Lecture 3.2 Manipulating the latent space of a GAN model

Dimitrios Papadopoulos
Associate Professor, DTU Compute

Yesterday

- Generative Adversarial Networks
- Applications of GANs
- How to train a GAN
- Different GAN models
- Exercise 3

Today

- Progressive GANs, StyleGAN, StyleGAN2, StyleGAN2-ADA
- Interpolation in latent space
- Reconstructing real images
- Learn and apply latent directions
- Generate images from text prompts (DALLE, DALLE2, Imagen)
- CLIP + StyleGAN
- Project 2

Updated Schedule

Part 3: GANs

Tuesday 13.6	09:00-11:00 Lecture - 3.1: Introduction to GANs 11:00-12:00 Exercise 3: GANs 13:00-17:00 Continue exercise	9-12 Dimitrios 13-15 Manxi 14-16 Paraskevas 15-17 Thanos
Wednesday 14.6	09:00-10:30 Lecture - 3.2: Manipulate image generation 10:30-17:00 Work on project	9-12 Dimitrios 11-12 Thanos 13-15 Manxi 14-16 Paraskevas 16-17 Thanos
Thursday 15.6	09:00-10:00 Lecture - 3.3: Intro to Diffusion Models *10:00-11:00 Project 2 Poster session* - Give peer feedback to designated groups 11:00-17:00 Work on project **Project 3 deadline at midnight**	9-12 Dimitrios 13-15 Manxi 14-16 Paraskevas 15-17 Thanos


BREAKING
NEWS

[New Courses] 02514 → 02516

02516 | Information

02516 Introduction to Deep Learning in Computer Vision 2023/2024

Course information

Danish title	Introduktion til Deep Learning i Computer Vision
Language of instruction	English
Point(ECTS)	5
Course type	MSc
Schedule	January
Location	Campus Lyngby
Scope and form	The course consists of lectures, practical exercises.
Duration of Course	3 weeks
Type of assessment	Written or oral examination The exam will be oral, unless there are so many participants that it becomes impractical. In this case, there will be a written exam.
Exam duration	Written exam: 3 hours
Aid	All Aid : All aids except the internet
Evaluation	7 step scale , internal examiner
Recommended prerequisites	02456 , 02456 or their equivalents (in particular, we expect that you are familiar with training and regularization of neural networks using PyTorch)
Participants restrictions	Minimum 8
Responsible	Aasa Feragen , afhar@dtu.dk
Course co-responsible	Dimitrios Papadopoulos , Lyngby Campus, Building 324 , dimp@dtu.dk Morten Rieger Hannemose , Lyngby Campus, Building 324 , mohan@dtu.dk
Department	01 Department of Applied Mathematics and Computer Science
Registration Sign up	At the Studyplanner

General course objectives

Many computer vision tasks were traditionally solved based on hand-crafted features, but are now solved using deep learning. Examples of such tasks include object detection, segmentation and classification. In this course the student will learn about some of these tasks and how to solve them with deep learning. This will enable the student to identify and describe a problem within computer vision, solve it using a suitable neural network architecture and analyze the performance of the solution.

Learning objectives

A student who has met the objectives of the course will be able to:

- Define what is deep learning and explain the difference between deep learning models and traditional machine learning based on hand-crafted features in computer vision.
- Select and implement suitable convolutional neural network architectures for classification, localization and segmentation.
- Interpret and illustrate the internal representations of a convolutional neural network
- Demonstrate knowledge of and use of popular techniques used for training neural networks.
- Find appropriate training data for a given problem as well as apply relevant data augmentation.
- Setting up convolutional neural networks to be trained with a custom dataloader on a remote server
- Assess and analyze the performance of convolutional neural networks.
- Define and implement adversarial attacks to highlight instabilities in neural networks
- Define, implement and discuss attribution methods from explainable AI

Content

The course contains lectures and exercises about neural networks for computer vision. The exercises takes the student from a concrete problem within computer vision to a solution based on neural networks. Topics covered include image recognition and understanding tasks such as image classification, object detection and semantic segmentation.

Last updated

04. maj, 2023

https://kurser.dtu.dk/course/2023-2024/02516

[New Courses] 02501 Advanced DLCV

02501 Advanced Deep Learning in Computer Vision 2023/2024

Course information

Danish title	Videregående Deep Learning i Computer Vision
Language of instruction	English
Point(ECTS)	5
Course type	MSc
Schedule	Spring F4A (Tues 13-17)
Location	Campus Lyngby
Scope and form	Lectures, practical exercises and projects.
Duration of Course	13 weeks
Date of examination	F4A
Type of assessment	Written or oral examination Oral or written examination. The exam is oral, unless there are so many participants that it becomes impractical. In this case, there will be a written exam.
Exam duration	Written exam: 3 hours
Aid	All Aid
Evaluation	7 step scale , internal examiner
Recommended prerequisites	02450.02456.02516 , 02450, 02456, 02516 or their equivalents (in particular, we expect that you are well experienced with training neural networks for computer vision on custom data using PyTorch on a remote server)
Participants restrictions	Minimum 8
Responsible	Dimitrios Papadopoulos , Lyngby Campus, Building 324 , dimp@dtu.dk
Course co-responsible	Aasa Feragen , afhar@dtu.dk Morten Rieger Hannemose , Lyngby Campus, Building 324 , mohan@dtu.dk
Department	01 Department of Applied Mathematics and Computer Science
Registration Sign up	At the Studyplanner

General course objectives

To give knowledge of advanced deep learning methods and models for computer vision, and give competence in applying these techniques in different applications.

Learning objectives

A student who has met the objectives of the course will be able to:

- Select, implement, and utilize state-of-the-art deep learning architectures for classical computer vision tasks such as classification, segmentation, and recognition
- Utilize and examine deep learning models that combine image data with other modalities, such as text
- Implement deep learning models for temporal image data, such as videos
- Explain and implement deep generative models for image synthesis
- Describe, implement and compare alternative methods for training deep learning models in limited data settings
- Assess the quality of deep learning models for computer vision from viewpoints of both performance and responsibility/ethics.
- Present projects and associated results in writing as well as oral presentation.
- Discuss strengths, weaknesses and societal implications of state-of-the-art deep learning models.

Content

The course gives an introduction to advanced topics within deep learning for computer vision. Therefore, focus in the exercises is on implementing algorithms and using these for solving practical computer vision problems within the following topics: image recognition, sequential data, generative models, video understanding, explainability, and fairness.

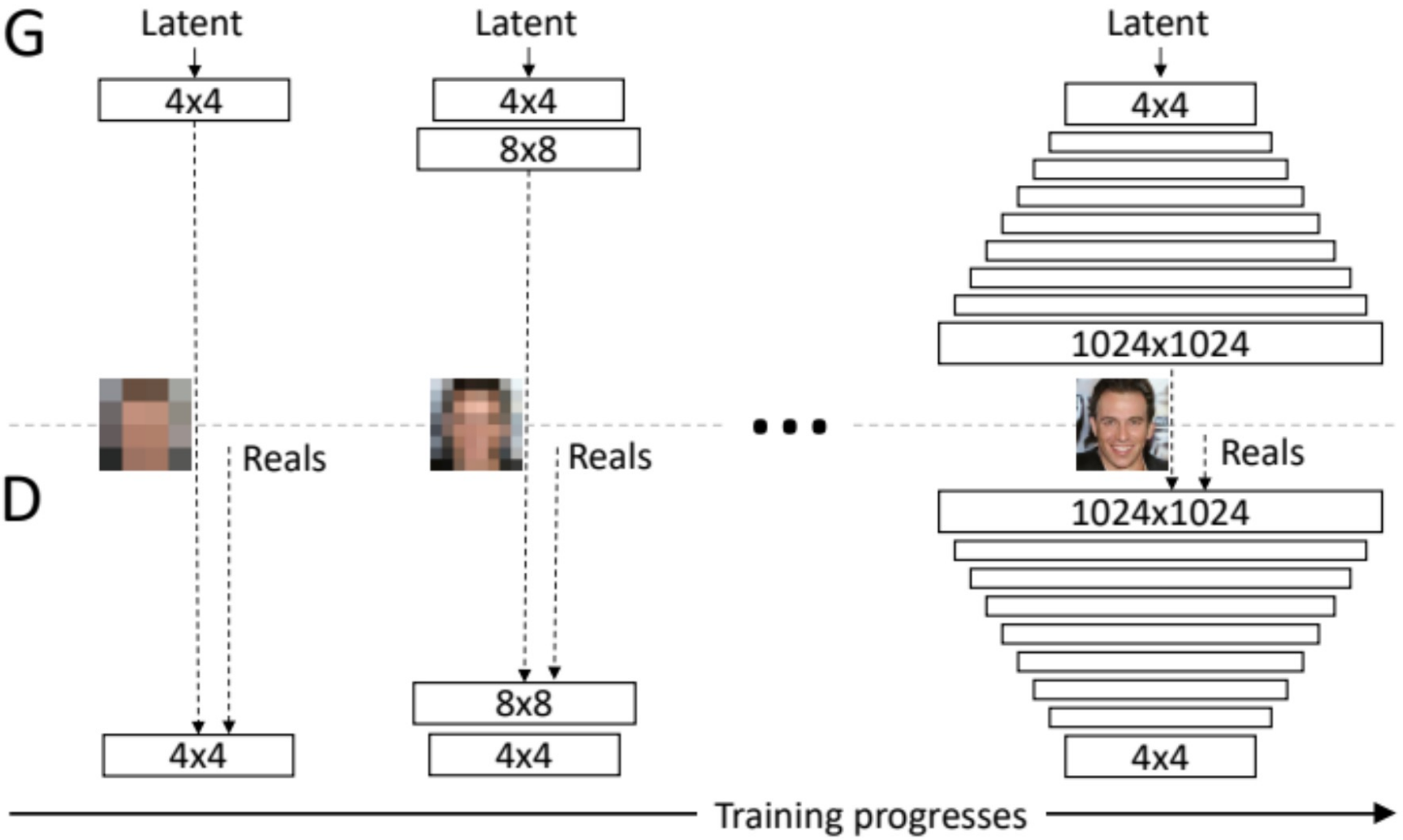
Last updated

17. maj, 2023

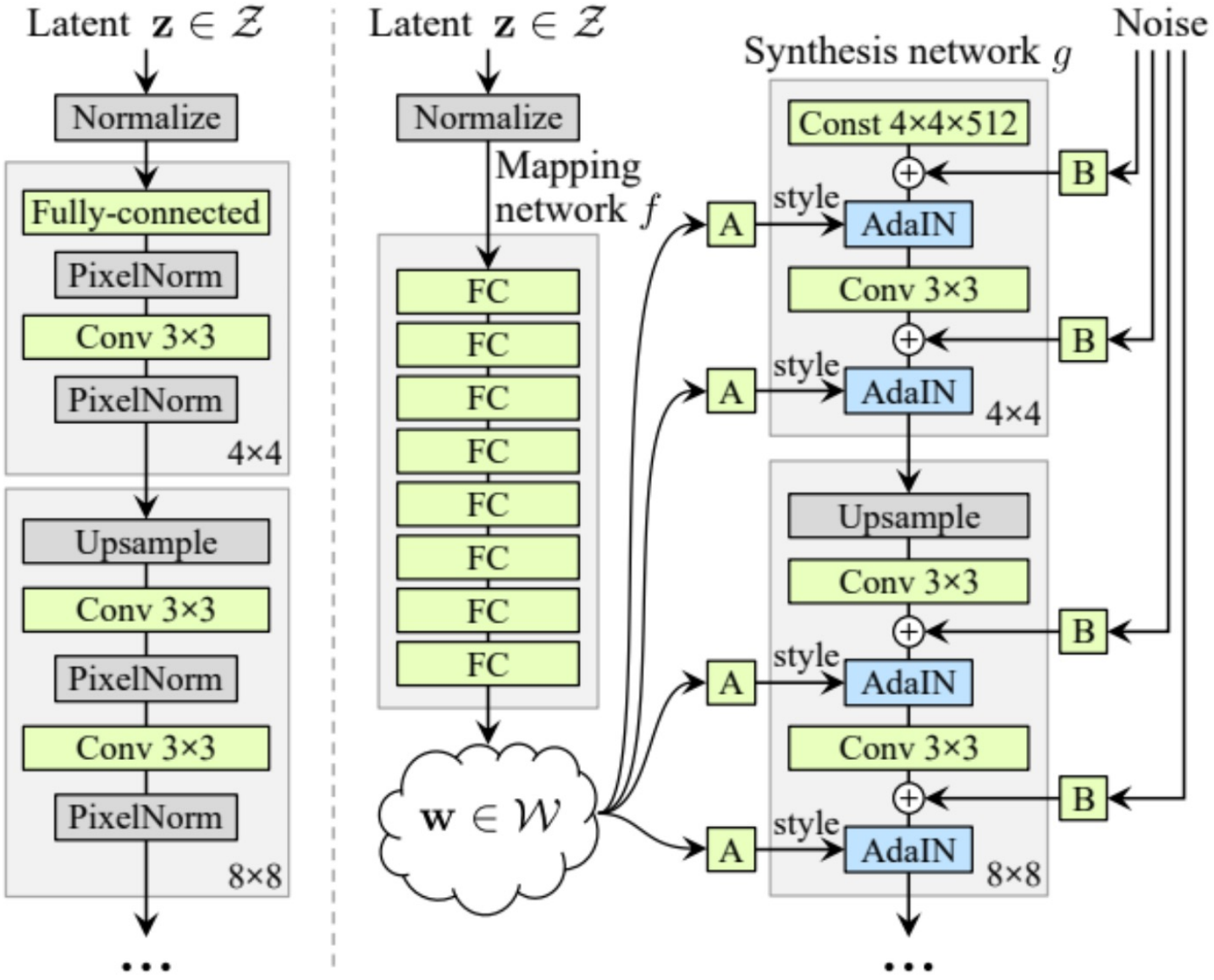
Also: Special course Autumn 2023 !!!

https://kurser.dtu.dk/course/2023-2024/02501

Progressive GAN, Karras et al ICLR 2018



StyleGAN, Karras et al CVPR 2019



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

[Huang and Belongie, ICCV 2017]

Figure 1. While a traditional generator [30] feeds the latent code though the input layer only, we first map the input to an intermediate latent space $\mathcal{W}$, which then controls the generator through adaptive instance normalization (AdaIN) at each convolution layer. Gaussian noise is added after each convolution, before evaluating the nonlinearity. Here "A" stands for a learned affine transform, and "B" applies learned per-channel scaling factors to the noise input. The mapping network f consists of 8 layers and the synthesis network g consists of 18 layers — two for each resolution ($4^2 - 1024^2$). The output of the last layer is converted to RGB using a separate 1×1 convolution, similar to Karras et al. [30]. Our generator has a total of 26.2M trainable parameters, compared to 23.1M in the traditional generator.

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.

StyleGAN2, Karras et al CVPR 2020

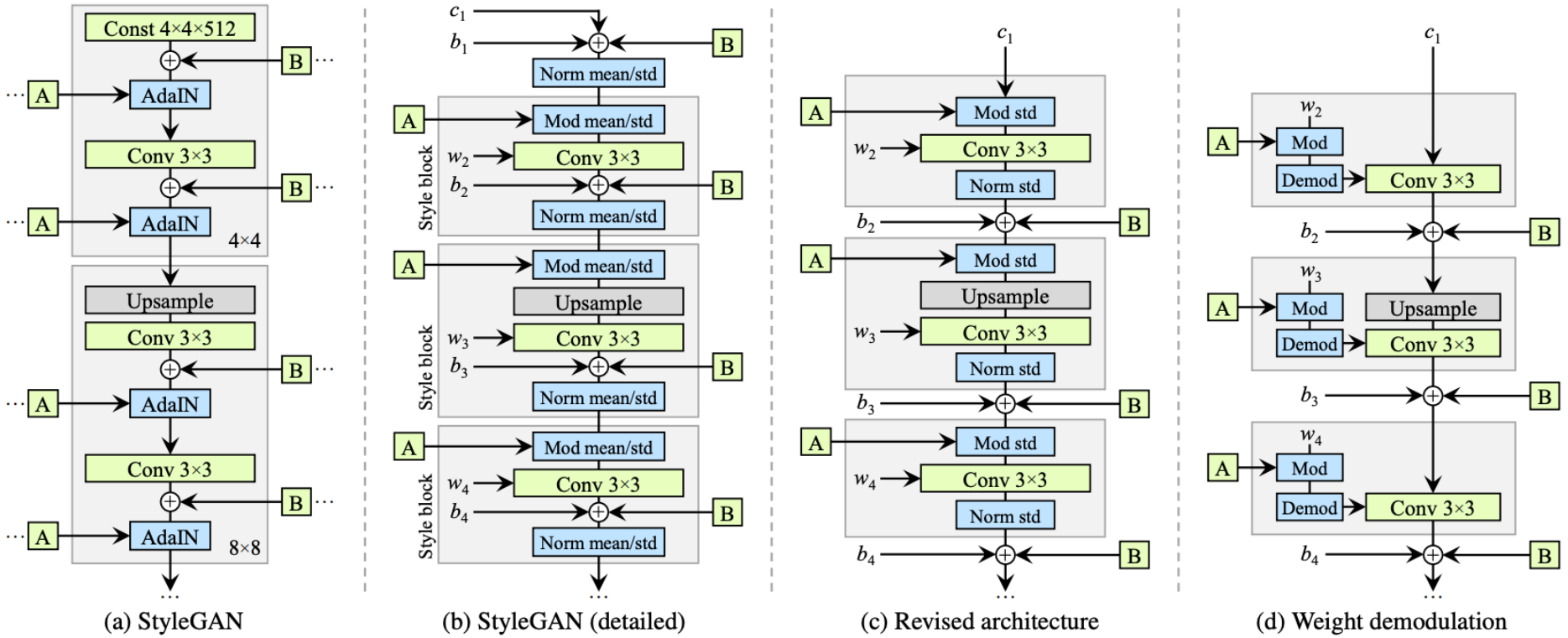


Figure 2. We redesign the architecture of the StyleGAN synthesis network. (a) The original StyleGAN, where A denotes a learned affine transform from $\mathcal{W}$ that produces a style and B is a noise broadcast operation. (b) The same diagram with full detail. Here we have broken the AdaIN to explicit normalization followed by modulation, both operating on the mean and standard deviation per feature map. We have also annotated the learned weights (w), biases (b), and constant input (c), and redrawn the gray boxes so that one style is active per box. The activation function (leaky ReLU) is always applied right after adding the bias. (c) We make several changes to the original architecture that are justified in the main text. We remove some redundant operations at the beginning, move the addition of b and B to be outside active area of a style, and adjust only the standard deviation per feature map. (d) The revised architecture enables us to replace instance normalization with a "demodulation" operation, which we apply to the weights associated with each convolution layer.

StyleGAN2 - ADA, Karras et al NeurIPS 2020

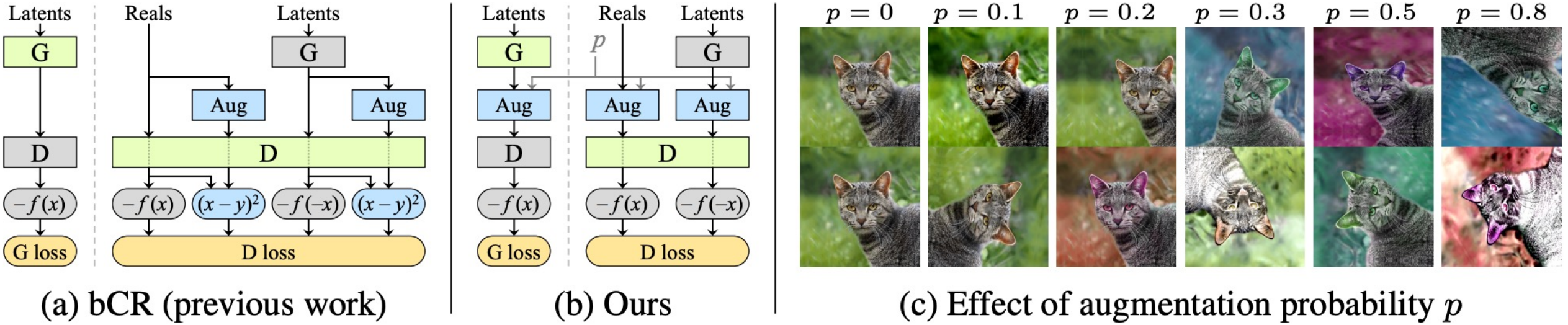
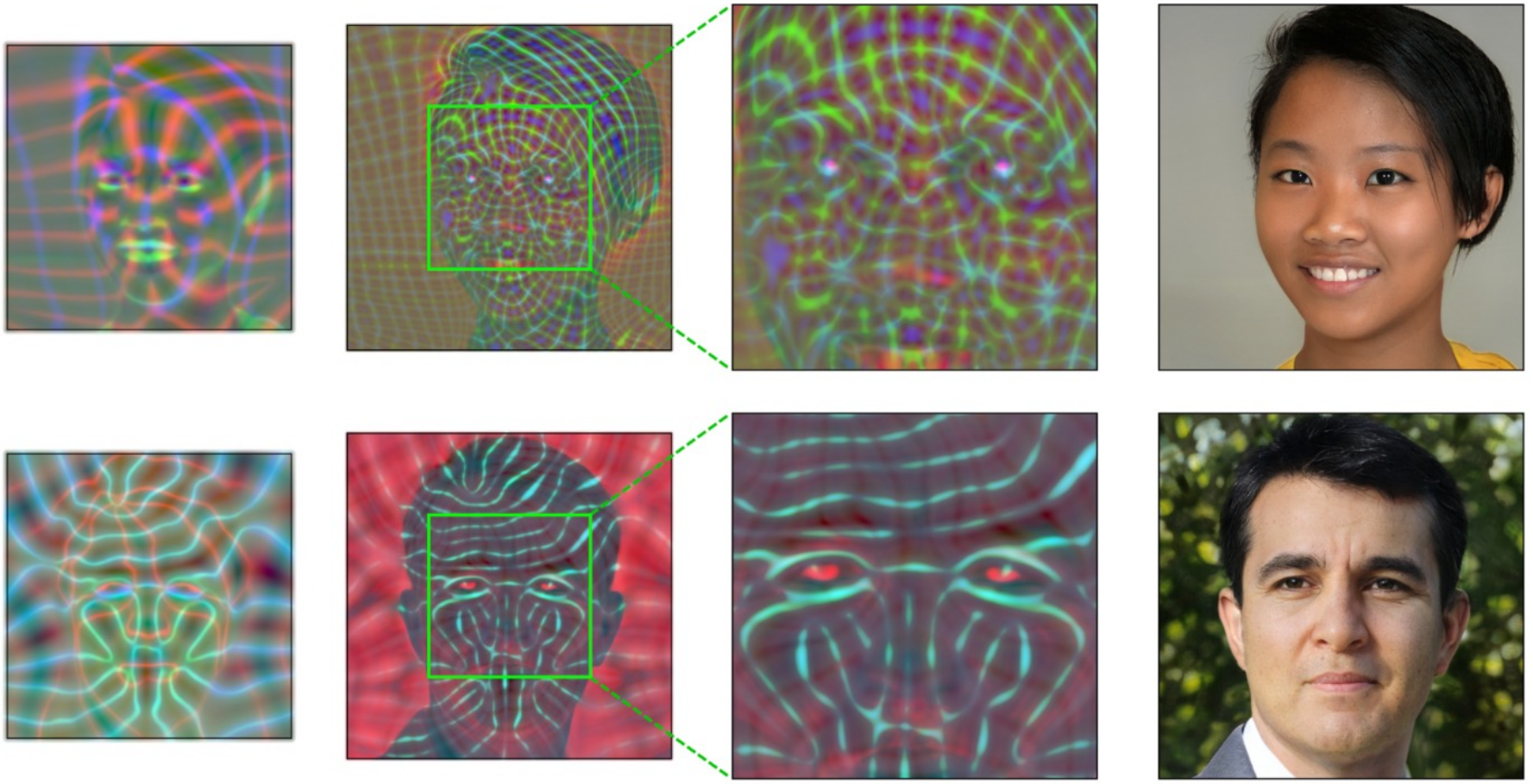


Figure 2: (a,b) Flowcharts for balanced consistency regularization (bCR) [43] and our stochastic discriminator augmentations. The blue elements highlight operations related to augmentations, while the rest implement standard GAN training with generator G and discriminator D [12]. The orange elements indicate the loss function and the green boxes mark the network being trained. We use the non-saturating logistic loss [12] $f(x) = \log(\text{sigmoid}(x))$. (c) We apply a diverse set of augmentations to every image that the discriminator sees, controlled by an augmentation probability p.

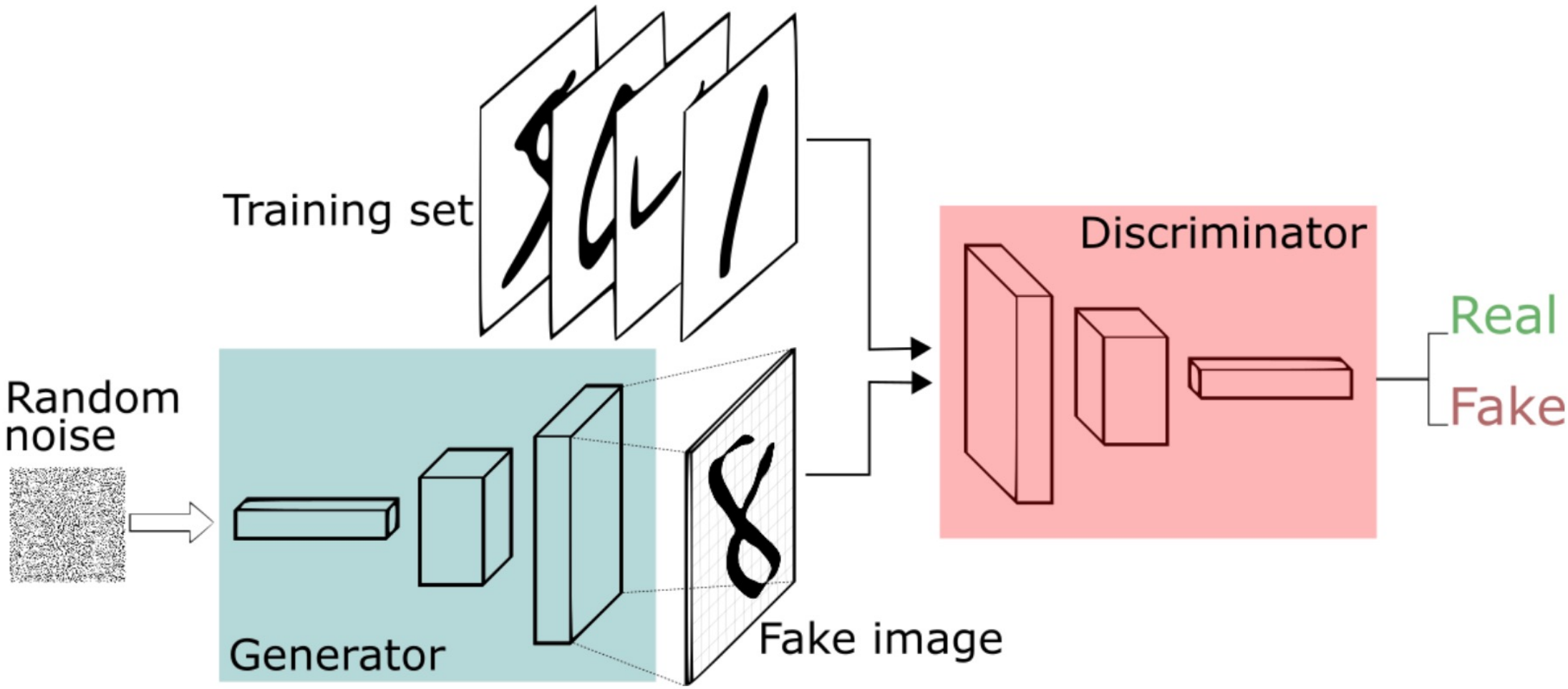
https://github.com/NVlabs/stylegan2-ada

StyleGAN3, Karras NeurIPS 2021



https://nvlabs.github.io/stylegan3/

Latent space of a GAN



Generate random images

```
python generate.py --outdir=out --trunc=1 --seeds=666-700 --network=https://nvlabs-fi-cdn.nvidia.com/stylegan2-ada-pytorch/pretrained/ffhq.pkl
```

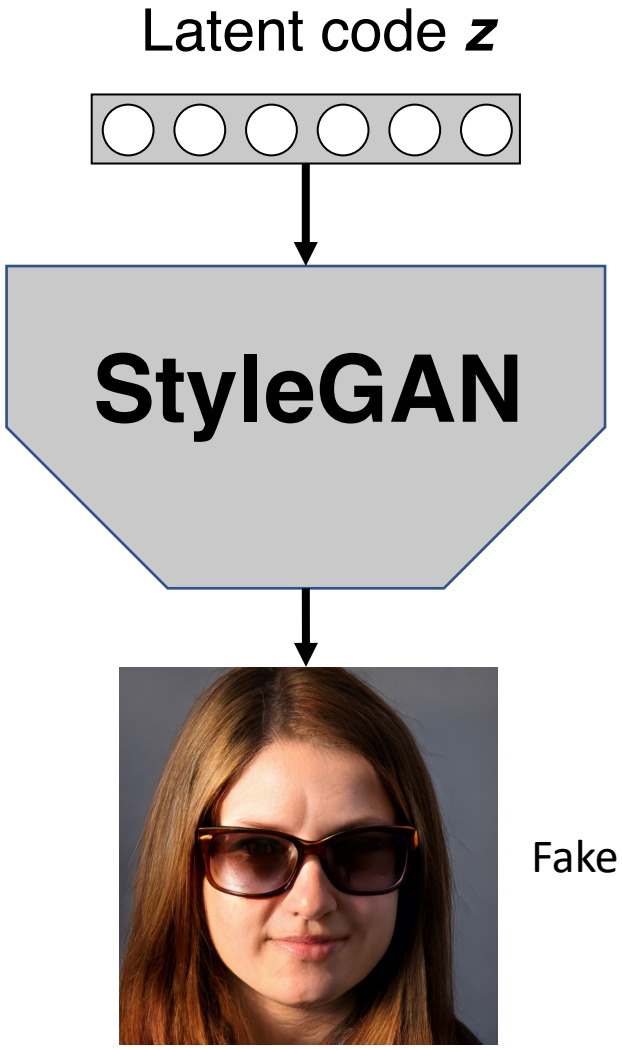
Interpolation

Latent space of a GAN

z_1 $a*z_1+(1-a)*z_2$ z_2

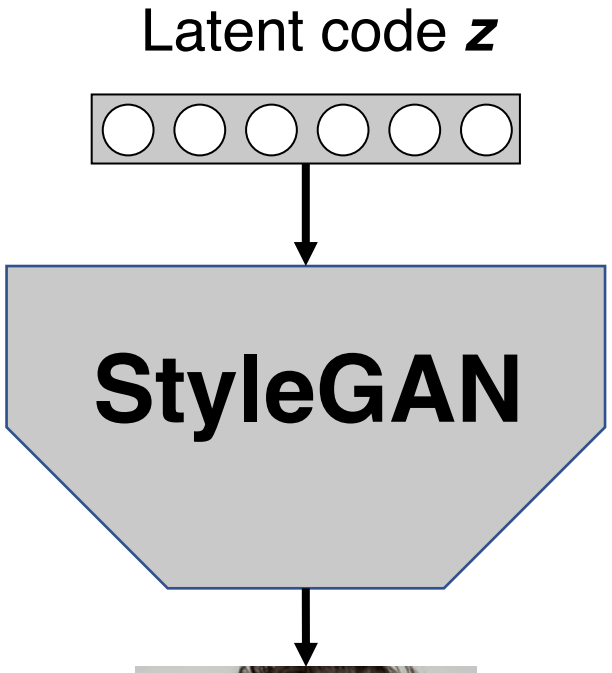
```
python generate.py --outdir=out --dlatents=out/dlatents.npz --network=https://nvlabs-fi-cdn.nvidia.com/stylegan2-ada/pretrained/ffhq.pkl
```

Real images: GAN inversion



Fake generated image

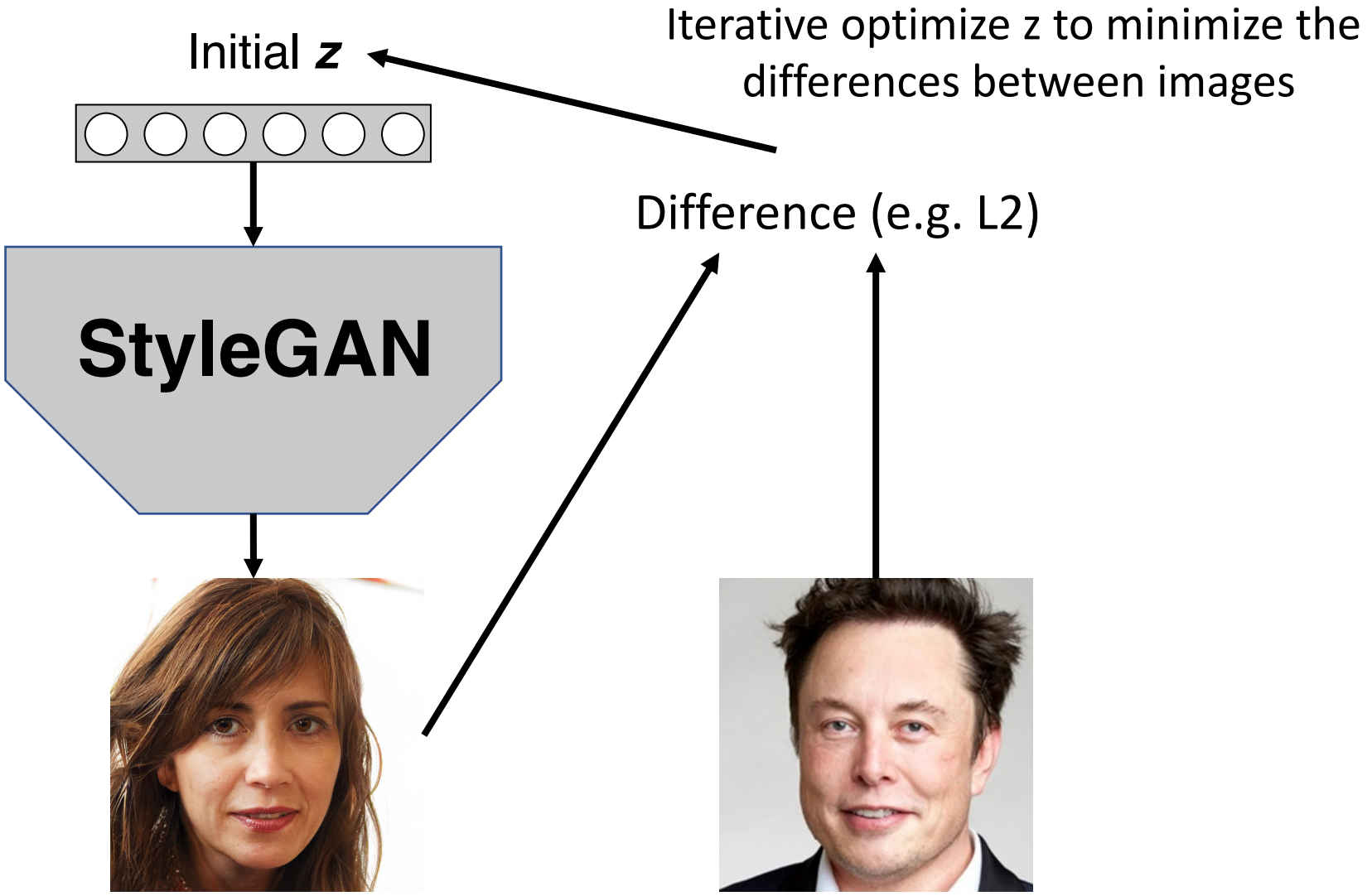
Real images: GAN inversion



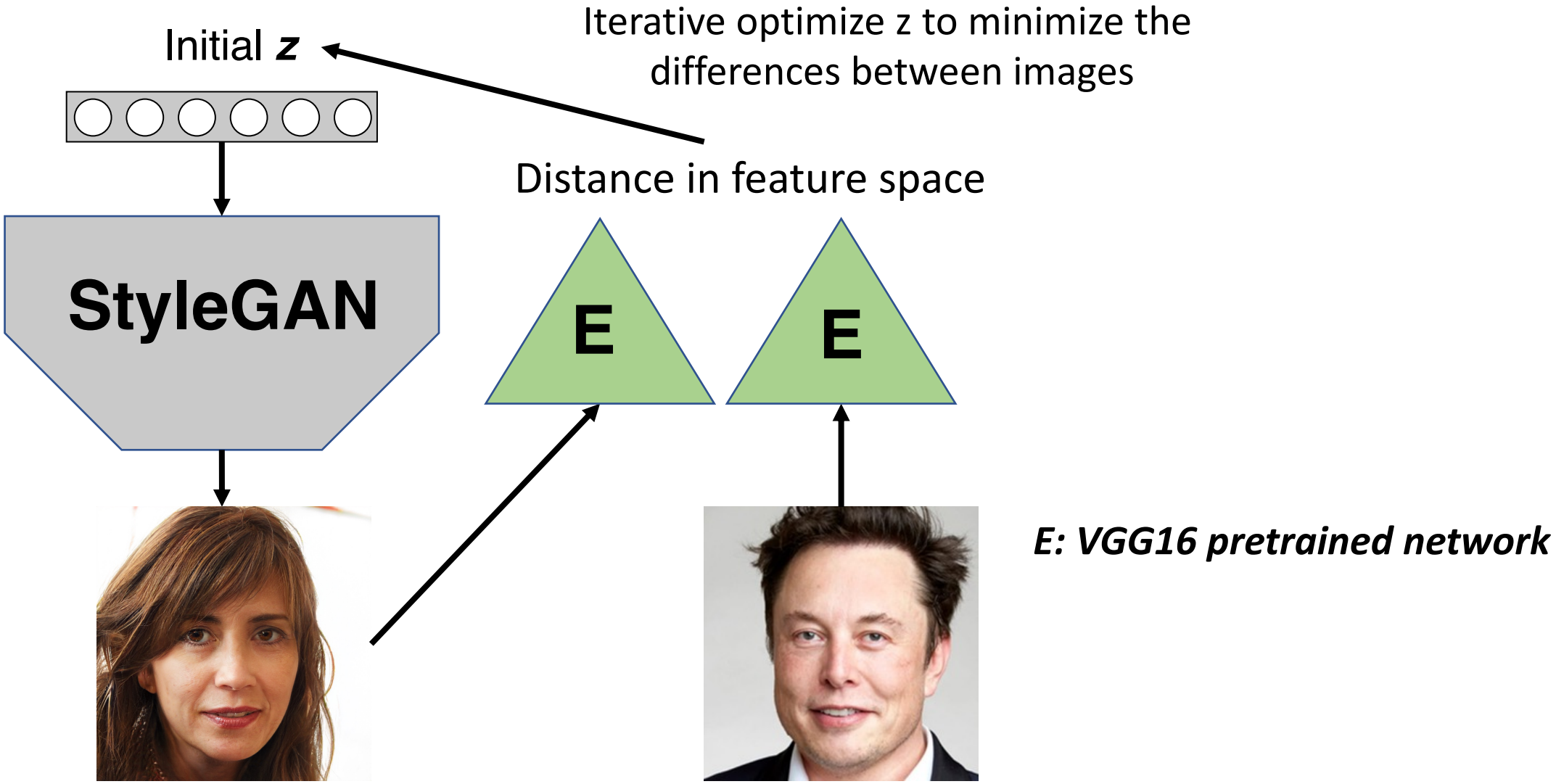
Real image

How to find z_o that generates Elon Musk?

Real images: GAN inversion



Real images: GAN inversion



Real images: GAN inversion

```
python projector.py --outdir=out --target=targetimg.png \ --
network=https://nvlabs-fi-cdn.nvidia.com/stylegan2-ada/pretrained/ffhq.pkl
```

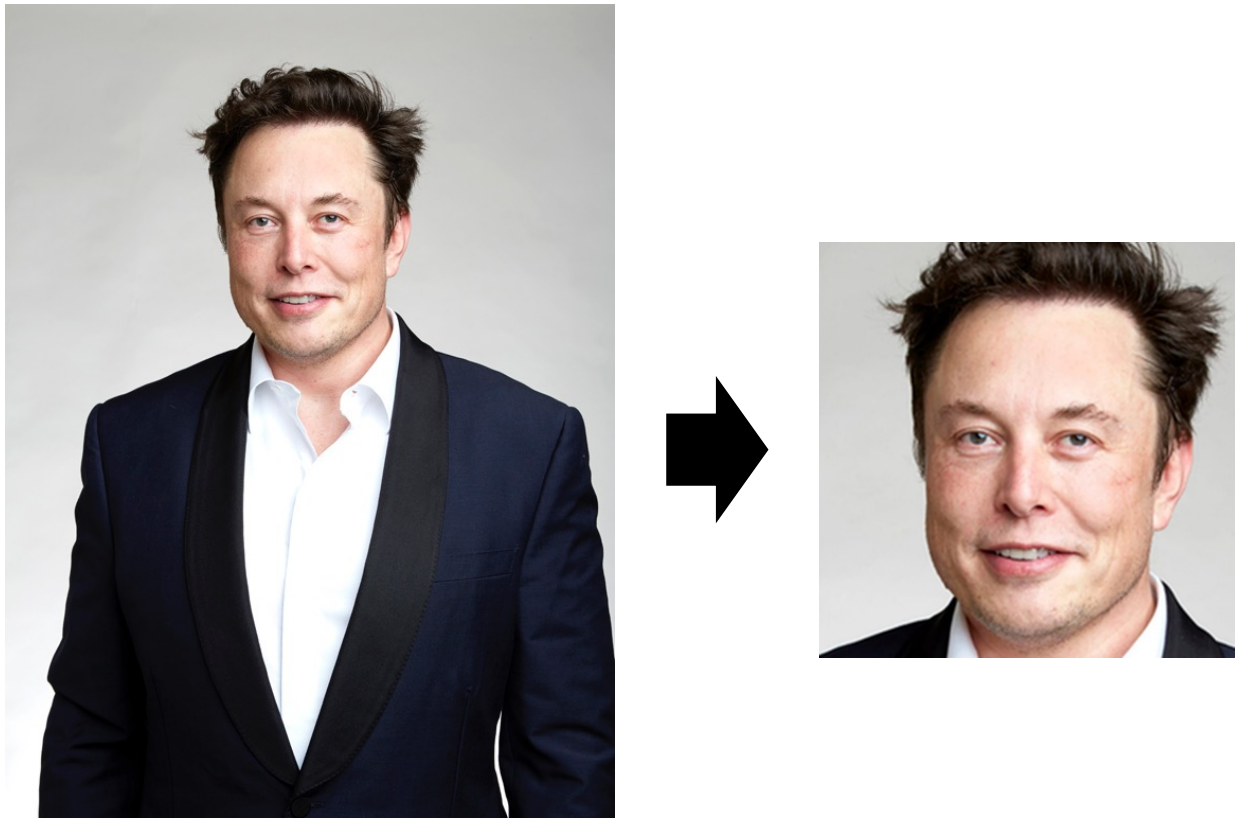
Real images: GAN inversion

```
python projector.py --outdir=out --target=targetimg.png \ --
network=https://nvlabs-fi-cdn.nvidia.com/stylegan2-ada/pretrained/ffhq.pkl
```

Align Images with FFHQ

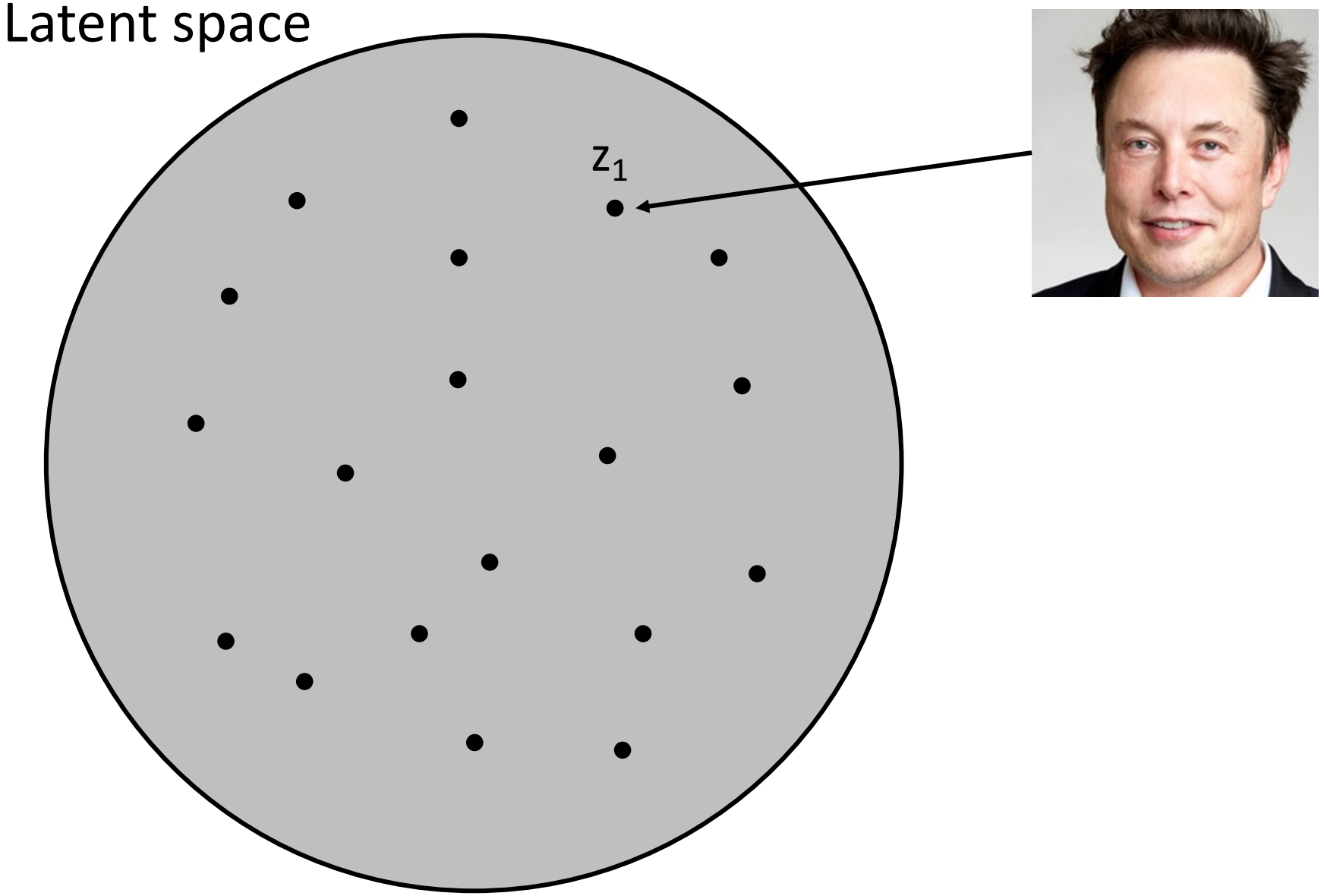
https://github.com/happy-jihye/FFHQ-Alignment

Align Images with FFHQ

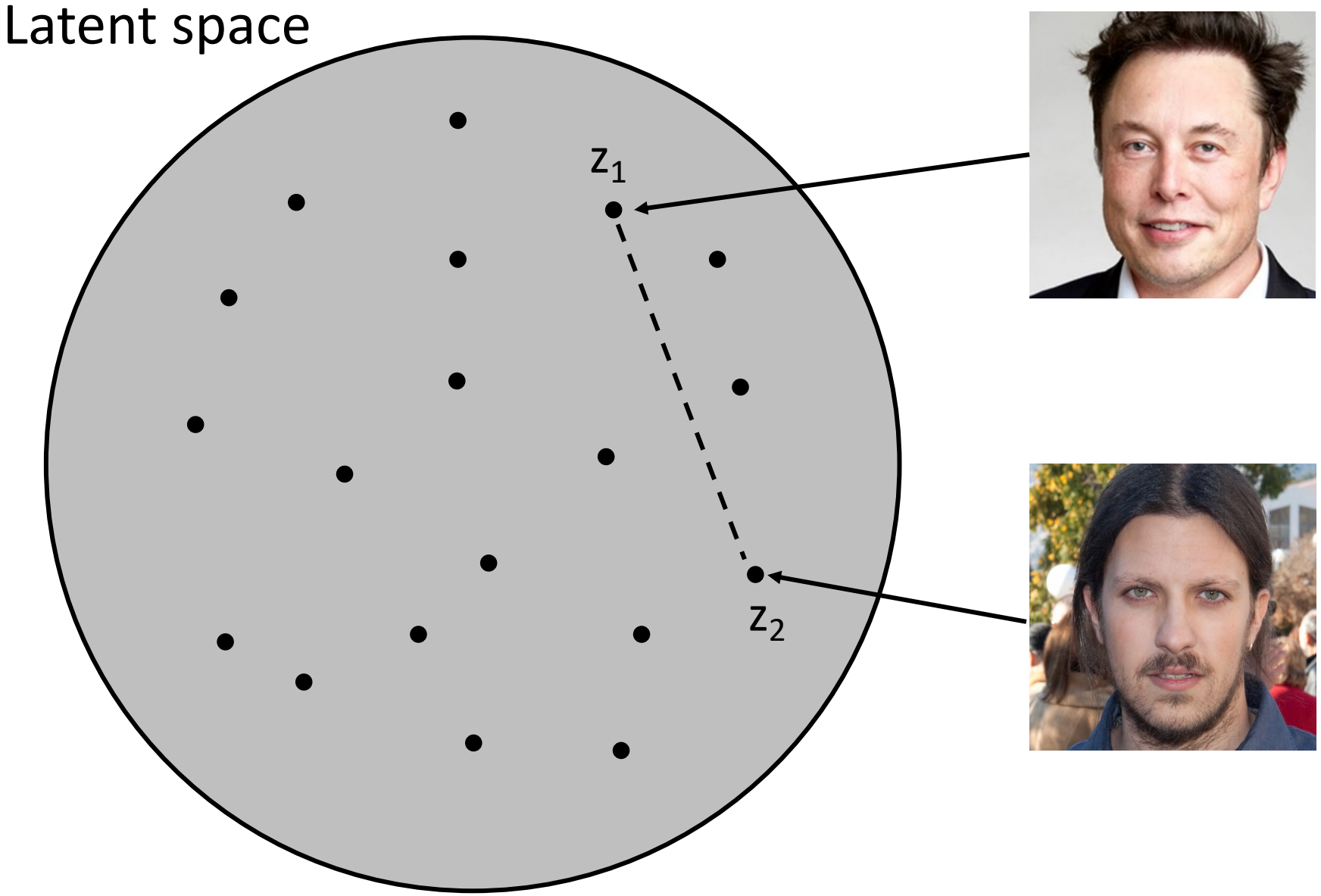


https://github.com/happy-jihye/FFHQ-Alignment

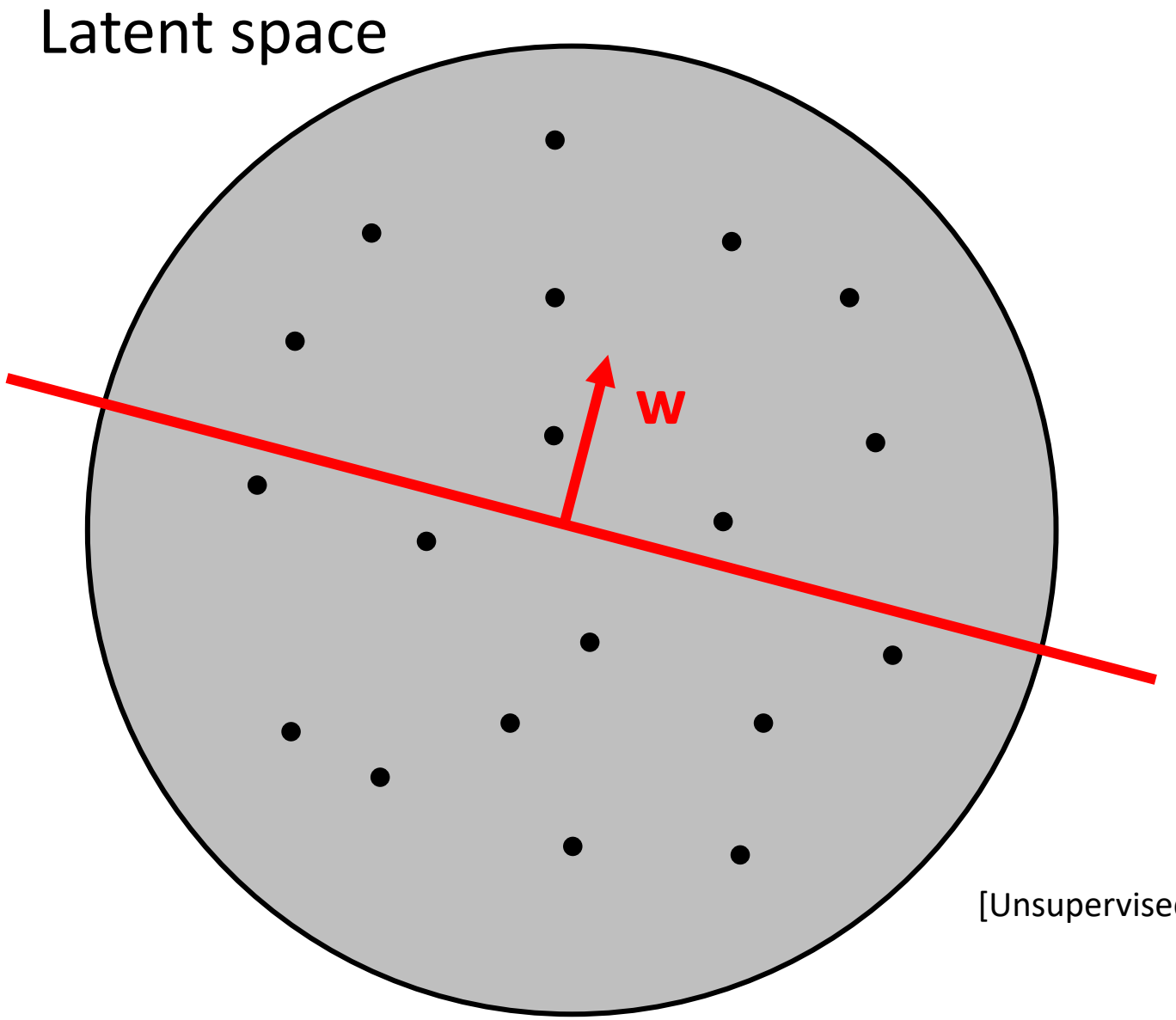
Latent directions



Latent directions



Latent directions

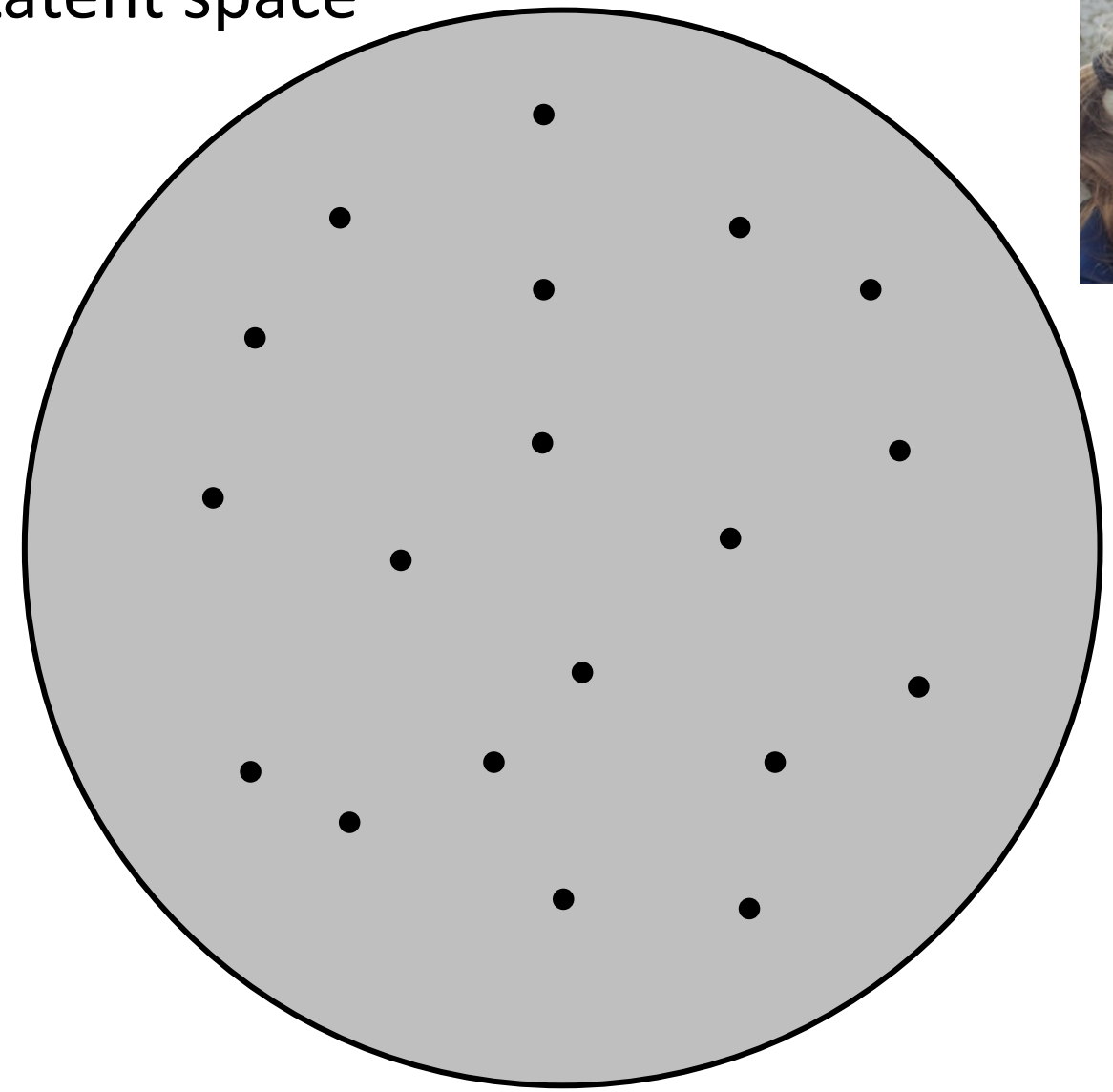


- Learn meaningful latent directions
- Either supervised or unsupervised
- Apply these directions in any image
- Example of directions in faces:
  - Aging
  - Smiling
  - Hair or skin color
  - Gender

[Unsupervised: Voynov and Babenko. "Unsupervised discovery of interpretable directions in the gan latent space." In ICML 2020]

Supervised Learning of Latent directions

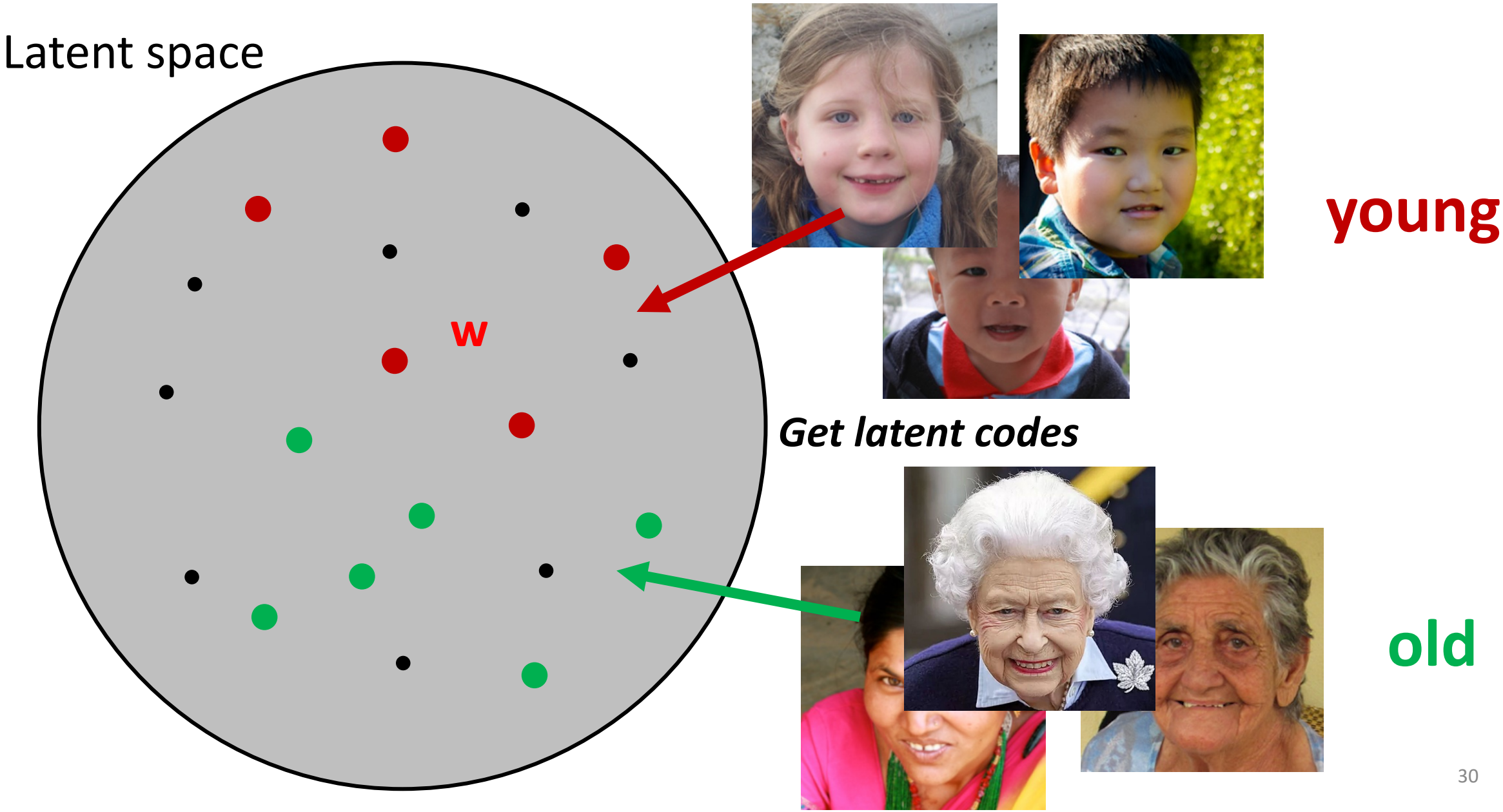
Latent space



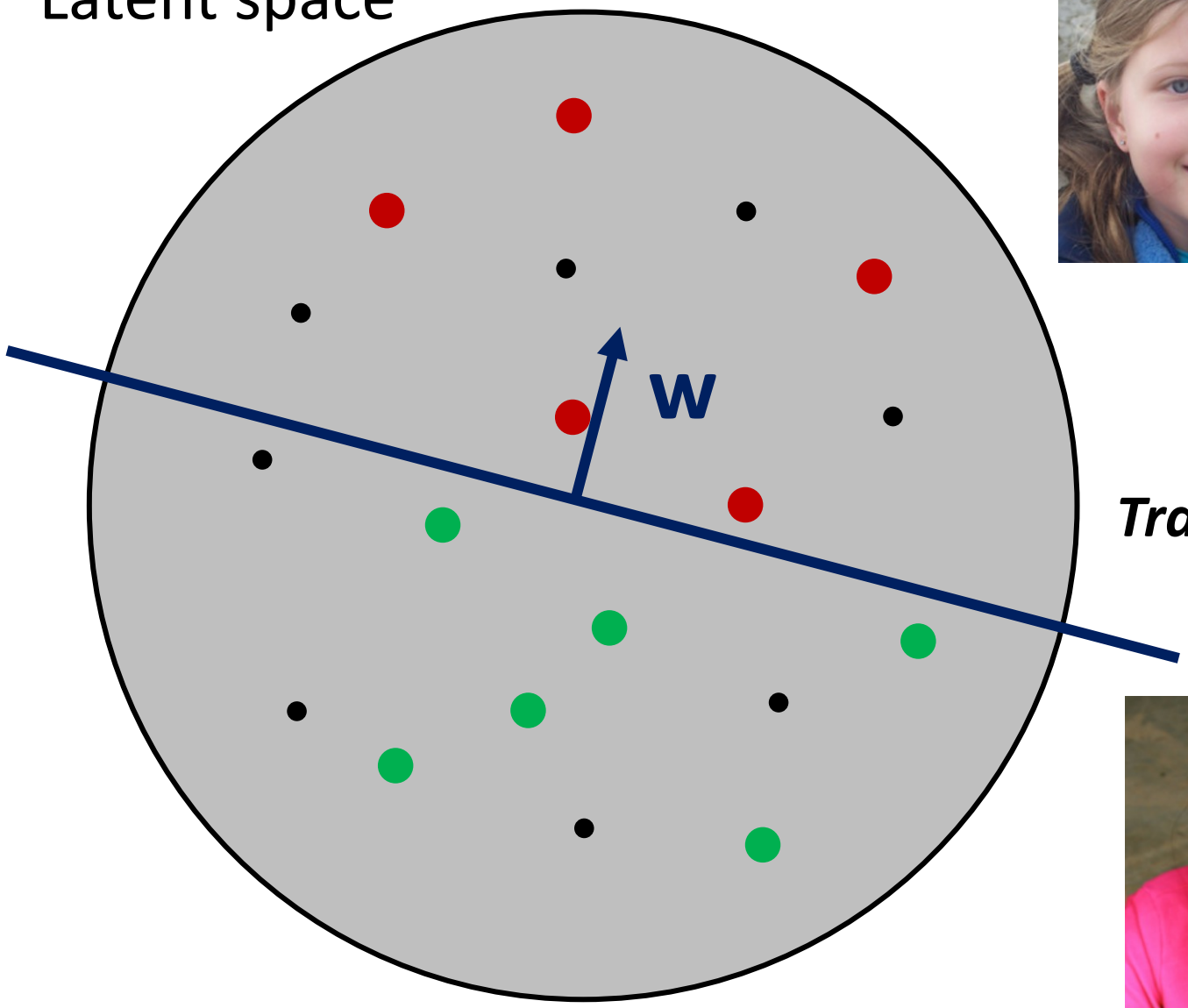
young

old

Supervised Learning of Latent directions



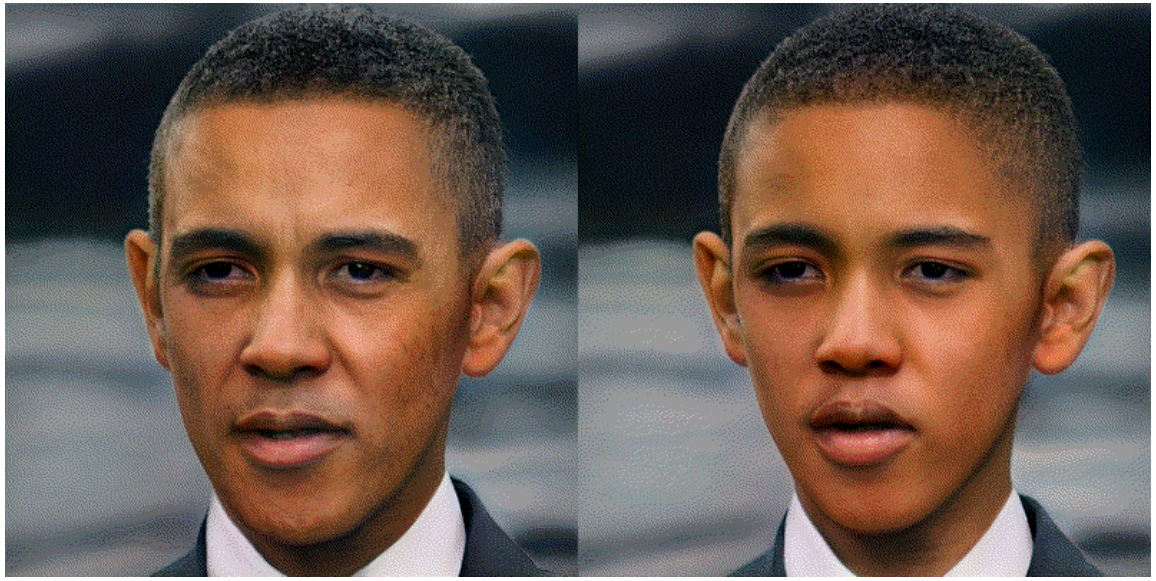
Supervised Learning of Latent directions



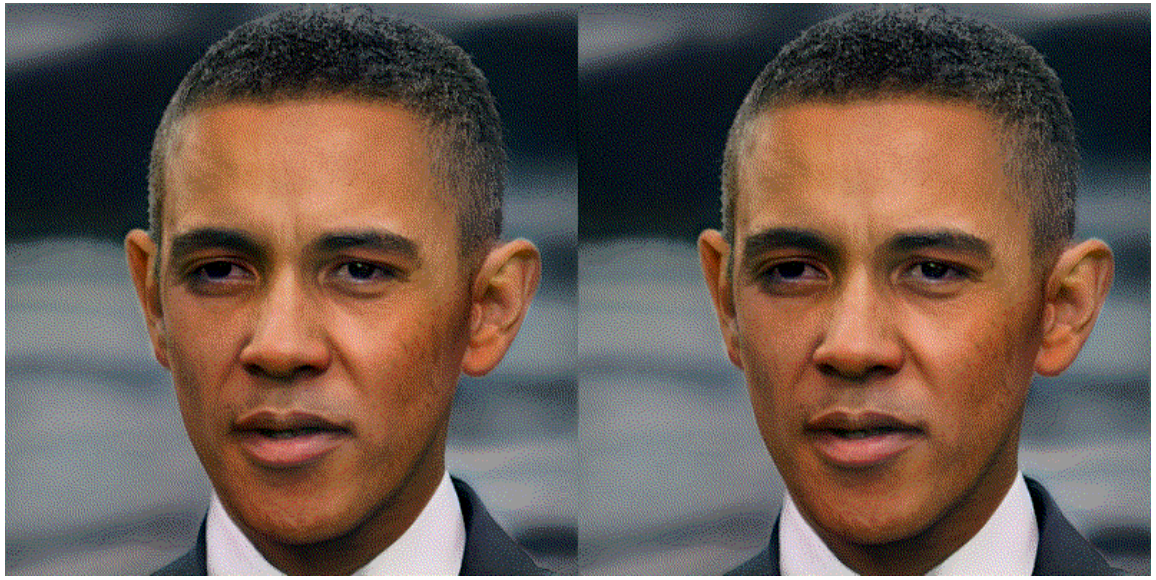
Train a linear classifier on the latent codes

Supervised Learning of Latent directions

aging

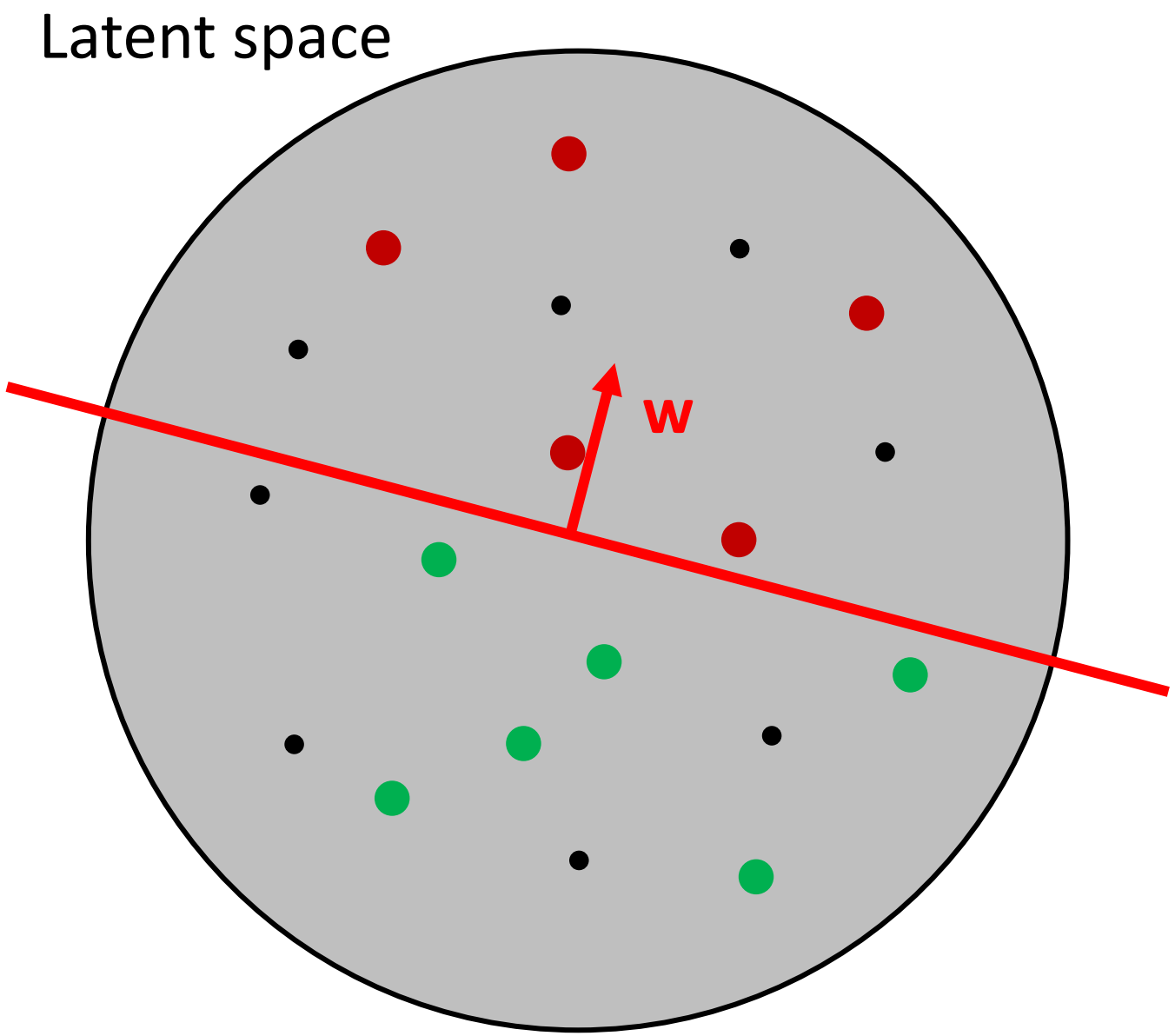


smiling



$z_{new} = z + w*m$

Supervised Learning of Latent directions



$z_{new} = z + w*m$

m:magnitude

DALLE

TEXT PROMPT an armchair in the shape of an avocado. an armchair imitating an avocado.

AI-GENERATED IMAGES

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

DALLE

TEXT PROMPT

a soap dispenser in the style of a pikachu. a soap dispenser imitating a pikachu.

AI-GENERATED IMAGES

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

DALLE 2

TEXT DESCRIPTION

An astronaut Teddy bears A bowl of soup

mixing sparkling chemicals as mad scientists shopping for groceries working on new AI research

as kids' crayon art on the moon in the 1980s underwater with 1990s technology

→

DALL·E 2

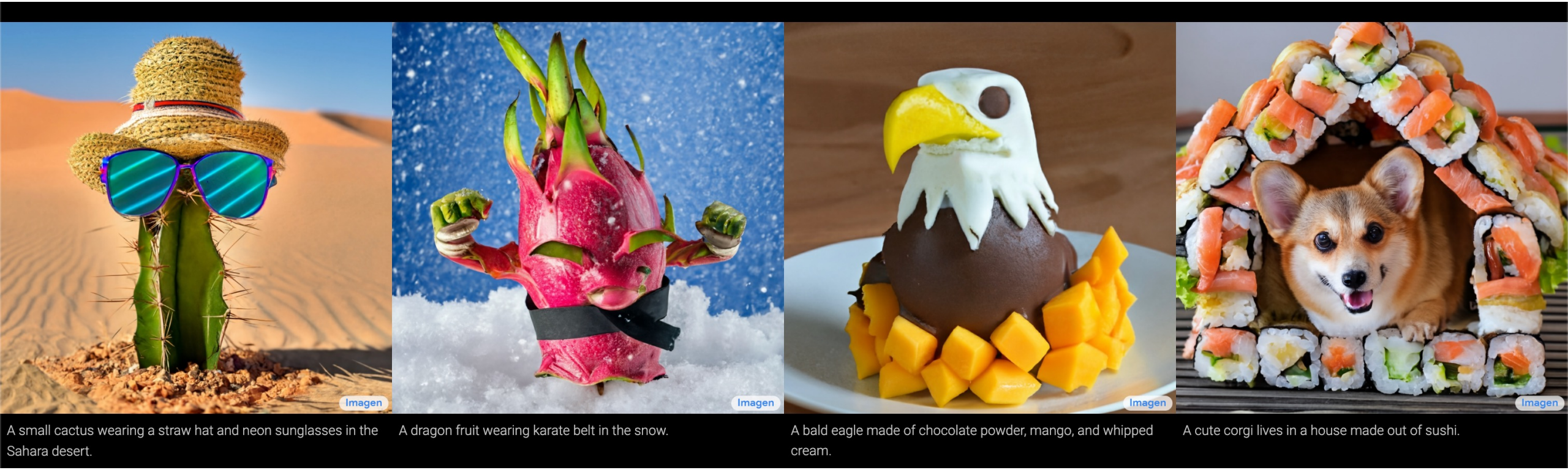
https://openai.com/dall-e-2/

Imagen by Google AI

A photo of a Corgi dog riding a bike in Times Square. It is wearing sunglasses and a beach hat.

https://imagen.research.google/

Imagen by Google AI



A small cactus wearing a straw hat and neon sunglasses in the Sahara desert.

A dragon fruit wearing karate belt in the snow.

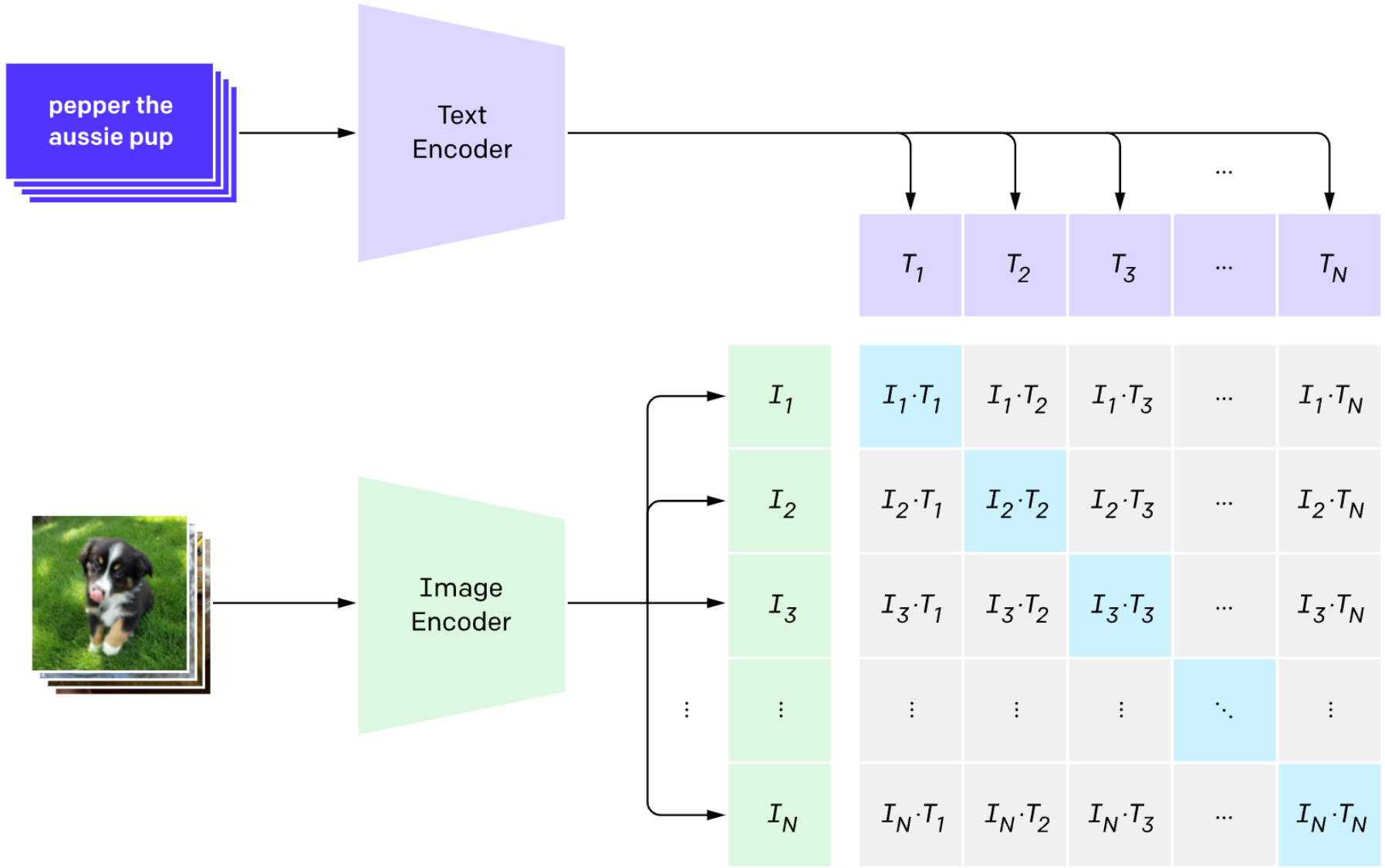
A bald eagle made of chocolate powder, mango, and whipped cream.

A cute corgi lives in a house made out of sushi.

https://imagen.research.google/

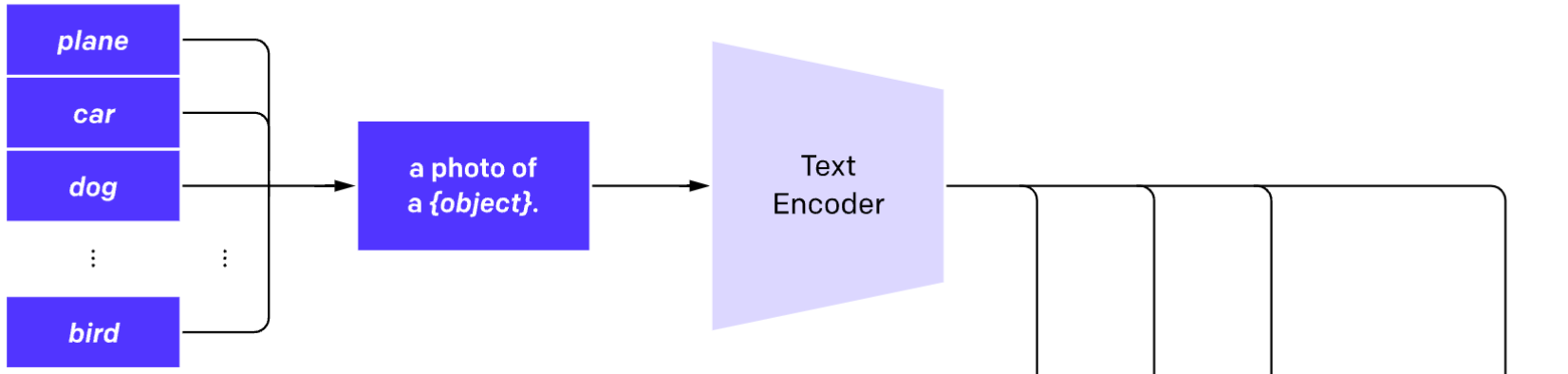
CLIP

1. Contrastive pre-training



CLIP

2. Create dataset classifier from label text



3. Use for zero-shot prediction

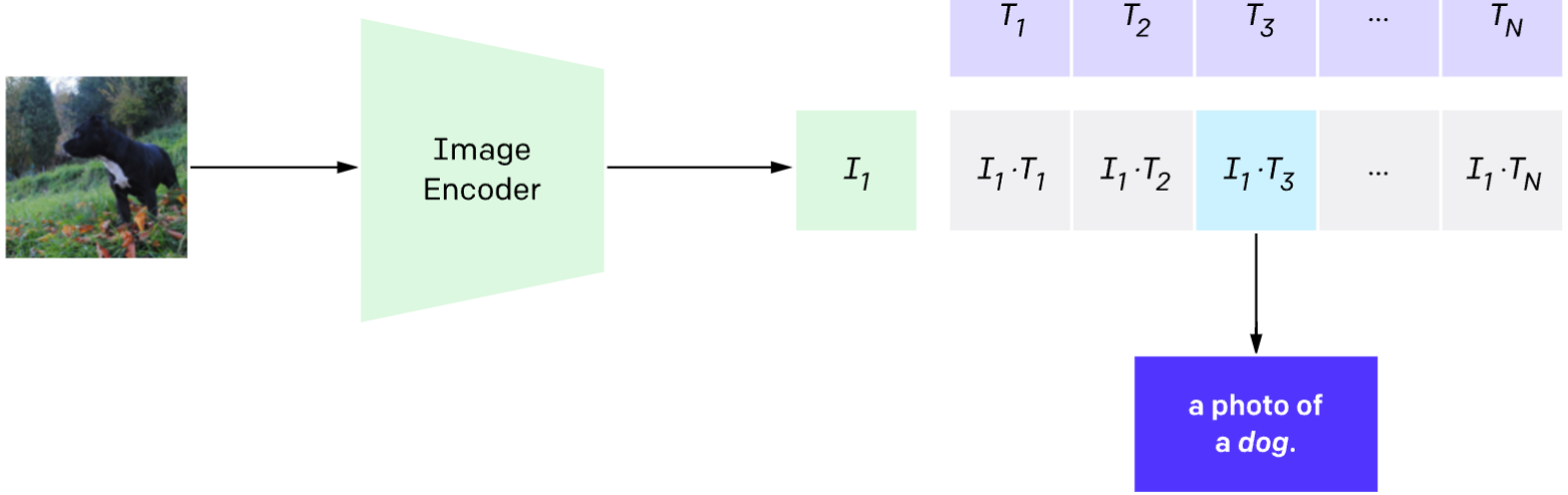


Image generation from text

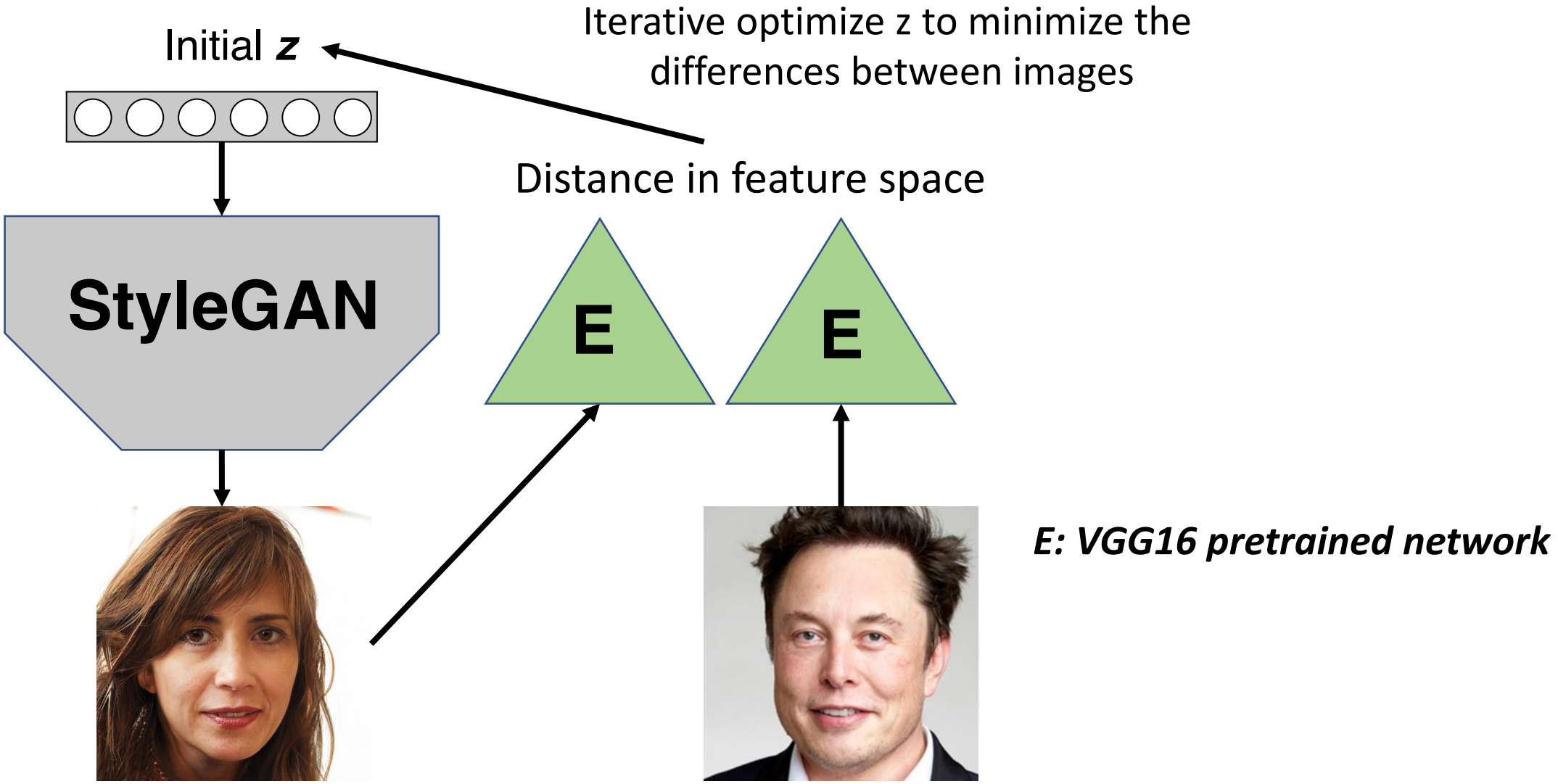


Image generation from text

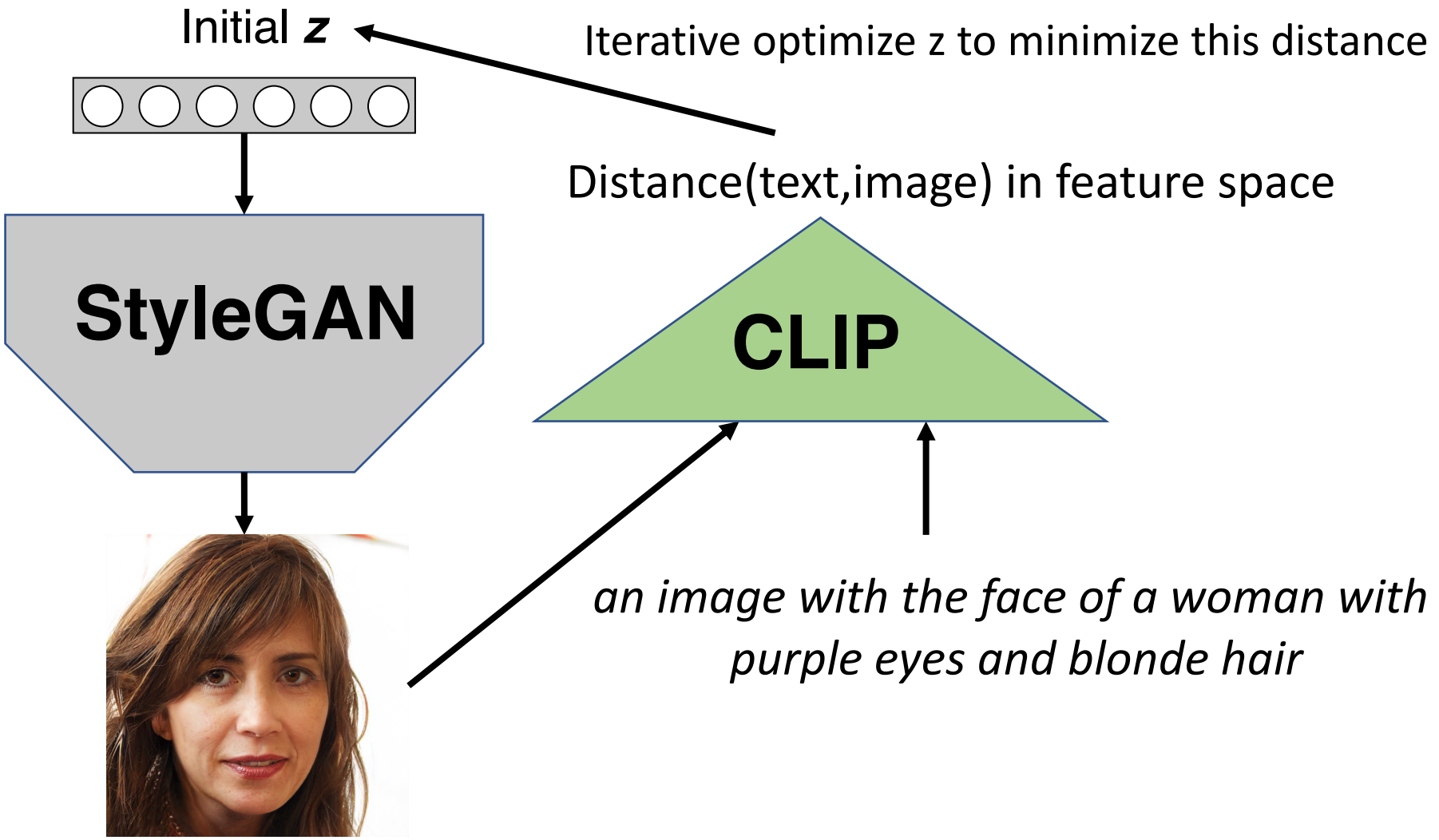


Image generation from text

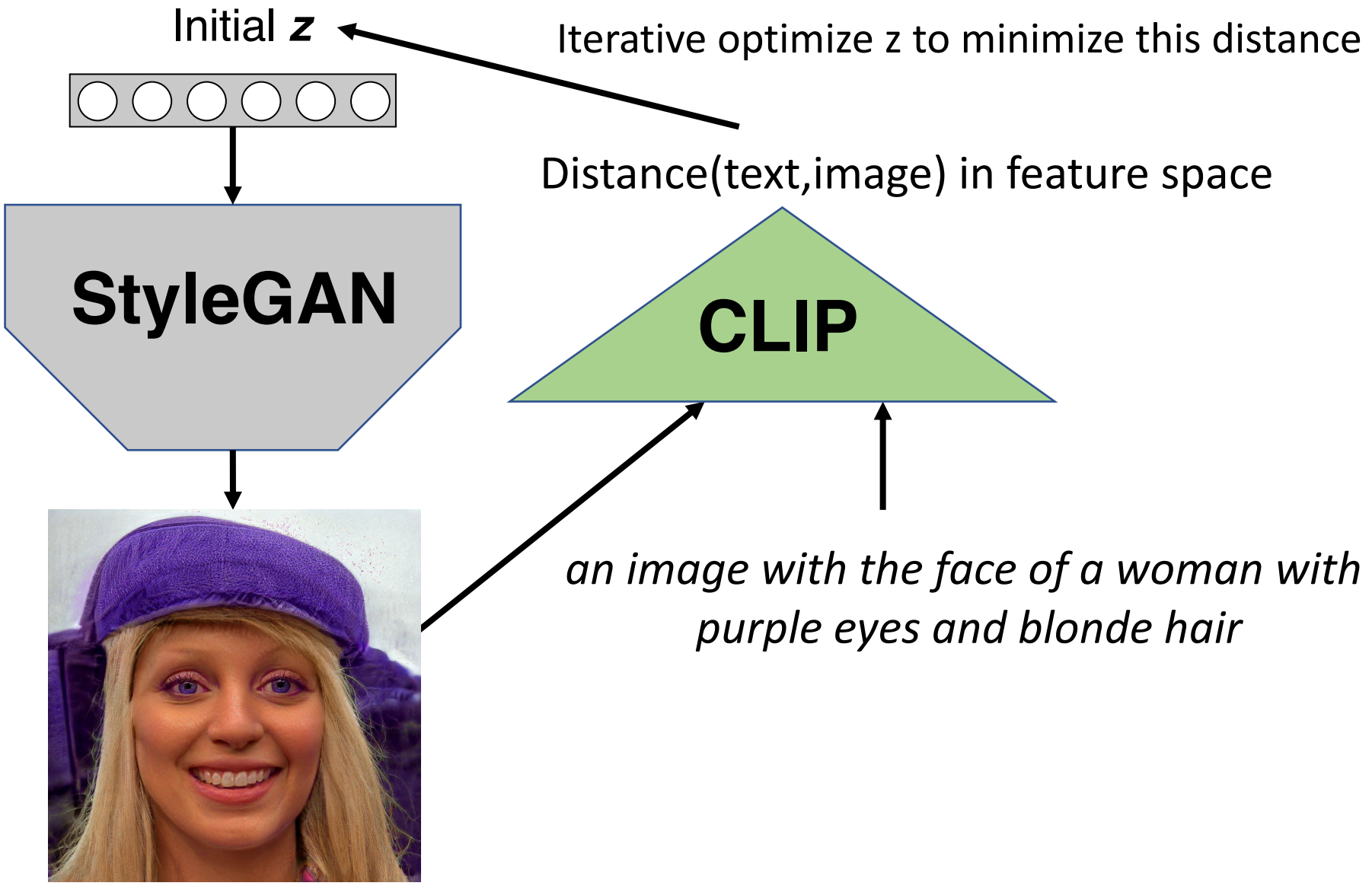
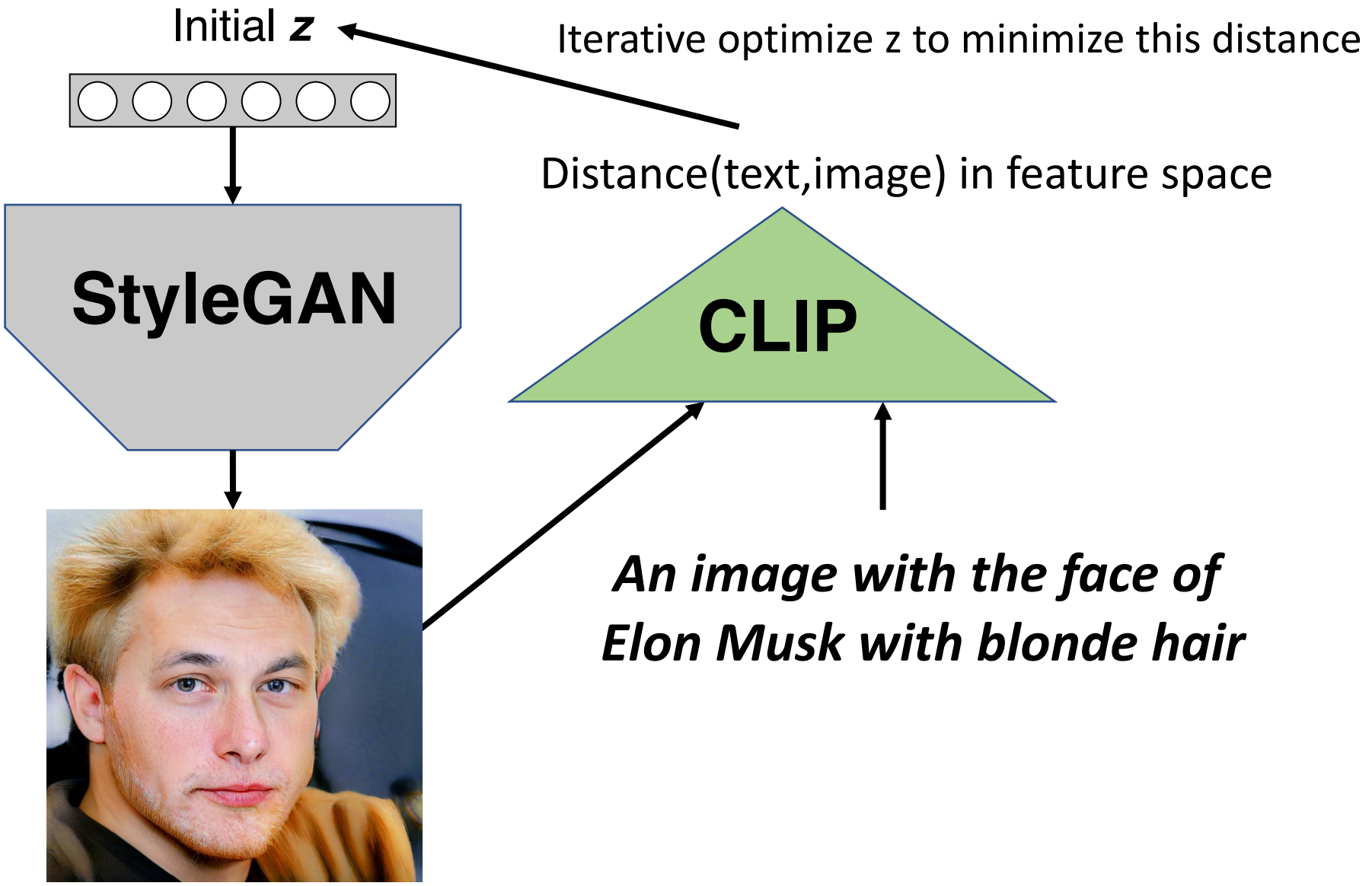


Image generation from text



Project 2

Generative Adversarial Networks

Project 3

Deep Learning in Computer Vision

June 2022

The main goal of this project is to understand and manupulate the latent space of a GAN model.

MNIST

You have also been working on creating networks that can generate digits from the MNIST dataset. If you are in need of more content for your poster, you are allowed to also include these on your poster. For this you could: show results from these networks, describe their architectures, the training process, and describe your process of making these choices.

Pre-trained GAN

In this project, you will work with a pretrained StyleGAN2 model which was trained on the FFHQ dataset. The pretrained network can be found here: https://nvlabs-fi-cdn.nvidia.com/stylegan2-ada-pytorch/pretrained/ffhq.pkl. If you want to explore pre-trained StyleGAN2 models on other domains (e.g., cats, dogs, art faces), you can find these weights here: https://nvlabs-fi-cdn.nvidia.com/stylegan2-ada/pretrained/

Useful links

- StyleGAN2-ADA official implementation in pytorch: https://github.com/NVlabs/stylegan2-ada
- Align images to FFHQ: https://github.com/happy-jihye/FFHQ-Alignment
- Pre-trained latent directions (e.g. age, gender, smile): https://hostb.org/NCM

Tasks

1. Generate random images from the pre-trained GAN model
2. Reconstruct your own images and get the latent codes using the projector.py script from the official stylegan2 implementation.
3. Interpolate between two real images that you reconstructed.
4. Apply some pre-trained latent directions to your reconstructed images (e.g. aging, smiling, gender)
5. Learn your own latent direction
6. Generate images from text prompts using CLIP (e.g. “an image with the face of a woman with purple eyes and blonde hair”)

Feedback