Interpreting the Latent Space of GANs for Semantic Face Editing

Authors: Yujun Shen, Jinjin Gu, Xiaoou Tang, Bolei Zhou

Presented by Anat Kleiman, Gustaf Ahdritz, Cynthia Chen, and Luke Bailey

Presentation Roadmap

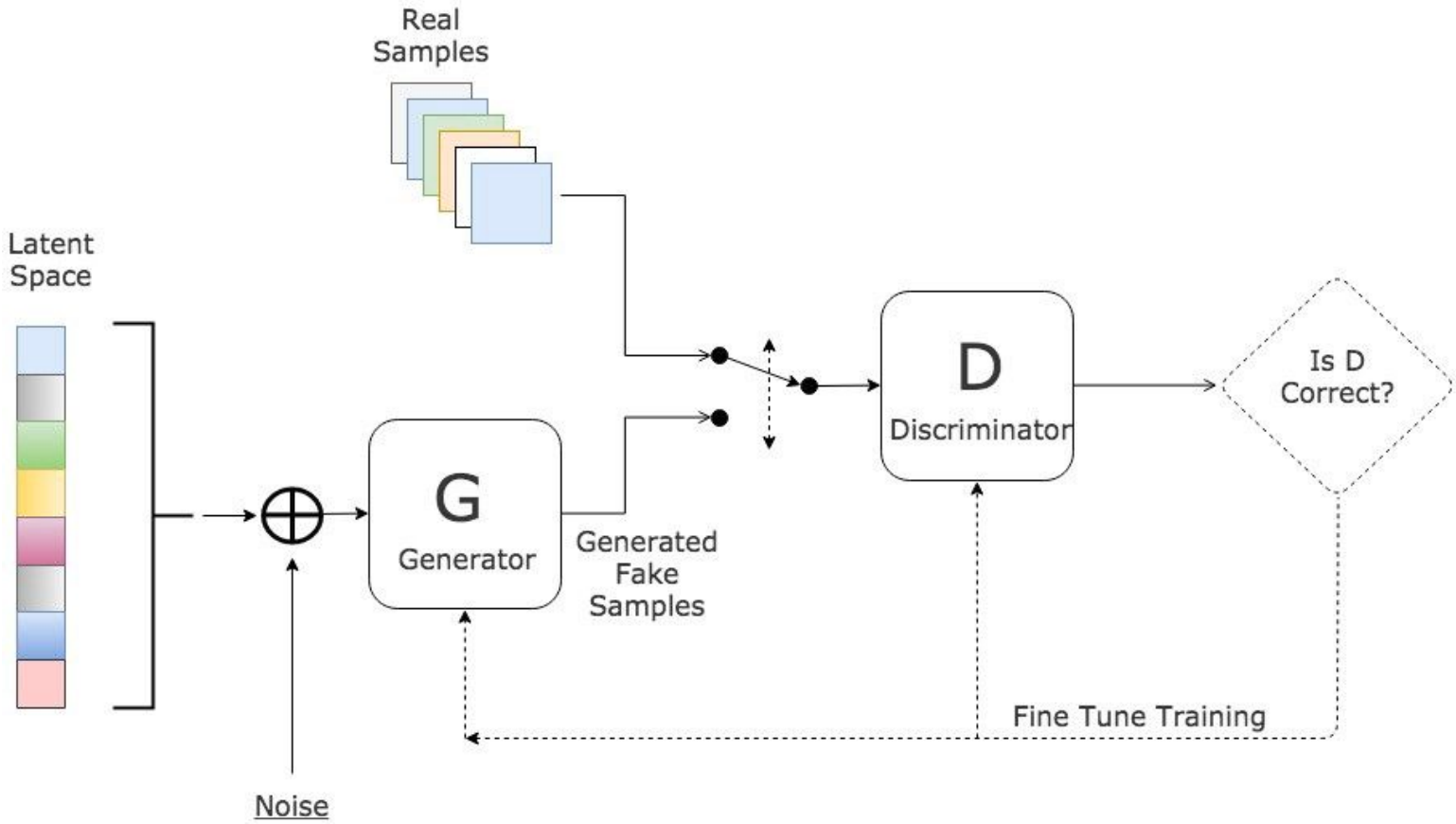
- **Introduction**
- **Method**
- **Experiments**

Background: What are GANs?

- **Generative Adversarial Network (GAN)**: generative models that generate incredibly realistic synthetic data that mimic underlying training data
    - **Generator**: tries to generate samples similar to training data
    - **Discriminator**: tries to distinguish between generated samples (fake) and training data (real)

- **Latent representation:** a low-dimensional representation of an image (eg: a 5-dimensional vector)
- **Latent space:** the set of all possible latent representations
    - Through adversarial training, the generator learns a mapping (“latent distribution”) from points in the latent space to real images

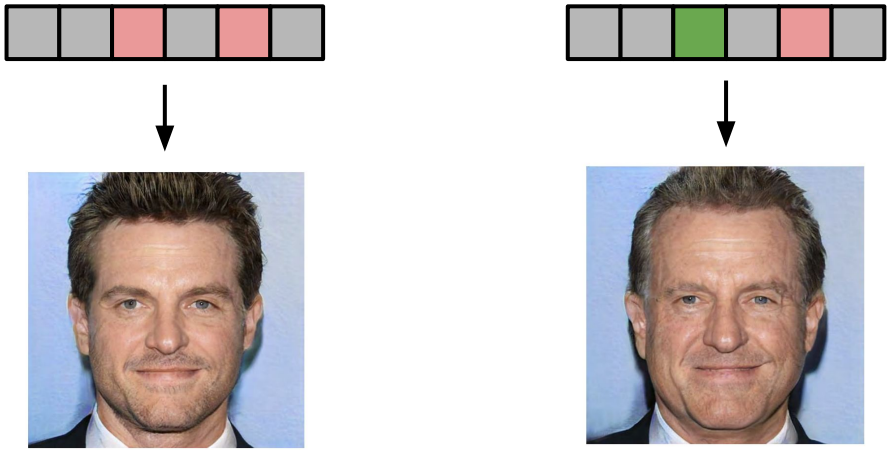
Source: https://paperswithcode.com/method/gan



Source: https://paperswithcode.com/method/gan

Background: GANs for Image Editing

- **Latent code:** the version of an image in the latent space
  - GAN inversion: identifying the latent code for a given image such that the generator could construct the original image
- **Semantic editing**: editing the latent code to manipulate some features of the resulting image, such that only the desired features are changed



Related Work

- Exploring latent spaces in GANs
  - Laine et al: exploring making the output smoothly vary from one synthesis to another in the latent space
  - Radford et al observe the *vector arithmetic property*
    - Can be thought of as an additive property in the embedding space, e.g.:

      `vector("King") - vector("Man") + vector("Woman") = vector("Queen")`
  - Applications in camera motion (Jahanian et al.), scene synthesis (Yang et al.), memorability (Goetschalckx et al.)
- Semantic face editing with GANs
  - Previous methods involved carefully designed loss functions or training new models with specialized architectures
  - No existing method to perform controlled facial editing using GANs by varying latent codes

Research Questions

- **How do semantics in the latent space originate, and how are they organized?**
  - Does there exist structure in the latent space relating to disentangled semantic representations?

- **What does a GAN actually learn with respect to the latent space?**
  - How does the GAN connect the latent space and the image semantic space?

- **How can the latent code be used for image editing?**
  - How are various semantic attributes of an individual's face (gender, age) determined and entangled with each other?

Contributions and Main Findings

- Proposed ***InterFaceGAN***, a framework to identify the semantics encoded in the latent space of face synthesis models
    - Observed that **GANs learn latent subspaces corresponding to specific attributes**

- Showed that InterFaceGAN **enables semantic face editing with any pre-trained GAN**
    - Found theoretical and experimental results to verify that linear subspaces align with semantics emerging in the latent space

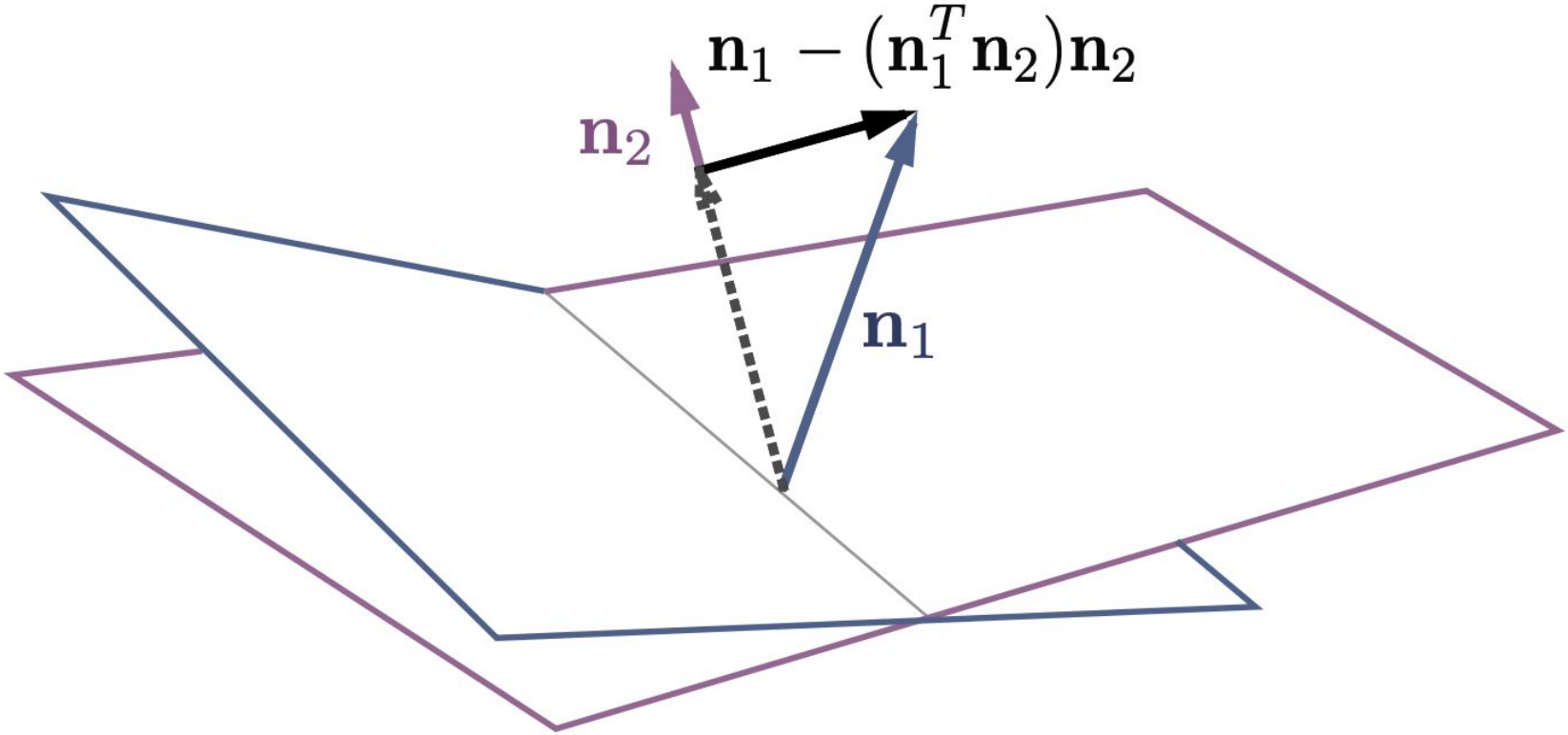
- Applied InterFaceGAN to **real image editing**
    - Successfully edited and manipulated the attributes of real faces by varying the latent code

Presentation Roadmap

- Introduction
- **Method**
- Experiments

$$\mathrm{d}(\mathbf{n}, \mathbf{z}) = \mathbf{n}^T \mathbf{z}.$$

$$f(g(\mathbf{z})) = \lambda \mathrm{d}(\mathbf{n}, \mathbf{z})$$

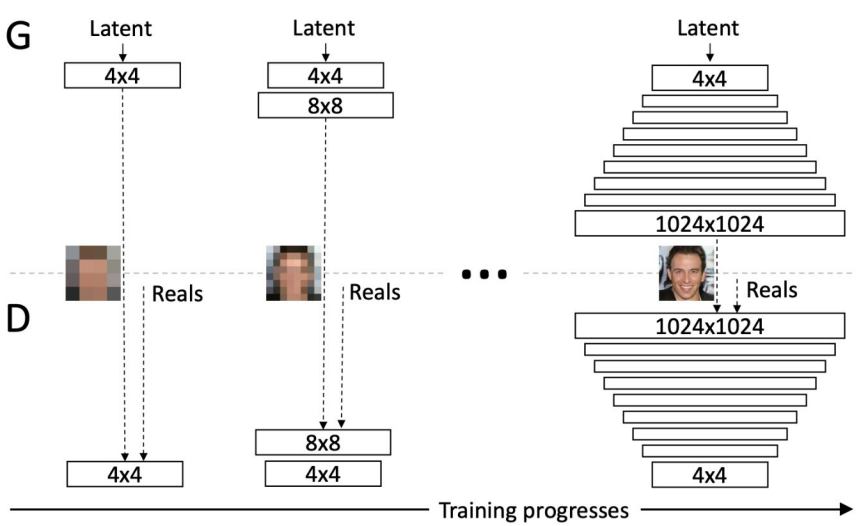

$\mathbf{n}_1 - (\mathbf{n}_1^T \mathbf{n}_2)\mathbf{n}_2$
$\mathbf{n}_2$
$\mathbf{n}_1$

Presentation Roadmap

- Introduction
- Method
- **Experiments**

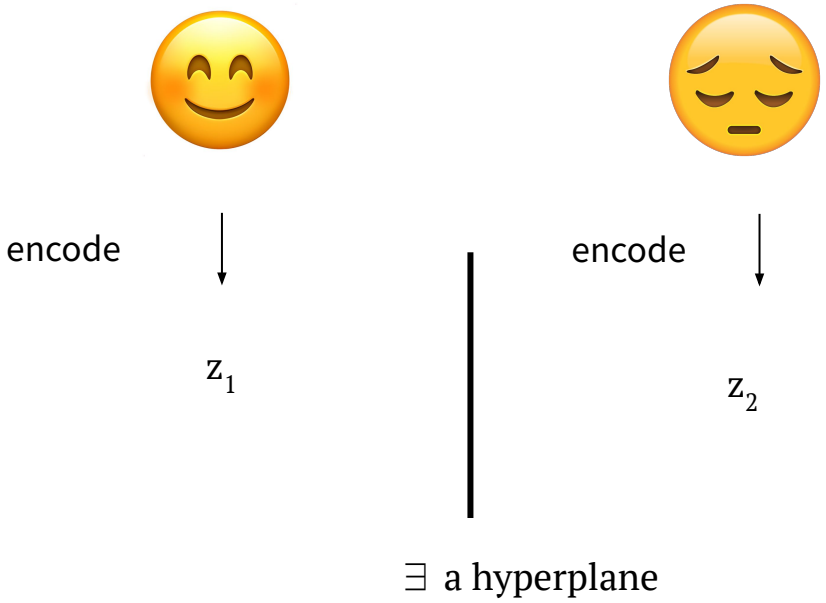
Experiment 1: Latent Space Separation

- **Model -** PGGAN (for experiment 2 and 3 also)
  - trained by growing both the generator and discriminator progressively: starting from a low resolution, add new layers that model increasingly fine details as training progresses.
  - Trained on CelebA-HQ face attributes dataset (each face is labeled with a subset of 40 attribute annotations). 30,000 images in total.



Experiment 1: Latent Space Separation

The framework is based on an assumption that for any binary attribute, there exists a hyperplane in latent space such that all samples from the same side have the same attribute.



Experiment 1: Latent Space Separation

Train five independent linear SVMs on pose, smile, age, gender, eyeglasses, latent codes and then evaluate them on the validation set (6K samples with high confidence level on attribute scores) as well as the entire set (480K random samples).

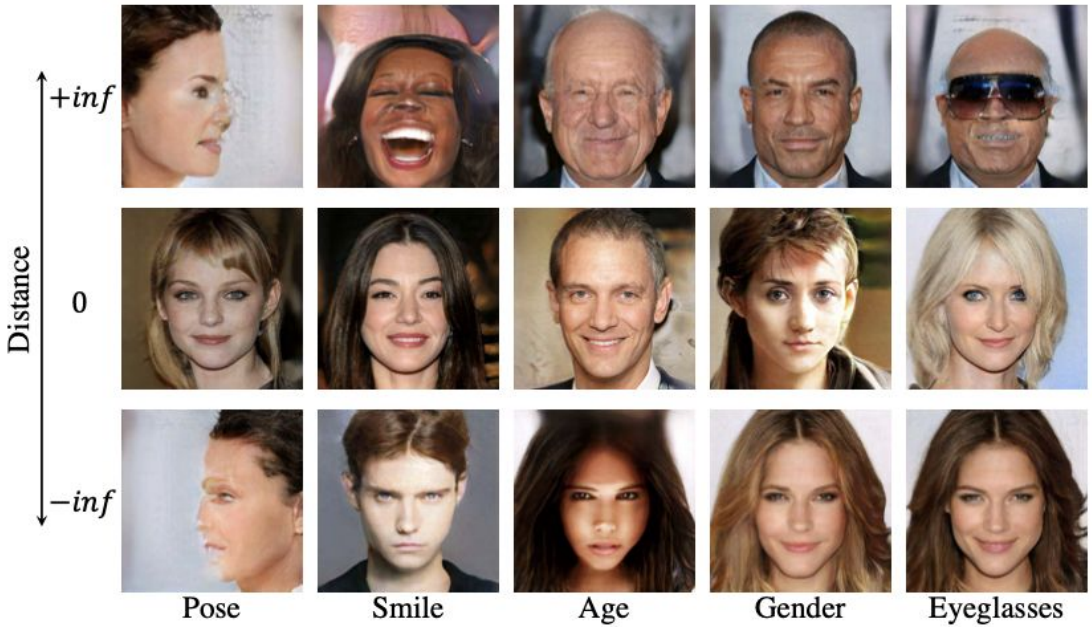
Datasets created by generating from GAN and classifying with ResNet-50 trained on the CelebA-HQ dataset.

Table 1: Classification accuracy (%) on separation boundaries in latent space with respect to different attributes.

Dataset	Pose	Smile	Age	Gender	Eyeglasses
Validation	100.0	96.9	97.9	98.7	95.6
All	90.3	78.5	75.3	84.2	80.1

Experiment 1: Latent Space Separation

Choose a z far away from and on decision boundary and generate from it



Experiment 2: Latent Space Manipulation

Verify whether the semantics found by InterFaceGAN are manipulable

Single attribute
Generate central image then move away from boundary

Works in both positive and negative directions.

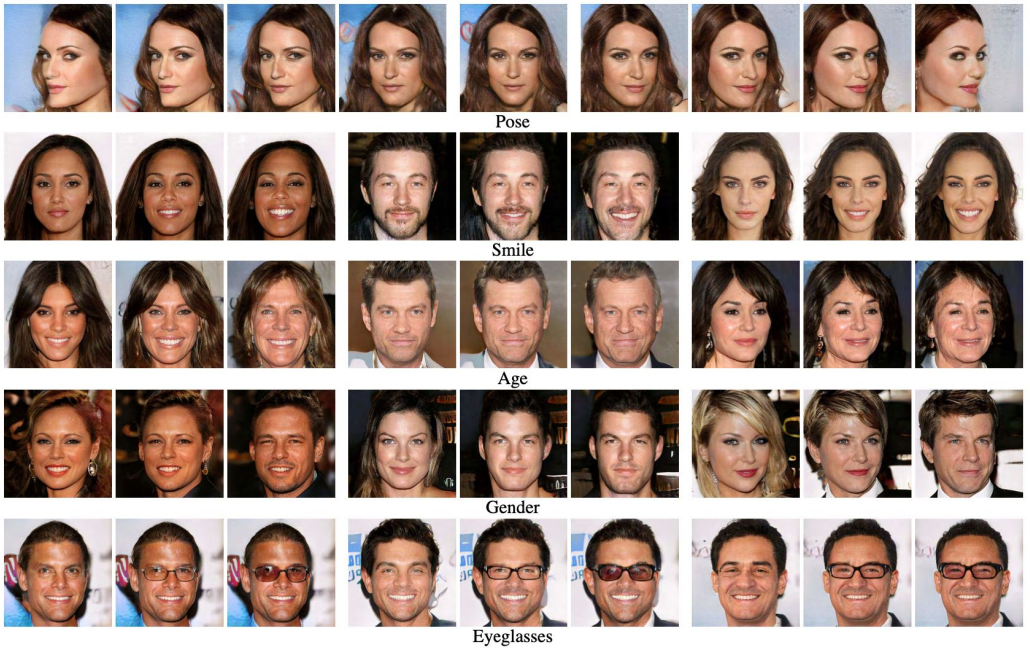


Figure 4: Single attribute manipulation results. The first row shows the same person under gradually changed poses. The following rows correspond to the results of manipulating four different attributes. For each set of three samples in a row, the central one is the original synthesis, while the left and right stand for the results by moving the latent code along negative and positive direction respectively.

Experiment 2: Latent Space Manipulation

Distance Effect of Semantic Subspace

Observe an interesting distance effect that the samples will suffer from severe changes in appearance if being moved too far from the boundary,

Extreme samples are very unlikely to be directly drawn from a standard normal distribution

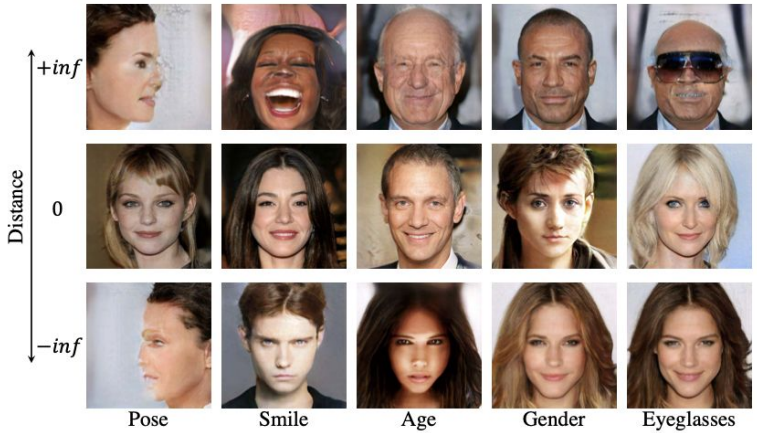


Figure 5: Illustration of the distance effect by taking gender manipulation as an example. The image in the red dashed box stands for the original synthesis. Our approach performs well when the latent code locates close to the boundary. However, when the distance keeps increasing, the synthesized images are no longer like the same person.

Experiment 2: Latent Space Manipulation

Artifacts Correction. Manually labeled 4K bad synthesis (have artifacts) and then trained a linear SVM to find the separation hyperplane

Figure 6: Examples on fixing the artifacts that GAN has generated. First row shows some bad generation results, while the following two rows present the gradually corrected synthesis by moving the latent codes along the positive "quality" direction.

Experiment 3: Conditional Manipulation

Study the disentanglement between different attributes and evaluate the conditional manipulation approach.

Correlation between Attributes

Cosine similarity of normal vectors

Correlation coefficient

$$\rho_{A_1 A_2} = \frac{Cov(A_1, A_2)}{\sigma_{A_1} \sigma_{A_2}}$$

Metrics agree, reflects the attribute correlation in the training dataset (male old people are more likely to wear eyeglasses)

Table 2: Correlation matrix of attribute boundaries.

	Pose	Smile	Age	Gender	Eyeglasses
Pose	1.00	-0.04	-0.06	-0.05	-0.04
Smile	-	1.00	0.04	-0.10	-0.05
Age	-	-	1.00	0.49	0.38
Gender	-	-	-	1.00	0.52
Eyeglasses	-	-	-	-	1.00

Table 3: Correlation matrix of synthesized attribute distributions.

	Pose	Smile	Age	Gender	Eyeglasses
Pose	1.00	-0.01	-0.01	-0.02	0.00
Smile	-	1.00	0.02	-0.08	-0.01
Age	-	-	1.00	0.42	0.35
Gender	-	-	-	1.00	0.47
Eyeglasses	-	-	-	-	1.00

Experiment 3: Conditional Manipulation

Effect of conditional manipulation

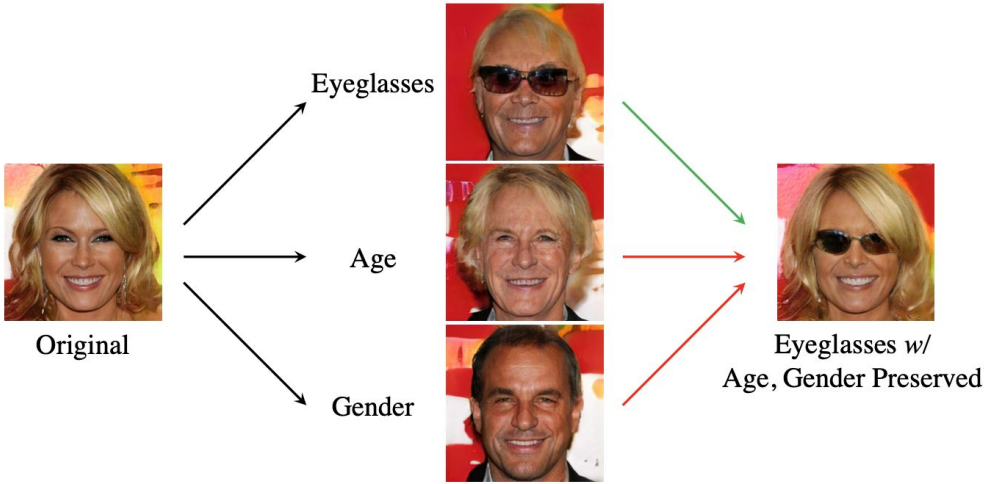


Figure 8: Examples for conditional manipulation with more than one conditions. Left: Original synthesis. Middle: Manipulations along single boundary. Right: Conditional manipulation. **Green** arrow: Primal direction. **Red** arrows: Projection subtraction.

Experiment 3: Conditional Manipulation

Effect of conditional manipulation

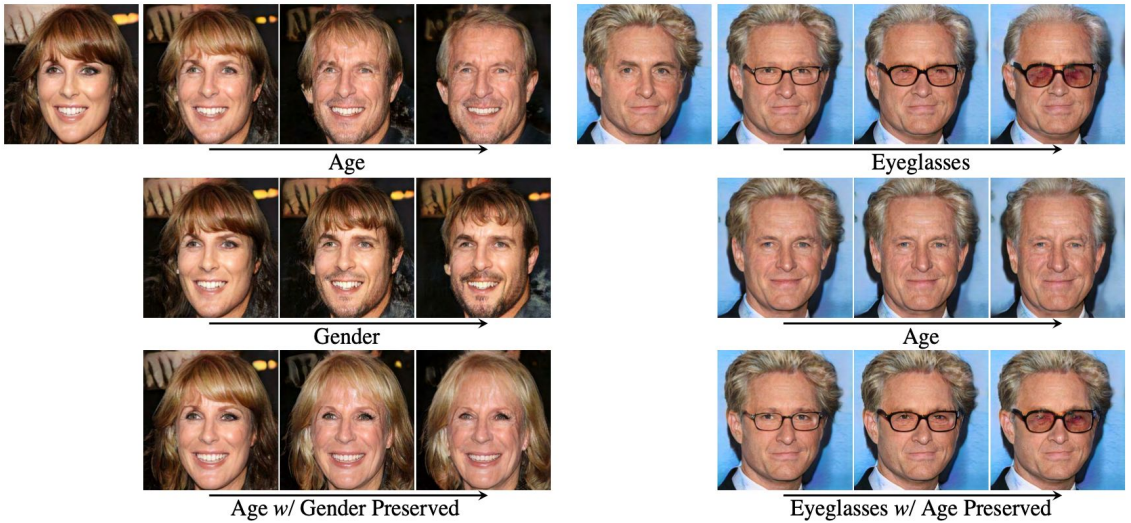


Figure 7: Examples for conditional manipulation. The first two rows show the manipulation results along with the original directions learned by SVMs for two attributes independently. The last row edits the faces by varying one attribute with the other one unchanged.

Experiment 4: Results on StyleGAN

StyleGAN Key Attributes**:**

PGGan: Model learns to generate larger images over time

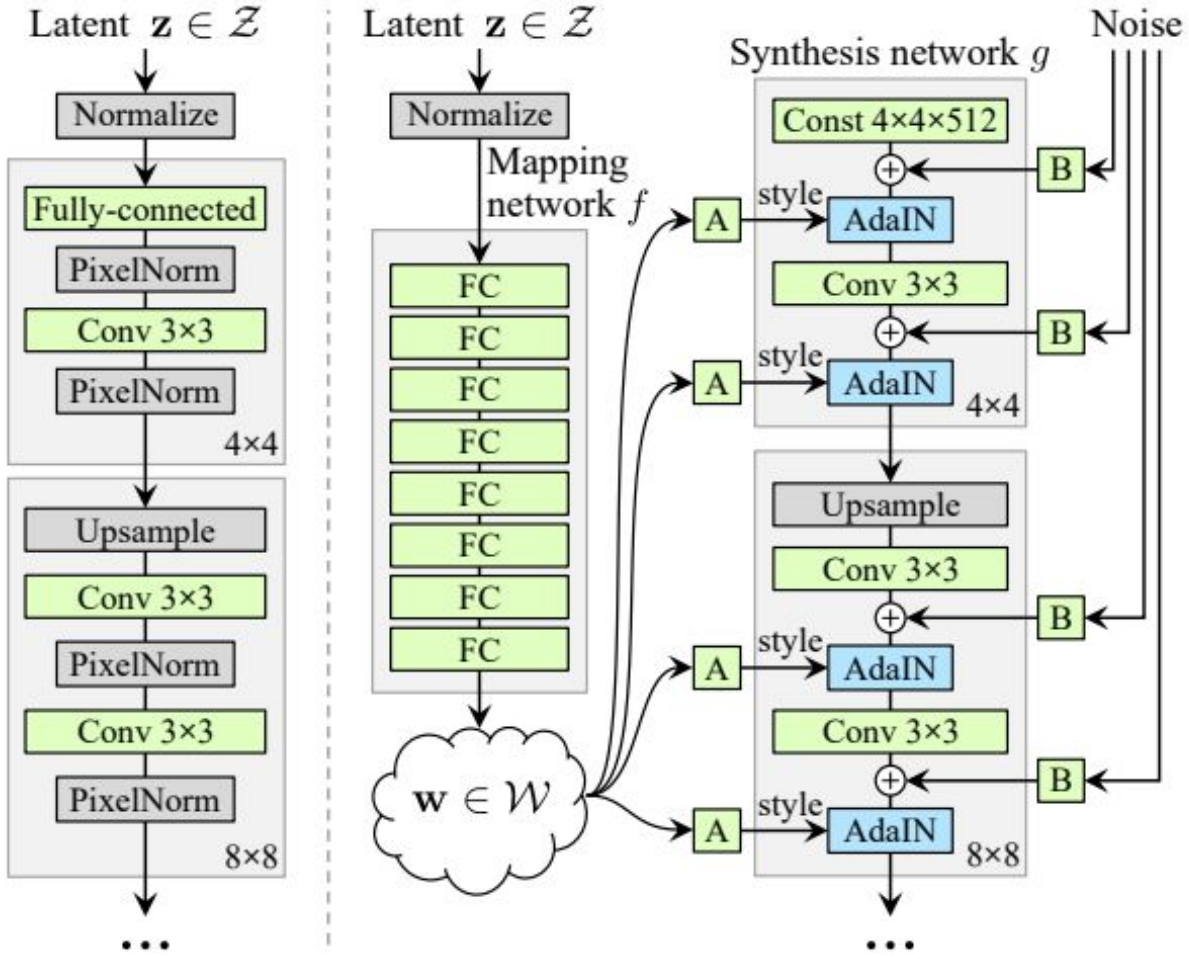
Mapping Network: Cast random input z as w (“style vector”)

Synthesis Network: Uses both style vector and noise across layers with AdaIN

Bilinear Sampling: Replace nearest neighbor upsampling

Mixing Regularization: Use 2 latent codes instead of one to generate

Datasets: trained on CelebA-HQ and FFHQ (new dataset)

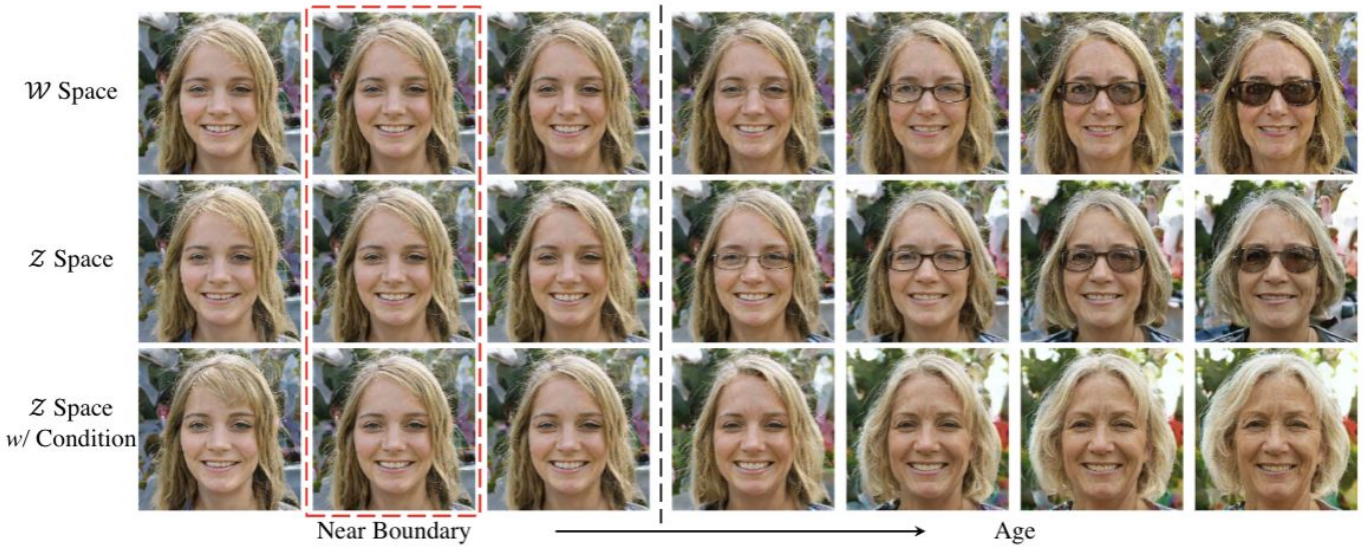


(a) Traditional

(b) Style-based generator

Experiment 4: Results on StyleGAN

Eyeglasses/age direction from StyleGAN



Analysis on Z and W reveals:

-W learns more disentangled representation

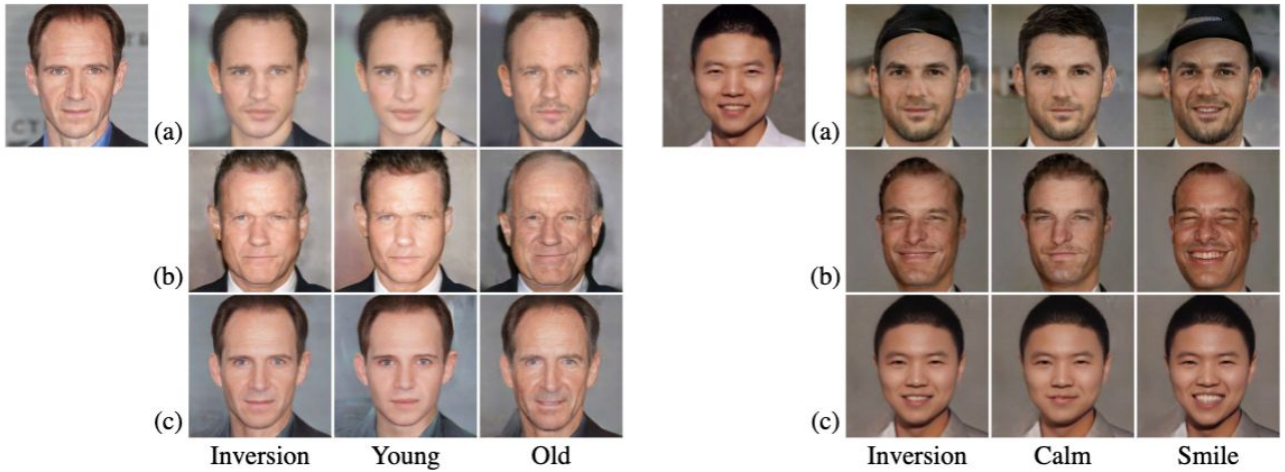
-W performs better in long-distance manipulation

-W capture attribute correlations

Experiment 5: Real Image Manipulation

-Take a real face image, invert to latent code, and use InterFaceGAN to face edit

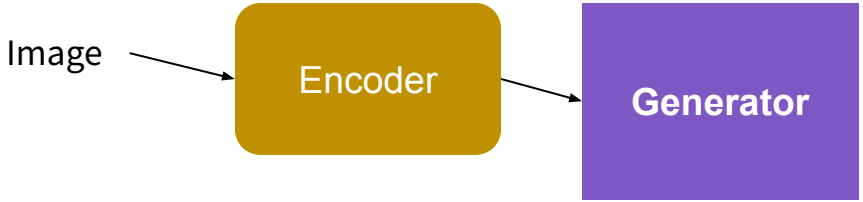
Latent Code Optimization: Optimize latent code with fixed generator to minimize pixel-wise reconstruction error



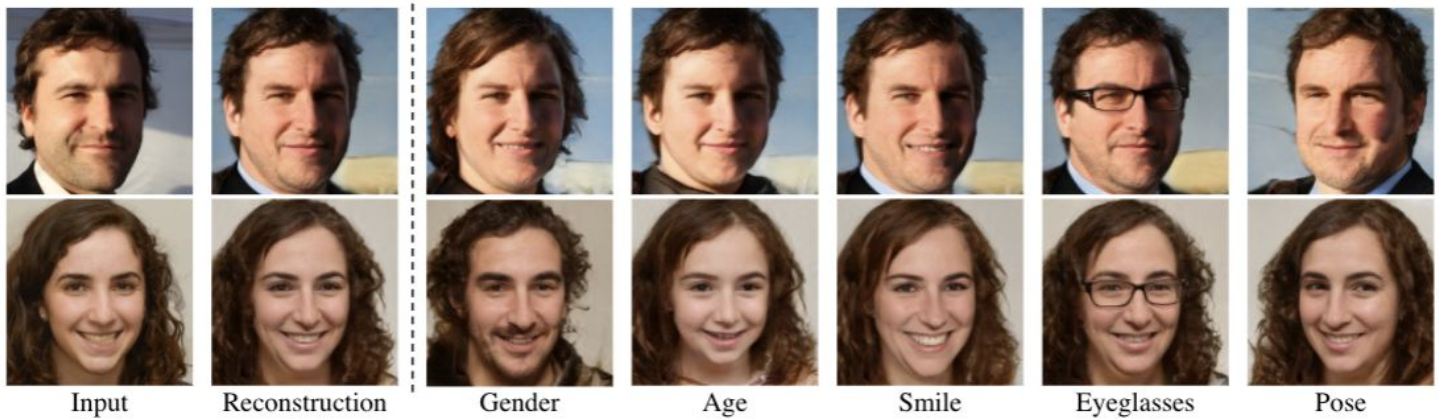
(a) PGGAN with optimization-based inversion method, (b) PGGAN with encoder-based inversion method, (c) StyleGAN with optimization-based inversion method.

Experiment 5: Real Image Manipulation

Encoder-Based: Extra encoder learns inverse mapping from image to latent code by training with generator and discriminator



Testing latent space of LIA



Discussion Questions

1) Do you believe the use of GANs for generating photo-realistic images should be regulated? (Potential misuse of such tools)
2) Why do we think there are such simple linear boundaries separating binary semantics in the latent space of GANs (is it only for face GANs)?
3) How can analyzing semantic representations be used to detect biases in the training data?
4) Have you ever used GANs for photo generation, if so how realistic did you find them?
5) How relevant is this paper given that GANs are progressively going out of style (diffusion taking over)