Advanced Section #X:
Modern Generative Adversarial Networks

Camilo Fosco

CS209B Advanced topics in Data Science

Pavlos Protopapas, Mark Glickman, Chris Tanner

Outline

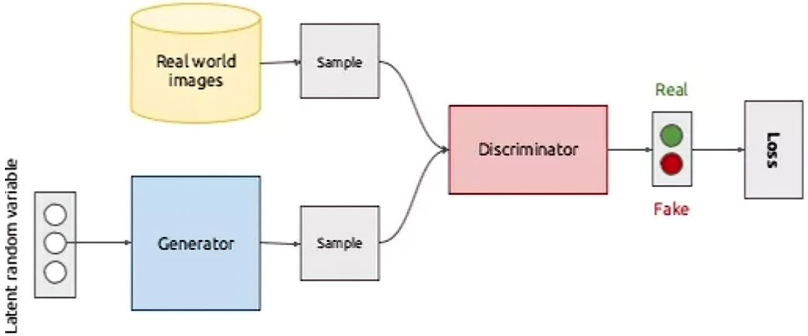
- A Refresher on GANs
- Where are GANs today?
- Description of new generation of Image-based GANs
- Towards generative models of language and vision

GAN Refresher: What is a GAN?

GAN Refresher: What is a GAN?

Generator tries to **approximate the distribution of real world images** as best as possible to fool discriminator.

Usually creates images from a **latent random variable**, but can also generate images given other images as inputs (copying their style, modifying their appearance, etc)



GAN Refresher: How do we train them?

GAN Objective:

Zero-sum non-coop game. Gans converge when G and D reach a Nash equilibrium.

$$\min_G \max_D \mathbb{E}_{x \sim q_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Training Procedure

- Generate fake images by sampling from latent space
- Train discriminator only with batches of real and fakes (N iterations)
- Sample new fake images
- Train full model (update the generator) with these new fake images and targets of 1
- Repeat

GAN Refresher: How do we evaluate them?

Evaluation methods

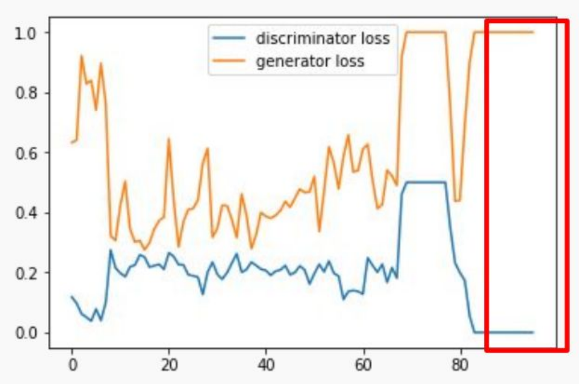
- Inception score
- Frechet Inception Distance
- Earth mover's distance
- Perceptual path length

Generating samples

- Truncation trick
- Latent space interpolation

GAN Refresher: Common Problems

- Mode collapse
- Evaluation metrics
  - Inception score?
  - TSTR?
  - Frechet Inception Distance?
- Oscillation
- Vanishing gradient
- Normalization procedures
  - If no normalization, increasingly stronger activations in G and D hinder stability



Where are GANs today?

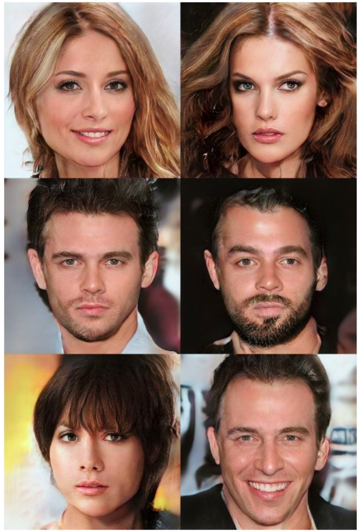
- Massive improvements in terms of **quality, stability, speed, diversity**
- GANs available in **many different areas**, not just vision
  - Tabular GANs
  - Language GANs
  - Audio GANs
- Research is exploring both **increasingly massive** nets (BigGAN) as well as **increasingly small/efficient** alternatives (MobileStyleGAN)
- Ideas related to **combining vision & language** are becoming more prominent

Modern GANs

Too many to list, but we will go over some of the most important ones and their contributions.

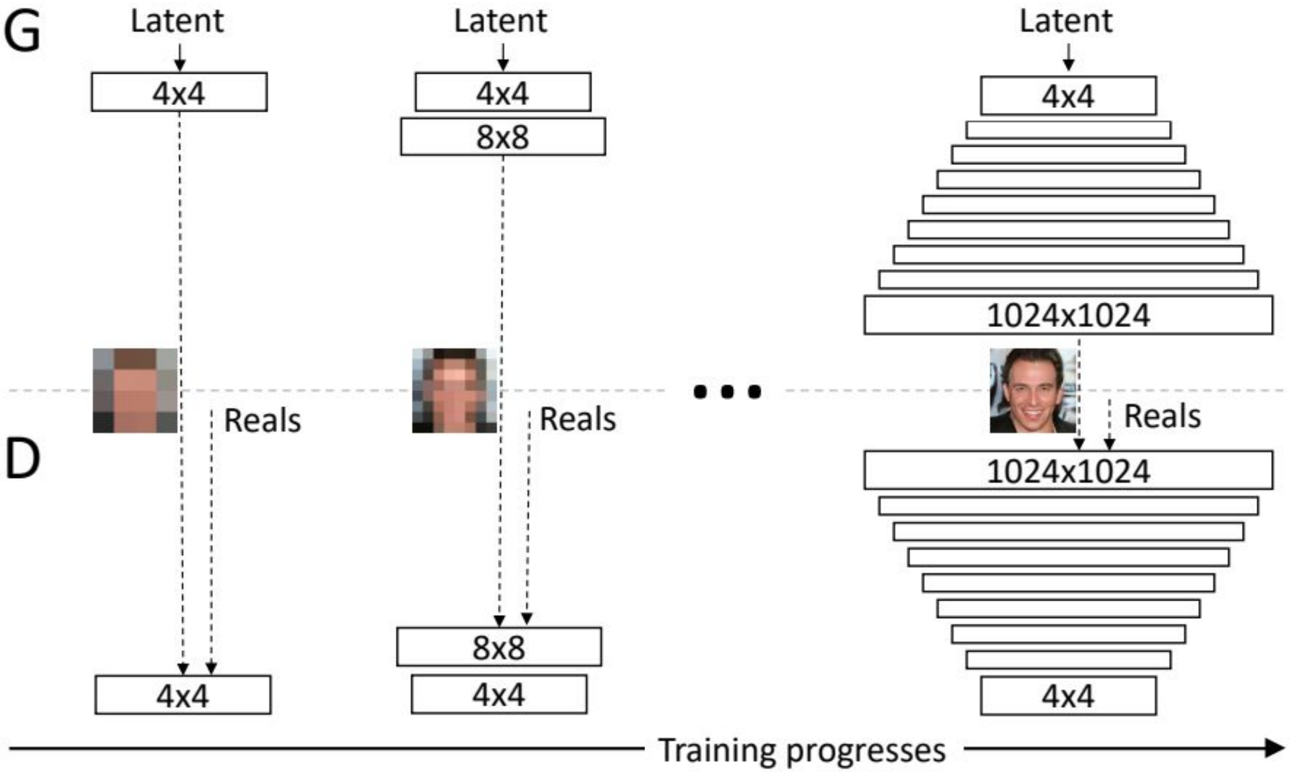
- ProGAN
- StyleGAN
- BigGAN
- GauGAN
- DALL-E

ProGAN (2017)



- First GAN generating realistic high resolution images
- Developed a “progressive growing” training methodology.
  - Makes training faster
  - Stabilizes training
  - Improves quality + variation
- Introduced various tricks to improve learning.

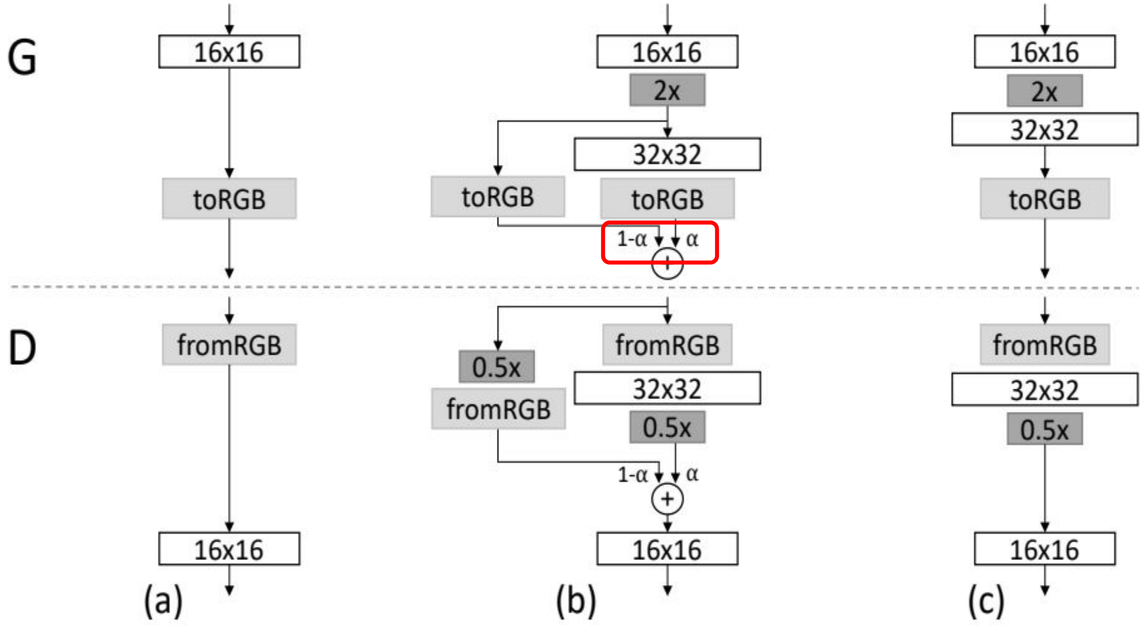
Progressive growing



Notable tricks

Smooth fading in of larger layers

When adding larger layers, they are initially appended as residual connections with a small multiplying factor that progressively grows.



Notable tricks

Smooth fading in of larger layers

When adding larger layers, they are initially appended as residual connections with a small multiplying factor that progressively grows.

Appending minibatch standard deviation

Minibatch standard deviation is computed, replicated to the size of a channel, and appended at the level of the discriminator input.

Notable tricks

Smooth fading in of larger layers

When adding larger layers, they are initially appended as residual connections with a small multiplying factor that progressively grows.

Appending minibatch standard deviation

Minibatch standard deviation is computed, replicated to the size of a channel, and appended at the level of the discriminator input.

Pixelwise Feature Vector Normalization

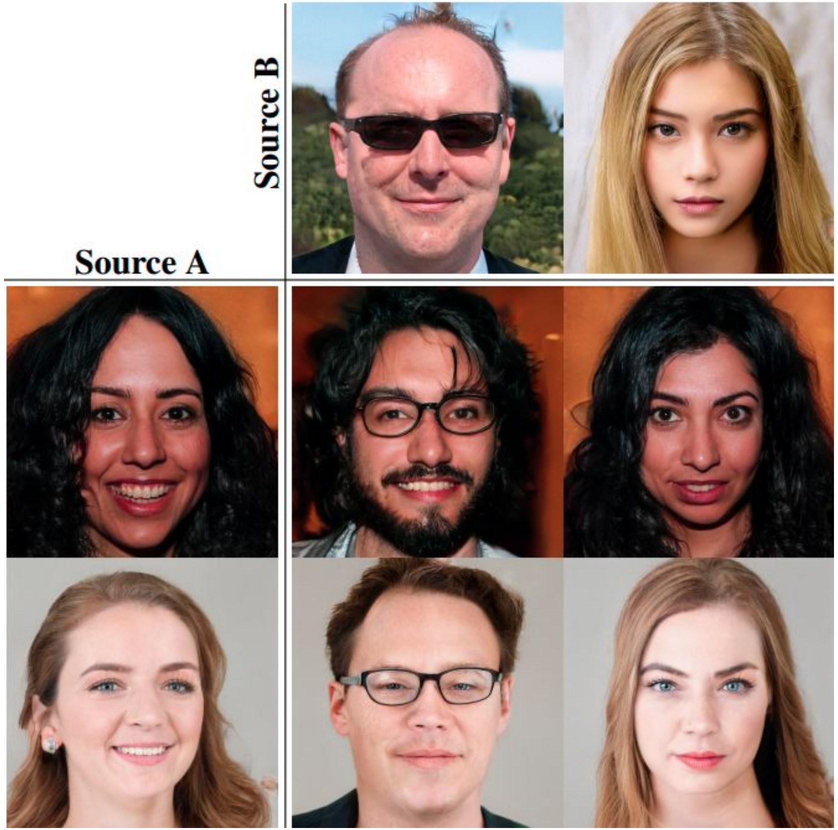
The feature vector for each pixel is normalized following:

$$b_{x,y} = a_{x,y} / \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} (a_{x,y}^{j})^2 + \epsilon}$$

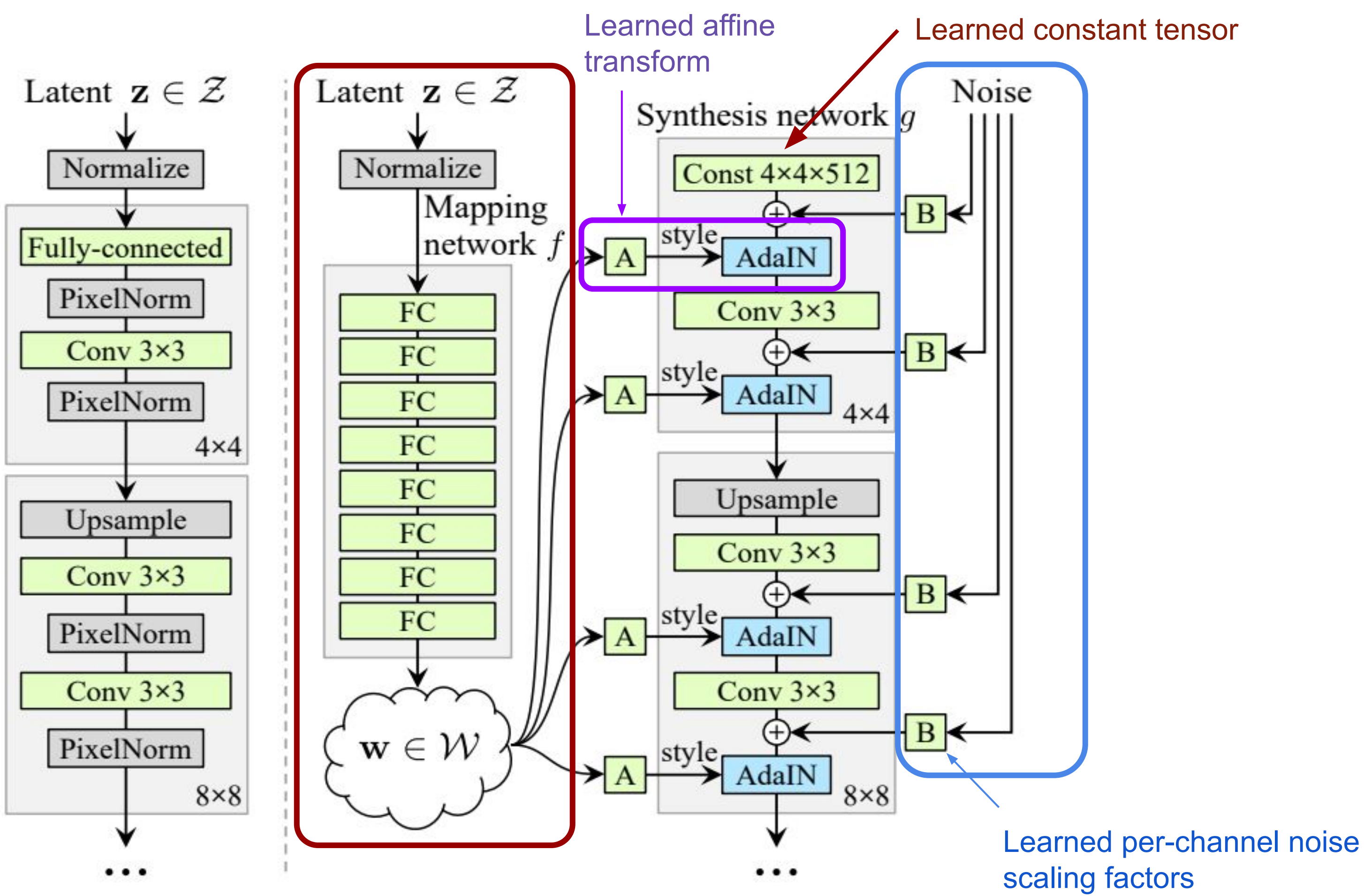
ProGAN

StyleGAN (2018)

- Introduces more control during generation: “styles” are automatically learned and can be modified at different levels
- Generator automatically learns to separate attributes like pose, identity from stochastic variation like freckles
- Effects of each style are localized in the network
- The paper introduced a highly relevant new dataset, FFHQ



https://arxiv.org/pdf/1812.04948.pdf



(a) Traditional

(b) Style-based generator

Notable tricks

AdaIN: Adaptive Instance Normalization

Normalizes feature maps, then multiplies by **style scale** and adds **style bias**

Feature map *i*

$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i},$$

Styles $\mathbf{y} = (\mathbf{y}_s, \mathbf{y}_b)$

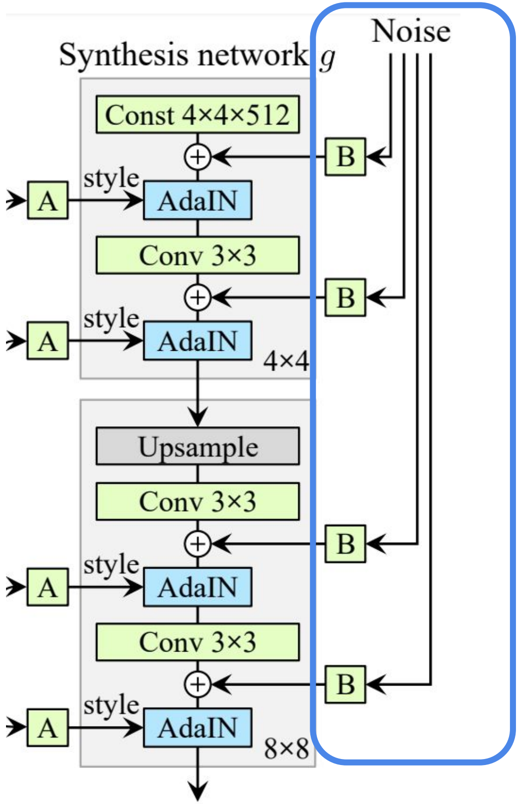
Styles have **twice the dimensionality** of corresponding feature maps at each level

This **helps localize effect of styles**: The new per-channel statistics introduced by the styles **modify the relative importance** of features for the following conv

Notable tricks

Separate noise inputs:

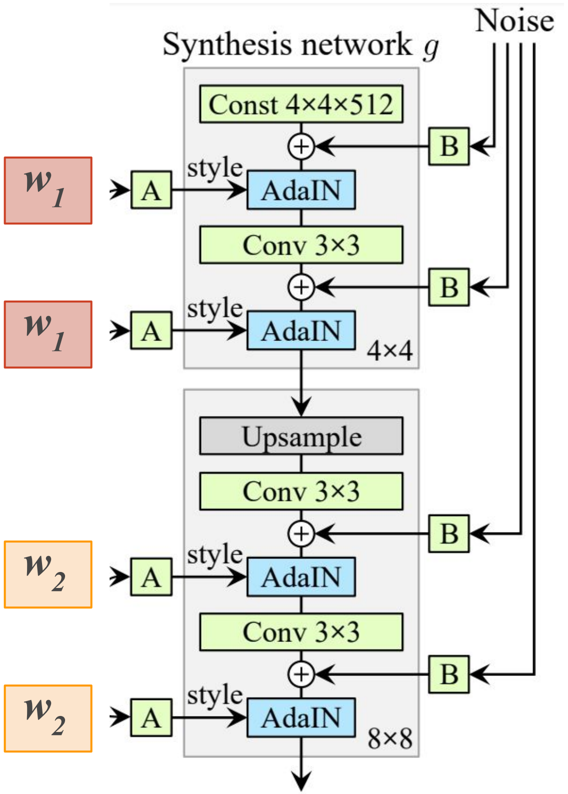
- allows for explicit stochastic detail
- Single-channel “images” with uncorrelated gaussian noise
- Dedicated noise image for each layer
- Broadcasted to all feature maps and added to output of convolution
- They show that stochastic variation ends up being localized (automatically focuses on hair and other details)



Notable tricks

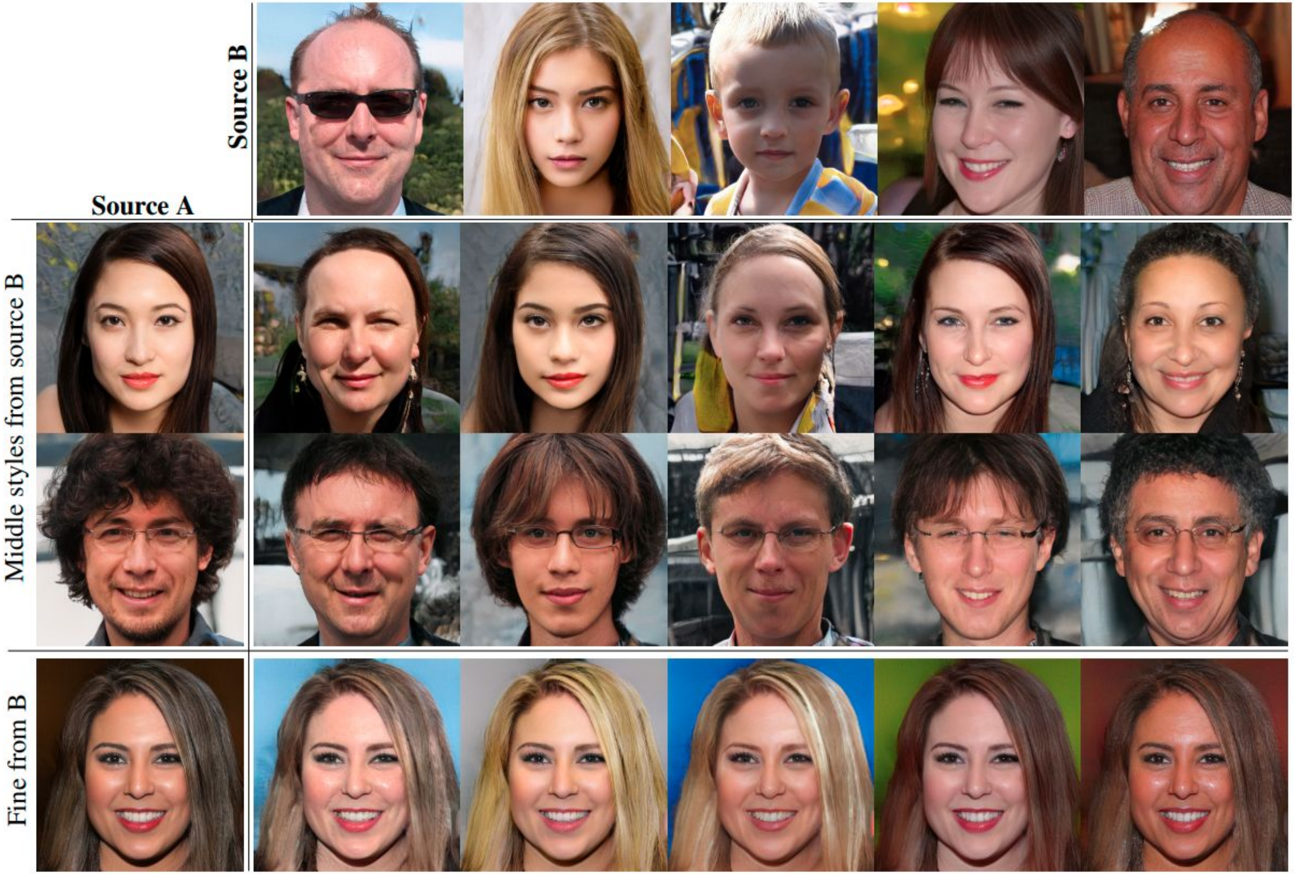
Mixing Regularization

- Percentage of images are generated using **two random latent codes instead of one** during training
- Code w_1 is used up to layer *L*, then code w_2 is used for the remaining ones
- This **prevents** the network from assuming that consecutive styles will be correlated.
- They show that FID improves (although slightly) when testing with many mixed Ws



Mixing styles

Mixing styles



Other cool things introduced by StyleGAN

- 2 new methods to quantify disentanglement
  - **Perceptual path length:** sum of perceptual differences for pairs of close images generated along an interpolation path
  - **Linear separability:** classify GAN outputs based on binary attributes (e.g. male/female), then use an SVM to see if they can be easily separated by a hyperplane. If they can -> more disentanglement.
- Truncation trick in W
- FFHQ dataset

StyleGANv2 (2019)

- Corrected several StyleGAN **artifacts**
    - Through new way of applying AdaIN
- Revisited progressive growing and generator normalization (they *dropped progressive growing*)
- Introduced **Perceptual Path Length regularization**: a new way of regularizing the generator to ensure smoother W spaces
    - Makes it easier to invert (map image back into latent space)

Notable contributions

Droplet artifact: present in most of StyleGAN's images. They traced it back to AdaIN.

Authors hypothesize that it corresponds to a strong signal that the generator crafts to dominate local statistics and leak information to subsequent layers.

They correct it by **modifying the structure of AdaIN.** They replace it entirely by a dynamic normalization on the convolution weights:

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

Generation examples

Generation examples

In the case of cat images, meme-like text can also be generated - the training set contains many cat memes!

BigGAN (2018)

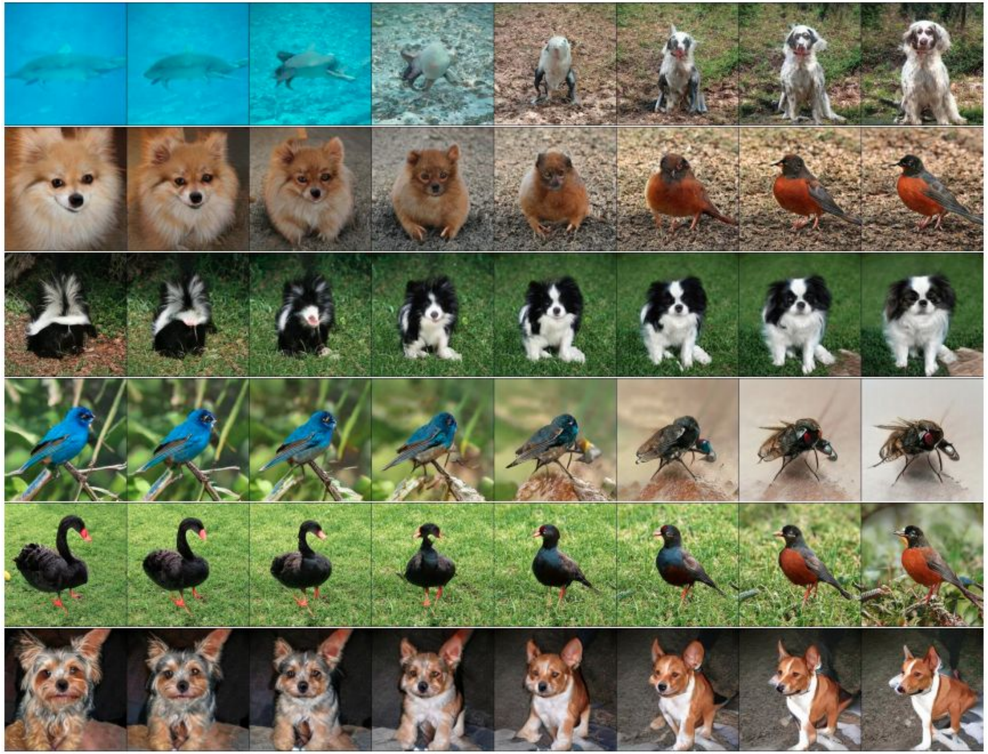
- GANs at a massive scale
- State of the art in class-conditional image synthesis
  - Massive IS and FID improvements (ISx3, FID/2)
- They make the truncation trick work for their setup
- Discover and characterize instabilities

https://arxiv.org/pdf/1809.11096.pdf

Basics of BigGAN

- SA-GAN architecture with hinge loss
- Class information through class-conditional BatchNorm in G, projection in D
- Orthogonal initialization
- Spectral Norm in G
- Two D steps per G step
- Progressive growing unnecessary
- Batch size x8 -> +46% IS, but collapse after initial convergence

Interpolations between z, c pairs



GauGAN / SPADE (2019)

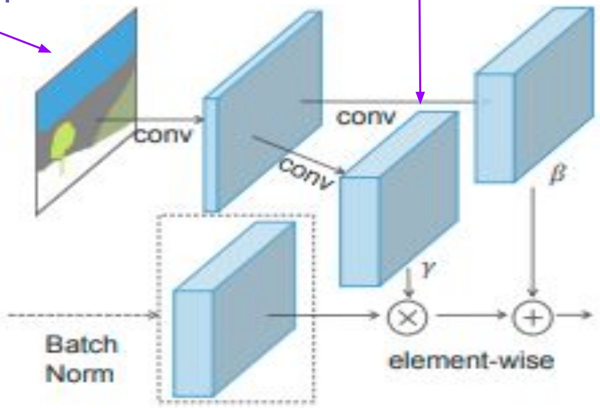
- Realistic images from **segmentation maps.**
- Spatially Adaptive Normalization: instead of feeding semantic layout as input to net, they use it to modulate activations in normalization layers.
- Proposed model allows control over both semantics and style.

How does SPADE work?

Tensor of coefficients

Input Segmentation Map

Tensor of biases



Output of previous layer

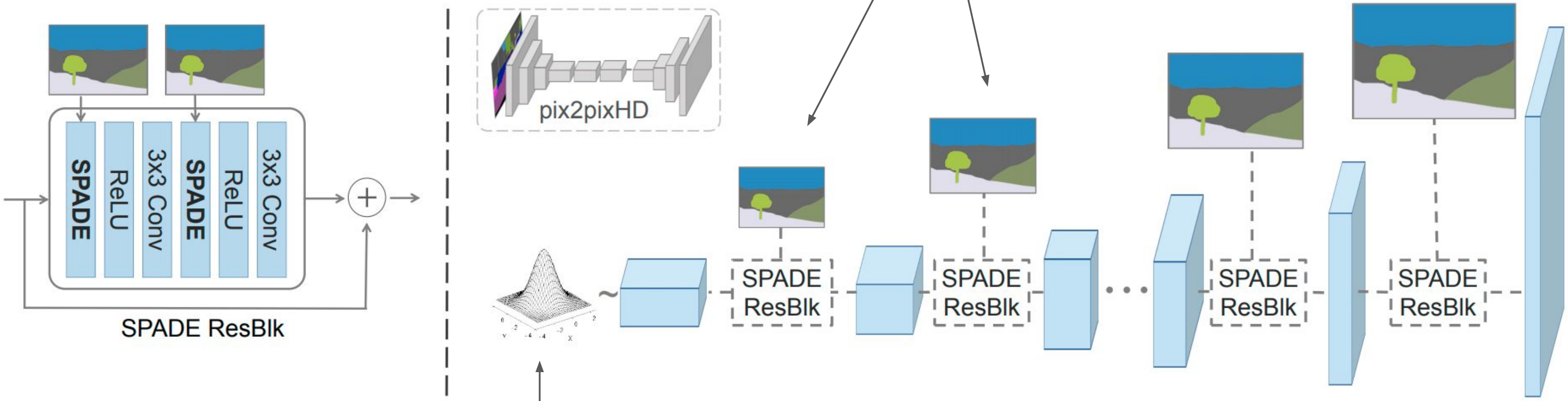
SPADE is generalization of several normalization layers.

- Generalizes to Conditional BatchNorm
- Generalizes to AdaIN if input is image (style transfer) or noise (styleGAN)

Modulation, but no normalization: enjoys benefits of normalization without losing semantic input information

How is SPADE used in a network?

Segmentation map is fed repeatedly to each layer

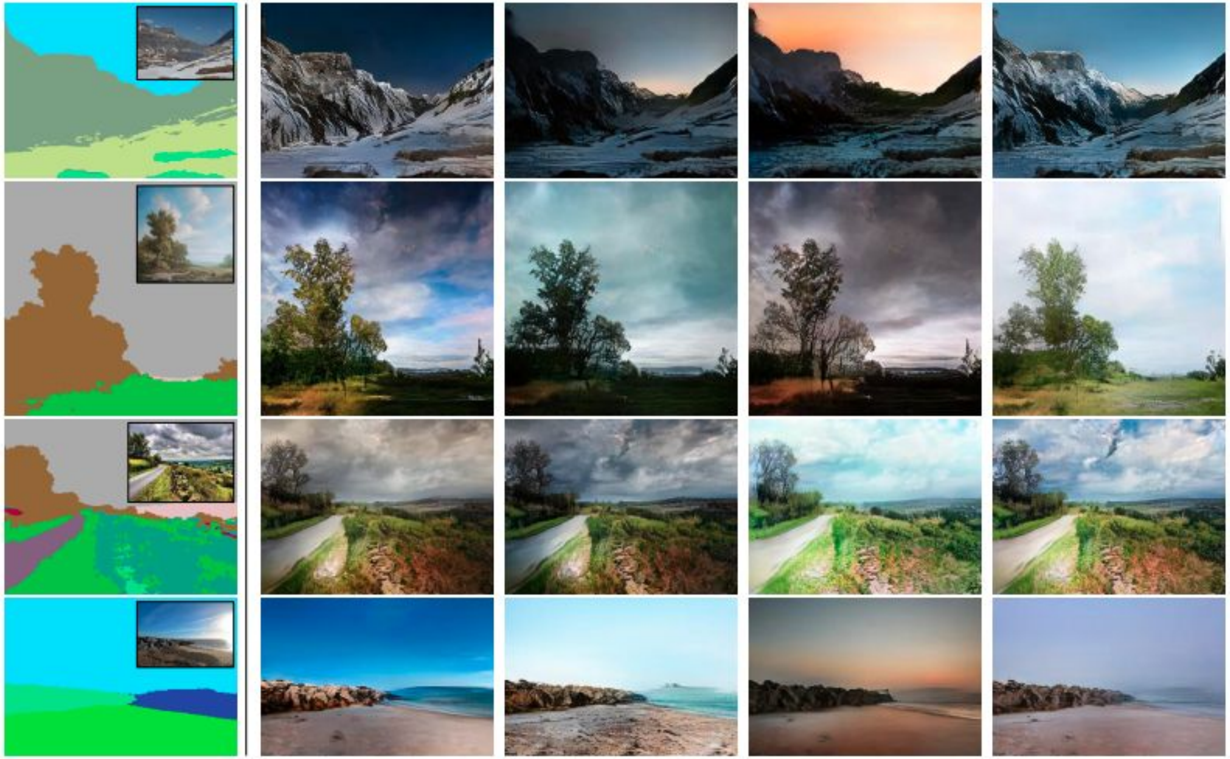


Gaussian noise as input, or style vector derived from an image

Lightweight compared to traditional image to image nets: no downsampling layers

Outputs

Correctly reproduces style, and is capable of generating different versions by sampling different latent vectors



Demo

Amazing live demo (recommended):
http://nvidia-research-mingyuliu.com/gaugan/

DALL-E (2021)

- Latest work from OpenAI. **Creates images from a text prompt**.
- Based on **GPT-3** (but with "just" 12 billion parameters) and **CLIP** (measures similarity between text and image)
- Not a GAN: it's missing the adversarial part. Essentially a big transformer-based generative model.
- Too cool to skip.

TEXT PROMPT

an armchair in the shape of an avocado [...]

AI-GENERATED IMAGES

What can it do?

- Impressive stuff.
- Creating unknown objects by merging attributes (e.g. cube made of porcupine)
- Realistic images in a wide variety of settings (objects, humans, drawings, abstract shapes)
- Generating missing parts of an image
- Zero-shot im to im translation
    - “Same cat as a sketch on bottom”
- Geographic and temporal knowledge

What can it do?

an emoji of a baby penguin wearing a blue hat, red gloves, green shirt, and yellow pants

a plain white cube looking at its own reflection in a mirror. a plain white cube gazing at itself in a mirror.

a painting of a capybara sitting in a field at sunrise

https://openai.com/blog/dall-e/

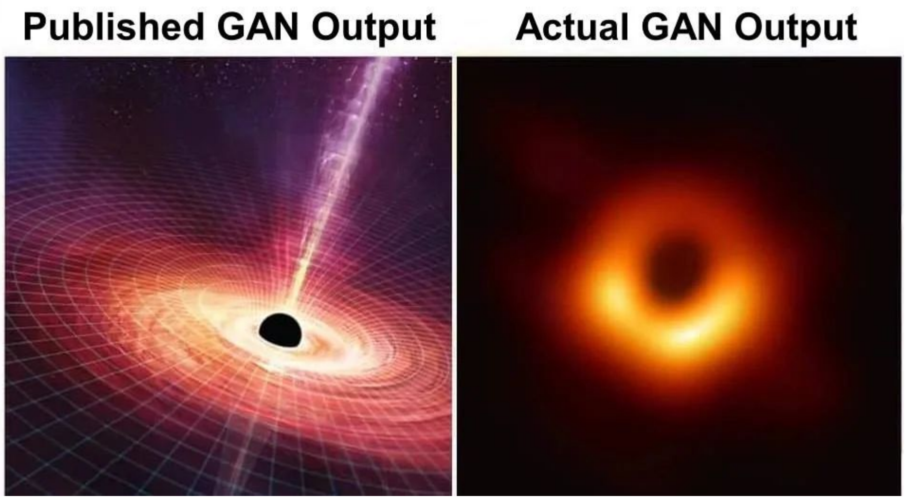
How it works

- Text and Image patches are fed as tokens
  - Words are tokenized normally (vocab size 16k)
  - Images are represented as stream of 1024 tokens (patches) with vocab size 8k
- Transformer models the stream of tokens as single stream of data
- 12B parameter model trained on 250M image-text pairs
- Training is done in two stages:
  - 1. Train discrete VAE to embed images
  - 2. Train 12 billion param transformer to model sequences of text+image
- Mixed precision training, distributed optimization (challenging)

https://arxiv.org/pdf/2102.12092.pdf

What's next for GANs?

- GAN Inversion
- Making them more efficient
- Video GANs
- More language-based fusion
- Better metrics
- Improved outputs in conditional settings



What’s next for GANs?

Dank Learning: Generating Memes Using Deep Neural Networks

Abel L. Peirson V
Department of Physics
Stanford University
alpv95@stanford.edu

E. Meltem Tolunay
Department of Electrical Engineering
Stanford University
meltem.tolunay@stanford.edu

Thank you!

Questions?