Recent GAN architectures, trends

BigGAN (2018)
Recent progress in GANs

4.5 years of GAN progress on face generation.
Recent progress in GANs

EBGAN (2017)

BigGAN (2018)
Outline

• Progressive GAN
• StyleGAN
• Self-attention GAN, BigGAN
• Visualizing and controlling GANs
Progressive GANs

Realistic face images up to 1024 x 1024 resolution

Progressive GANs

- Key idea: train lower-resolution models, gradually add layers corresponding to higher-resolution outputs

Progressive GANs

- Key idea: train lower-resolution models, gradually add layers corresponding to higher-resolution outputs

Transition from 16x16 to 32x32 images

Progressive GANs: Implementation details

- Loss: WGAN-GP loss (preferred) or LSGAN
- Architectures:
  - Nearest neighbor upsampling (2x2 replication) followed by regular convolutions instead of transposed conv layers
  - Average pooling instead of striding for downsampling in discriminator
  - Leaky ReLUs used in discriminator and generator
  - Per-pixel response normalization in generator: rescale feature vector in each pixel to unit length after each conv layer
- Use of minibatch standard deviation in discriminator (append to feature map)
- Exponential moving average of generator weights for display

Progressive GANs: Results

256 x 256 results for LSUN categories
StyleGAN

- Built on top of Progressive GAN
- Start generation with constant (instead of noise vector)
- Noise vector is transformed to latent vector $w$ that is later specialized to style codes
- Style codes control *adaptive instance normalization* (AdaIN) or scaling and biasing of each feature map
- Add noise after each convolution and before nonlinearity (enables stochastic detail)

StyleGAN: Results

Mixing styles

“Two sets of images were generated from their respective latent codes (sources A and B); the rest of the images were generated by copying a specified subset of styles from source B and taking the rest from source A.”
Mixing styles
Mixing styles
DeepFakes

That smiling LinkedIn profile face might be a computer-generated fake

March 27, 2022 - 7:00 AM ET

https://www.npr.org/2022/03/27/1088140809/fake-linkedin-profiles
StyleGAN: Bedrooms

StyleGAN: Cars

StyleGAN2

- Change normalization, remove progressive growing to address StyleGAN artifacts

Figure 1. Instance normalization causes water droplet -like artifacts in StyleGAN images. These are not always obvious in the generated images, but if we look at the activations inside the generator network, the problem is always there, in all feature maps starting from the 64x64 resolution. It is a systemic problem that plagues all StyleGAN images.

Figure 6. Progressive growing leads to “phase” artifacts. In this example the teeth do not follow the pose but stay aligned to the camera, as indicated by the blue line.

T. Karras et al. Analyzing and Improving the Image Quality of StyleGAN. CVPR 2020
Figure 1: Examples of “texture sticking”. **Left:** The average of images generated from a small neighborhood around a central latent (top row). The intended result is uniformly blurry because all details should move together. However, with StyleGAN2 many details (e.g., fur) stick to the same pixel coordinates, showing unwanted sharpness. **Right:** From a latent space interpolation (top row), we extract a short vertical segment of pixels from each generated image and stack them horizontally (bottom). The desired result is hairs moving in animation, creating a time-varying field. With StyleGAN2 the hairs mostly stick to the same coordinates, creating horizontal streaks instead.

StyleGAN3

Random latent walk using directions from StyleCLIP, GANSpace, and SeFa.


Videos
Outline

• Progressive GAN
• StyleGAN
• Self-attention GAN, BigGAN
Self-attention GAN

- Adaptive receptive fields to capture non-local structure

Self-attention GAN

- Adaptive receptive fields to capture non-local structure (based on Wang et al., 2018)

How much to attend to location $1$ while synthesizing feature at location $2$
Attention map visualization
Self-attention GAN: Implementation details

- Both generator and discriminator are conditioned on class label $y$
Self-attention GAN: Implementation details

- Both generator and discriminator are conditioned on class label $y$
  - Conditioning the discriminator: projection (Miyato & Koyama, 2018)
  - Conditioning the generator: conditional batch norm
Self-attention GAN: Implementation details

• Both generator and discriminator are conditioned on class label $y$
• Hinge loss formulation
  • Discriminator: Drive discriminator score on real data above 1, on generated data below $-1$
    \[
    L_D = -E_{(x,y)\sim p_{data}} \left[ \min(0, D(x, y) - 1) \right] \\
    -E_{z\sim p_z, y\sim p_{data}} \left[ \min(0, -D(G(z, y), y) - 1) \right]
    \]
  • Generator: maximize discriminator score on generated data
    \[
    L_G = -E_{z\sim p_z, y\sim p_{data}} D(G(z, y), y)
    \]
Self-attention GAN: Implementation details

• Both generator and discriminator are conditioned on class label $y$
• Hinge loss formulation
• *Spectral normalization* for generator and discriminator (Miyato et al., 2018) – divide weight matrices by largest singular value (estimated)
• Different learning rates for generator and discriminator (TTUR or two-timescale update rule – Heusel et al., 2017)
Self-attention GAN: Results

- 128 x 128 ImageNet

- goldfish

- indigo bunting

- redshank

- Saint Bernard
BigGAN

- Scale up SA-GAN to generate ImageNet images up to 512 x 512 resolution

A. Brock, J. Donahue, K. Simonyan, *Large scale GAN training for high fidelity natural image synthesis*, ICLR 2019
BigGAN: Implementation details

- 8x larger batch size, 50% more channels (2x more parameters) than baseline SA-GAN
- Hierarchical latent space: feed (transformations of) $z$ vector into multiple layers of the generator
BigGAN: Implementation details

- 8x larger batch size, 50% more channels (2x more parameters) than baseline SA-GAN
- Hierarchical latent space: feed (transformations of) $z$ vector into multiple layers of the generator
- Truncation trick: at test time, resample the components of the $z$ vector whose magnitude falls above a certain threshold
- Trade off diversity for image quality

“The effects of increasing truncation. From left to right, the threshold is set to 2, 1, 0.5, 0.04.”
BigGAN: Implementation details

• 8x larger batch size, 50% more channels (2x more parameters) than baseline SA-GAN
• Hierarchical latent space: feed (transformations of) $z$ vector into multiple layers of the generator
• Truncation trick: at test time, resample the components of the $z$ vector whose magnitude falls above a certain threshold
• Lots of other tricks (initialization, training, etc.)
• Training observed to be unstable, but good results are achieved “just before collapse”
• Evidence that discriminator memorizes the training data, but the generator doesn’t
BigGAN: Implementation details

This is your machine learning system?

Yup! You pour the data into this big pile of linear algebra, then collect the answers on the other side.

What if the answers are wrong?

Just stir the pile until they start looking right.

https://xkcd.com/1838/
BigGAN: Results

- Samples at 256 x 256 resolution:
BigGAN: Results

- Samples at 512 x 512 resolution:
BigGAN: Results

• Interpolation between $c$ with $z$ held constant:
BigGAN: Results

- Interpolation between $c, z$ pairs:
BigGAN: Results

• Difficult classes:
Outline

• Progressive GAN
• StyleGAN
• Self-attention GAN, BigGAN
• Visualizing and controlling GANs
GAN Dissection

D. Bau et al. GAN Dissection: Visualizing and understanding generative adversarial networks. ICLR 2019
GAN Dissection

• Dissection:

D. Bau et al. GAN Dissection: Visualizing and understanding generative adversarial networks. ICLR 2019
GAN Dissection

- Interpreting units at different levels of a progressive GAN (trained on “bedroom”):

D. Bau et al. **GAN Dissection: Visualizing and understanding generative adversarial networks**. ICLR 2019
GAN Dissection

• Intervention:

D. Bau et al. GAN Dissection: Visualizing and understanding generative adversarial networks. ICLR 2019
GANPaint demo

https://ganpaint.io/demo/?project=church

D. Bau et al. Semantic Photo Manipulation with a Generative Image Prior. SIGGRAPH 2019
Traversing the GAN latent space

- Try to find simple “walks” in the latent space of GANs to achieve various meaningful transformations to explore structure of that space and test GANs’ ability to interpolate between training samples

Traversing the GAN latent space

• Goal: learn a set of directions inducing “disentangled” image transformations that are easy to distinguish from each other

A.Voynov and A. Babenko. Unsupervised Discovery of Interpretable Directions in the GAN Latent Space. ICML 2020
GAN editing

- Supervised training to find latent space directions corresponding to pose, smile, age, gender, eyeglasses.

StyleCLIP

- Combine powerful recent image-text embedding technique (CLIP) with pre-trained StyleGAN for text-based image editing

O. Patashnik et al.. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. ICCV 2021
GAN Inversion

W. Xia et al. **GAN Inversion: A Survey**. arXiv 2022
GANs for representation learning

- Bidirectional GAN (BiGAN): simultaneously train generator and encoder (mapping from images to $z$ vectors or approximate inverse of the generator), show that the encoder creates a latent representation useful for other tasks

GANs for representation learning: BigBiGaN

• Train BiGAN with BigGAN architecture on ImageNet
• Show that bidirectional framework improves image generation
• Encoder representation gives results comparable to (then) state-of-the-art self-supervised models on ImageNet classification

J. Donahue, K. Simonyan, Large Scale Adversarial Representation Learning, NeurIPS 2019
GANs for representation learning: BigBiGAN

Real images $x$

Reconstructions $G(E(x))$

J. Donahue, K. Simonyan, Large Scale Adversarial Representation Learning, NeurIPS 2019