

Generative Adversarial Networks (GAN)

Matt Hyatt
Jakob Veselsky



2014

GAN

Goodfellow, Ian, et al. "Generative adversarial networks." Communications of the ACM 63.11 (2020): 139-144

Generative Adversarial Nets (GAN)

- Deep learning models at this point were discriminative
 - Lots of inputs resulting in a class label
 - Rely on backpropagation and drop out algorithms for tuning
 - Largest Impact
- Generative models had also been introduced
 - Difficulty in approximating probabilities that come from likelihood estimation and related strategies

GAN

- Why not use both
 - Generative model could generate results from what it was trained on
 - A discriminative model can then go through and check these results
 - “Adversarial Nets”
 - “The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable from the genuine articles.”

GAN

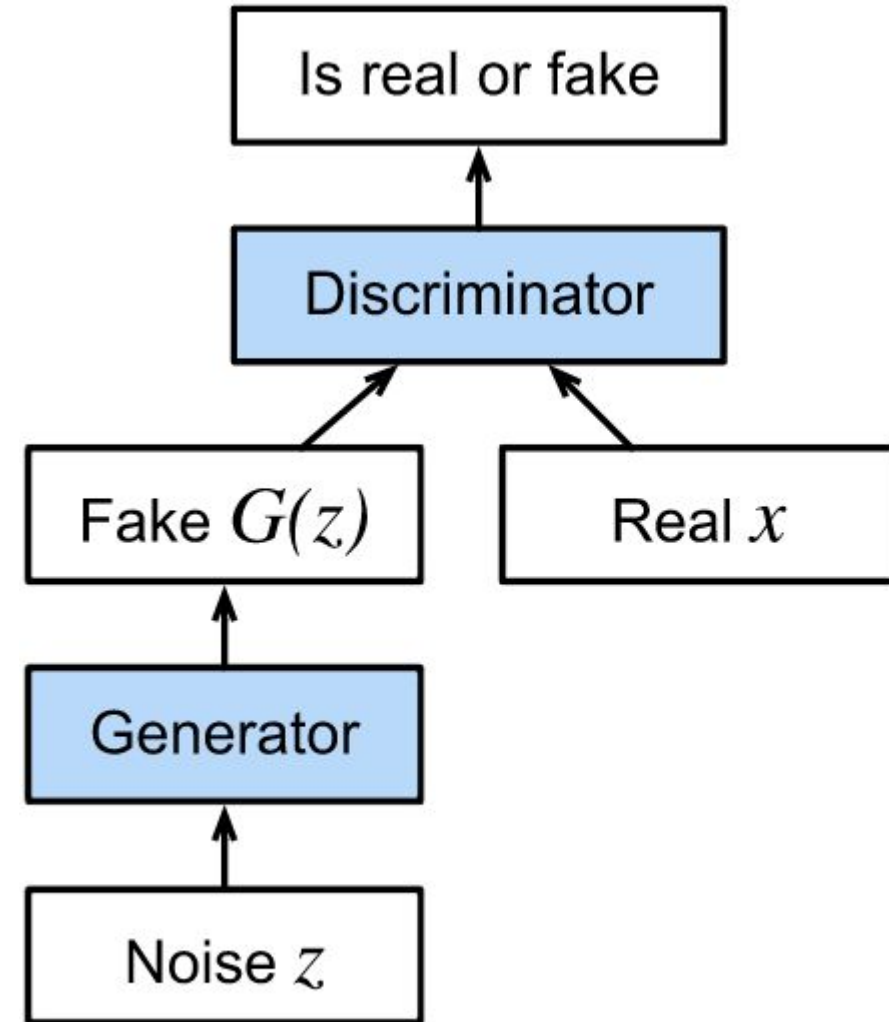
2 networks

- G makes data from noise
- D classifies real data and fake data

Loss

- L_D makes D better at separating $G(z)$ from x
- L_G makes $G(z)$ look more like x

As D gets harder to fool it provides better feedback to G for making realistic samples



GAN

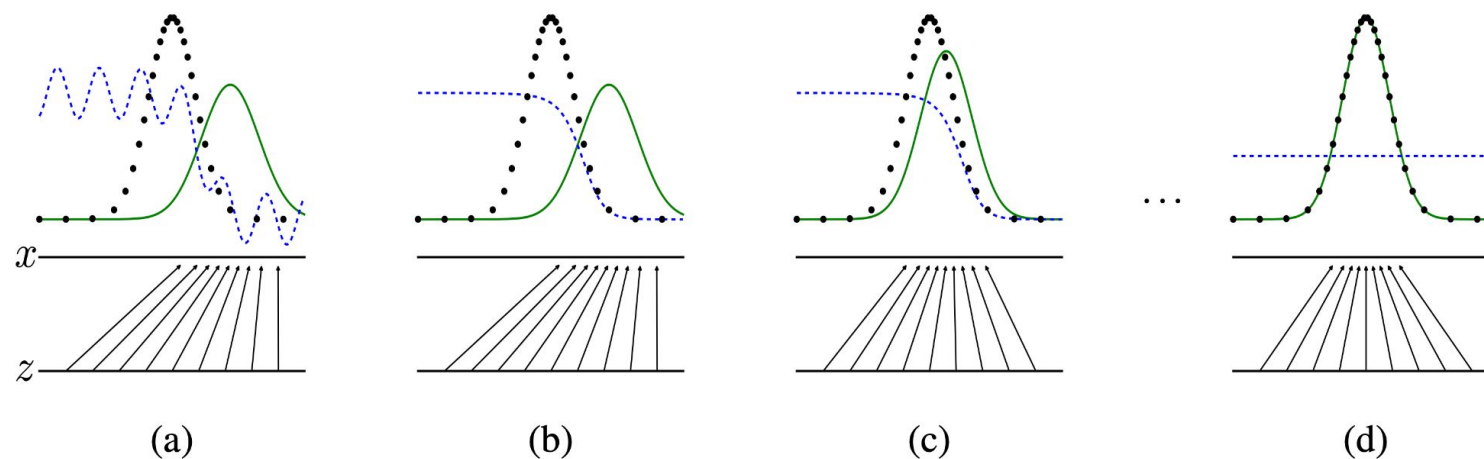
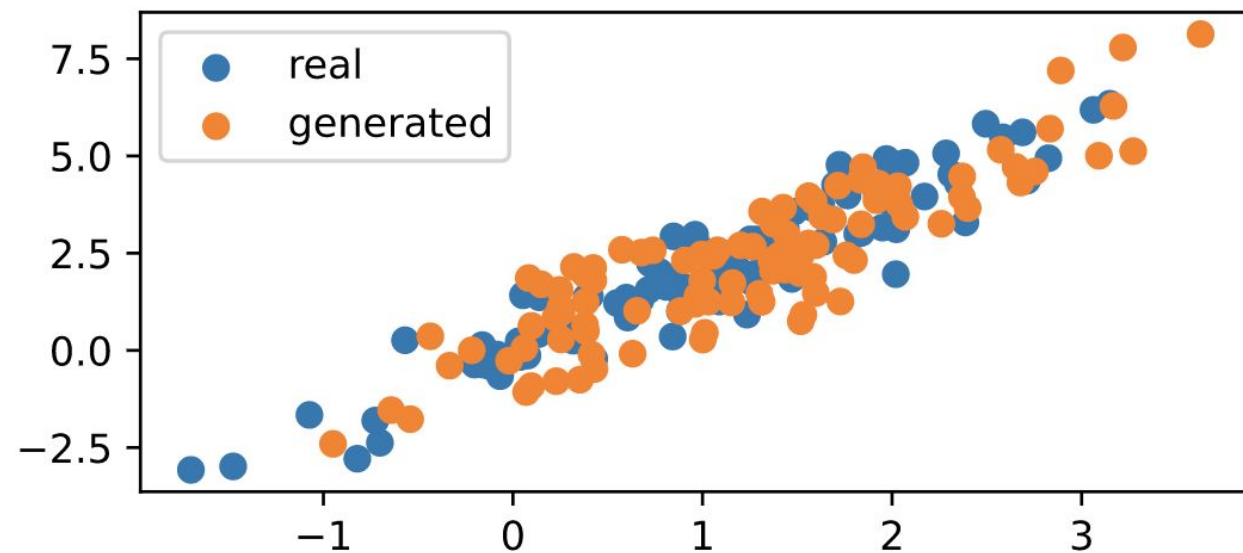
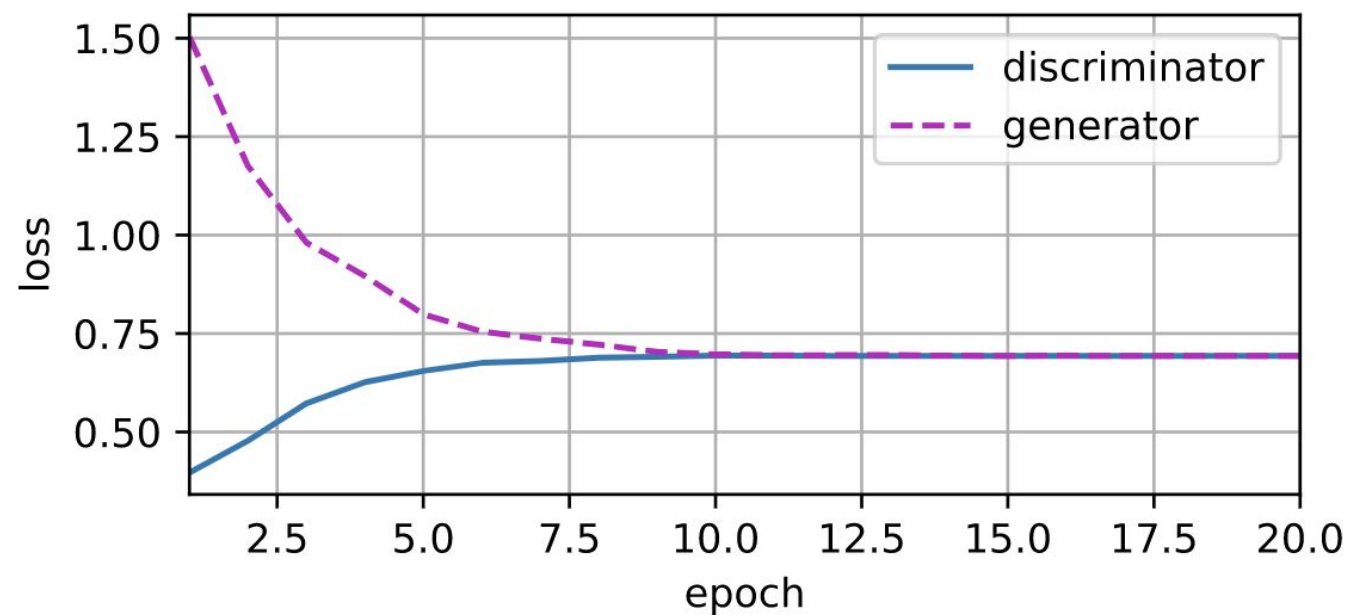


Figure 1: Generative adversarial nets are trained by simultaneously updating the discriminative distribution (D , blue, dashed line) so that it discriminates between samples from the data generating distribution (black, dotted line) $p_{\mathbf{x}}$ from those of the generative distribution p_g (G) (green, solid line). The lower horizontal line is the domain from which \mathbf{z} is sampled, in this case uniformly. The horizontal line above is part of the domain of \mathbf{x} . The upward arrows show how the mapping $\mathbf{x} = G(\mathbf{z})$ imposes the non-uniform distribution p_g on transformed samples. G contracts in regions of high density and expands in regions of low density of p_g . (a) Consider an adversarial pair near convergence: p_g is similar to p_{data} and D is a partially accurate classifier. (b) In the inner loop of the algorithm D is trained to discriminate samples from data, converging to $D^*(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_g(\mathbf{x})}$. (c) After an update to G , gradient of D has guided $G(\mathbf{z})$ to flow to regions that are more likely to be classified as data. (d) After several steps of training, if G and D have enough capacity, they will reach a point at which both cannot improve because $p_g = p_{\text{data}}$. The discriminator is unable to differentiate between the two distributions, i.e. $D(\mathbf{x}) = \frac{1}{2}$.

GAN Metrics

SSIM

FID



2014

Conditional GAN

Mirza et al, Conditional generative adversarial nets. ArXiv preprint (<https://arxiv.org/abs/1411.1784>), 2014.

Conditional GAN (CGAN)

- Generative Adversarial Nets with conditional
 - Add conditions to both the generator and discriminator
- Novelty Claims
 - Can generate specific digits
 - Can be used in other applications
 - Image tagging

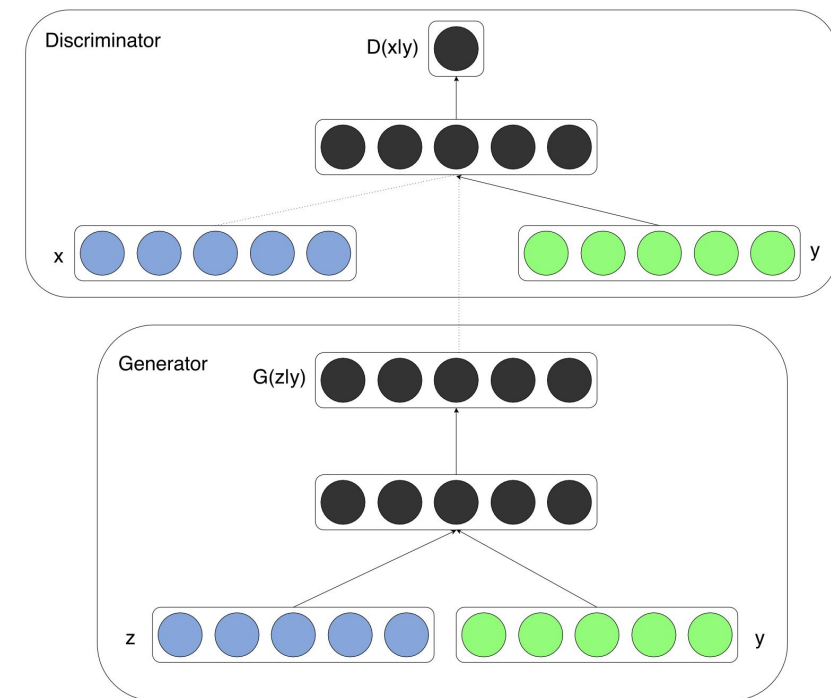


Figure 1: Conditional adversarial net

CGAN

- “In an unconditioned generative model, there is no control on modes of the data being generated. However, by conditioning the model on additional information it is possible to direct the data generation process.”

CGAN

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x}|\mathbf{y})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z}|\mathbf{y})))]$$

CGAN Results



Figure 2: Generated MNIST digits, each row conditioned on one label

CGAN Results




	User tags + annotations	Generated tags
	montanha, trem, inverno, frio, people, male, plant life, tree, structures, transport, car	taxi, passenger, line, transportation, railway station, passengers, railways, signals, rail, rails
		chicken, fattening, cooked, peanut, cream, cookie, house made, bread, biscuit, bakes
	water, river	creek, lake, along, near, river, rocky, treeline, valley, woods, waters
	people, portrait, female, baby, indoor	love, people, posing, girl, young, strangers, pretty, women, happy, life

Table 2: Samples of generated tags

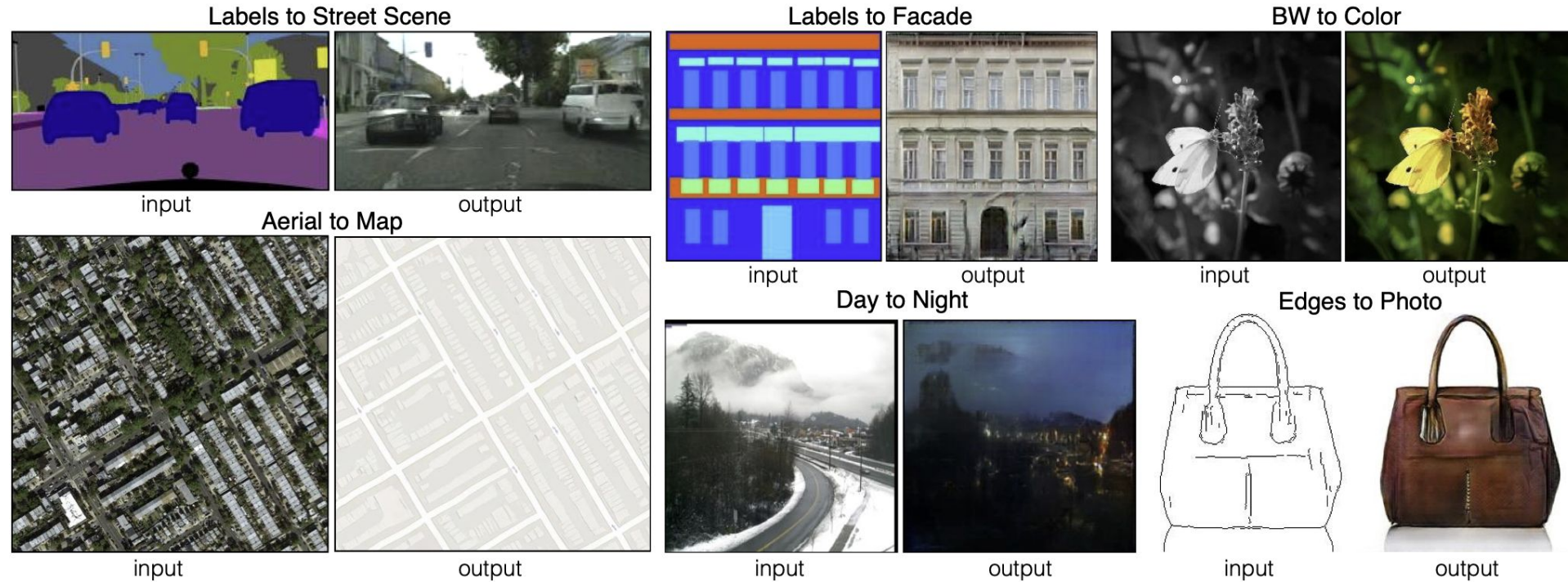
Questions?

2017, CVPR

Pix2pix

Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017

Pix2pix Goal



“predict pixels from pixels”

Pix2pix Novelty

“Our primary contribution is to demonstrate that on a wide variety of problems, conditional GANs produce reasonable results.”

Pix2pix

- CNN's
 - Hand Tuned Loss Function
 - “Coming up with loss functions that force the CNN to do what we really want – e.g., output sharp, realistic images – is an open problem and generally requires expert knowledge.”
- GAN's
 - cGANS
 - Learn A Conditional Loss Function

Pix2pix Approach

- Input and Output will share some things
- Skip Connections
 - Allows data that will be shared to be pass without computation.

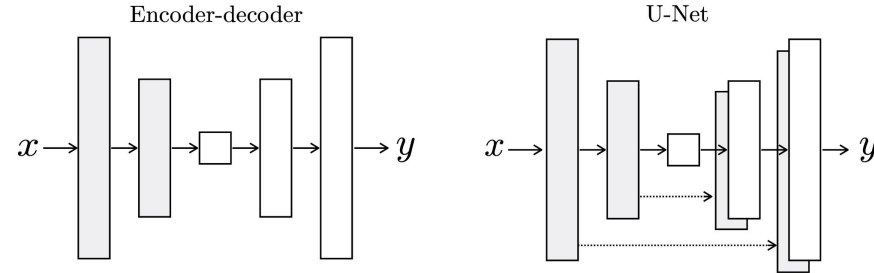


Figure 3: Two choices for the architecture of the generator. The “U-Net” [50] is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.



Pix2pix Approach

- PatchGAN
 - Classifier
 - only penalizes structure at the scale of image patches

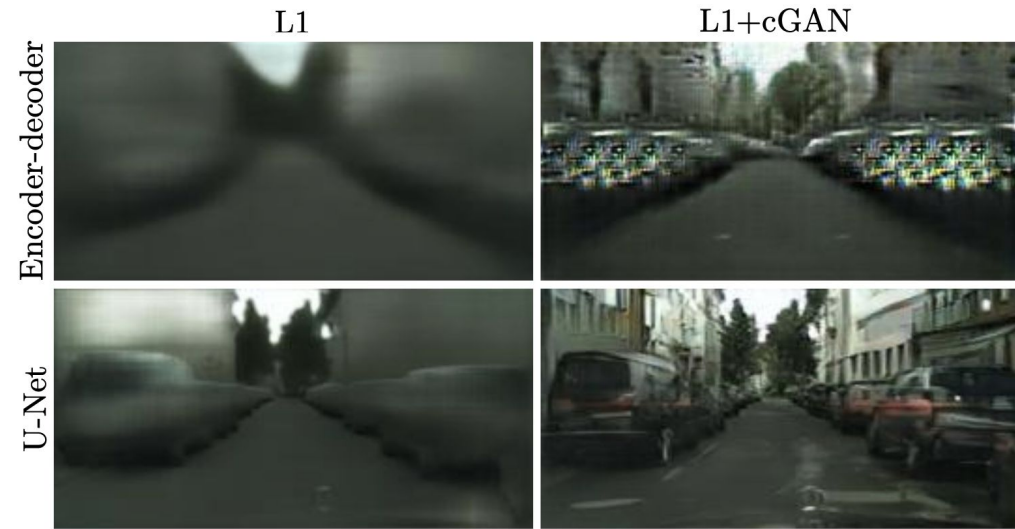


Figure 5: Adding skip connections to an encoder-decoder to create a “U-Net” results in much higher quality results.

Approach

PatchGAN



Results

- Semantic Labels to photos
- Architectural labels to photos
- Map to aerial photo
- BW to color
- Edges to photo
- Sketch to photo
- Day to night
- Thermal to color
- Missing pixels to not

Results



Figure 8: Example results on Google Maps at 512x512 resolution (model was trained on images at 256 × 256 resolution, and run convolutionally on the larger images at test time). Contrast adjusted for clarity.

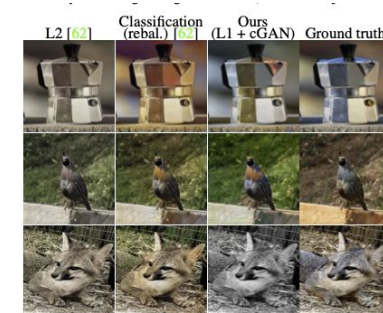


Figure 9: Colorization results of conditional GANs versus the L2 regression from [62] and the full method (classification with re-balancing) from [64]. The cGANs can produce compelling colorizations (first two rows), but have a common failure mode of producing a grayscale or desaturated result (last row).

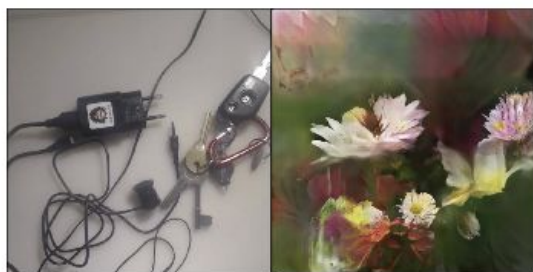


Figure 12: *Learning to see: Gloomy Sunday*: An interactive artistic demo developed by Memo Akten [8] based on our pix2pix codebase. Please click the image to play the video in a browser.

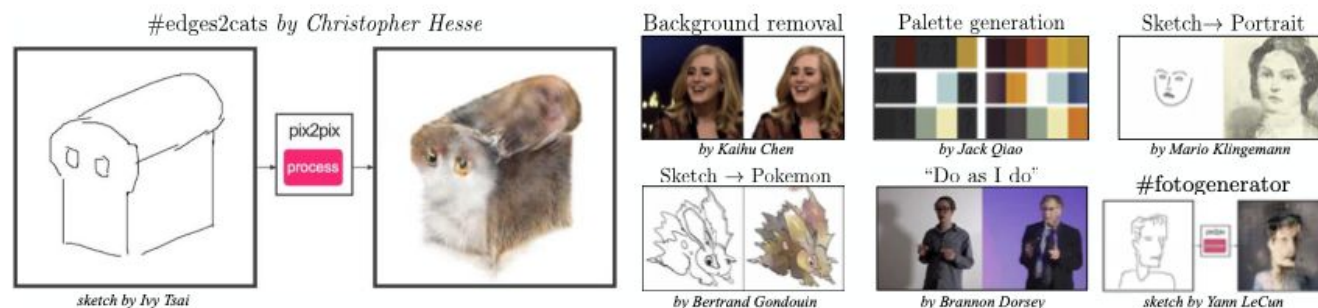


Figure 11: Example applications developed by online community based on our pix2pix codebase: #edges2cats [3] by Christopher Hesse, Background removal [6] by Kaihu Chen, Palette generation [5] by Jack Qiao, Sketch → Portrait [7] by Mario Klingemann, Sketch → Pokemon [1] by Bertrand Gondouin, “Do As I Do” pose transfer [2] by Brannon Dorsey, and #fotogenerator by Bosman et al. [4].

Questions?

2017, ICCV

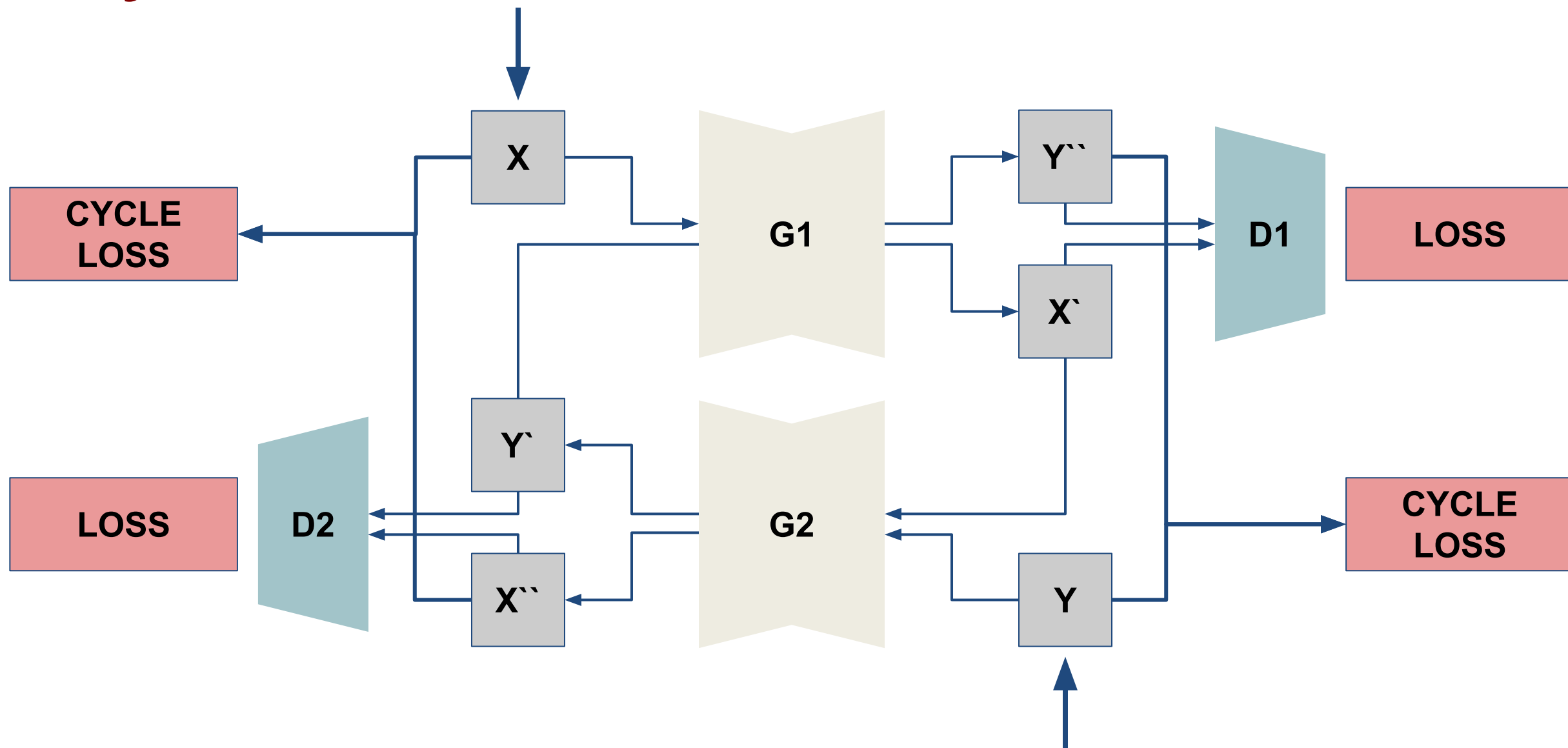
CycleGAN

Zhu et al, Unpaired image-to-image translation using cycle-consistent adversarial networks. IEEE International Conference on Computer Vision (ICCV), 2017.

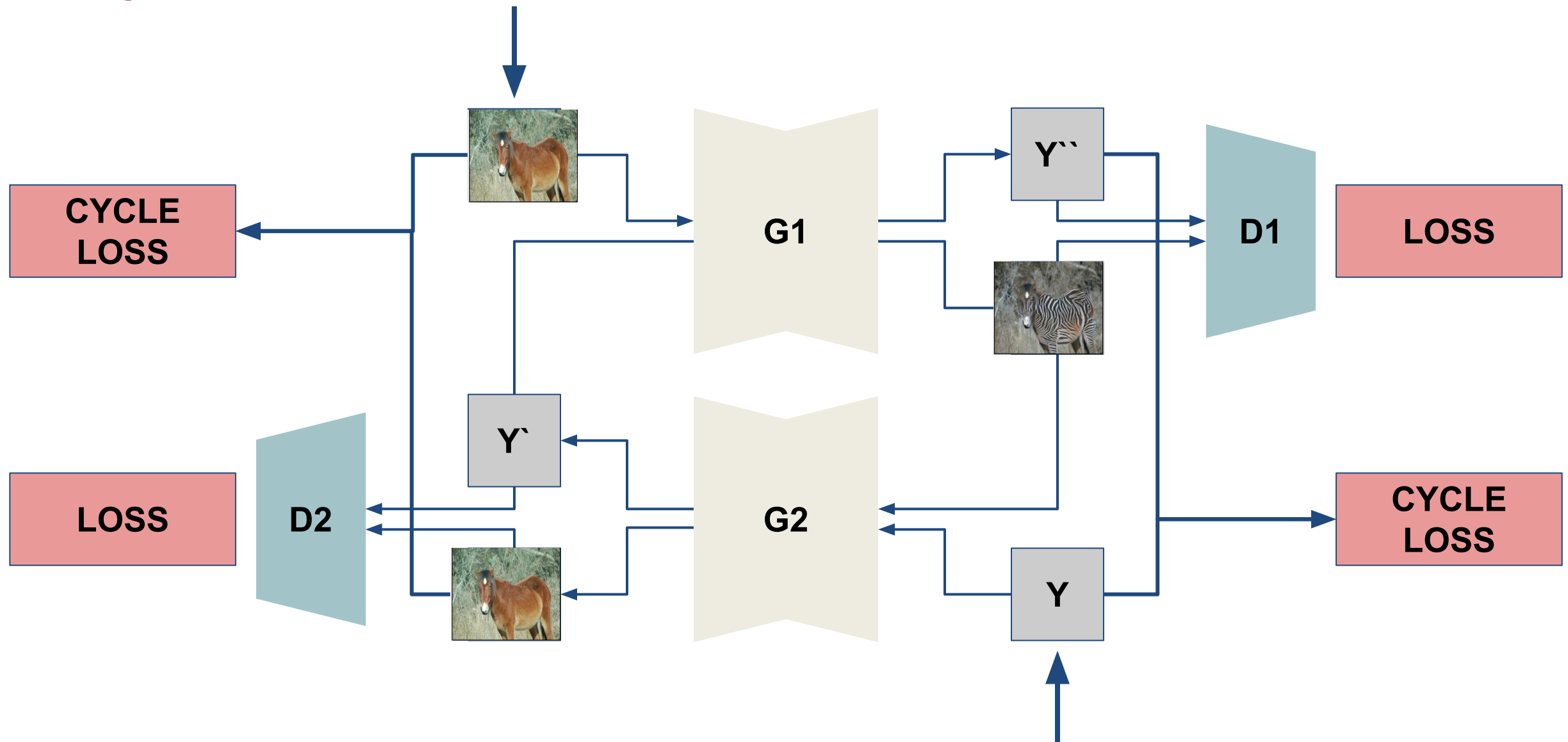
CycleGAN

- how do define a mapping from one distribution to another in an unsupervised manner?
 - 1-1 annotation of images is expensive and sometimes not feasible
- cycle consistent loss
 - if a gan turns $X \rightarrow X'$ and turns $X' \rightarrow X''$
 - then X'' should look like the original image X

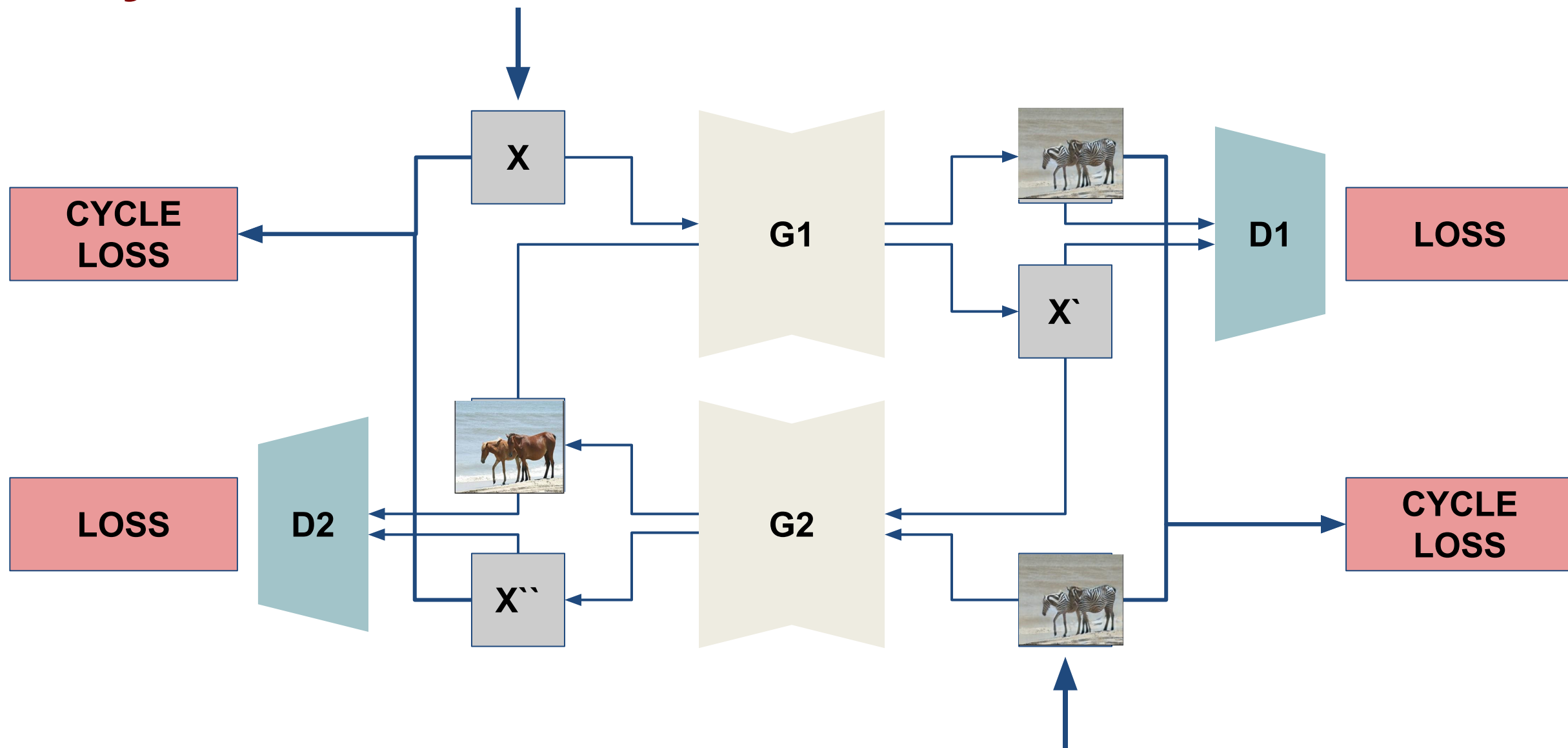
CycleGAN



CycleGAN



CycleGAN



CycleGAN Results



CycleGAN Results



CycleGAN Results



Questions?

2019, CVPR

StyleGAN

Karras, T., Laine, S., Aila, T. A style-based generator architecture for generative adversarial networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

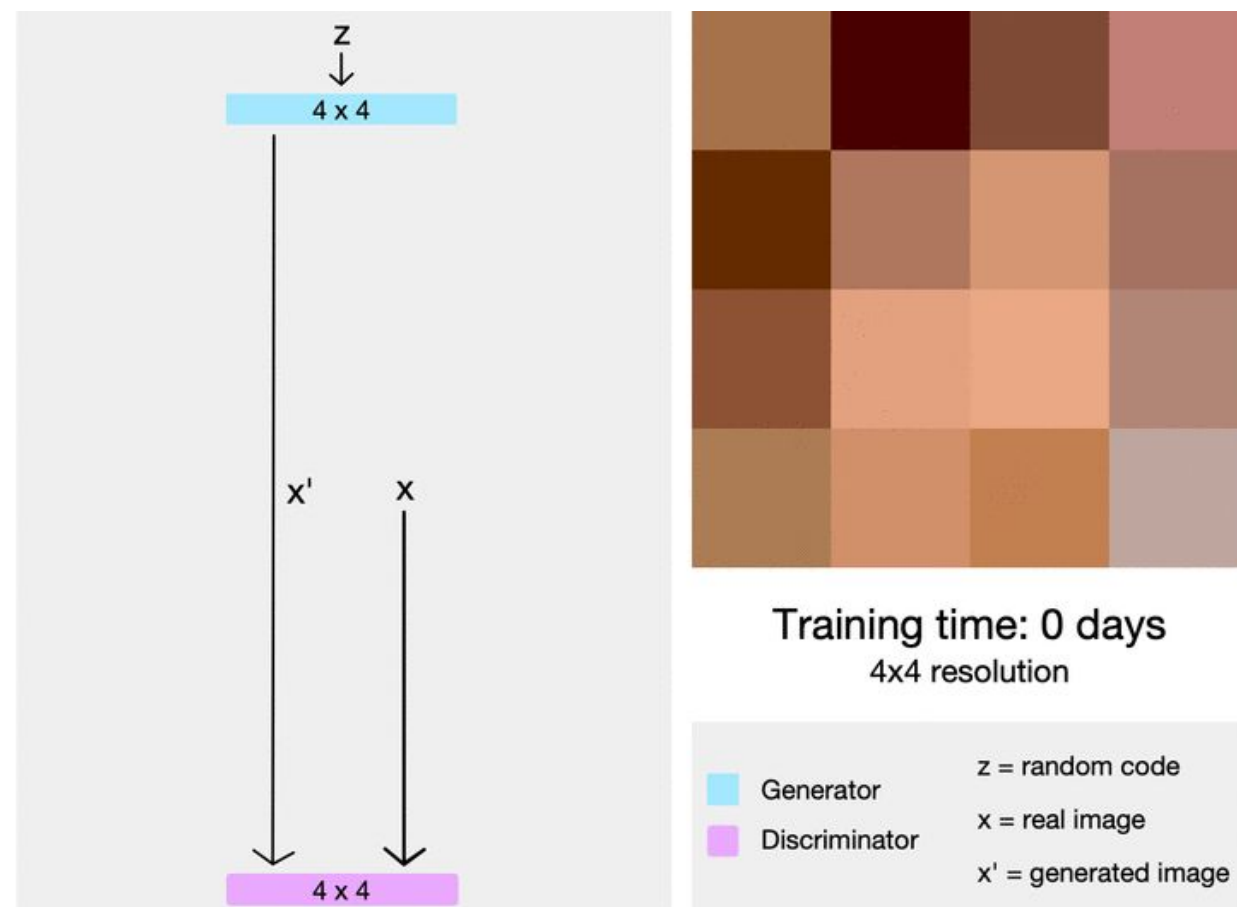
StyleGAN

- How can we get better control of styles?
 - disentanglement
 - path length
1. progressive growing
 2. learned constant input
 3. style mapping network
 4. adaptive instance normalization
 5. mixing regularization

StyleGAN Architecture (Progressive GAN)

Progressive Growing of GANs

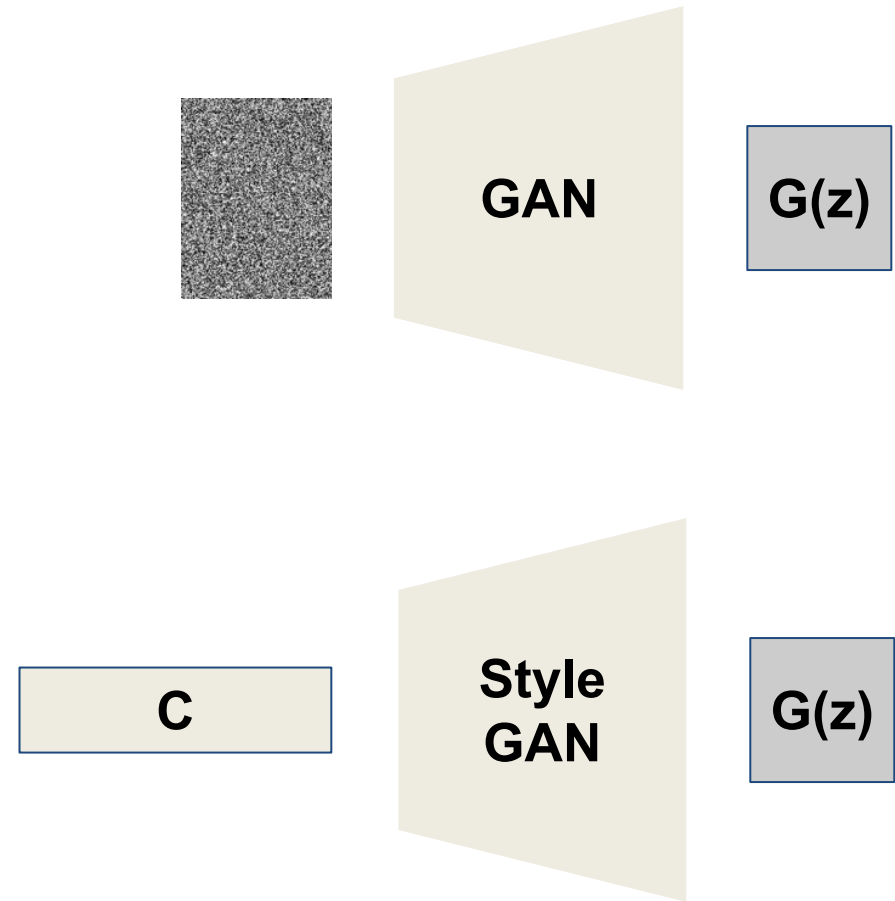
- learning simple problems helps with complex problems
- more stable results by building large models from converged smaller models
- trains in stages



StyleGAN Architecture (Noise)

Noise sampling

- Learning the starting vector helps the model become more stable
- Outputs are still unique because of noise added later



StyleGAN Architecture (Mapping Network)

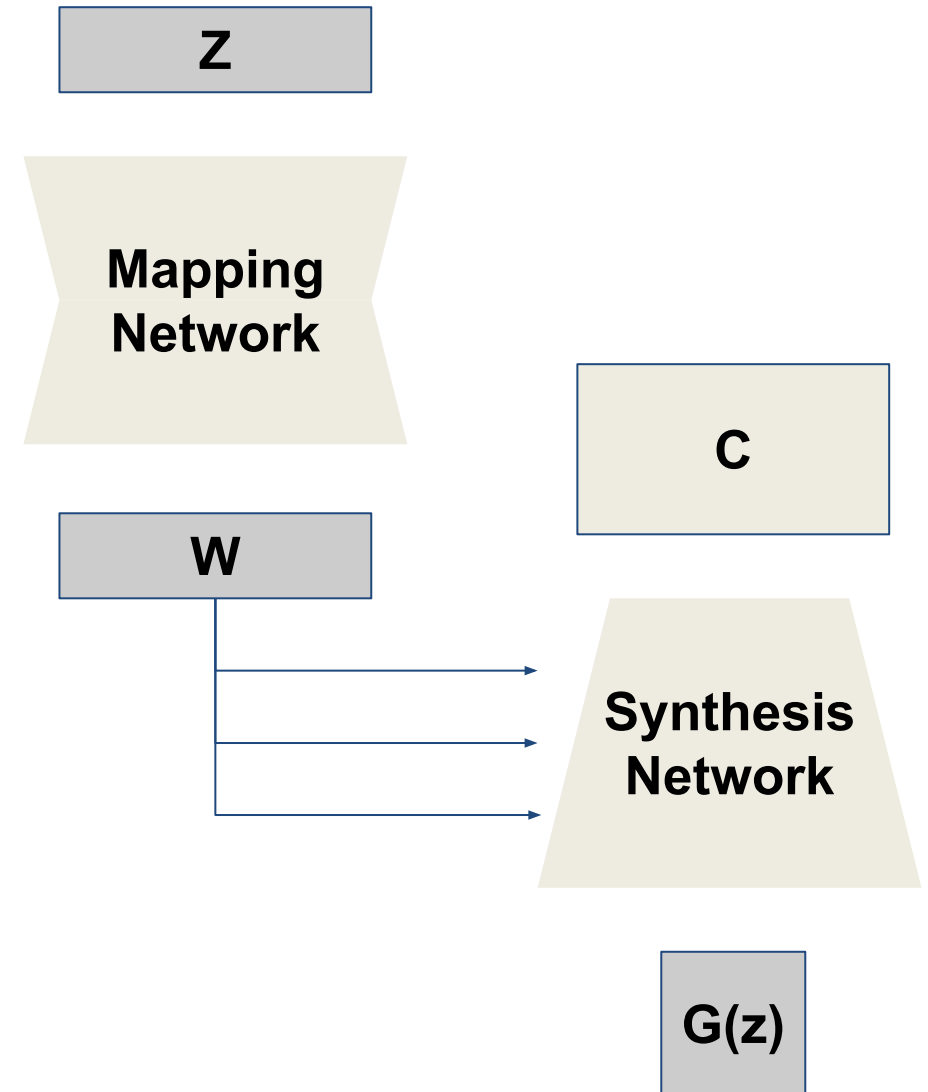
Making good images and controlling how they look are two different tasks

Mapping Network

- learn to control the style

Synthesis Network

- learn to make good images



StyleGAN Affine Transform

0 To understand mathematically what a linear and affine transformations are, read Pratik_Katte's answer.

But in machine learning what are called linear layers are actually mathematically affine transformations $\mathbb{R}^n \rightarrow \mathbb{R}^m$ (i.e. transformations on the features as a vector). What are called affine layers are actually n affine transformations $\mathbb{R} \rightarrow \mathbb{R}$ (i.e. transformations on the coordinates of the features)

```
class Affine(nn.Module):
    def __init__(self, dim):
        super().__init__()
        self.alpha = nn.Parameter(torch.ones(dim))
        self.beta = nn.Parameter(torch.zeros(dim))

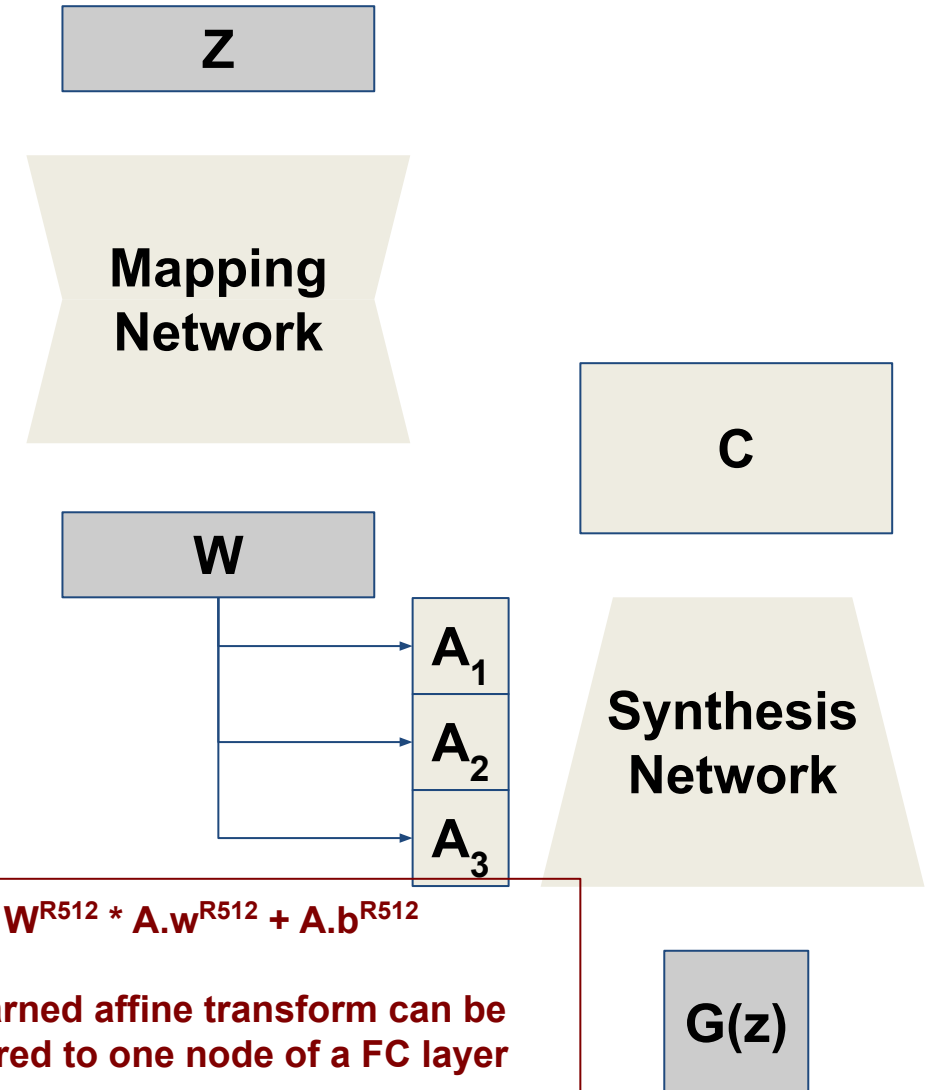
    def forward(self, x):
        return self.alpha * x + self.beta
```

https://github.com/facebookresearch/deit/blob/main/resmlp_models.py

Where as the pseudo-code for linear layers might go

```
class Linear(Module):
    def __init__(self, in_features: int, out_features: int):
        super(Linear, self).__init__()
        self.weight = Parameter(torch.rand((out_features, in_features)))
        self.bias = Parameter(torch.rand(out_features))
```

<https://datascience.stackexchange.com/questions/13405/what-is-affine-transformation-in-regard-to-neural-networks>



StyleGAN Adaptive Instance Normalization

Adaptive Instance Normalization (AdaIN)

- Integrates the “style” of one image with that of another by carrying its mean and std from one to the other

$$\text{AdaIN}(x, y) = \sigma(y) \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y) \quad (8)$$

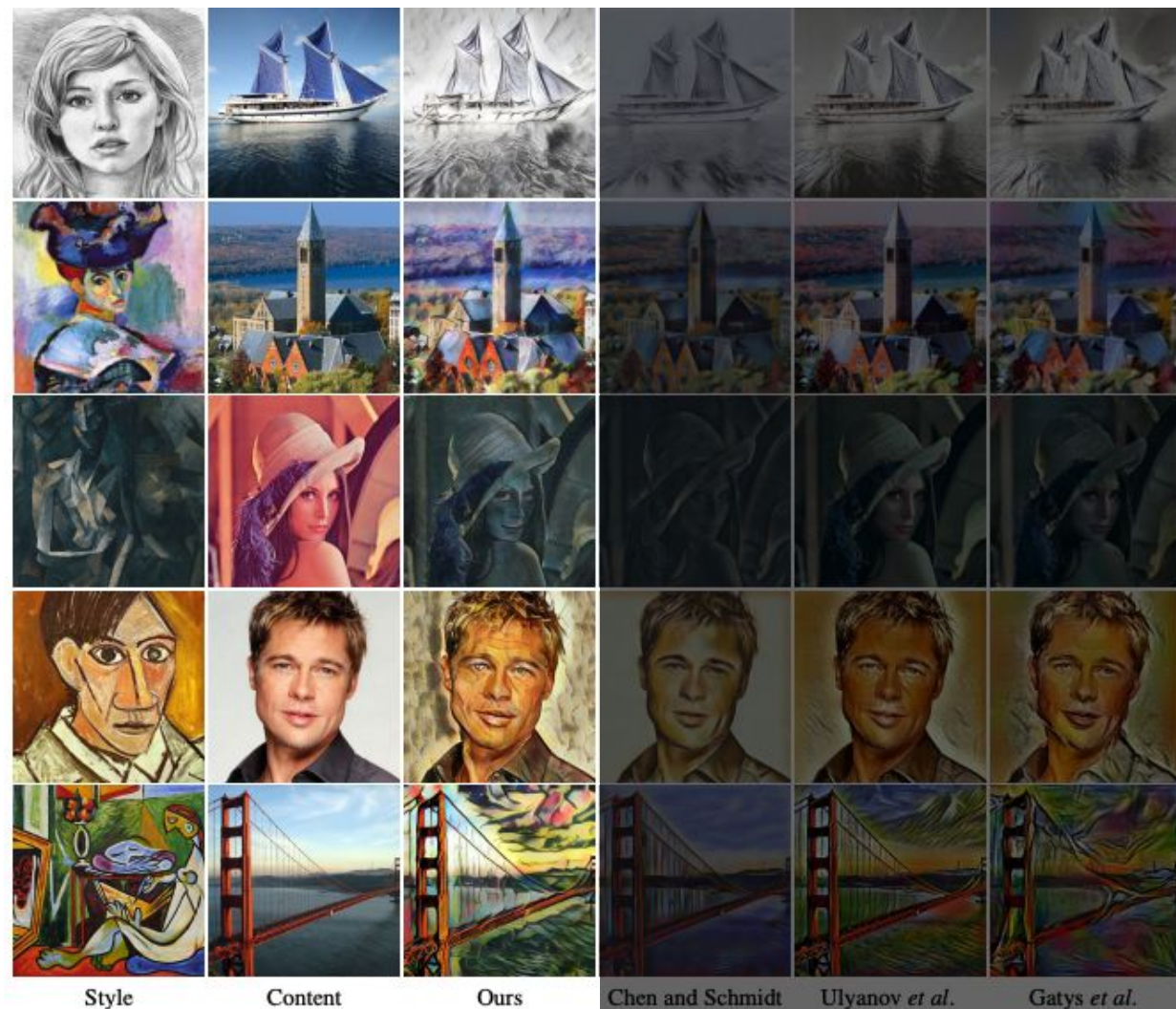
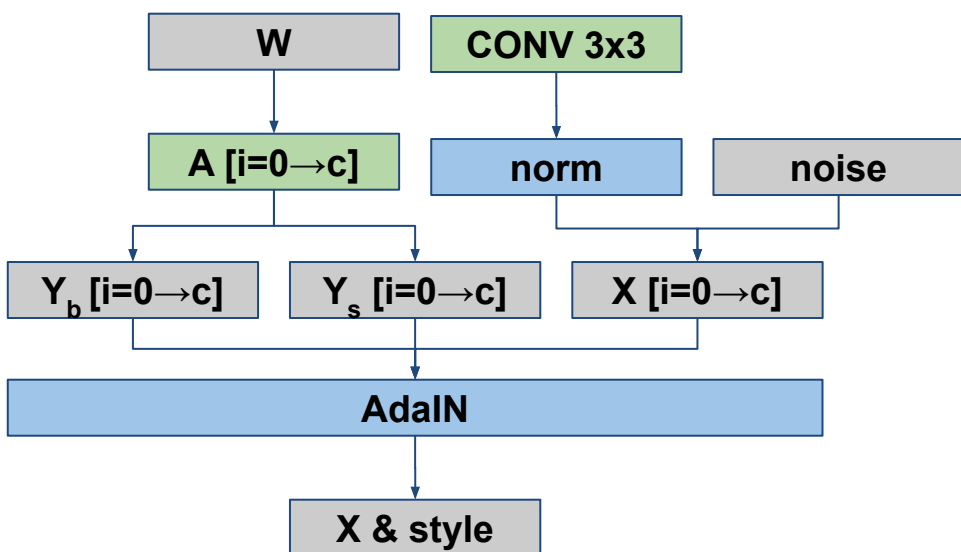


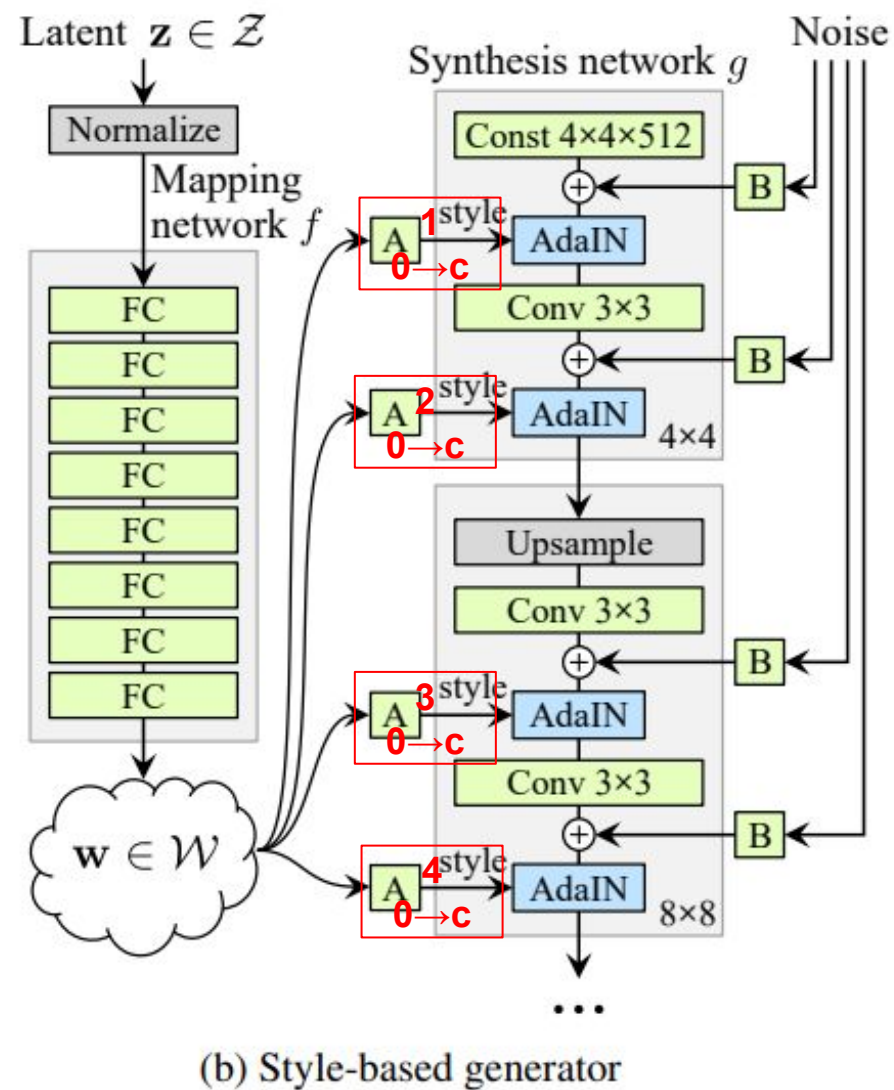
Figure 4. Example style transfer results. All the tested content and style images are never observed by our network during training.

StyleGAN Architecture (AdaIN)

Adaptive Instance Normalization (AdaIN)



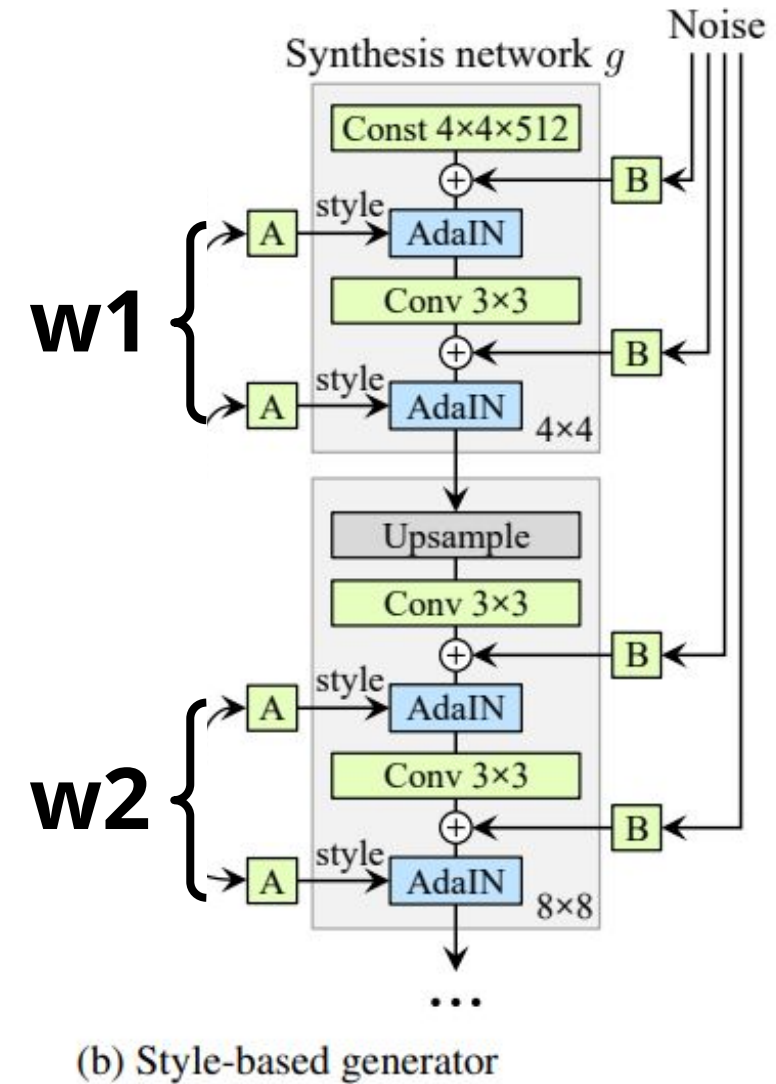
$$\text{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}, \quad (1)$$



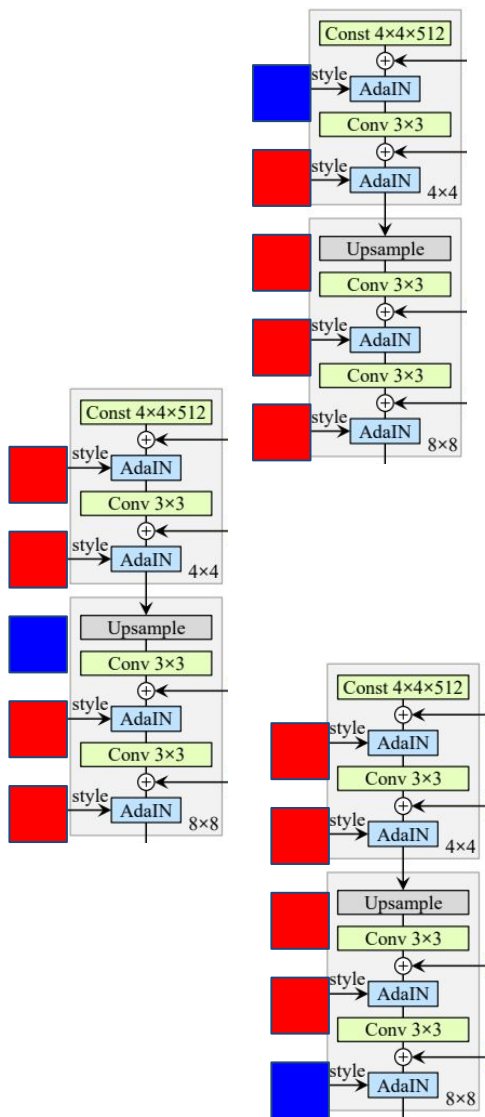
StyleGAN Training (Style Mixing)

Style Mixing

- 2 latent vectors $\mathbf{z}_1, \mathbf{z}_2$
- 2 style vectors $\mathbf{w}_1, \mathbf{w}_2$
- teaches the model that styles are not correlated



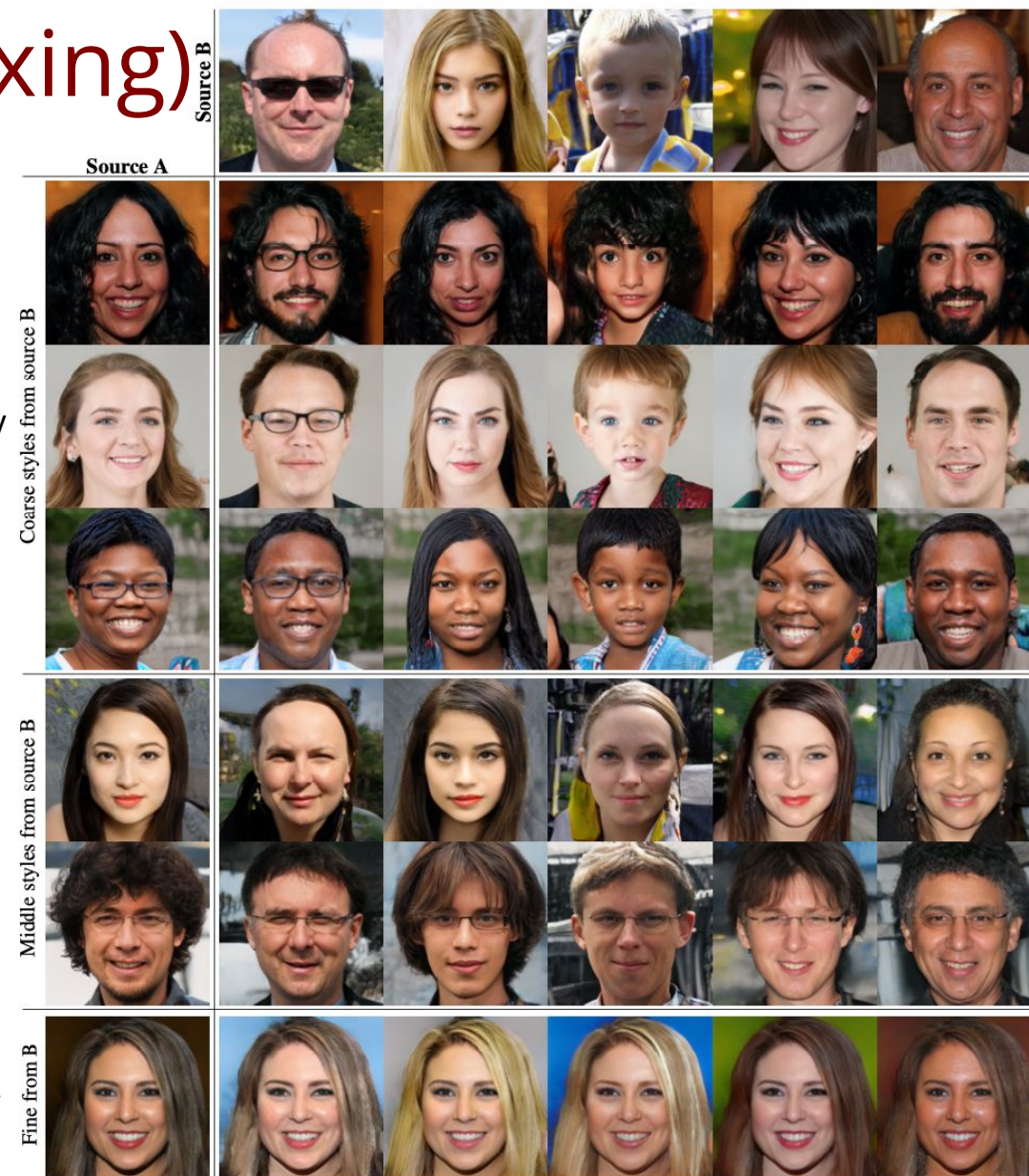
StyleGAN Training (Style Mixing)



W_B for first few layers only

W_B for middle layers only

W_B for last few layers only

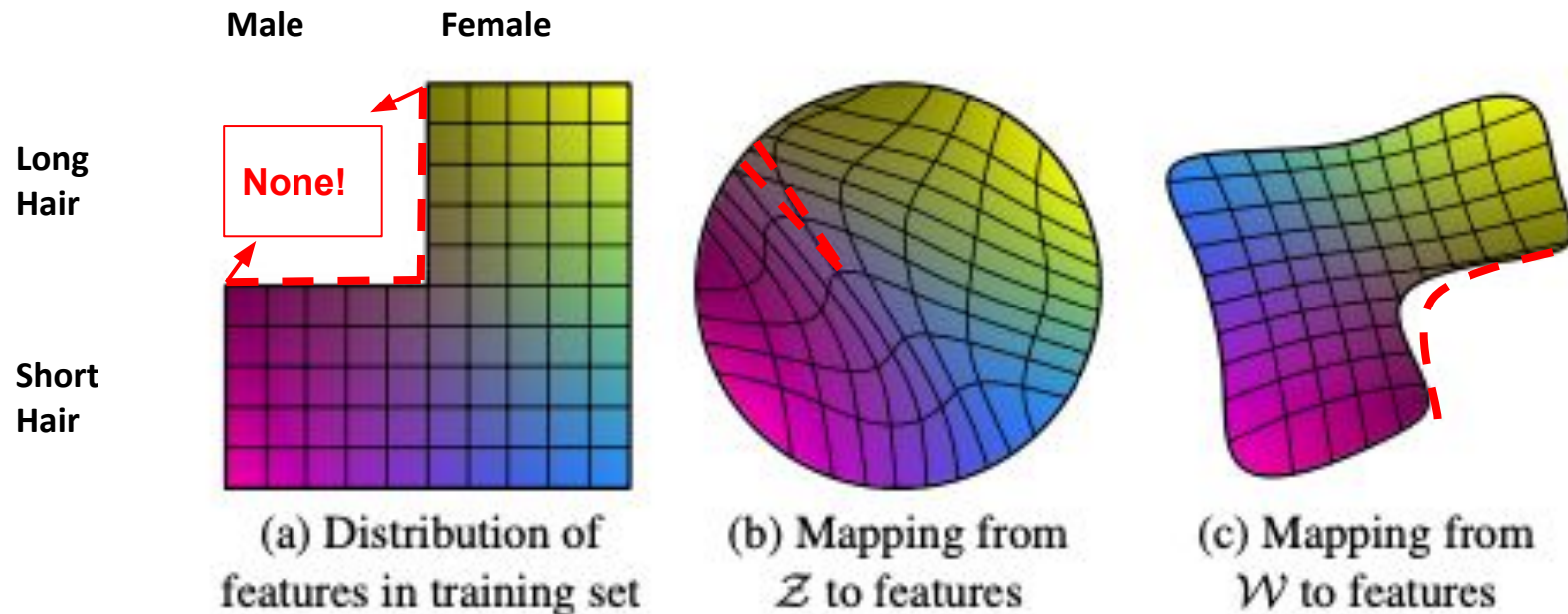


StyleGAN Disentanglement

without the mapping network,
 Z into a feature where the forbidden
combination doesn't exist

...
 W immediately changes from
purple \rightarrow yellow

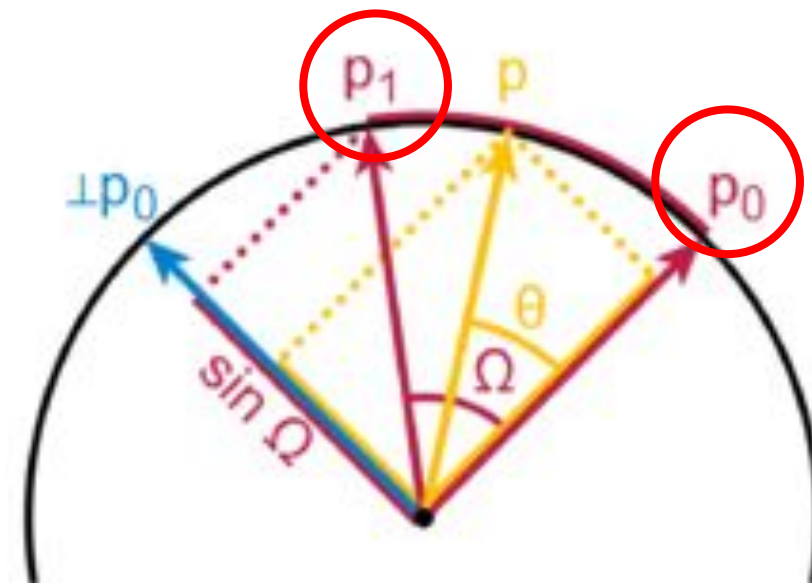
with mapping, $M(Z) = W$
 W is a space where you can
smoothly move around the
forbidden combination
although you may not get samples
which accurately represent the
under-represented combo



StyleGAN Metrics

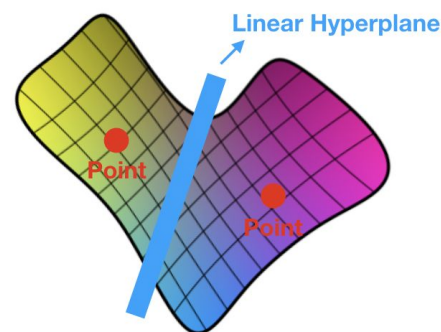
Perceptual Path Length (PPL)

As you walk small steps along the latent space, how much does the image visually change? Measured by distance of VGG16 embeddings. *(lower is better)*

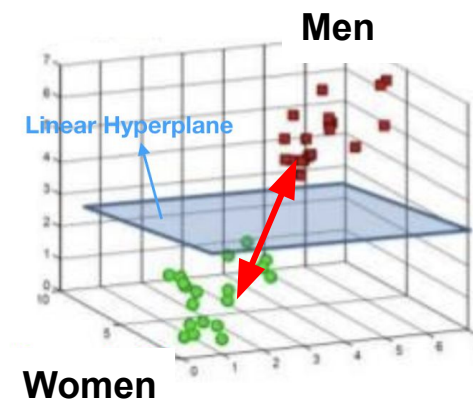


Separability

How well can you linearly separate visual qualities in the latent space? How much additional info is required to separate images with a SVM? *(lower is better)*



Latent space



Women

StyleGAN Interpolation via truncation

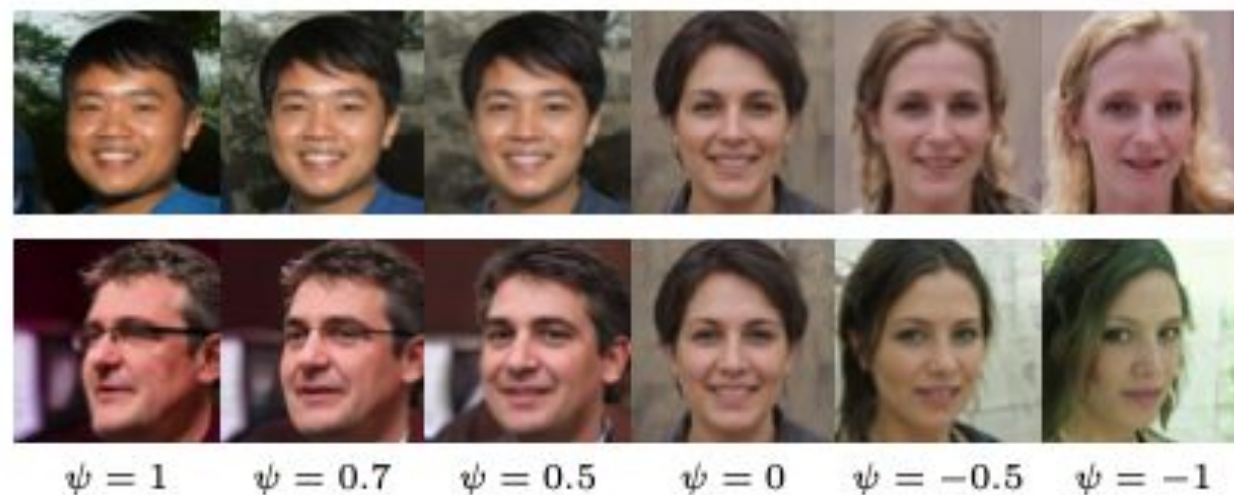
Find W center of mass $E(W)$

- $E(W)$ = the average face
- scale face towards or away from the mean

$$w' = E(W) + \psi (w - E(W))$$

$\psi = 1$: original w vector

$\psi = -1$: the “anti-face”



Questions?

Response Questions

- What ethical and moral complications arise with the usage of GANs?
- Application
 - NLP?



Preparing people to lead extraordinary lives