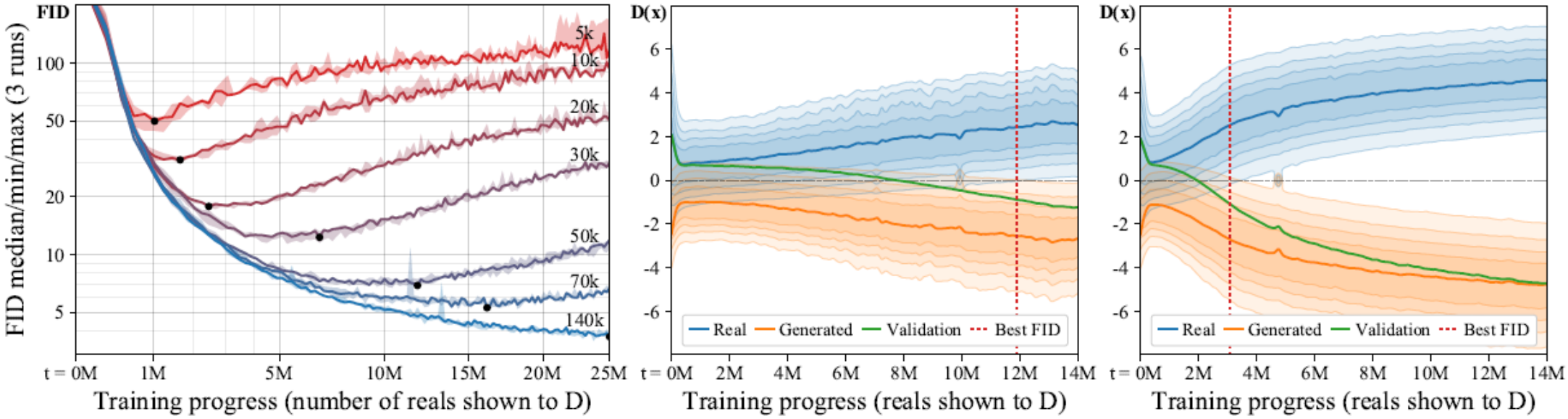
Data and Visual Analytics Lab

Motivation

- In order to train a high-quality, high-resolution GAN, we need to collect the custom dataset: acquiring, processing, and distributing the ~ $10^5 \sim 10^6$
- The key problem with small datasets is that the discriminator overfits to the training examples: its feedback to the generator becomes meaningless and training starts to diverge.
- Data augmentation is the standard solution against overfitting.
- However, a GAN trained under similar dataset augmentations learns to generate the augmented distribution.

Overfitting in GANs

[StyleGAN2 - 256x256 version of FFHQ dataset]



(a) Convergence of FFHQ (256 × 256) (b) Discriminator outputs, 50k (c) Discriminator outputs, 20k

Figure 1: (a) Convergence with different training set sizes. "140k" means that we amplified the 70k dataset by 2× through x-flips; we do not use data amplification in any other case. (b,c) Evolution of discriminator outputs during training. Each vertical slice shows a histogram of $D(x)$, i.e., raw logits.

Balanced Consistency Regularization

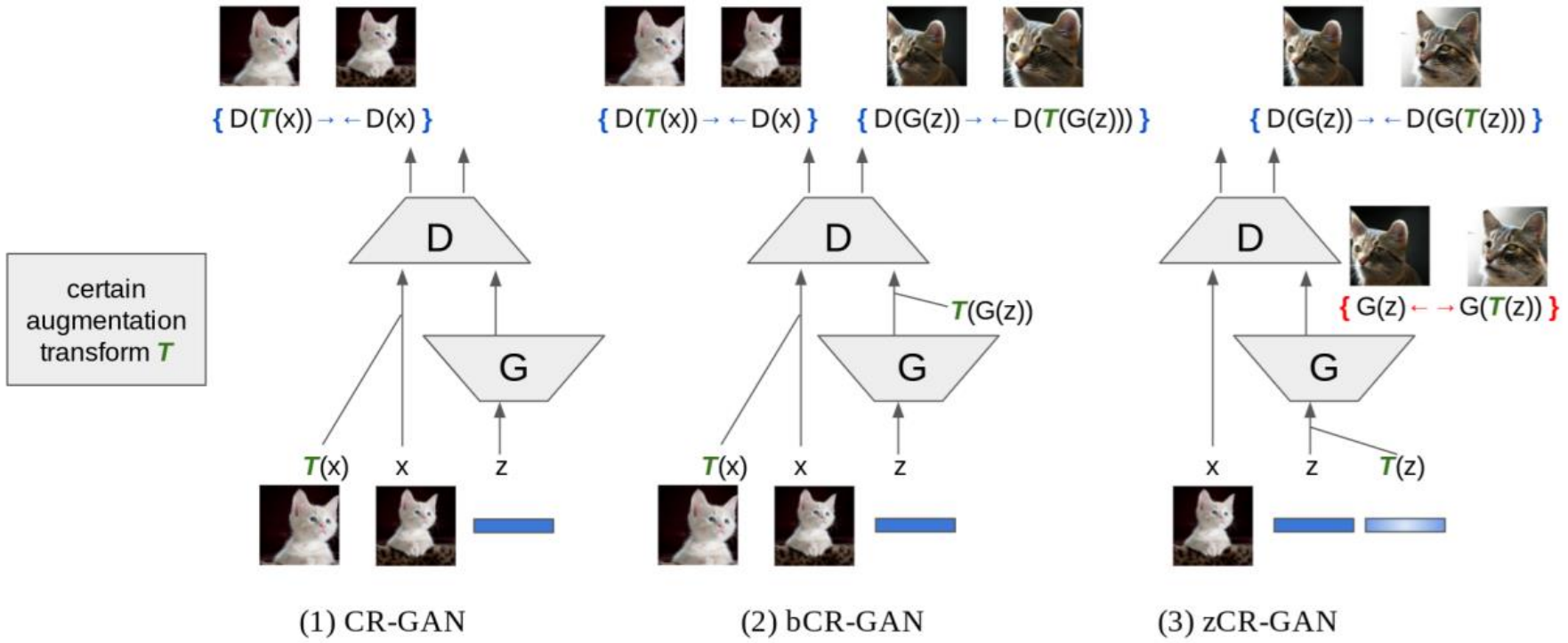


Figure 1. Illustrations comparing our methods to the baseline. (1) CR-GAN (Zhang et al., 2020) is the baseline, with consistency regularization applied only between real images and their augmentations. (2) In Balanced Consistency Regularization (bCR-GAN), we also introduce consistency regularization between generated fake images and their augmentations. With consistency regularization on both real and fake images, the discriminator is trained in a balanced way and less augmentation artifacts are generated. (3) Furthermore, we propose Latent Consistency Regularization (zCR-GAN), where latent z is augmented with noise of small magnitude. Then for the discriminator, we regularize the consistency between corresponding pairs; while for the generator we encourage the corresponding generated images to be more diverse. Note that $\{\rightarrow\leftarrow\}$ indicates a loss term encouraging pairs to be closer together, while $\{\leftarrow\rightarrow\}$ indicates a loss term pushing pairs apart.

Stochastic Discriminator Augmentation

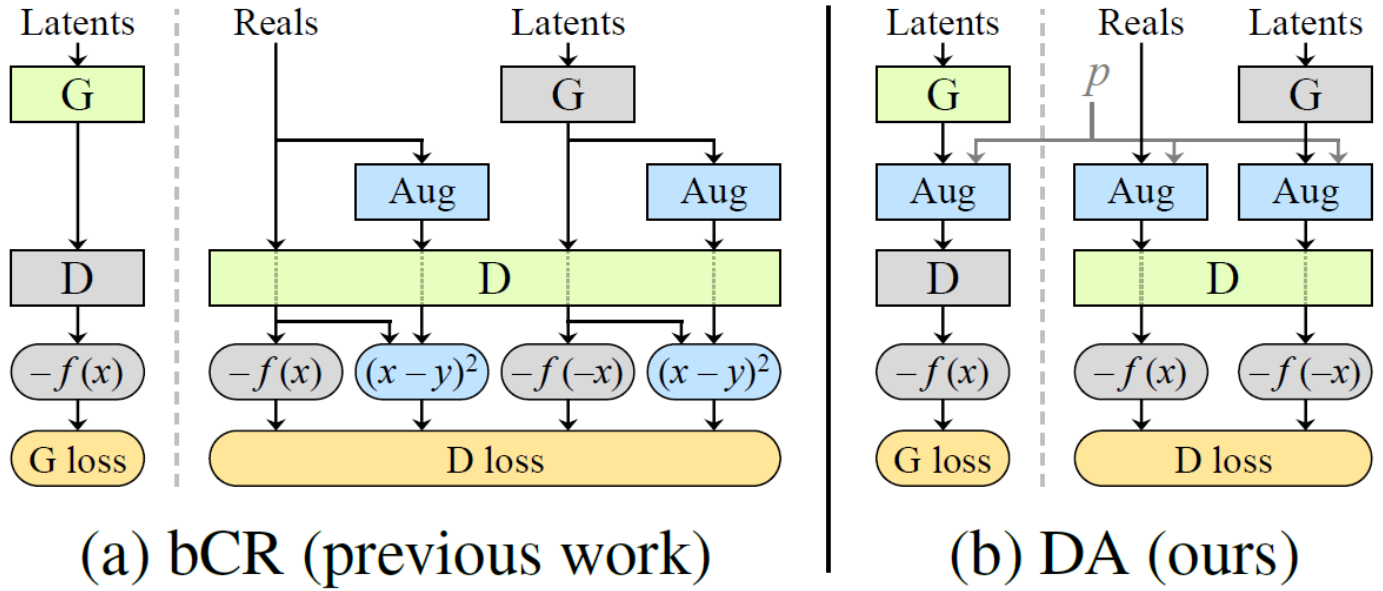
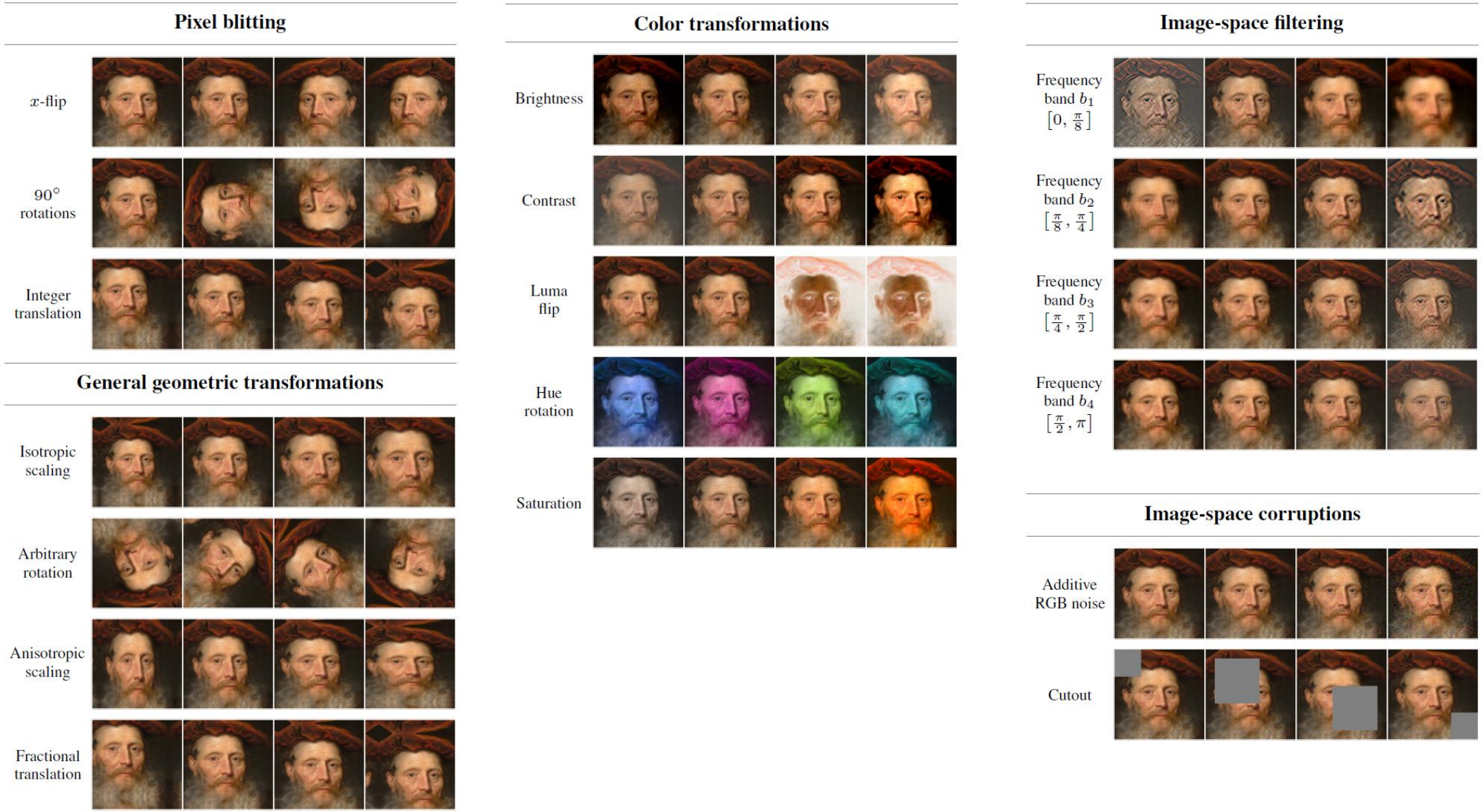
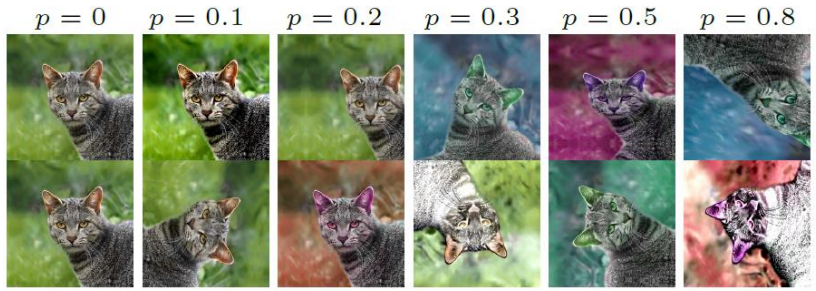


Figure 2: (a,b) Flowcharts for balanced consistency regularization (bCR) [49] and our stochastic discriminator augmentations (DA). The blue elements highlight operations related to augmentations, while the rest implement standard GAN training with generator G and discriminator D [13]. We use the non-saturating logistic loss [13] $f(x) = \log(\text{sigmoid}(x))$.

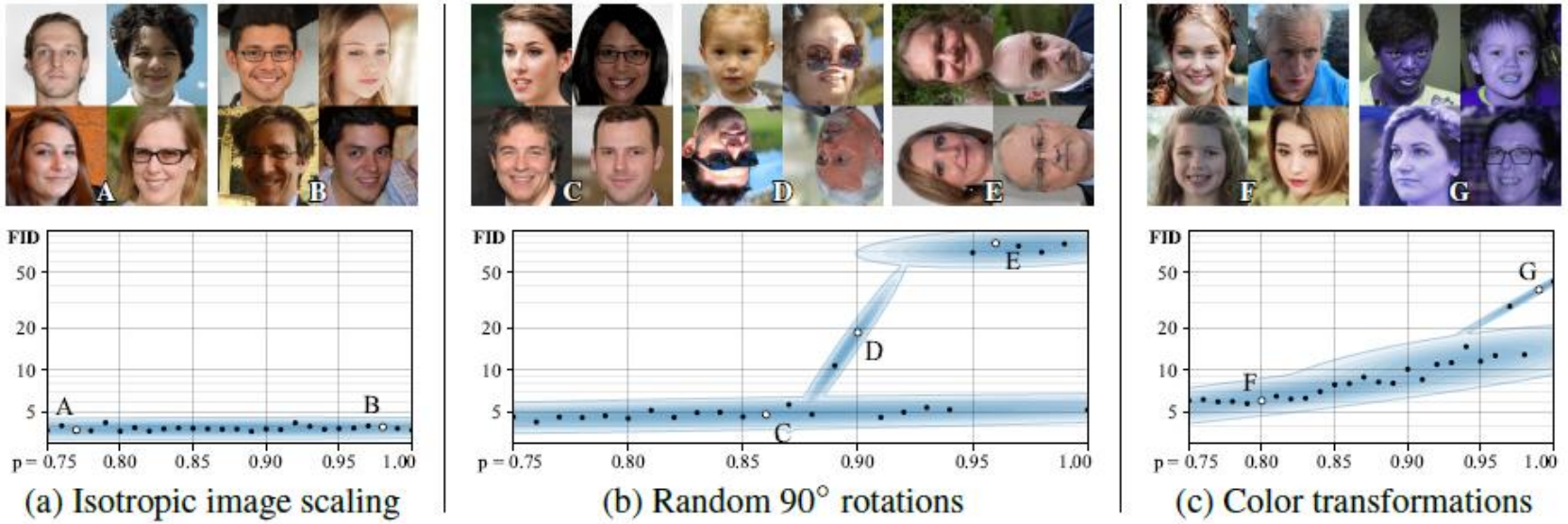
Data Augmentations



Designing augmentations



(c) Effect of augmentation probability p



(a) Isotropic image scaling (b) Random 90° rotations (c) Color transformations

Figure 3: Leaking behavior of three example augmentations, shown as FID w.r.t. the probability of executing the augmentation. Each dot represents a complete training run, and the blue Gaussian mixture is a visualization aid. The top row shows generated example images from selected training runs, indicated by uppercase letters in the plots.

Impact for p for different augmentation

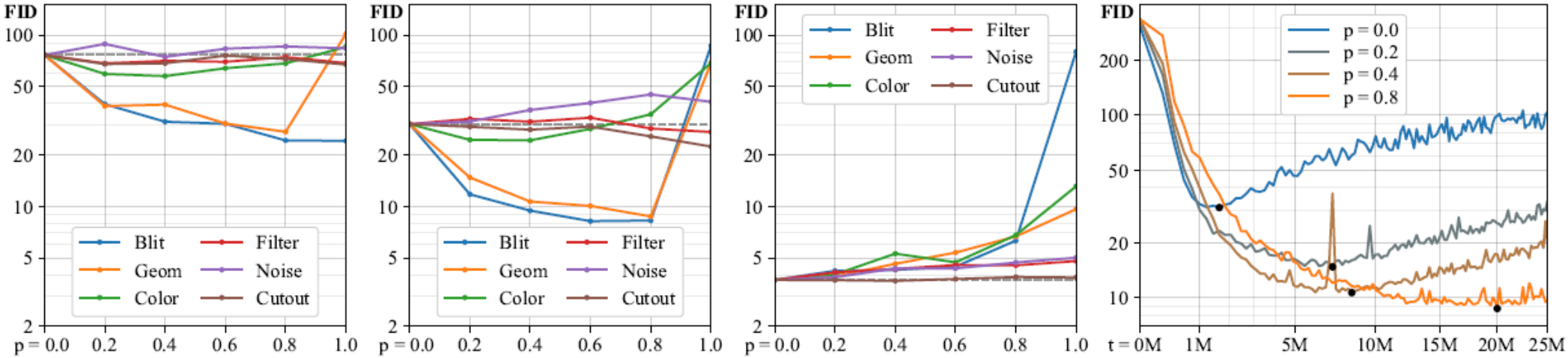


Figure 4: (a-c) Impact of p for different augmentation categories and dataset sizes. The dashed gray line indicates baseline FID without augmentations. (d) Convergence curves for selected values of p using geometric augmentations with 10k training images.

Impact for p for different augmentation

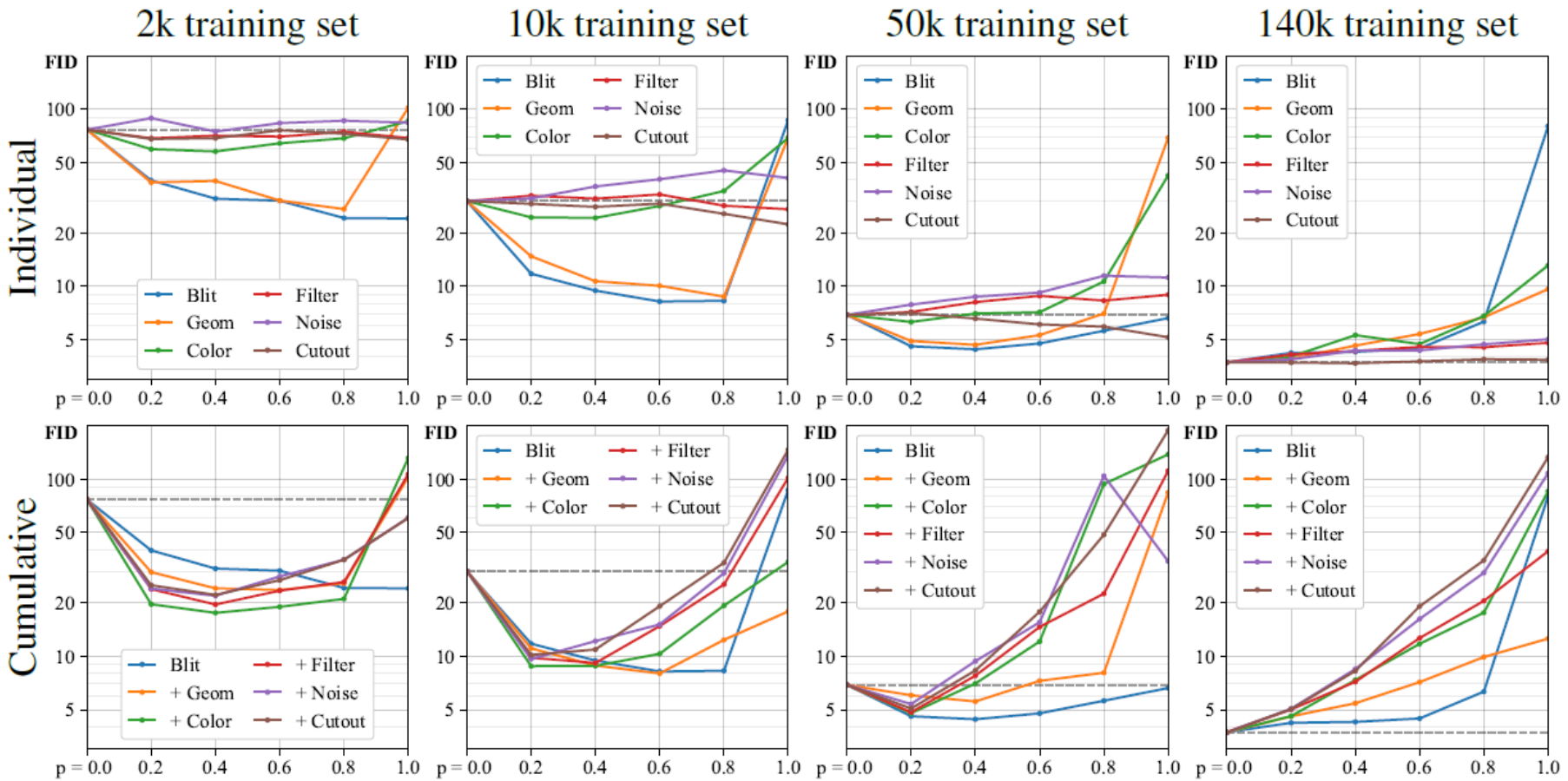


Figure 19: Extended version of Figure 4, illustrating the individual and cumulative effect of different augmentation categories with increasing augmentation probability p.

Adaptive Discriminator Augmentation

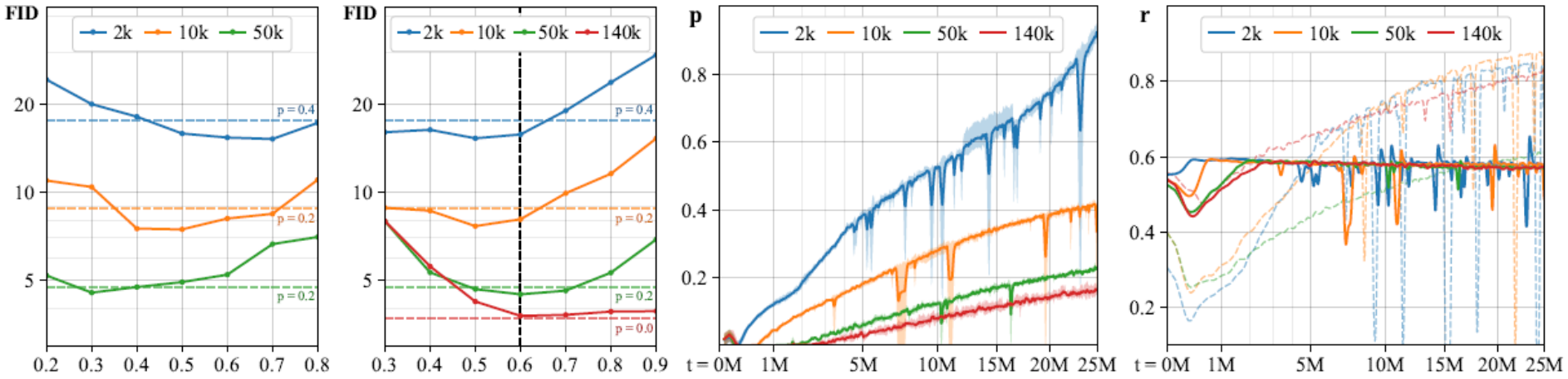
- Ideally, we would like to avoid manual tuning of the augmentation strength and instead control it dynamically based on the degree of overfitting.
- The authors introduced two overfitting heuristics:

$$r_v = \frac{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{validation}}]}{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{generated}}]} \qquad r_t = \mathbb{E}[\text{sign}(D_{\text{train}})]$$

- For both heuristics, $r = 0$ means no overfitting and $r = 1$ indicates complete overfitting.
- Goal is to adjust the augmentation probability p so that the chosen heuristic matches a suitable target value.
- r_v expresses the output for a validation set relative to the training set and generated images.
- r_t estimates the portion of the training set that gets positive discriminator outputs.

Adaptive Discriminator Augmentation

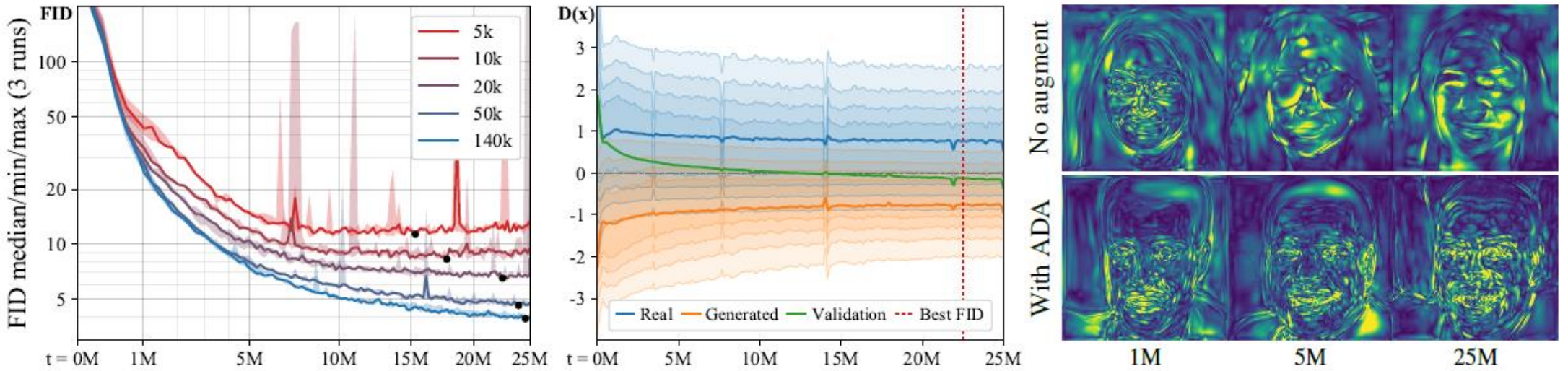
- We initialize p to zero and adjust its value once every four minibatches based on the overfitting heuristic. If the heuristic indicates too much/little overfitting, we counter by incrementing/decrementing p by a fixed amount.



(a) r_v target sweep (b) r_t target sweep (c) Evolution of p over training (d) Evolution of r_t

Figure 5: Behavior of our adaptive augmentation strength heuristics in FFHQ. (a,b) FID for different training set sizes as a function of the target value for r_v and r_t. The dashed horizontal lines indicate the best fixed augmentation probability p found using grid search, and the dashed vertical line marks the target value we will use in subsequent tests. (c) Evolution of p over the course of training using heuristic r_t. (d) Evolution of r_t values over training. Dashes correspond to the fixed p values in (b).

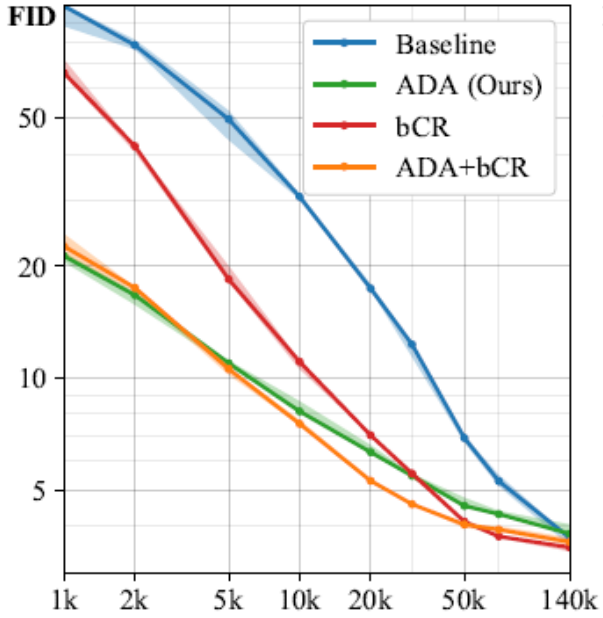
Adaptive Discriminator Augmentation



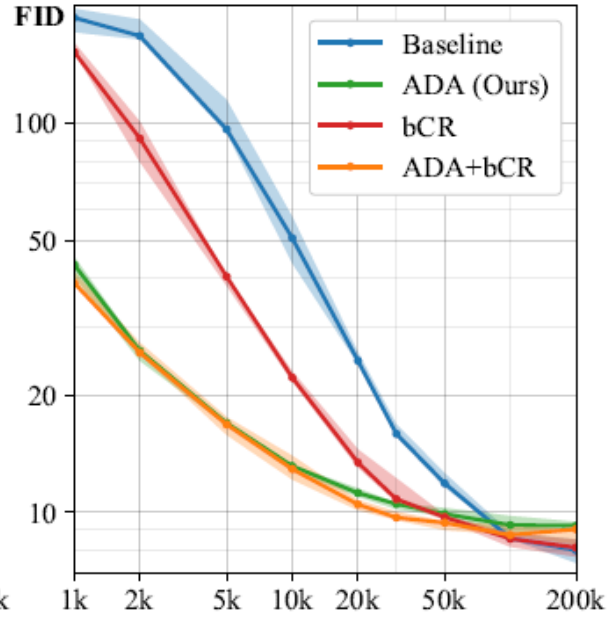
(a) With adaptive augmentation (b) Discriminator outputs, 20k (c) Discriminator gradients, 10k

Figure 6: (a) Training curves for FFHQ with different training set sizes using adaptive augmentation. (b) The supports of real and generated images continue to overlap. (c) Example magnitudes of the gradients the generator receives from the discriminator as the training progresses.

Experiments: Training from Scratch



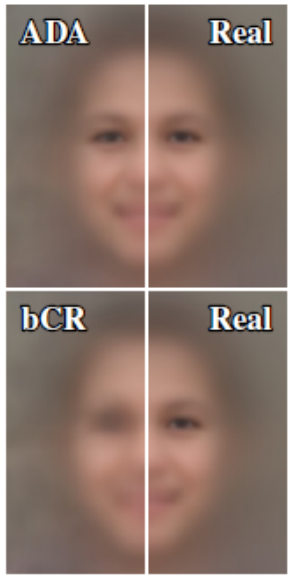
(a) FFHQ (256 × 256)



(b) LSUN CAT (256 × 256)

Dataset		Baseline	ADA	+ bCR
FFHQ	1k	100.16	**21.29**	22.61
	5k	49.68	10.96	**10.58**
	10k	30.74	8.13	**7.53**
	30k	12.31	5.46	**4.57**
	70k	5.28	4.30	**3.91**
	140k	3.71	3.81	**3.62**
LSUN CAT	1k	186.91	43.25	**38.82**
	5k	96.44	16.95	**16.80**
	10k	50.66	13.13	**12.90**
	30k	15.90	10.50	**9.68**
	100k	**8.56**	9.26	8.73
	200k	**7.98**	9.22	9.03

(c) Median FID



(d) Mean image

Figure 7: (a-c) FID as a function of training set size, reported as median/min/max over 3 training runs. (d) Average of 10k random images generated using the networks trained with 5k subset of FFHQ. ADA matches the average of real data, whereas the xy-translation augmentation in bCR [49] has leaked to the generated images, significantly blurring the average image.

Experiments: Training from Scratch

FFHQ (256 × 256)	**2k**	**10k**	**140k**
Baseline	78.58	30.74	3.71
PA-GAN [45]	54.00	28.81	3.81
WGAN-GP [14]	78.12	35.26	6.47
zCR [49]	66.39	23.08	**3.52**
Aux. rotation [6]	67.57	25.20	4.16
Spectral norm [29]	89.83	38.21	4.59
Shallow mapping	72.00	26.64	3.60
Adaptive dropout	65.98	23.43	4.18
ADA (Ours)	**16.71**	**8.13**	3.81

(a) Comparison methods

Dataset	**Scratch** FID	**Scratch** KID $\times 10^3$	**Transfer** FID	**Transfer** KID $\times 10^3$
METFACES	57.26	35.66	19.47	3.16
+ Freeze-D	–	–	17.06	2.05
ADA (Ours)	**18.22**	**2.41**	**15.34**	**0.81**
BRECAHAD	97.72	89.76	30.58	18.07
+ Freeze-D	–	–	19.38	6.94
ADA (Ours)	**15.71**	**2.88**	**16.33**	**3.36**
AFHQ DOG	19.37	9.62	11.64	4.63
+ Freeze-D	–	–	9.02	2.80
ADA (Ours)	**7.40**	**1.16**	**7.65**	**1.40**

(b) Results for other datasets

Method	**Uncond.** FID ↓	**Uncond.** IS ↑	**Cond.** FID ↓	**Cond.** IS ↑
ProGAN [18]	15.52	8.56	–	–
AutoGAN [12]	12.42	8.55	–	–
BigGAN [5]	–	–	14.73	9.22
+ Tuning [21]	–	–	8.47	9.07
MultiHinge [21]	–	–	6.40	9.58
FQ-GAN [48]	–	–	5.59	8.48
Baseline	9.90	8.99	9.37	9.16
+ ADA (Ours)	4.65	**9.92**	3.31	**10.14**
+ Tuning	**3.26**	9.74	**2.67**	10.06

(c) CIFAR-10

Figure 8: (a) Comparison methods; median of 3 runs. (b) Other datasets trained with StyleGAN2 config F, StyleGAN2 + Freeze-D, and ADA. We report the best KID, and compute FID using the same snapshot. (c) CIFAR-10 results reported as averages of the best scores over 5 training runs. For the comparison methods we report the average scores when available, and the single best score otherwise. The best IS and FID were searched separately [21], and often came from different snapshots.

Experiments: Transfer Learning

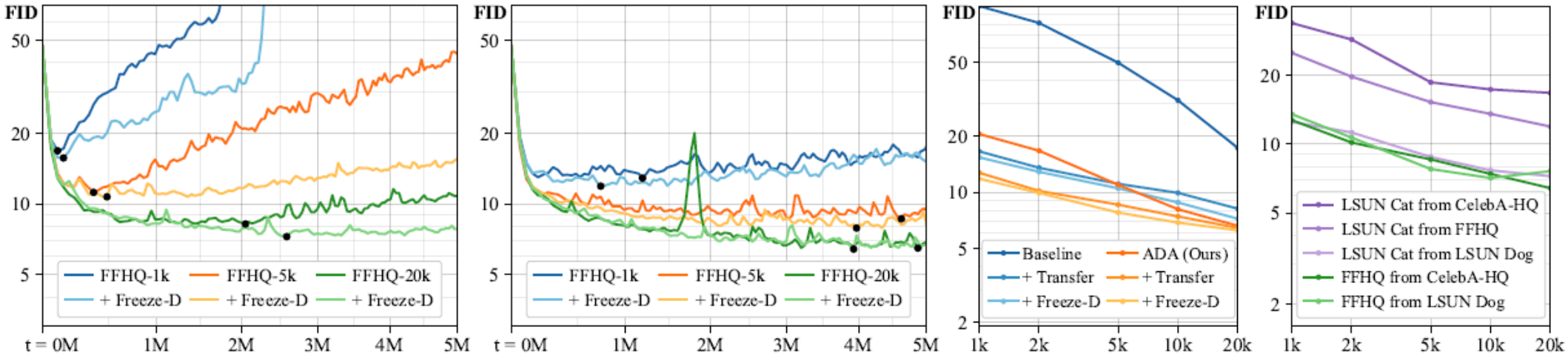


Figure 9: Transfer learning FFHQ starting from a pre-trained CELEBA-HQ model, both 256×256. (a) Training convergence for our baseline method and Freeze-D [31]. (b) The same configurations with ADA. (c) FIDs as a function of dataset size. (d) Effect of source and target datasets.

Experiments: Generated Images

Figure 11: Uncurated 1024×1024 results generated for METFACES (1336 images) with and without ADA, along with real images from the training set. Both generators were trained using transfer learning, starting from the pre-trained StyleGAN2 for FFHQ. We recommend zooming in.

Experiments: Generated Images

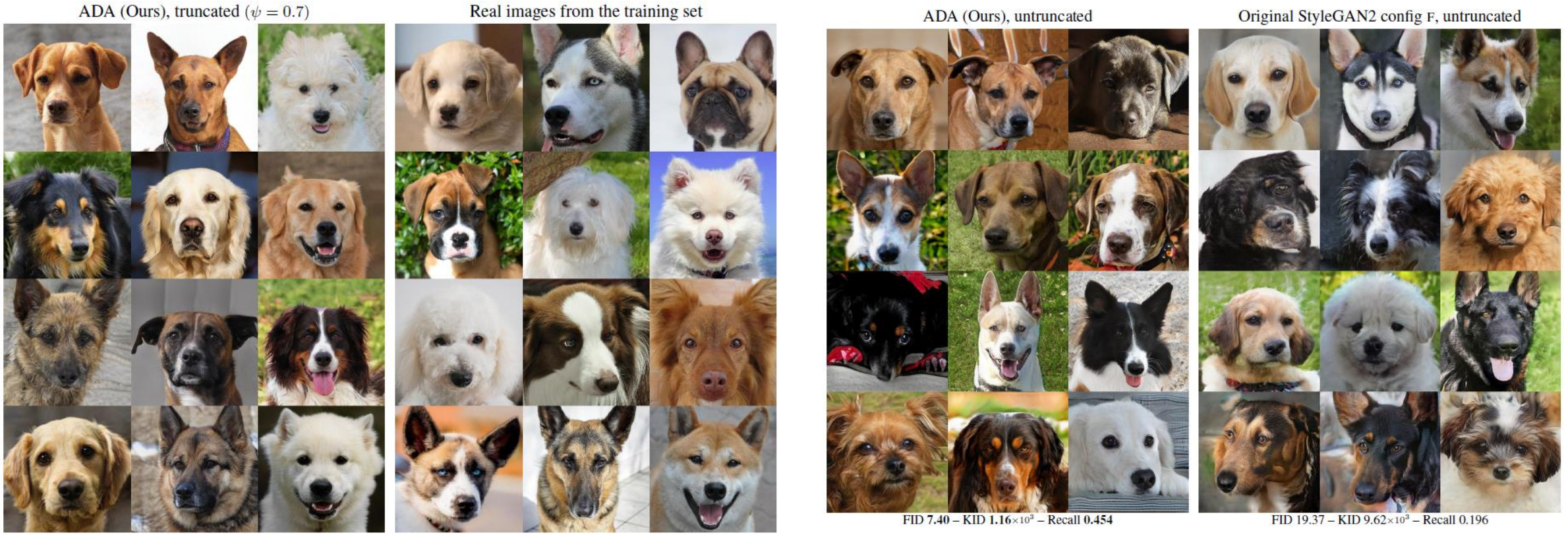


Figure 12: Uncurated 512×512 results generated for AFHQ Dog [7] (4739 images) with and without ADA, along with real images from the training set. Both generators were trained from scratch. We recommend zooming in to inspect the image quality in detail.

Experiments: Generated Images

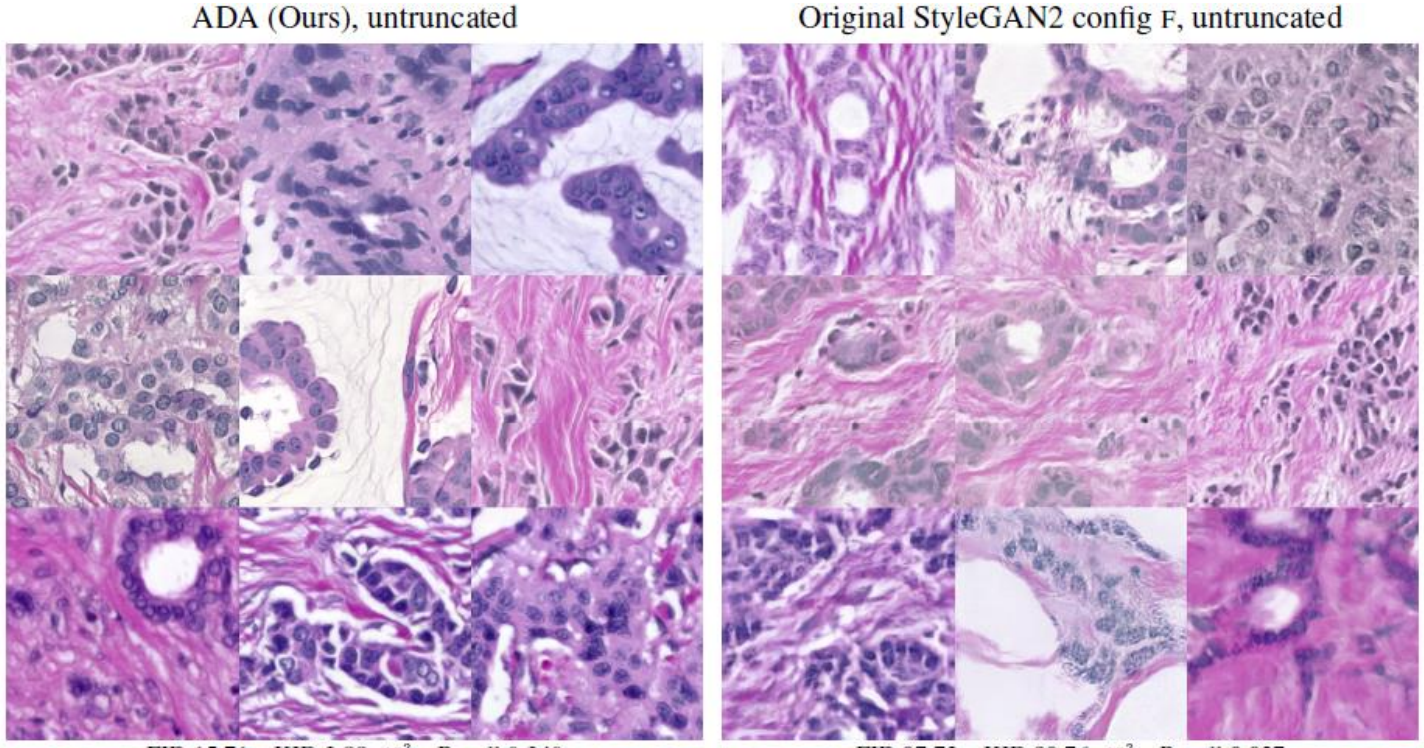
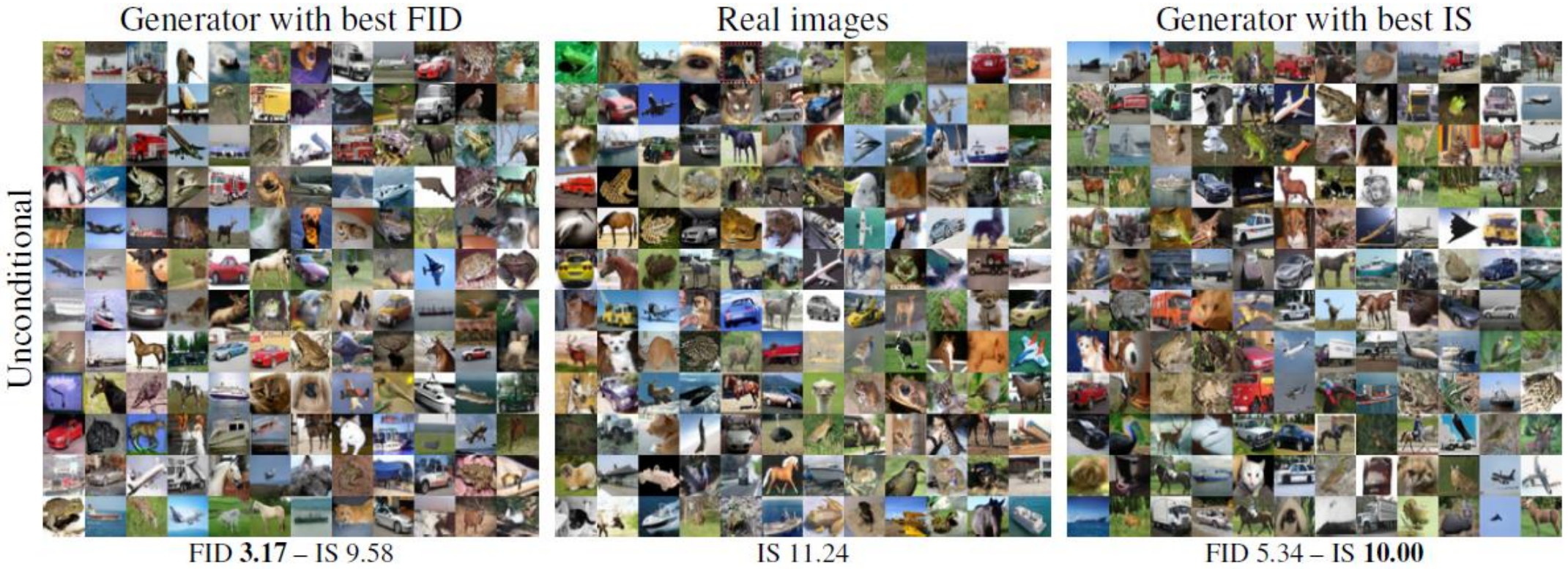


Figure 13: Uncurated 512×512 results generated for BRECAHAD [1] (1944 images) with and without ADA, along with real images from the training set. Both generators were trained from scratch. We recommend zooming in to inspect the image quality in detail.

Experiments: Generated Images



Thank you!