個人報告3

應名宥

outline

- Analyzing and Improving the Image Quality of StyleGAN (Style-GAN-2)
- Training Generative Adversarial Networks with Limited Data (Style-GAN-2-ada)
- Improving Text-to-Image Synthesis Using Contrastive Learning
- Training experiment

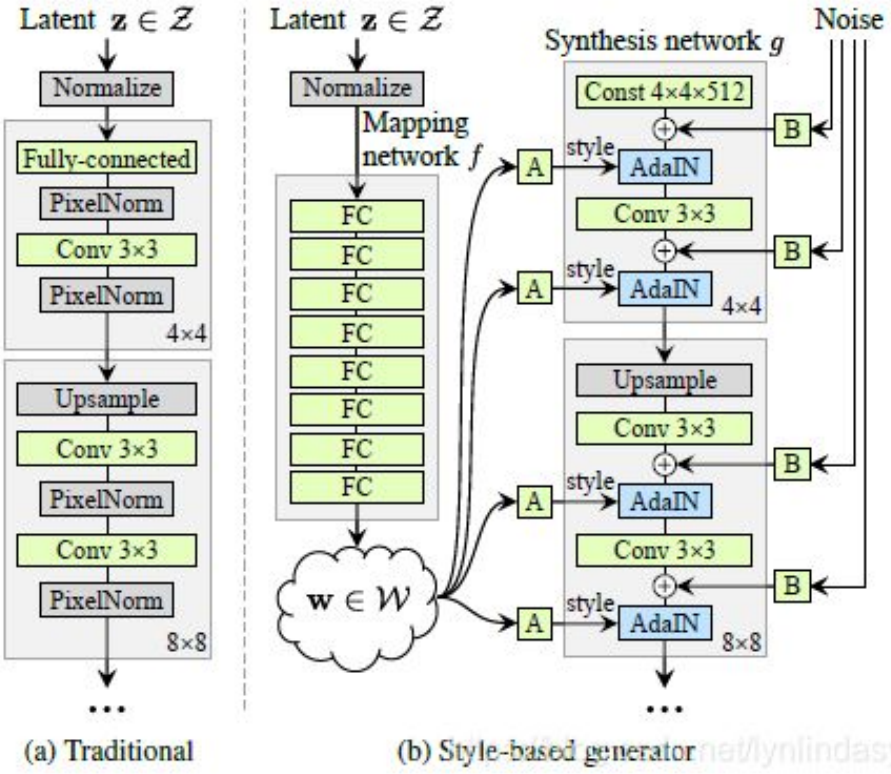
Style-GAN-2

Analyzing and Improving the Image Quality of StyleGAN

StyleGAN structure

- Instead of feeding the input latent code $z \in Z$ only to the beginning of a the network, the mapping network f first transforms it to an intermediate latent code $w \in W$
- Affine transforms then produce styles that control the layers of the synthesis network G via adaptive instance normalization
- intermediate latent space W to be much less entangled (更好被理解) than the input latent space Z

StyleGAN structure



(a) Traditional

(b) Style-based generator

problem

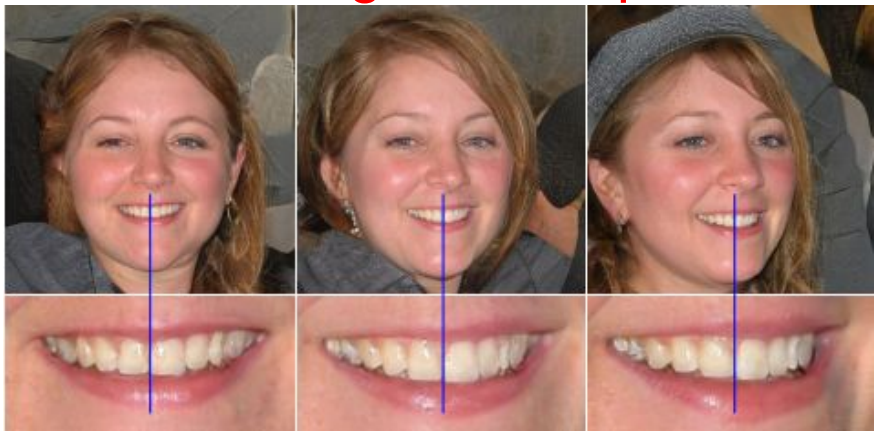
- common droplet-like artifacts
- generator creates them to circumvent a design flaw in its architecture

- analyze artifacts related to progressive growing that has been highly successful in stabilizing high-resolution GAN training
- propose an alternative design that achieves the same goal. training starts by focusing on low-resolution images and then progressively shifts focus to higher and higher resolutions

droplet-like artifacts

progressive growing

- Progressive growing has been very successful in stabilizing high-resolution image synthesis, but it causes its own characteristic artifacts.
- The key issue is that the progressively grown generator appears to have a strong location preference for details.



Removing normalization artifacts

- starts to appear around 64×64 resolution, is present in all feature maps, and becomes progressively stronger at higher resolutions.
- the discriminator should be able to detect it.

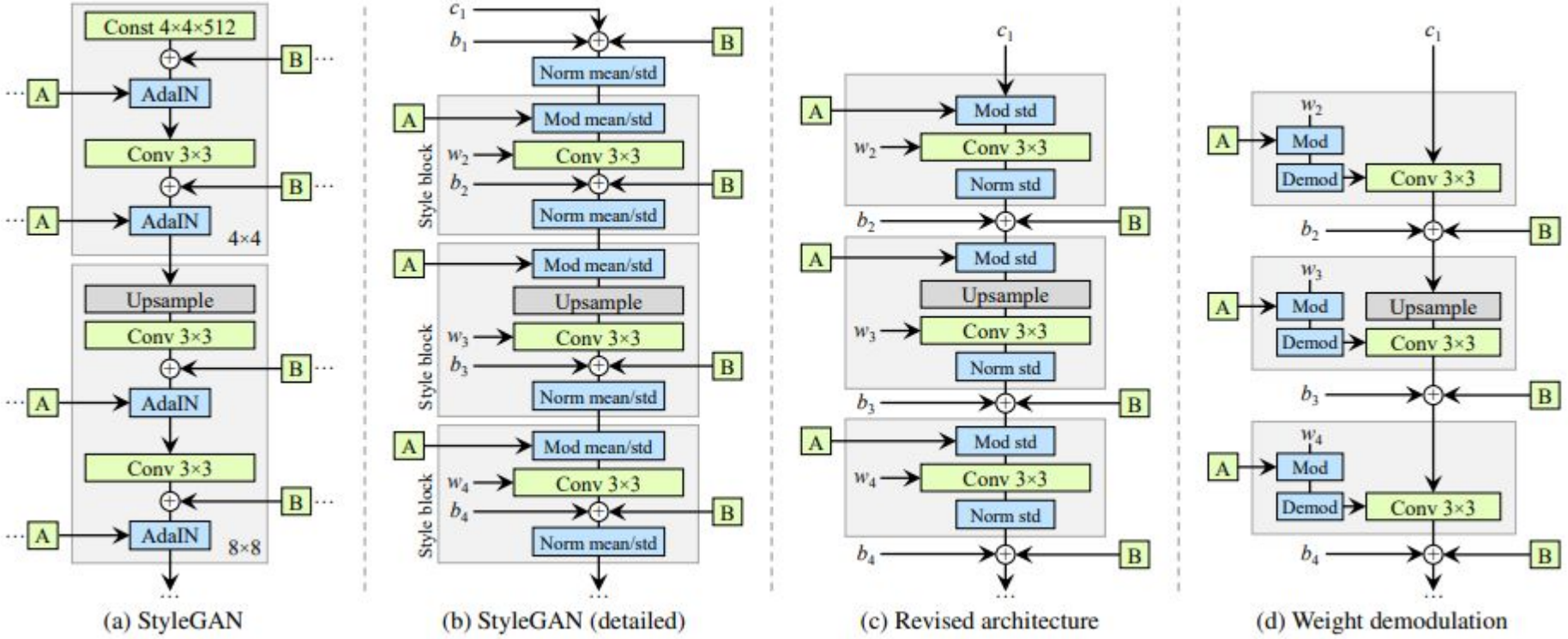
normalization artifacts cont.

- the problem to the AdaIN operation that normalizes the mean and variance of each feature map separately, thereby potentially destroying any information found in the magnitudes of the features relative to each other.
- generator intentionally sneaking signal strength information past instance normalization.
- when the normalization step is removed from the generator, as detailed below, the droplet artifacts disappear completely.

Generator architecture

- sufficient for the normalization and modulation to operate on the standard deviation alone. (the mean is not needed)

Generator architecture revisited

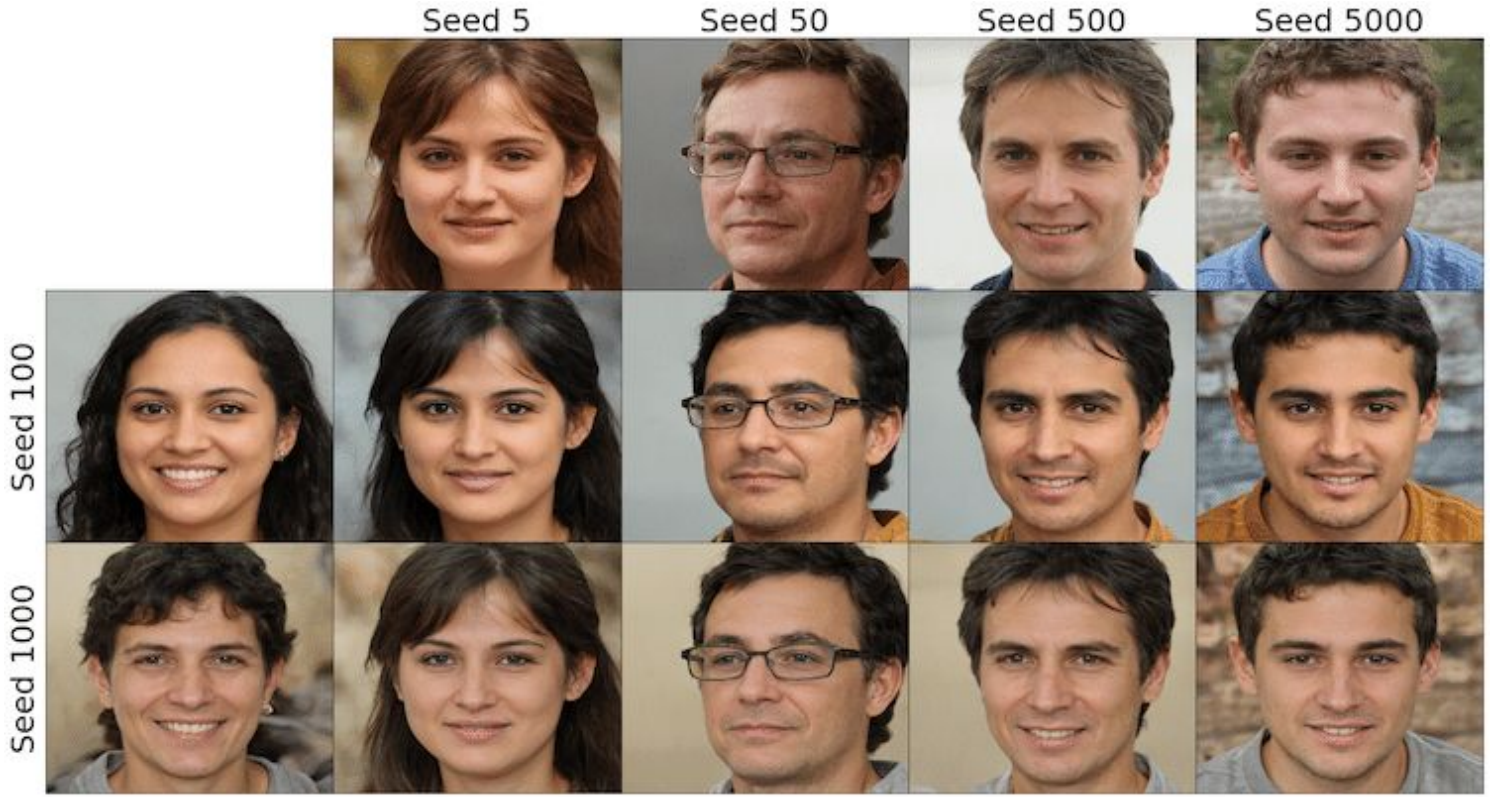


(a) StyleGAN (b) StyleGAN (detailed) (c) Revised architecture (d) Weight demodulation

style-mixing

- One of the main strengths of StyleGAN is the ability to control the generated images via style mixing, i.e., by feeding a different latent w to different layers at inference time. In practice, style modulation may amplify certain feature maps by an order of magnitude or more. For style mixing to work, we must explicitly counteract this amplification on a per-sample basis — otherwise the subsequent layers would not be able to operate on the data in a meaningful way

style-mixing example



modulation

- The main idea is to base normalization on the expected statistics of the incoming feature maps, but without explicit forcing.
- The modulation scales each input feature map of the convolution based on the incoming style, which can alternatively be implemented by scaling the convolution weights

$$w'_{ijk} = s_i \cdot w_{ijk},$$

demodulation

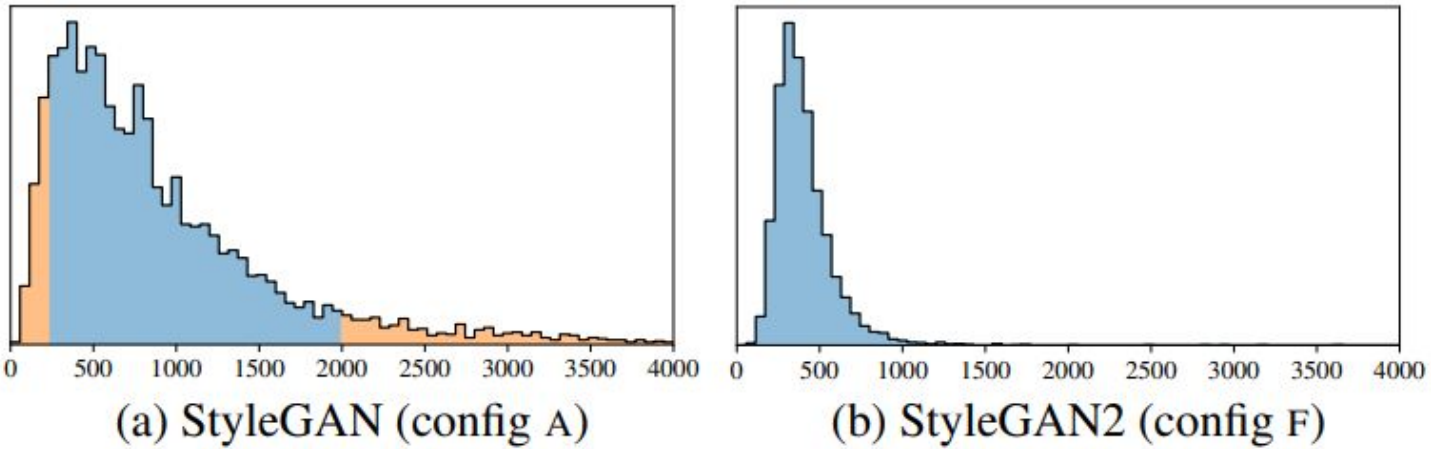
- The subsequent normalization aims to restore the outputs back to unit standard deviation. Based on Equation 2, this is achieved if we scale (“demodulate”) each output feature map.

$$\sigma_j = \sqrt{\sum_{i,k} {w'_{ijk}}^2},$$

$$w''_{ijk} = w'_{ijk} \Big/ \sqrt{\sum_{i,k} {w'_{ijk}}^2 + \epsilon},$$

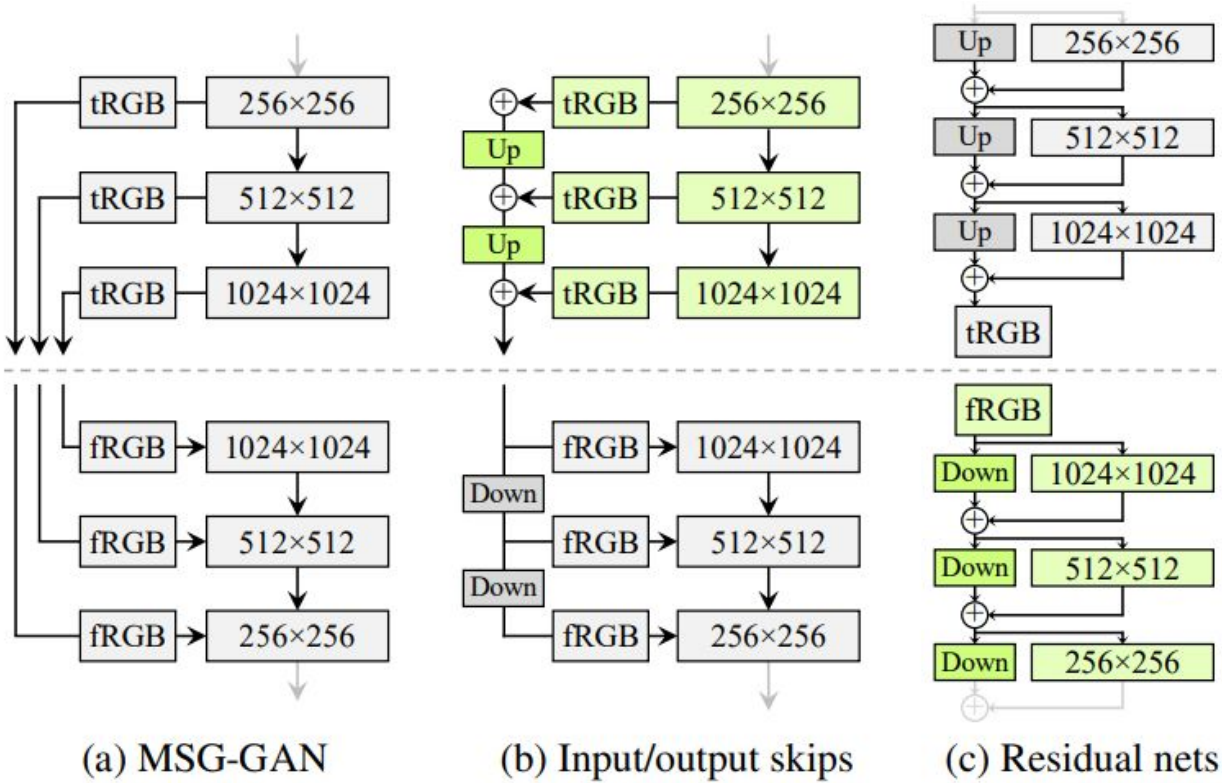
perceptual path length

- achieve a clear improvement in quality.
- executing all regularizations less frequently.
- quantifying the smoothness of the mapping from a latent space to the output image.



(a) StyleGAN (config A) (b) StyleGAN2 (config F)

Alternative network architectures



Alternative network architectures

- we use a skip generator and a residual discriminator, without progressive growing.

FFHQ	D original		D input skips		D residual	
	FID	PPL	FID	PPL	FID	PPL
G original	4.32	265	4.18	235	3.58	269
G output skips	4.33	169	3.77	127	**3.31**	**125**
G residual	4.35	203	3.96	229	3.79	243

LSUN Car	D original		D input skips		D residual	
	FID	PPL	FID	PPL	FID	PPL
G original	3.75	905	3.23	758	3.25	802
G output skips	3.77	544	3.86	**316**	3.19	471
G residual	3.93	981	3.40	667	**2.66**	645

Style-GAN-2-ada

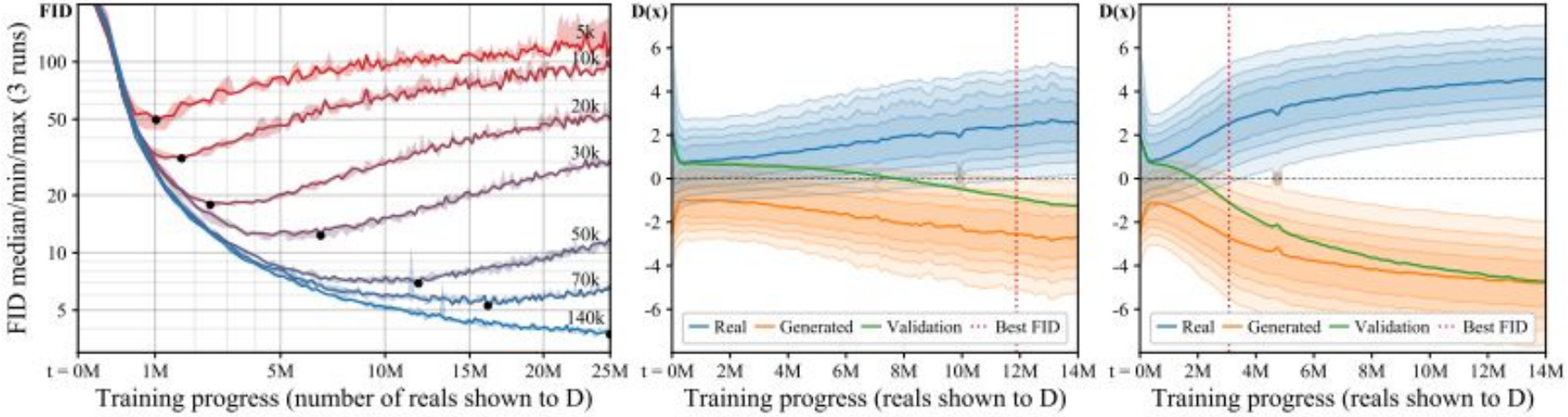
Adaptive discriminator augmentation

Introduction

- generative adversarial networks (GAN) are fueled by the seemingly unlimited supply of images available online.
- custom dataset: acquiring, processing, and distributing the ~ 10^5 − 10^6 images required to train a modern high-quality, high-resolution GAN is a costly undertaking.
- A significant reduction in the number of images required therefore has the potential to considerably help many applications.

key problem with small datasets

- discriminator overfits to the training examples, feedback to the generator becomes meaningless and training starts to diverge.
- dataset augmentation is the standard solution against overfitting.
- a GAN trained under similar dataset augmentations learns to generate the augmented distribution (noise augmentation leads to noisy results)

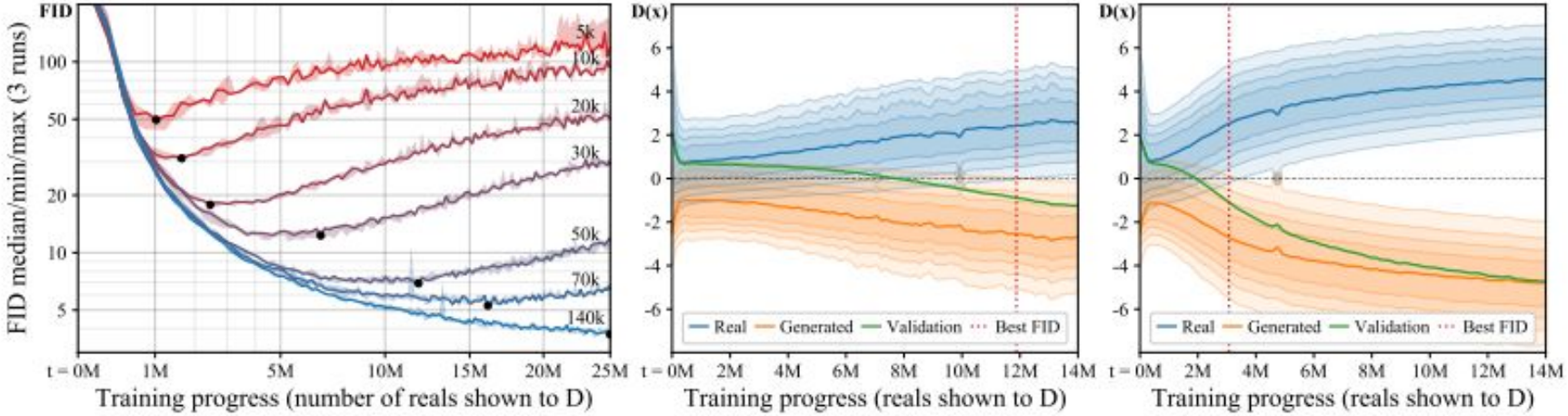


(a) Convergence of FFHQ (256 × 256) (b) Discriminator outputs, 50k (c) Discriminator outputs, 20k

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.

We demonstrate, on several datasets, that good results are now possible using only a few thousand images

Training starts the same way in each case, but eventually the progress stops and FID starts to rise, The less training data there is, the earlier this happens



(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.

The distributions overlap initially but keep drifting apart as the discriminator becomes more and more confident, and the point where FID starts to deteriorate is consistent with the loss of sufficient overlap between distributions.

Stochastic discriminator augmentation

- By definition, any augmentation that is applied to the training dataset will get inherited to the generated images.
- recently proposed balanced consistency regularization (bCR) as a solution that is not supposed to leak augmentations to the generated images. Consistency regularization states that two sets of augmentations, applied to the same input image, should yield the same output.

how to do

- add consistency regularization terms for the discriminator loss, and enforce discriminator consistency for both real and generated images, whereas no augmentations or consistency loss terms are applied when training the generator.

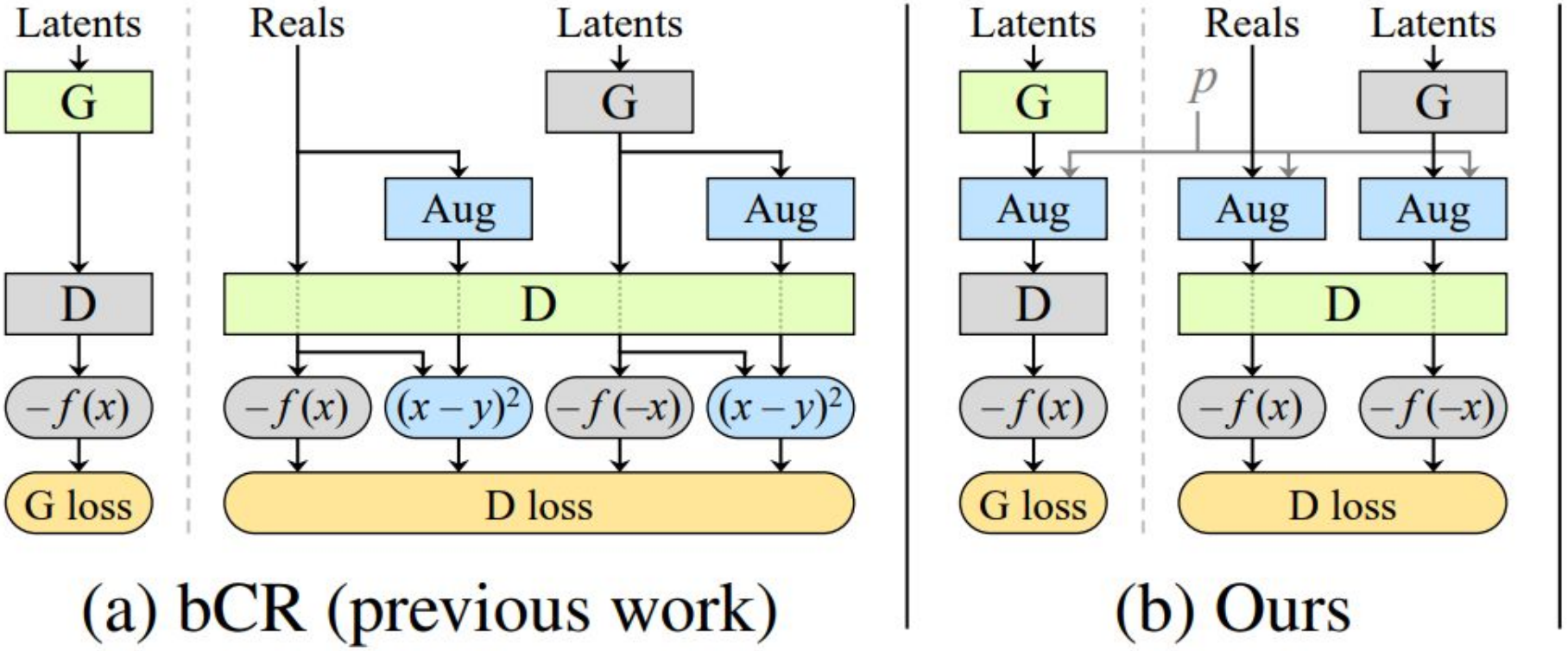
problem

- As such, their approach effectively strives to generalize the discriminator by making it blind to the augmentations used in the CR term. However, meeting this goal opens the door for leaking augmentations, because the generator will be free to produce images containing them without any penalty

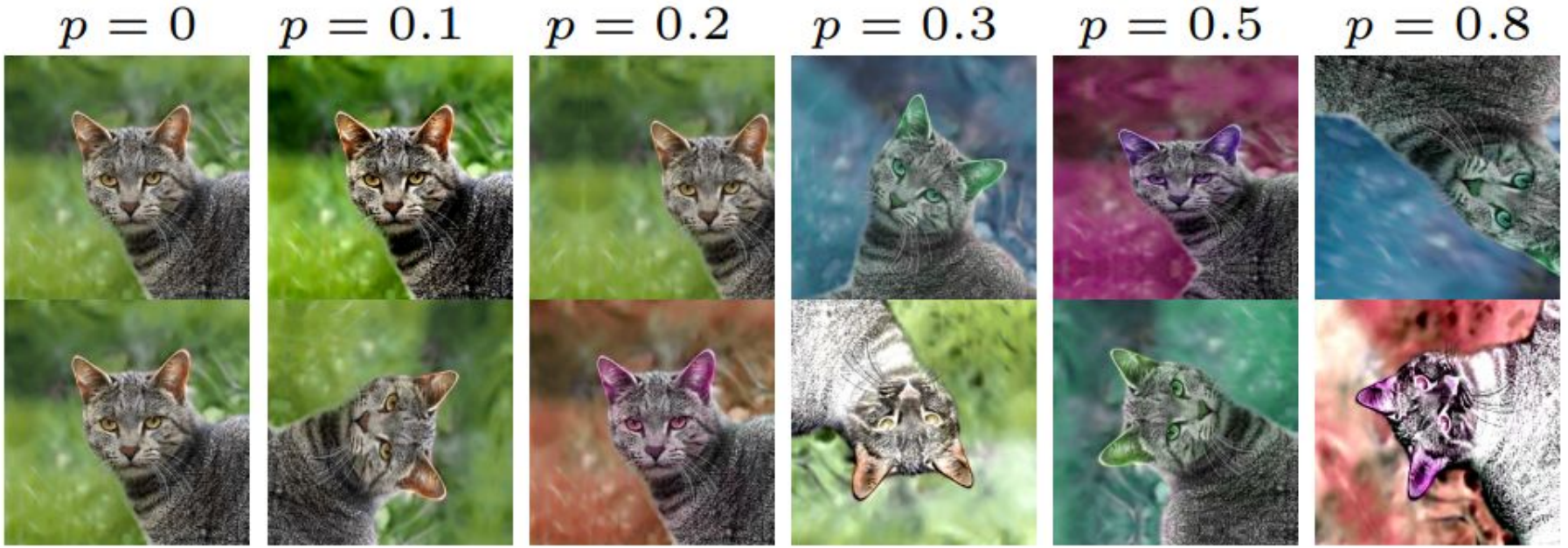
solution

- apply a set of augmentations to all images shown to the discriminator
- instead of adding separate CR loss terms, we evaluate the discriminator only using augmented images, and do this also when training the generator
- This approach that we call *stochastic discriminator augmentation* is therefore very straightforward.

structure


Latents
G
D
$-f(x)$
G loss
Reals
Latents
G
Aug
Aug
D
$-f(x)$
$(x-y)^2$
$-f(-x)$
$(x-y)^2$
D loss
(a) bCR (previous work)
Latents
Reals
Latents
p
G
G
Aug
Aug
Aug
D
D
$-f(x)$
$-f(x)$
$-f(-x)$
G loss
D loss
(b) Ours

augmentation probability



(c) Effect of augmentation probability p

but … ?

- if the discriminator never sees what the training images really look like, it is not clear if it can guide the generator properly.

Designing augmentations that do not leak

- Discriminator augmentation corresponds to putting distorting, perhaps even destructive goggles on the discriminator, and asking the generator to produce samples that cannot be distinguished from the training set when viewed through the goggles
- the training implicitly undoes the corruptions and finds the correct distribution, as long as the corruption process is represented by an invertible transformation of probability distributions over the data space.
- We call such augmentation operators *non-leaking*.

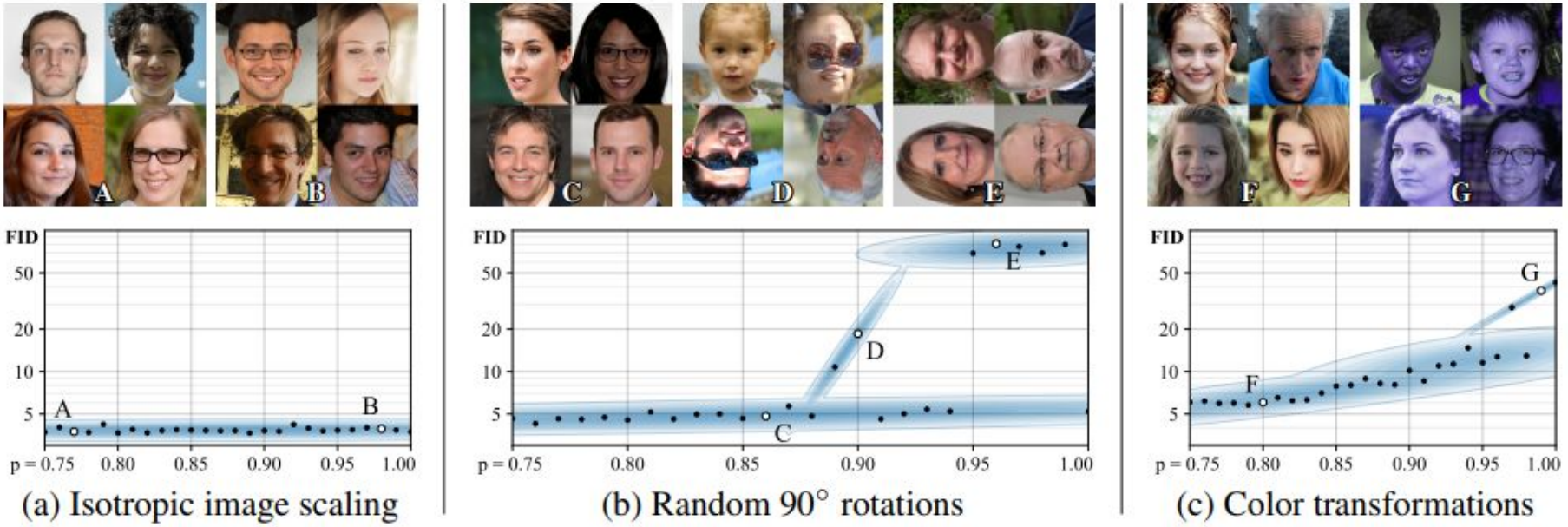
operator examples

- set to zero is non-leaking
  - setting the input image to zero 90% of the time is invertible in the probability distribution sense
- random rotation is leaking
  - impossible to discern differences among the orientations after the augmentation
- composing non-leaking augmentations in a fixed order yields an overall non-leaking augmentation.

augmentation probability setting

- When p is too high, the generator cannot know which way the generated images should face and ends up picking one of the possibilities at random.
- When p remains below ~0.85, the generated images are always oriented correctly.
- in the stylegan2-ada official implementation, p = 0.6

augmentation probability setting cont.

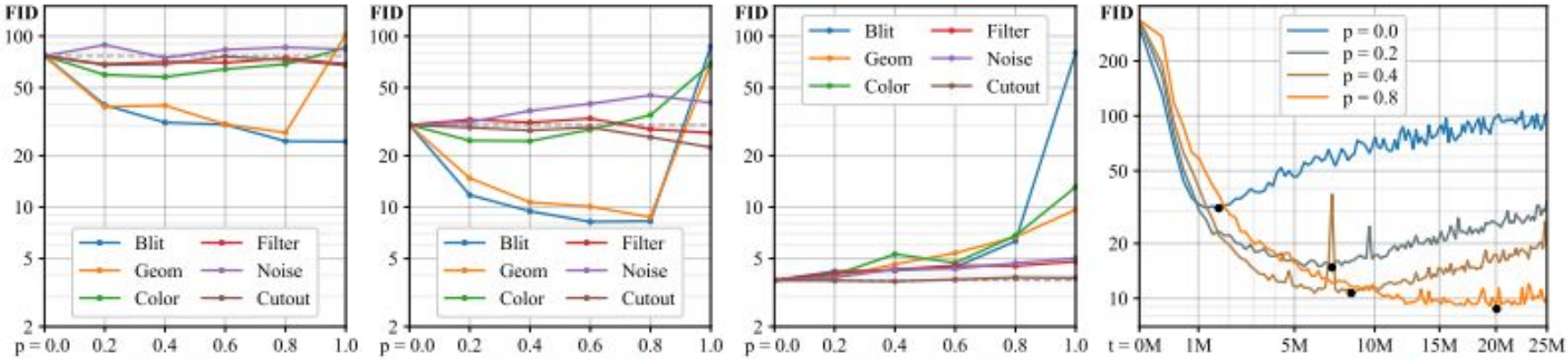


augmentation pipeline

- maximally diverse set of augmentations is beneficial.
- During training, we process each image shown to the discriminator using a pre-defined set of transformations in a fixed order.
- The strength of augmentations is controlled by the scalar $p \in [0, 1]$, so that each transformation is applied with probability p or skipped with probability $1 - p$.
- always use the same value of p for all transformations.
- the generator is guided to produce only clean images as long as p remains below the practical safety limit.

in different situations

- With a 2k training set, the vast majority of the benefit came from pixel blitting and geometric transforms.
- The curves also indicate some of the augmentations becoming leaky when $p \rightarrow 1$. With a 10k training set, the higher values of p were less helpful
- with 140k the situation was markedly different: all augmentations were harmful.



(a) FFHQ-2k (b) FFHQ-10k (c) FFHQ-140k (d) Convergence, 10k, Geom

Adaptive discriminator augmentation (ADA)

- avoid manual tuning of the augmentation strength and instead control it dynamically based on the degree of overfitting.
- when overfitting kicks in, the validation set starts behaving increasingly like the generated images.
- For both heuristics, r = 0 means no overfitting and r = 1 indicates complete overfitting

plausible overfitting heuristics

outputs $\mathbb{E}[\cdot]$: mean over N consecutive minibatche

D_{train}

$D_{\text{validation}}$

$$r_v = \frac{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{validation}}]}{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{generated}}]}$$

validation set relative to the training set and generated images.

$D_{\text{generated}}$

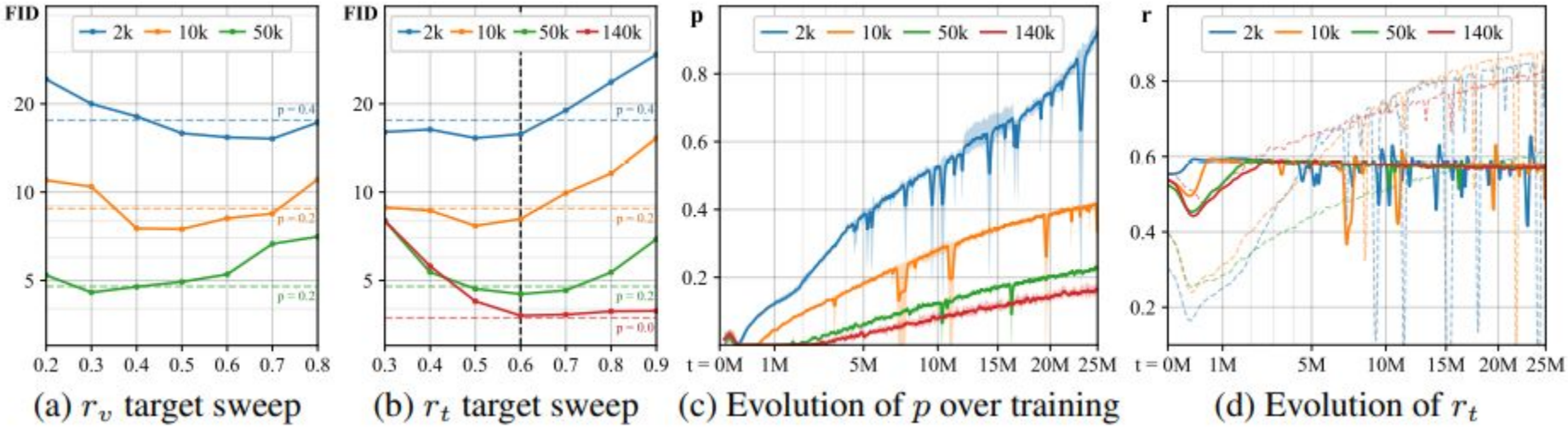
$$r_t = \mathbb{E}[\text{sign}(D_{\text{train}})]$$

portion of the training set that gets positive discriminator outputs

ADA cont.

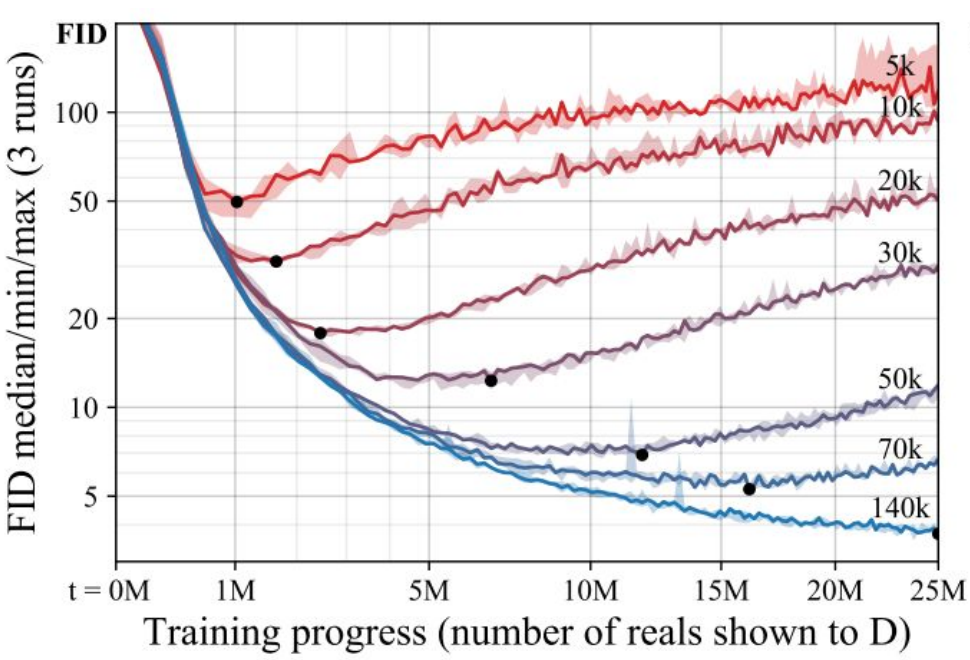
- initialize p to zero and adjust its value once every four minibatches based on the chosen overfitting heuristic.
- If the heuristic indicates too much/little overfitting, we counter by incrementing/decrementing p by a fixed amount.
- After every step clamp p from below to 0.
- effective in preventing overfitting, and that they both improve the results over the best fixed p found using grid search.
- the evolution of r_t with adaptive vs fixed p, showing that a fixed p tends to be too strong in the beginning and too weak towards the end.

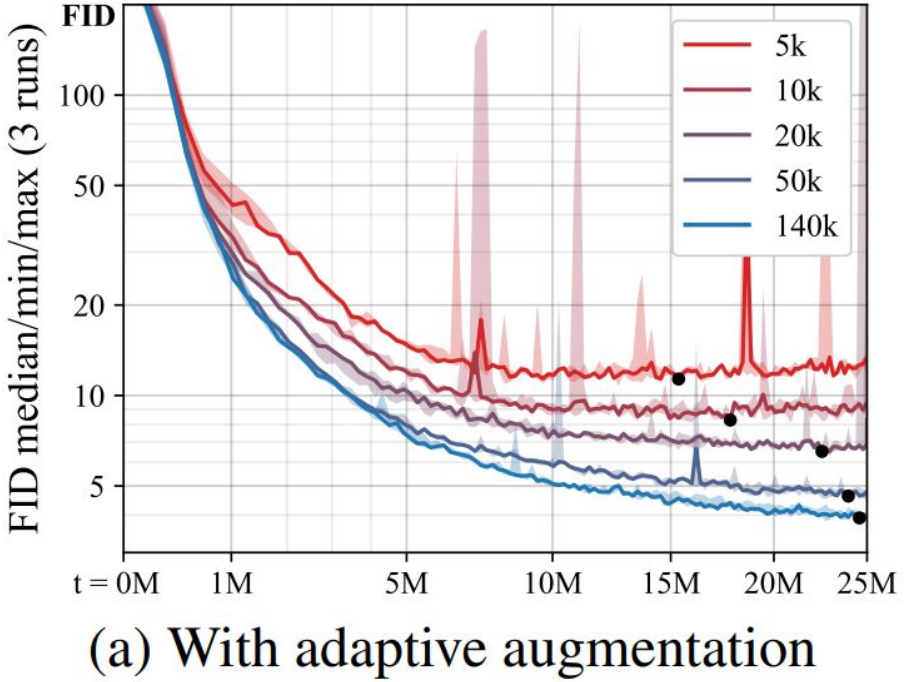
Behavior of ADA



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

results





(a) With adaptive augmentation

solve

- It thereby primarily contributes to the deep technical question of how much data is enough for generative models to succeed in picking up the necessary commonalities and relationships in the data.

implementation detail

- generally beneficial to use the highest learning rate that does not result in training instability
- larger minibatch size allows for a slightly higher learning rate
- moving average of generator weights, the natural choice is to parameterize the decay rate with respect to minibatches— not individual images— so that increasing the minibatch size results in a longer decay.

R1 regularization

- In practice, we have found that a good initial guess is given by $\gamma_0 = 0.0002 \cdot N/M$, where $N = w \times h$ is the number of pixels and M is the minibatch size. Nevertheless, the optimal value of γ tends to vary depending on the dataset, so we recommend experimenting with different values in the range $\gamma \in [\gamma_0/5, \gamma_0 \cdot 5]$.

Contrastive Learning

Improving Text-to-Image Synthesis Using Contrastive Learning

Abstract

- In practice, the captions annotated by humans for the same image have large variance in terms of contents and the choice of words.
- The linguistic discrepancy(語言差異) between the captions of the identical image leads to the synthetic images deviating from the ground truth.

example

- This wire metal rack holds several pairs of shoes and sandals
- A dog sleeping on a show rack in the shoes.
- Various slides and other footwear rest in a metal basket outdoors.
- A small dog is curled up on top of the shoes
- a shoe rack with some shoes and a dog sleeping on them

example

- A toy dinosaur standing on a sink next to a running faucet.
- a faucet running next to a dinosuar holding a toothbrush
- A toy lizard with a toothbrush in it's mouth standing next to a running water faucet in a bathroom.
- A fake toy dinousure has a green tooth brush in its mouth
- a sink with running water a mirror and a Godzilla toothbrush holder

structure

- In the pretraining stage, we utilize the contrastive learning approach to learn the consistent textual representations for the captions corresponding to the same image.
- in the following stage of GAN training, we employ the contrastive learning method to enhance the consistency between the generated images from the captions related to the same image.
- we learn the consistent textual representations by pushing together the captions of the same image and pushing way the captions of different images via the contrastive loss.

result

- Especially, on the challenging COCO dataset, our approach boosts the FID significantly by 29.60% over AttnGAN and by 21.96% over DM-GAN.

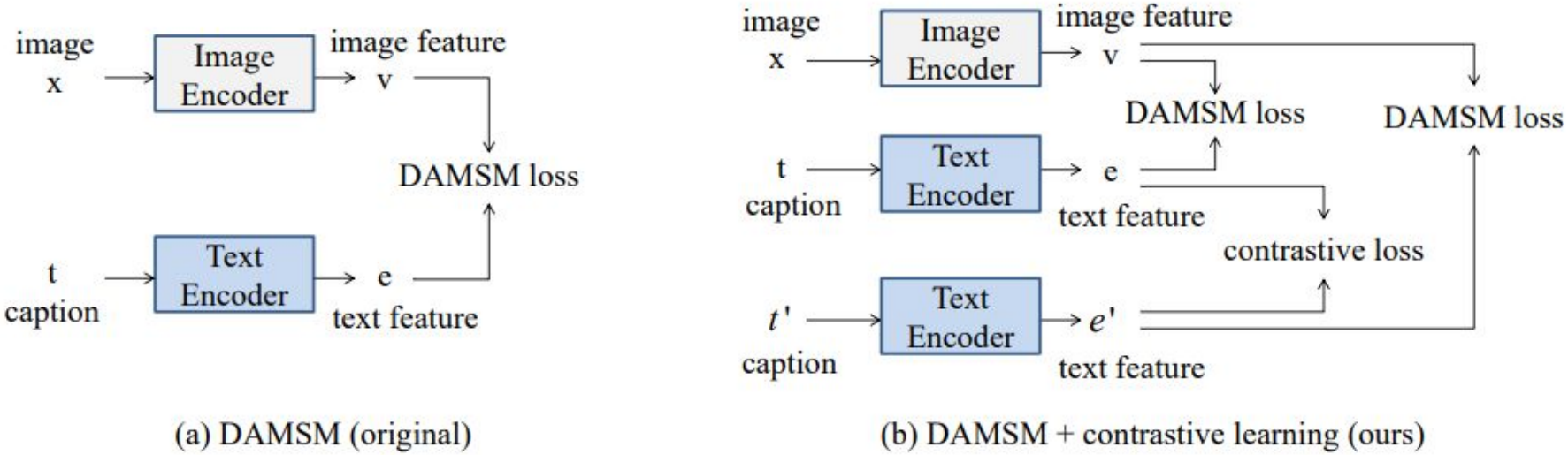
Method	Dataset	IS ↑	FID ↓	R-Precision ↑
DM-GAN*	CUB	4.66 ± .06	15.10	75.86 ± .83
DM-GAN + CL	CUB	**4.77 ± .05**	**14.38**	**78.99 ± .66**
DM-GAN*	COCO	32.37± .29	26.64	92.09 ± .50
DM-GAN + CL	COCO	**33.34 ± .51**	**20.79**	**93.40 ± .39**

Scenes

- In practice, one image is associated to multiple captions in the datasets.
- These text descriptions annotated by humans for the same image are highly subjective and diverse in terms of contents and choice of words.

Contrastive Learning for Pre-training

captions **t** and **t'** are corresponding to images **x**

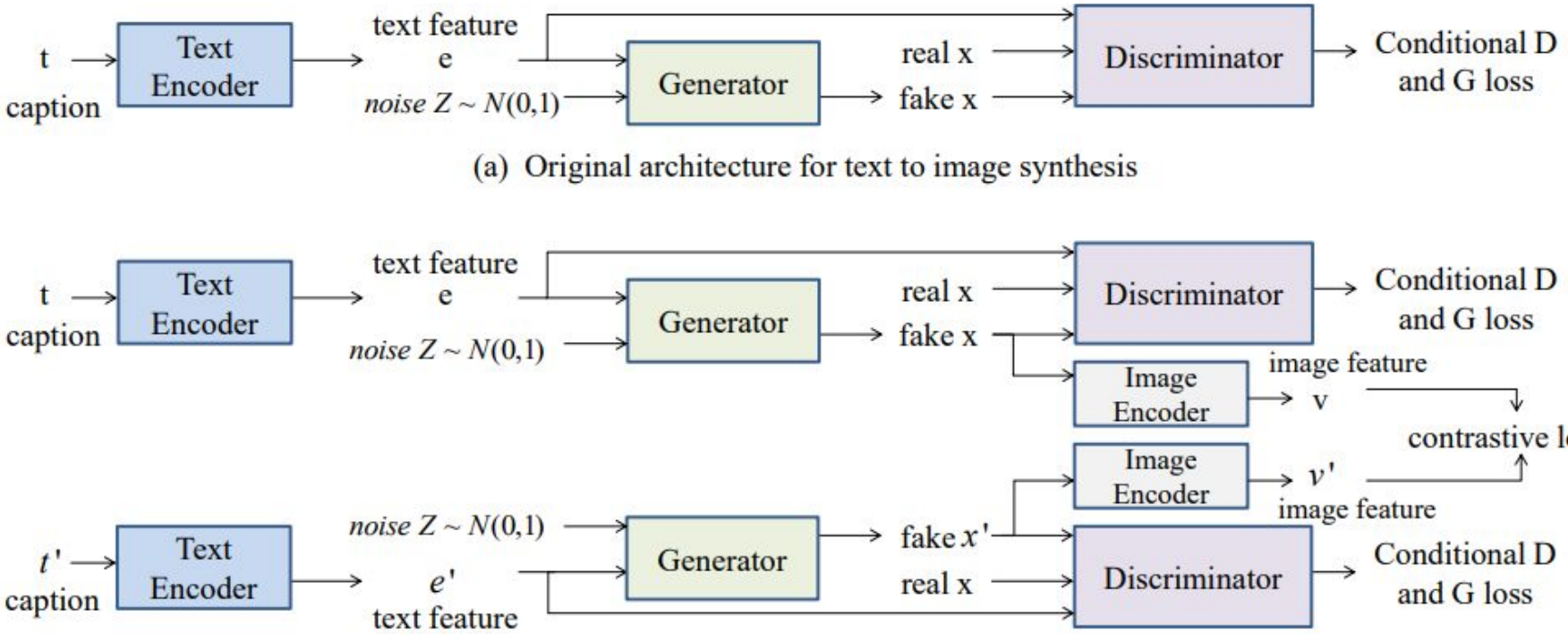


(a) DAMSM (original)

(b) DAMSM + contrastive learning (ours)

Figure 1. Architectures of original DAMSM and our approach for image-text matching.

Contrastive learning for GAN training



Ablation Study

CL1 : contrastive learning on pretraining

CL2 : contrastive learning on gan training

Method	Dataset	IS ↑	FID ↓	R-Precision ↑
AttnGAN†	CUB	4.29 ± .05	19.16	68.02 ± .98
+ CL1	CUB	4.31 ± .02	17.97	69.11 ± .63
+ CL1 + CL2	CUB	**4.42 ± .05**	**16.34**	**69.64 ± .63**
AttnGAN†	COCO	25.05 ± .64	30.67	84.24 ± .58
+ CL1	COCO	**25.87 ± .41**	26.89	85.93 ± .63
+ CL1 + CL2	COCO	25.70 ± .62	**23.93**	**86.55 ± .51**

Method	Dataset	IS ↑	FID ↓	R-Precision ↑
DM-GAN†	CUB	4.67 ± .06	15.55	75.88 ± .89
+ CL1	CUB	4.71 ± .05	14.56	76.74 ± .88
+ CL1 + CL2	CUB	**4.77 ± .05**	**14.38**	**78.99 ± .66**
DM-GAN†	COCO	31.53 ± .39	27.04	91.82 ± .49
+ CL1	COCO	30.98 ± .69	25.29	92.10 ± .56
+ CL1 + CL2	COCO	**33.34 ± .51**	**20.79**	**93.40 ± .39**

training experiment

different parameter set (reduce setting)

contrastive rate (at 140 kimg)

weights between two positive example

high contrastive rate
(0.2)

low contrastive rate
(0.02)

contrastive rate (at 280 kimg)

real image

high contrastive rate
(0.2)

low contrastive rate
(0.02)

itd (at 320 kimg)

img-txt-discriminator loss

real image

high itd (10)

low itd (5)

itd (at 320 kimg)

high itd (10)

low itd (5)

separate img and txt features (at 200 kimg)

real image

modified

normal

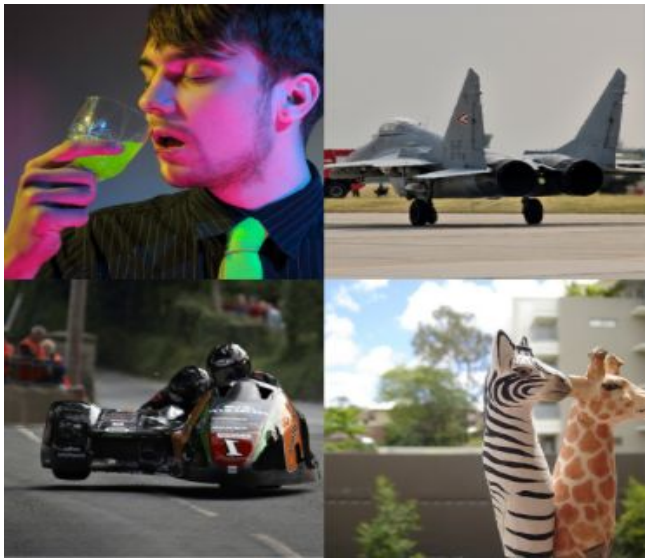
separate img and txt features (at 320 kimg)

real image

modified

normal

separate img and txt features (at 320 kimg)



real image

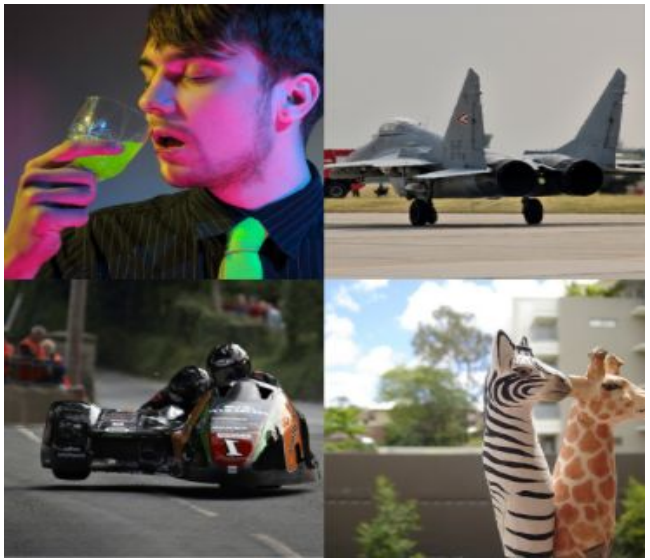
modified

normal

training experiment

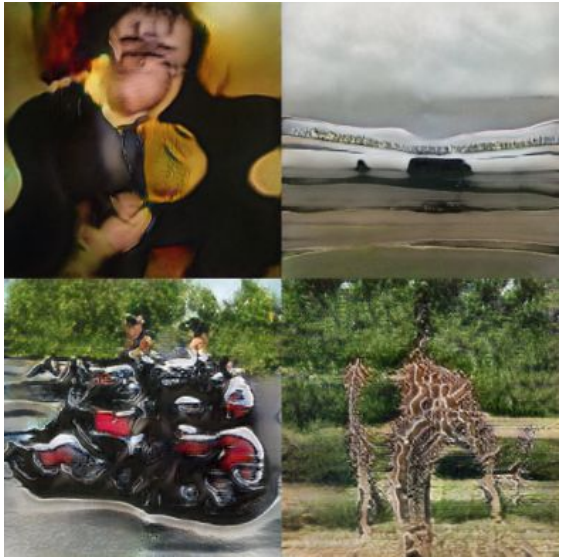
different parameter set (normal setting)

compare (at 320 kimg)



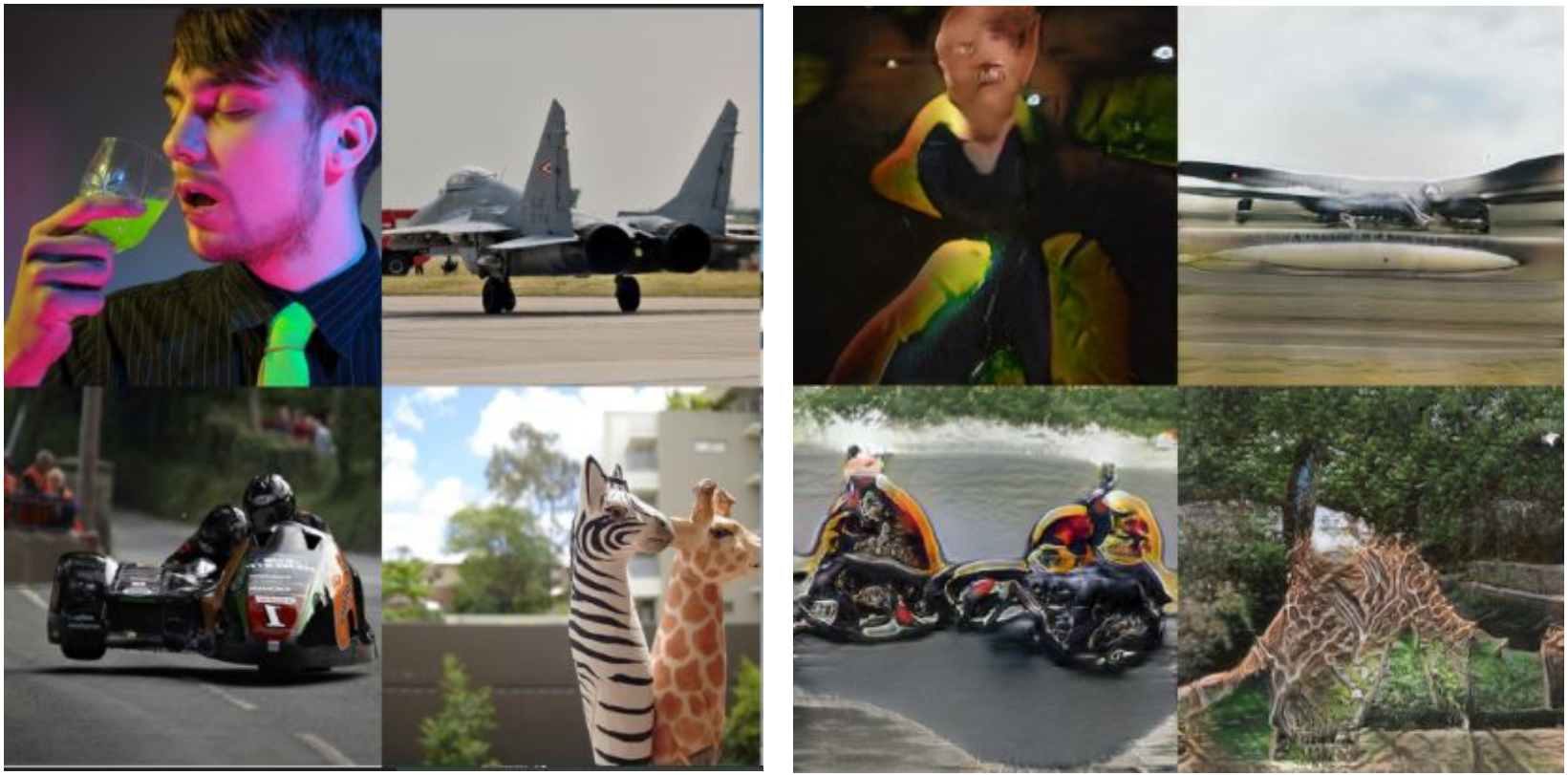
real image

reduced



normal

640 kimg



new stuff

- enable img-img loss
- separate img and txt features (discriminator)
- different generator layer (transform w)
- new contrastive learning structure
- mixing discriminator logits
- resnet guided discriminator

future work

- contrastive learning on text feature encoder
- tune gamma parameter
- discriminator embedding feature structure

study

- training gans with stronger augmentations via contrastive discriminator
  - https://arxiv.org/abs/2103.09742
- A Simple Framework for Contrastive Learning of Visual Representations
  - https://arxiv.org/abs/2002.05709
- NT-Xent loss
  - https://arxiv.org/abs/2011.02803
- On Self Modulation for Generative Adversarial Networks
  - https://arxiv.org/abs/1810.01365

reference

- Style-GAN-2
  - https://arxiv.org/abs/1912.04958
- Style-GAN-2 ada
  - https://arxiv.org/abs/2006.06676
- Improving Text-to-Image Synthesis Using Contrastive Learning
  - https://arxiv.org/abs/2107.02423