Generative Adversarial Networks (part 2)
Jun-Yan Zhu
16-726 Learning-based Image Synthesis, Spring 2022

many slides from Phillip Isola, Richard Zhang, Alyosha Efros
Fake image generation process:

- $z$: Random code input
- $G(z)$: Generator function
- $G$: Generator block

$G(z)$ produces a fake image.

[Goodfellow et al. 2014]
A two-player game:

- \( G \) tries to generate fake images that can fool \( D \).
- \( D \) tries to detect fake images.

[Goodfellow et al. 2014]
Learning objective (GANs)

$$\min_{G} \max_{D} \mathbb{E}_{z} [\log(1 - D(G(z)))]$$

[Goodfellow et al. 2014]
Learning objective (GANs)

\[
\min_G \max_D \mathbb{E}_z [\log(1 - D(G(z)))] + \mathbb{E}_x [\log D(x)]
\]

[Goodfellow et al. 2014]
Learning objective (GANs)

$$\min_G \max_D \mathbb{E}_z \left[ \log (1 - D(G(z))) \right] + \mathbb{E}_x \left[ \log D(x) \right]$$

[Goodfellow et al. 2014]
GANs Training Breakdown

G tries to synthesize fake images that fool D

D tries to identify the fakes

- Training: iterate between training D and G with backprop.
- Global optimum when G reproduces data distribution.

[Goodfellow et al., 2014]
What has driven GAN progress?

Ian Goodfellow @goodfellow_ian · Jan 14

4.5 years of **GAN progress** on face generation. [arxiv.org/abs/1406.2661](http://arxiv.org/abs/1406.2661)  
[arxiv.org/abs/1606.07536](http://arxiv.org/abs/1606.07536)  
[arxiv.org/abs/1710.10196](http://arxiv.org/abs/1710.10196)  
What has driven GAN progress?

Samples from StyleGAN2 [Karras et al., CVPR 2020]
GANs evaluation (FID)

Fréchet Inception Distance (FID)

\[ \text{FID} = \left\| \mu - \hat{\mu} \right\|_2^2 + \text{Tr}(\Sigma + \hat{\Sigma} - 2(\Sigma \hat{\Sigma})^{1/2}) \]
What has driven GAN progress?

A. Loss functions
B. Network architectures (G/D)
C. Training methods
D. Data
E. GPUs
F. Funding
Which topics are easy to publish?

A. Loss functions
B. Network architectures (G/D)
C. Training methods
D. Data
E. GPUs
F. Funding
Which topics are easy to publish?

A. Loss functions
B. Network architectures (G/D)
C. Training methods
D. Data
E. GPUs
F. Funding
Loss functions
Learning objective (GANs)

$$\min_G \max_D \mathbb{E}_z \left[ \log (1 - D(G(z))) \right] + \mathbb{E}_x \left[ \log D(x) \right]$$

[Goodfellow et al. 2014]
Learning objective (GANs variants)

\[
\min_G \max_{f_1, f_2} \mathbb{E}_z [f_1(G(z))] + \mathbb{E}_x [f_2(x)]
\]

EBGAN, WGAN, LSGAN, etc

[Goodfellow et al. 2014]
Other divergences?

Different choices of \( f_1 \) and \( f_2 \) correspond to different divergence measures:

- Original GAN \( \rightarrow \) JSD
- Least-squares GAN \( \rightarrow \) Pearson chi-squared divergence

\[
\min_G \max_{f_1, f_2} \left[ \mathbb{E}_z[f_1(G(z))] + \mathbb{E}_x[f_2(x)] \right] \quad \begin{array}{c}
\text{Convenient choice} \\
\end{array}
\]

\[
\begin{array}{c}
\min_D V_{\text{LSGAN}}(D) = \frac{1}{2} \mathbb{E}_{x \sim p_{\text{data}}}(x) [(D(x) - 1)^2] + \frac{1}{2} \mathbb{E}_{z \sim p_z}(z) \\
\min_G V_{\text{LSGAN}}(G) = \frac{1}{2} \mathbb{E}_{z \sim p_z}(z) [(D(G(z)) - 1)^2]. \\
\end{array}
\]
\[ p_g = p_{data} \] is the unique global minimizer of the GAN objective.

Proof

Optimal discriminator given fixed \( G \)

\[
C(G) = \mathbb{E}_{x \sim p_{data}} \left[ \log D^*_G(x) \right] + \mathbb{E}_{x \sim p_g} \left[ \log (1 - D^*_G(x)) \right] \\
= \mathbb{E}_{x \sim p_{data}} \left[ \log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right] + \mathbb{E}_{x \sim p_g} \left[ \log \frac{p_g(x)}{p_{data}(x) + p_g(x)} \right]
\]

\[
C(G') = - \log(4) + KL \left( p_{data} \parallel \frac{p_{data} + p_g}{2} \right) + KL \left( p_g \parallel \frac{p_{data} + p_g}{2} \right)
\]

\[
C(G) = - \log(4) + 2 \cdot JSD \left( p_{data} \parallel p_g \right)
\]

\[
\geq 0, \quad 0 \iff p_g = p_{data} \quad \square
\]

KLD (Kullback–Leibler divergence):

\[
\mathcal{KL}(p \parallel q) = \int p(x) \log \frac{p(x)}{q(x)} dx
\]

JSD (Jensen–Shannon divergence):

\[
JSD(p \parallel q) = \frac{1}{2} \mathcal{KL}(p \parallel \frac{p + q}{2}) + \frac{1}{2} \mathcal{KL}(q \parallel \frac{p + q}{2})
\]
Other divergences?

Different choices of $f_1$ and $f_2$ correspond to different divergence measures:

- Original GAN $\rightarrow$ JSD
- Least-squares GAN $\rightarrow$ Pearson chi-squared divergence

$$\min_G \max_{f_1, f_2} \mathbb{E}_z [f_1(G(z))] + \mathbb{E}_x [f_2(x)]$$

Convenient choice

$$f_1 = -f, \quad f_2 = f$$

$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2} \mathbb{E}_{x \sim p_{\text{data}}}(x) [(D(x) - 1)^2] + \frac{1}{2} \mathbb{E}_{z \sim p_z}(z)$$

$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2} \mathbb{E}_{z \sim p_z}(z) [(D(G(z)) - 1)^2].$$
Other divergences?

\[ KL(p_{\text{data}} \mid p_{\theta}) \] \[ \leftarrow \] \[ \mathbb{E}_{x \sim p_{\text{data}}} [\log p_{\theta}(x)] \]

\[ KL(p_{\theta} \mid p_{\text{data}}) \] \[ \leftarrow \] Reverse KL — mode seeking, intractable

\[ JS(p_{\text{data}}, p_{\theta}) \] \[ \leftarrow \] Jensen-Shannon, original GAN

\[ W(p_{\text{data}}, p_{\theta}) = \inf_{\gamma \in \Pi(p_{\text{data}}, p_{\theta})} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|] \] \[ \leftarrow \] Wasserstein

Earth-Mover (EM) distance / Wasserstein distance
Wasserstein GAN

\[\text{Lipschitz continuity}\]
\[|f(x) - f(y)| \leq |x - y|\]

\[
\arg \min_G \max_f \quad \mathbb{E}_{z,x} \left[ -f(G(z)) \right] + f(x)
\]

\[
W(p_{data}, p_\theta) = \inf_{\gamma \in \Pi(p_{data}, p_\theta)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|]
\]

wGAN GP [Gulrajani et al., 2018]:

\[
\arg \min_G \max_f \mathbb{E}_{z,x} \left[ -f(G(z)) \right] + f(x) + \lambda \mathbb{E}_{\hat{x} \sim P_\hat{x}} \left[ (\|\nabla_{\hat{x}} f(\hat{x})\|_2 - 1)^2 \right]
\]

Gradient penalty (GP)
Spectral Normalization

[Miyato, Kataoka, Koyama, Yoshida 2018]

\[
\tilde{W}_{SN}(W) := \frac{W}{\sigma(W)} \quad \sigma(A) := \max_{h: h \neq 0} \frac{\|Ah\|_2}{\|h\|_2}
\]

- \( W \) is the weight of one layer in the discriminator
- \( \sigma(A) \) (spectral norm) is the largest singular value of \( A \)
  (If \( A \) is a square matrix, the largest eigenvalue)
- Effect: limit the amount of changes each layer introduces

\[
\sigma(\tilde{W}_{SN}) = 1
\]

+ Lipschitz discriminator regularization (c.f. Wasserstein GAN)
Better objectives? optimizers?

Figure 4: A *wide range* hyperparameter search (100 hyperparameter samples per model). Black stars indicate the performance of suggested hyperparameter settings. We observe that GAN training is extremely sensitive to hyperparameter settings and there is no model which is significantly more stable than others.

[“Are all GANs Created Equal?”, Lucic*, Kurach*, et al. 2018]
Network architectures & Training methods
Better Architectures!

DCGAN
[Radford, Metz, Chintala 2016]

StyleGAN
[Karras, Laine, Aila 2019]
Better Architectures!

DCGAN
[Radford, Metz, Chintala 2016]

StyleGAN
[Karras, Laine, Aila 2019]
DCGAN
[Radford, Metz, Chintala 2015]

Convnet

also see LAPGAN [Denton*, Chintala*, Szlam, Fergus 2015], which used a convnet
DCGAN
[Radford, Metz, Chintala 2015]
DCGAN
[Radford, Metz, Chintala 2015]
Progressive GAN: Better Training Scheme!

[Karras, Aila, Laine, Lehtinen 2018]

Computer vision can help quality: Gaussian Pyramid (HW1)
Progressive GAN: Better Training Scheme!

[Karras, Aila, Laine, Lehtinen 2018]
Progressive GAN: Better Training Scheme!

[Karras, Aila, Laine, Lehtinen 2018]

- Coarse-to-fine
- Progressive training

Computer vision can help speed: Gaussian Pyramid (HW1)
StyleGAN: Quality+ Control

[Karras, Laine, Aila. CVPR 2019]

+ Multiscale “style” (noise)
+ AdaIN layers

\[
\text{AdaIN}(x) = \gamma(w)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \beta(w)
\]

w: style code

- batch/instance normalization:

\[
\text{BN}(x) = \gamma\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \beta
\]
StyleGAN: Quality+ Control

[Karras, Laine, Aila. CVPR 2019]
StyleGAN2 and StyleGAN3

Analyzing and improving individual layers

[Diagram of StyleGAN and StyleGAN3 layers]

Weight Modulation Layers

Alias-free layers


https://arxiv.org/abs/2106.12423
Data
Data alignment

- Work well for well-aligned objects and landscapes.

Figure 8: Creating the CELEBA-HQ dataset. We start with a JPEG image (a) from the CelebA in-the-wild dataset. We improve the visual quality (b, top) through JPEG artifact removal (b, middle) and 4x super-resolution (b, bottom). We then extend the image through mirror padding (c) and Gaussian filtering (d) to produce a visually pleasing depth-of-field effect. Finally, we use the facial landmark locations to select an appropriate crop region (e) and perform high-quality resampling to obtain the final image at $1024 \times 1024$ resolution (f).

See more details in Appendix of Progressive GANs [Karras et al., ICLR 2018]
Aligned vs. unaligned data

Real images from aligned FFHQ

StyleGAN2 samples

Photo credit: StyleGAN2 [Karras et al., CVPR 2020]
Aligned vs. unaligned data

Real images from unaligned CelebA

StyleGAN2 samples

Photo credit: William Peebles (UC Berkeley)
Data are Expensive

FFHQ dataset: 70,000 selective post-processed human faces  ImageNet dataset: millions of images from diverse categories

Months or even years to collect the data, along with *prohibitive* annotation costs.
GANs Heavily Deteriorate Given Limited Data

Generated samples of StyleGAN2 (Karras et al.) using only hundreds of images
Discriminator Overfitting

**D's Training Accuracy**

**D's Validation Accuracy**

- Teal: 100% training data
- Yellow: 20% training data
- Maroon: 10% training data

Axes:
- X-axis: \( \times 10^3 \) iterations
- Y-axis: Accuracy range from 0.0 to 1.0
Data Augmentation

Data augmentation: enlarge datasets without collecting new samples.
How to Augment GANs?
Augment reals only: the same artifacts appear on the generated images.
#2 Approach: Augment reals & fakes for $D$ only

Augment $D$ only: the unbalanced optimization cripples training.
Our approach (DiffAugment): Augment reals + fakes for both $D$ and $G$
CIFAR-10 (unconditional GANs)

Differentiable Augmentation for Data-Efficient GAN Training. Zhao et al., NeurIPS 2020

![Bar chart showing FID scores for StyleGAN2 baseline and StyleGAN2 baseline + DiffAugment.]
ImageNet Generation (25% training data)

BigGAN (baseline)

IS: 46.5  FID: 25.37

+ DiffAugment (ours)

IS: 74.2  FID: 13.28

Differentiable Augmentation for Data-Efficient GAN Training. Zhao et al., NeurIPS 2020
Low-Shot Generation

Obama: 100 images
Cat (Simard et al.): 160 images
Dog (Simard et al.): 389 images

StyleGAN2 (baseline)
StyleGAN2 + DiffAugment (ours)
100-Shot Interpolation

The smooth interpolation results suggest little overfitting of our method even given only 100 images of Obama, grumpy cat, panda, the Bridge of Sighs, the Medici Fountain, the Temple of Heaven, and Wuzhen.
from DiffAugment_pytorch import DiffAugment
# from DiffAugment_tf import DiffAugment
policy = 'color,translation,cutout'  # If your dataset is as small as ours (e.g.,
# hundreds of images), we recommend using the strongest Color + Translation + Cutout.
# For large datasets, try using a subset of transformations in ['color', 'translation', 'cutout'].
# Welcome to discover more DiffAugment transformations!

...  
# Training loop: update D
reals = sample_real_images()  # a batch of real images
z = sample_latent_vectors()
fakes = Generator(z)  # a batch of fake images
real_scores = Discriminator(DiffAugment(reals, policy=policy))
fake_scores = Discriminator(DiffAugment(fakes, policy=policy))
# Calculating D's loss based on real_scores and fake_scores...
...

...

# Training loop: update G
z = sample_latent_vectors()
fakes = Generator(z)  # a batch of fake images
fake_scores = Discriminator(DiffAugment(fakes, policy=policy))
# Calculating G's loss based on fake_scores...
...
Data Augmentation for GANs


### StyleGAN2-ADA

#### Pixel blitting

- $x$-flip
- $90^\circ$ rotations
- Integer translation

#### General geometric transformations

- Isotropic scaling
- Arbitrary rotation
- Anisotropic scaling
- Fractional translation

#### Color transformations

- Brightness
- Contrast
- Luma flip
- Hue rotation
- Saturation

#### Image-space filtering

- Frequency band $b_1$ $[0, \frac{\pi}{4}]$
- Frequency band $b_2$ $[\frac{\pi}{4}, \frac{\pi}{2}]$
- Frequency band $b_3$ $[\frac{\pi}{2}, \pi]$

#### Image-space corruptions

- Additive RGB noise
- Cutout

Training Generative Adversarial Networks with Limited Data [Karras et al., NeurIPS 2020]
StyleGAN2-ADA

Adaptative data augmentation

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

\( r_t = 0 \) no overfitting, decrease augmentation
\( r=1 \) complete overfitting, increase augmentation

Other metrics to consider:

\[
\frac{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{validation}}]}{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{generated}}]} \quad \mathbb{E}[D_{\text{train}}]
\]
Training methods
Discriminator is still Overfitting

![Graph showing accuracy over iterations for StyleGAN2-ADA, with training and validation lines]

- **StyleGAN2-ADA**

  - **Accuracy**
  - **Iterations**
  - **Training**
  - **Validation**

The graph illustrates the accuracy over iterations for StyleGAN2-ADA, with clear distinctions between training and validation lines.
Standard GAN training

Which pretrained models to use?

Off-the-shelf Models

- VGG-16 (Classification)
- Swin-T (Detection)
- Swin-T (Segmentation)
- Swin-T (MoBY)
- ViT (CLIP)
- ViT (DINO)

Trained from scratch

Loss

Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021
Model Selection

Fake samples

82%

Real samples

Off-the-shelf Models

VGG-16 (Classification)
VGG-16 (Classification)
Swin-T (Detection)
Swin-T (Detection)
ViT (CLIP)
ViT (CLIP)
ViT (DINO)
ViT (DINO)

Linear Probe Accuracy

Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021
Model Selection

Linear Probe Accuracy

- VGG-16 (Classification): 82%
- Swin-T (MoBY): 98%
- Swin-T (Segmentation): 88%
- ViT (CLIP): 99%
- Swin-T (Detection): 89%
- ViT (DINO): 98%

Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021
Vision-aided GAN training

\[ z \sim VGG-16 \text{ (Classification)} \quad \text{Swin-T (MoBY)} \quad \text{Swin-T (Segmentation)} \quad \text{ViT (CLIP)} \quad \text{Swin-T (Detection)} \quad \text{ViT (DINO)} \]

Off-the-shelf Models

Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021
Add 2nd Vision-aided discriminator

\[ z \sim G \]

\[ D \]

\[ D_1 \]

\[ \text{Loss} \]

Linear Probe Accuracy

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16 (Classification)</td>
<td>72%</td>
</tr>
<tr>
<td>Swin-T (MoBY)</td>
<td>82%</td>
</tr>
<tr>
<td>Swin-T (Segmentation)</td>
<td>77%</td>
</tr>
<tr>
<td>Swin-T (Detection)</td>
<td>79%</td>
</tr>
<tr>
<td>VIT (DINO)</td>
<td>86%</td>
</tr>
</tbody>
</table>

Highest performing

Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021
Add 2nd Vision-aided discriminator

\[ z \sim G \rightarrow D \rightarrow \text{Loss} \]

VGG-16 (Classification)  Swin-T (MoBY)  Swin-T (Segmentation)  Swin-T (Detection)  VIT (CLIP)  VIT (DINO)

Off-the-shelf Models

Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021
Better pretrained features lead to lower FID
Better pretrained features lead to lower FID
Better pretrained features lead to lower FID
Benefit with varying training samples

LSUN Cat

- **StyleGAN2 (ADA)**
- **DiffAugment**

- **FID**:
  - 41.1
  - 43.3

**Training Samples**: 1k
Benefit with varying training samples

![Graph showing FID with varying training samples for LSUN Cat. The graph includes bars for StyleGAN2 (ADA), DiffAugment, and Ours. The x-axis represents training samples, and the y-axis represents FID. The values for 1k training samples are: StyleGAN2 (ADA) - 41.1, DiffAugment - 43.3, Ours - 12.2.]
Benefit with varying training samples

LSUN Cat

<table>
<thead>
<tr>
<th>Training Samples</th>
<th>FID</th>
<th>StyleGAN2 (ADA)</th>
<th>DiffAugment</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1k</td>
<td>41.1</td>
<td>43.3</td>
<td>12.2</td>
<td>7.9</td>
</tr>
<tr>
<td>5k</td>
<td>17.5</td>
<td>16.7</td>
<td>7.9</td>
<td></td>
</tr>
</tbody>
</table>
Benefit with varying training samples

**LSUN Cat**

- **StyleGAN2 (ADA)**
- **DiffAugment**
- **Ours**

<table>
<thead>
<tr>
<th>Training Samples</th>
<th>FID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1k</td>
<td>41.1</td>
</tr>
<tr>
<td>5k</td>
<td>43.3</td>
</tr>
<tr>
<td>10k</td>
<td>12.2</td>
</tr>
<tr>
<td></td>
<td>17.5</td>
</tr>
<tr>
<td></td>
<td>16.7</td>
</tr>
<tr>
<td></td>
<td>7.9</td>
</tr>
<tr>
<td></td>
<td>13.2</td>
</tr>
<tr>
<td></td>
<td>12.6</td>
</tr>
<tr>
<td></td>
<td>6.9</td>
</tr>
</tbody>
</table>
Benefit with varying training samples

![Graph showing FID scores with different training samples for LSUN Cat dataset.](image)

- **StyleGAN2 (ADA)**
- **DiffAugment**
- **Ours**

Similar FID with ~0.7% data
Improved Samples

StyleGAN2-ADA

LSUN CAT 1k
Improved Samples

Ours

LSUN CAT 1k
Low-shot Generation with 100 samples
Low-shot Generation with 100 samples
Faster Convergence with Projected GANs

Projected GANs Converge Faster. Sauer et al., NeurIPS 2021

use a FastGAN generator (https://arxiv.org/abs/2101.04775)
Combining Perceptual Loss and GAN Loss

Idea 1: add them together (many papers did that. It works)

Idea 2: Pre-trained features + trainable MLP layers = Perceptual Discriminator

Image Manipulation with Perceptual Discriminators [Sungatullina et al. ECCV 2018]
Using multiple pre-trained models: Vision-aided GANs [Kumari et al., 2021]
Using random projection head: Projected GANs [Sauer et al., NeurIPS 2021]
Conditional discriminator: Enhancing photorealism enhancement [Richter et al., 2020]
What has driven GAN progress?

Ian Goodfellow @goodfellow_ian · Jan 14
4.5 years of GAN progress on face generation. arxiv.org/abs/1406.2661
arxiv.org/abs/1812.04948
What has driven GAN progress?

A. Loss functions
B. Network architectures (G/D)
C. Training methods
D. Data
E. GPUs