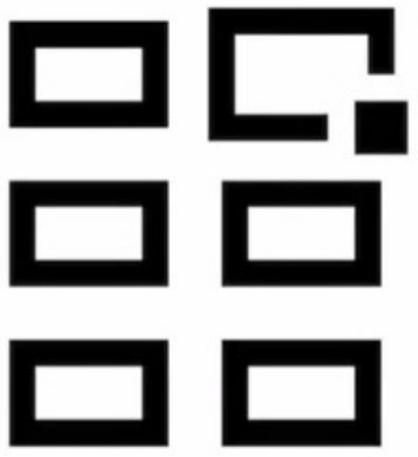
deeplearning.ai

Conditional Generation: Intuition

Outline

- Unconditional generation
- Conditional vs. unconditional generation



Unconditional Generation

Input

Generator

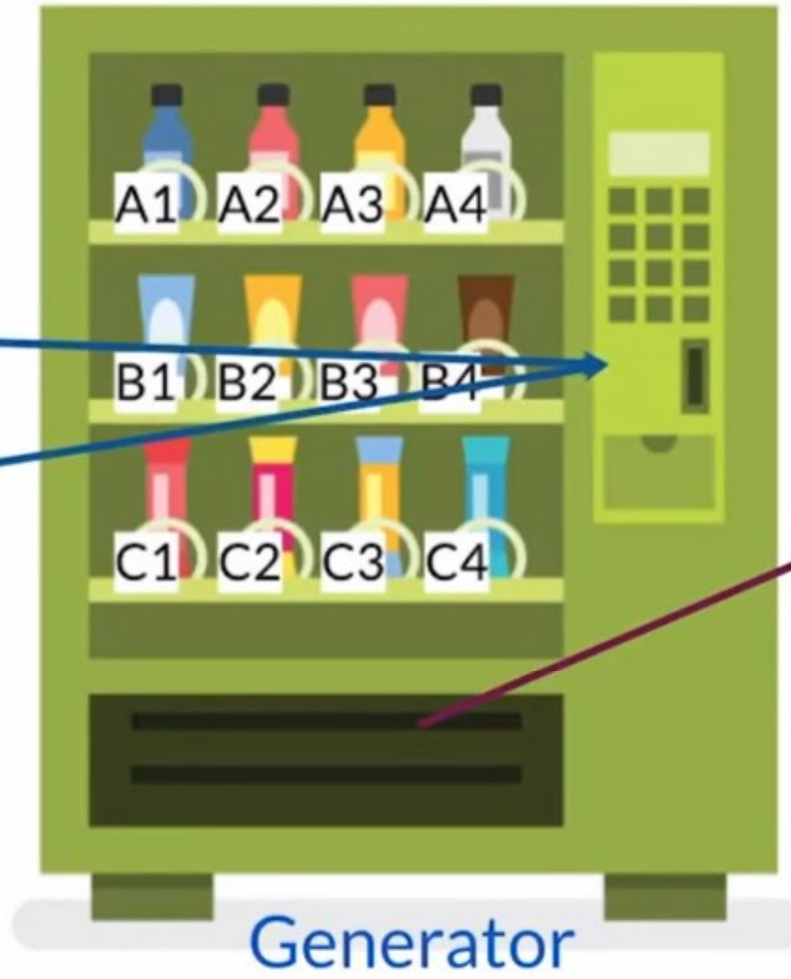
Generated Example

You get outputs from a random class

Conditional Generation

Input

Class A2



Generator

Generated Example

You get what you ask for

Conditional vs. Unconditional Generation

Conditional	Unconditional
Examples from **the classes you want**	Examples from *random classes*
Training dataset needs to be **labeled**	Training dataset *doesn't need to be labeled*

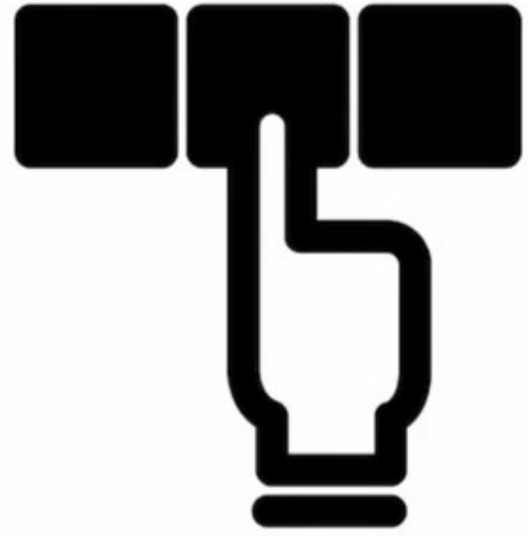
Summary

- Conditional generation requires labeled datasets
- Examples can be generated for the selected class

Conditional Generation: Inputs

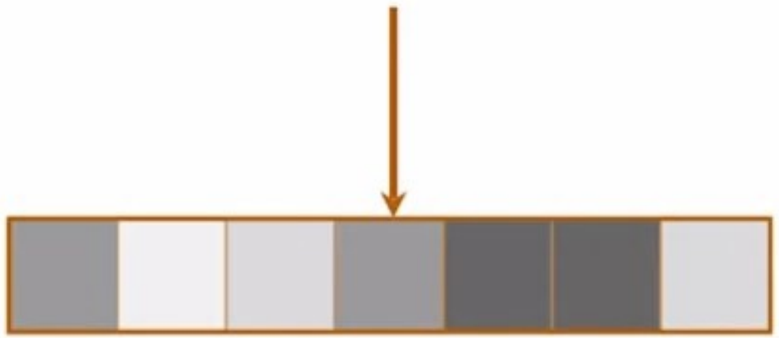
Outline

- How to tell the generator what type of example to produce
- Input representation for the discriminator



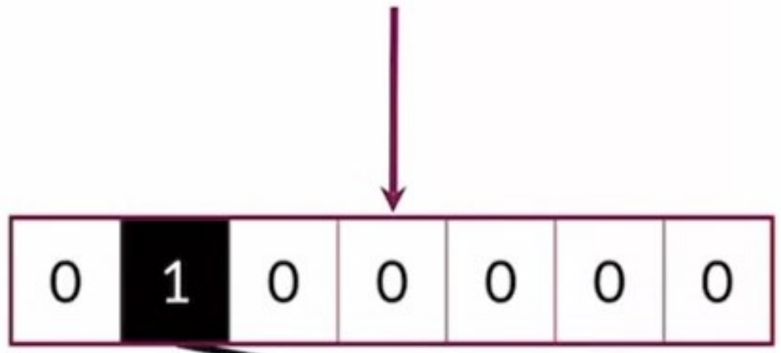
Generator Input

Noise vector



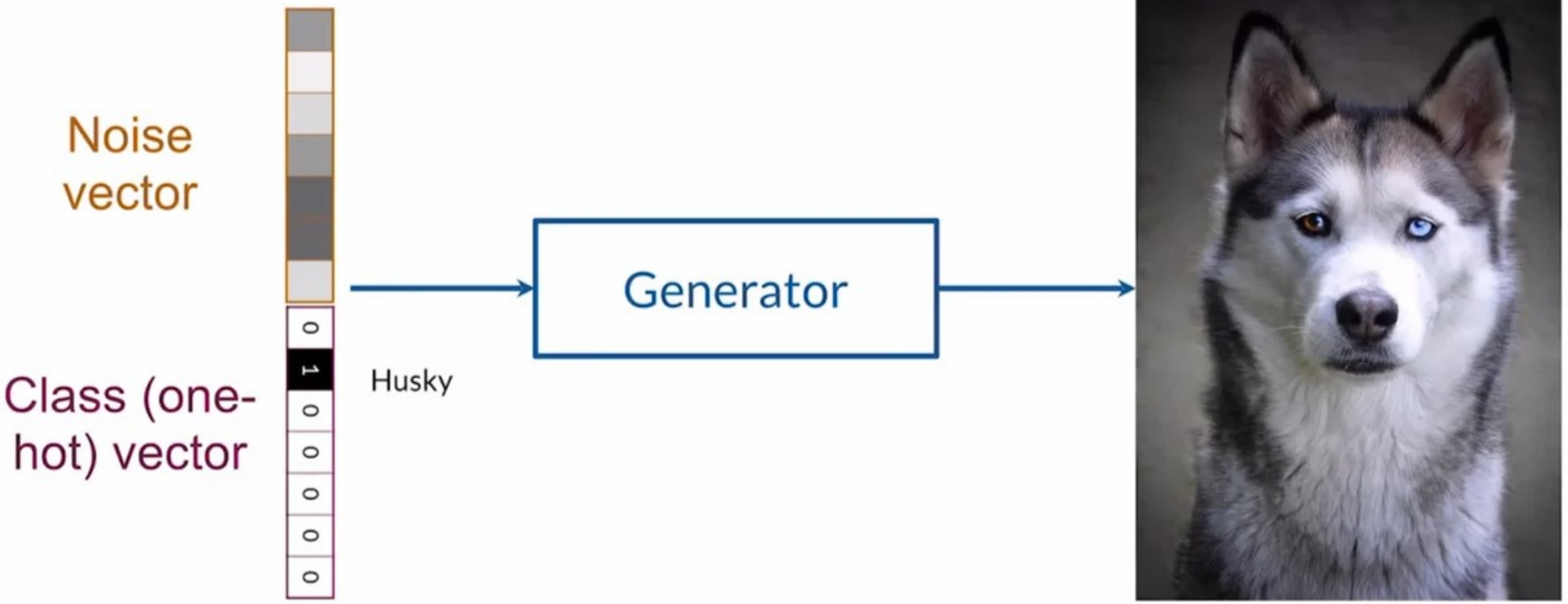
Randomness in the generation

Class (one-hot) vector



Control in the generation

Generator Input



Generator Input

Noise vector

Class (one-hot) vector

0
1
0
0
0
0
0

Husky

Generator

Output

Discriminator Input

Real image of some breed other than Golden Retriever

Golden Retriever

class

Discriminator

Output

5%
Real

Real image of different class is considered as a fake image by the discriminator.

Discriminator Input

Golden Retriever

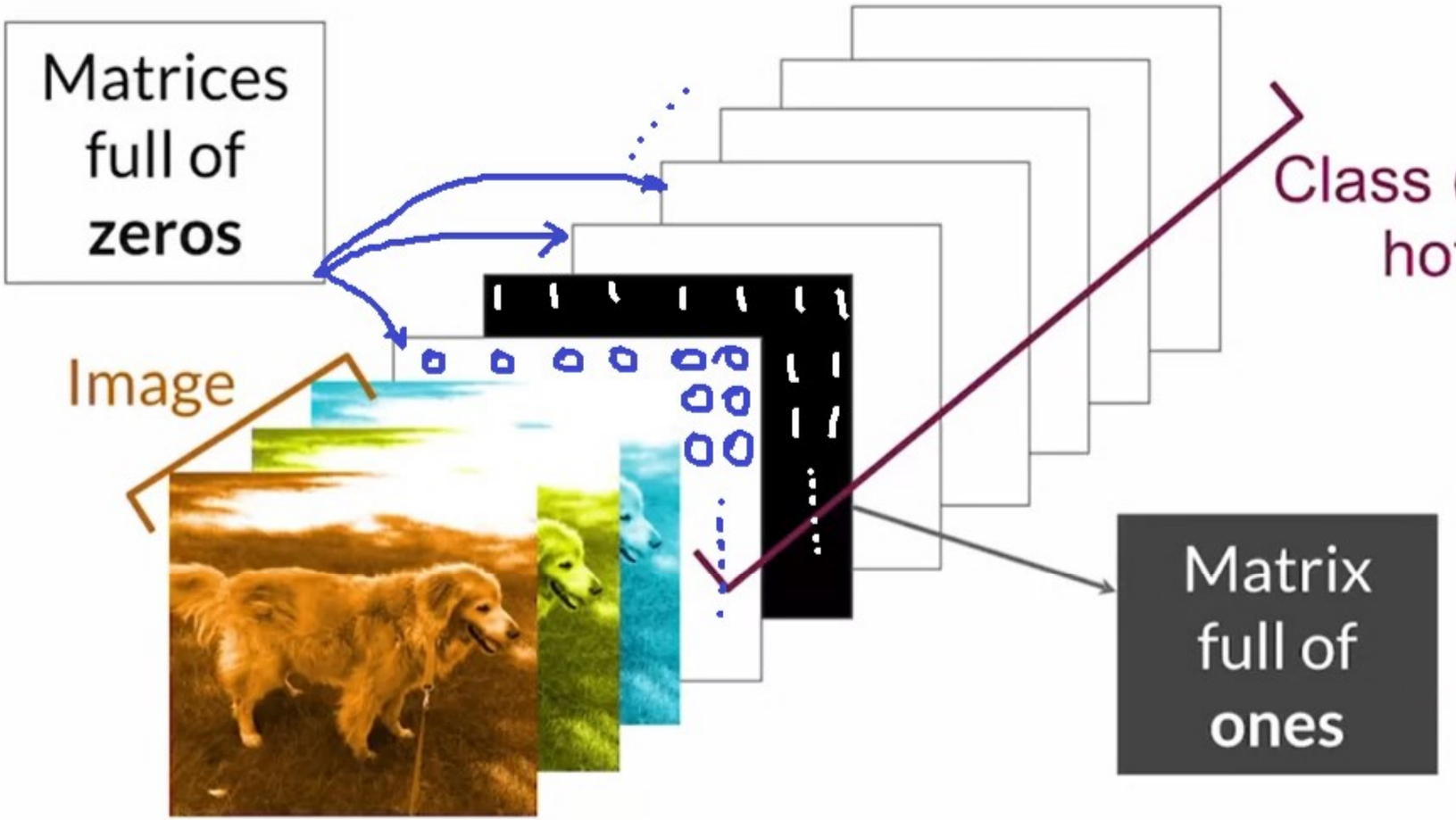
Discriminator

Output

98%
Real

Discriminator Input

We can feed the one hot vector (class) as channels. Other space efficient methods may also be used especially if there are lots of



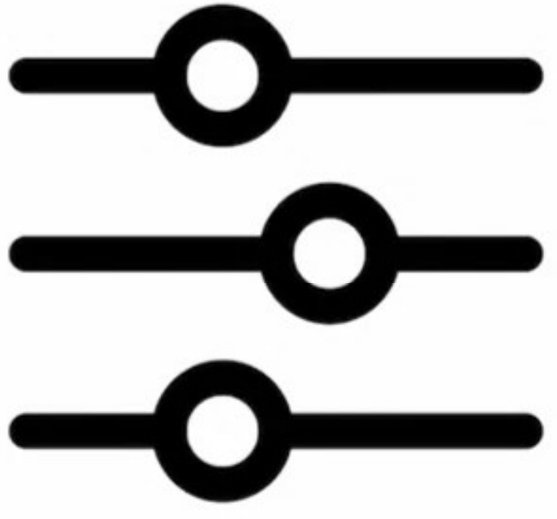
Summary

- The class is passed to the generator as one-hot vectors
- The class is passed to the discriminator as one-hot matrices
- The size of the vector and the number of matrices represent the number of classes

Controllable Generation

Outline

- What is controllable generation
- How it compares to conditional generation



Controllable Generation

Change specific features of the output

Available from: https://arxiv.org/abs/1907.10786

Controllable Generation

Noise vector

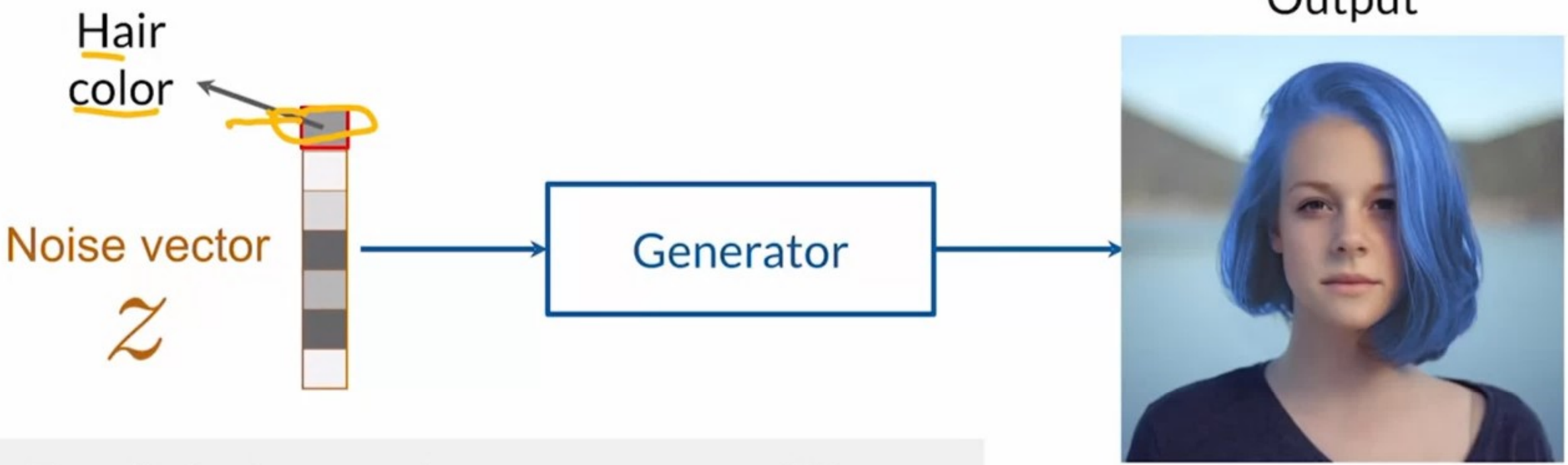
z

Generator

Controlled Output

Tweak the input noise vector to get different features on the output

Controllable Generation



Tweak the input noise vector to get different features on the output

Controllable Generation vs. Conditional Generation

Controllable	Conditional
Examples with the **features that you want**	Examples from *the classes you want*
Training dataset **doesn't need to be labeled**	Training dataset *needs to be labeled*
Manipulate the *z* vector input	*Append a class vector* to the input

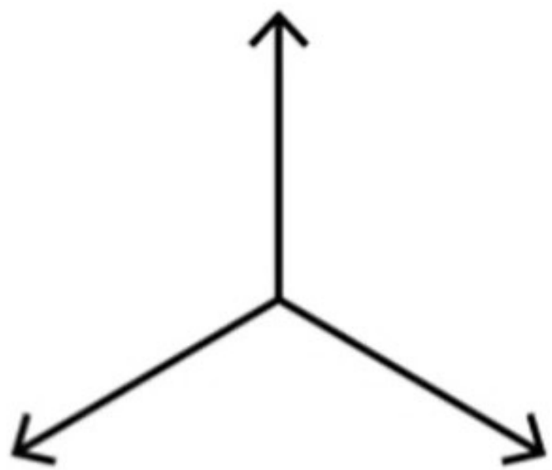
Summary

- Controllable generation lets you control the features of the generated outputs
- It does not need a labeled training dataset
- The input vector is tweaked to get different features on the output

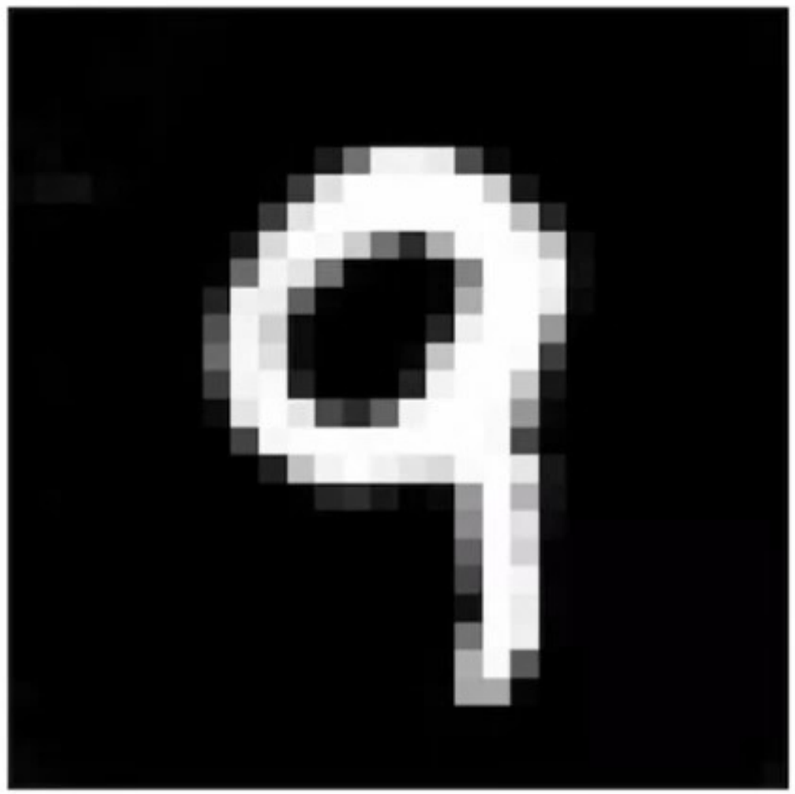
Vector Algebra in the Z-Space

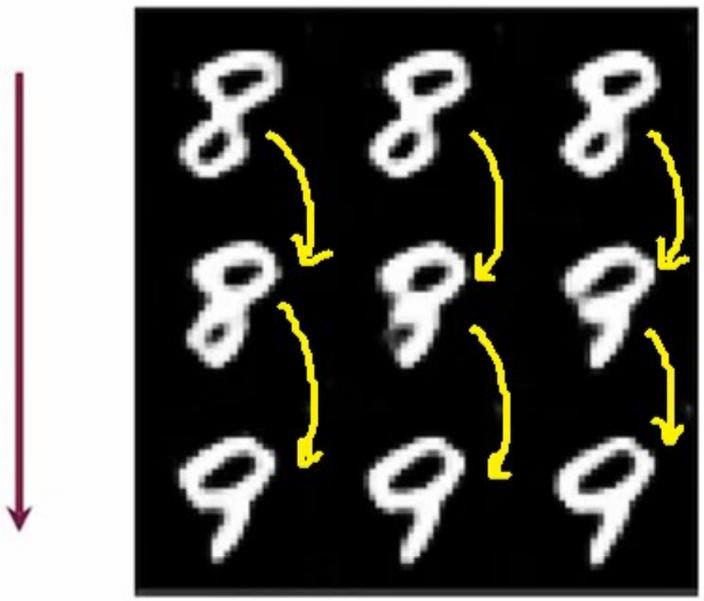
Outline

- Interpolation in the *Z*-space
- Modifying the noise vector *z* to control desired features



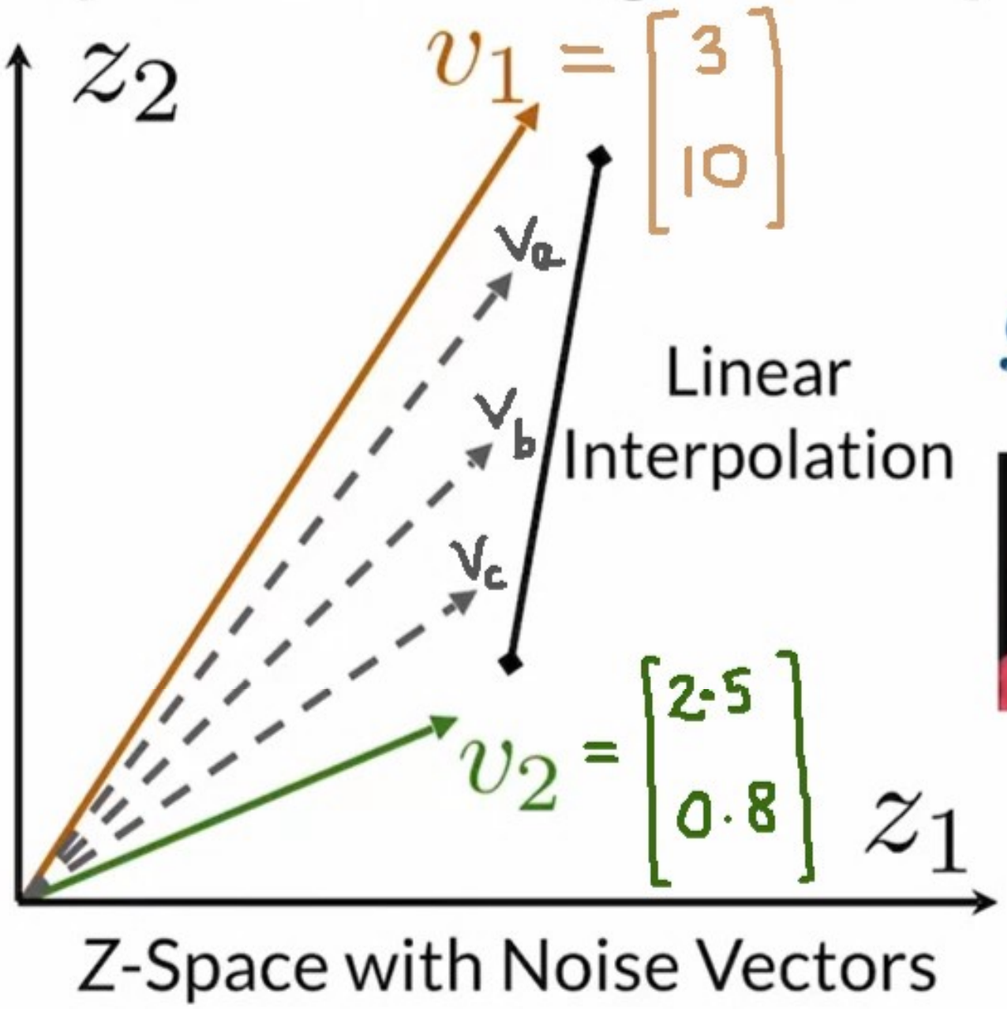
Interpolation Using the Z-Space

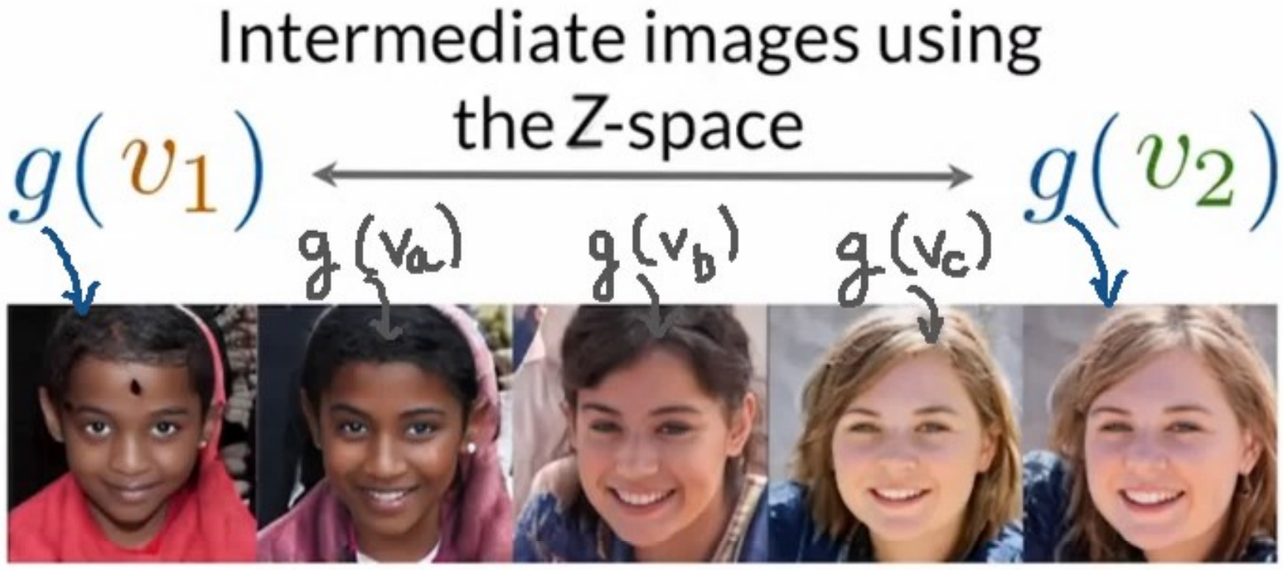




How an image morphs into another

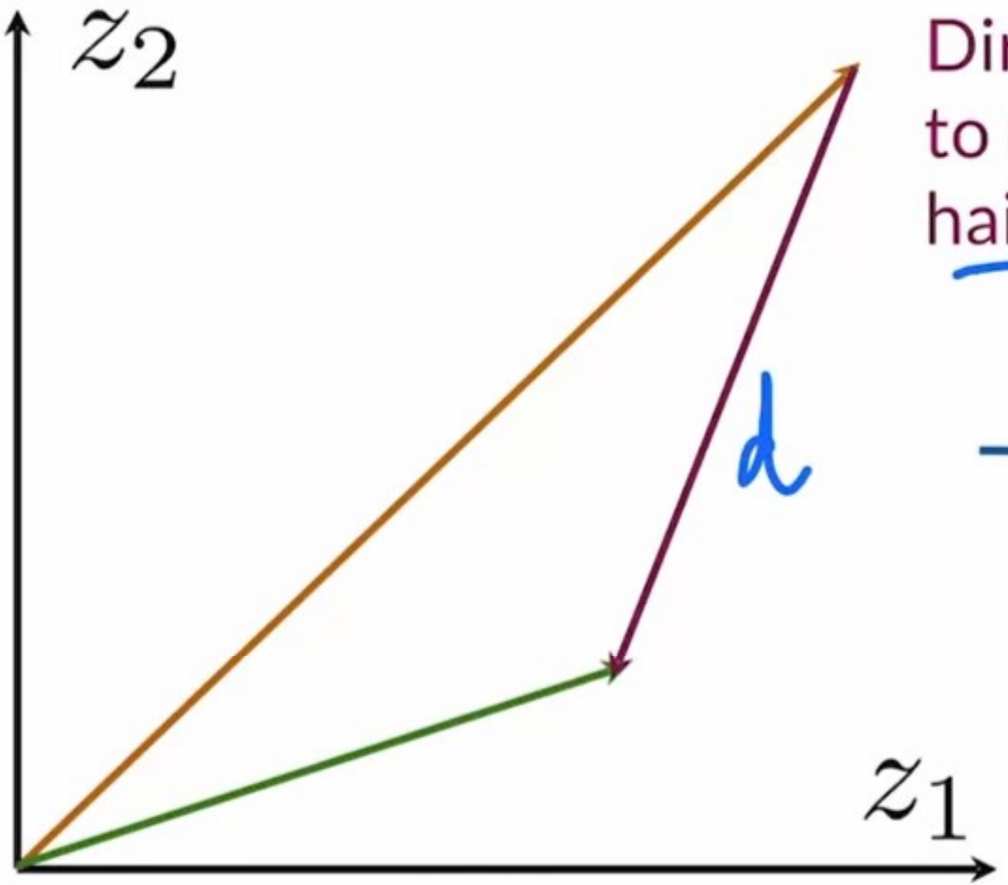
Interpolation Using the Z-Space



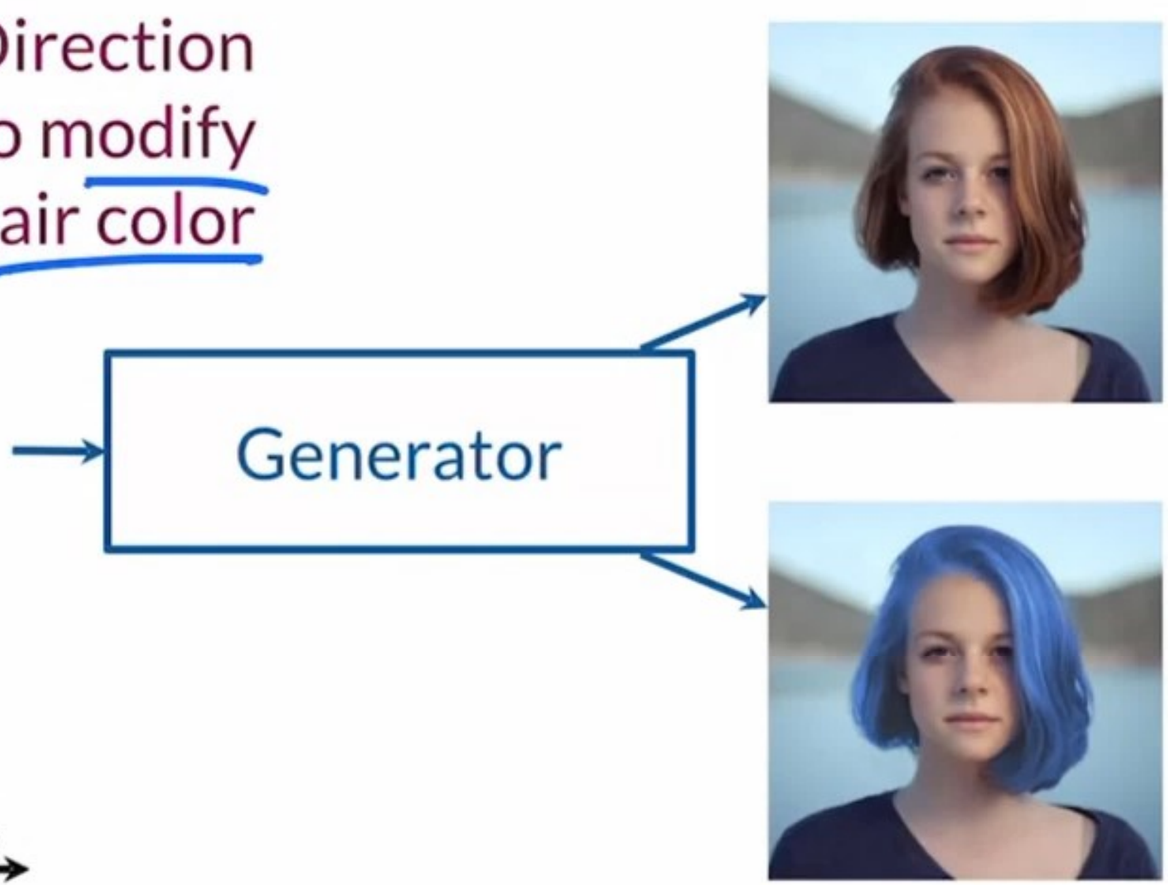


Identify the direction to control generation by hair color.

Z-Space and Controllable Generation

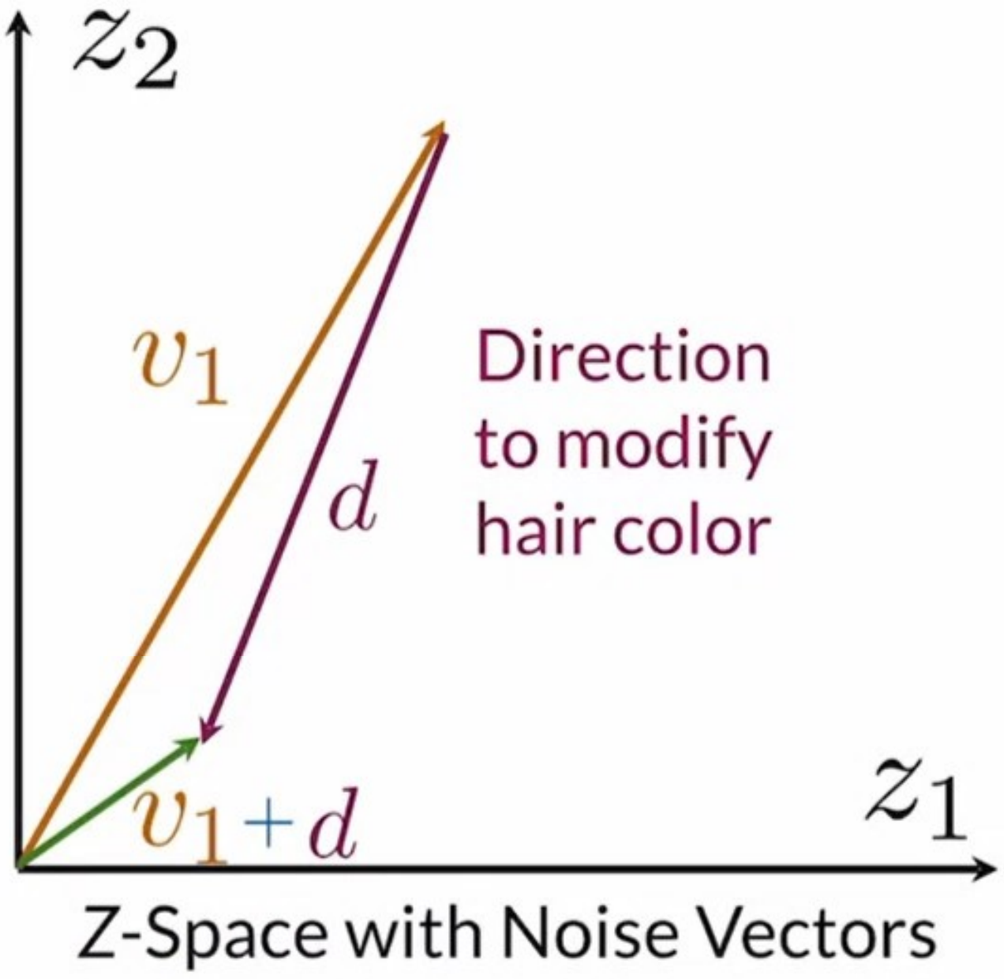


Z-Space with Noise Vectors



Output

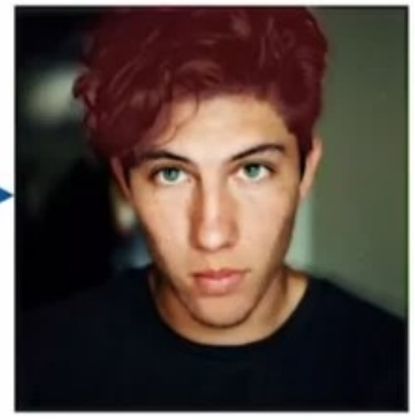
Z-Space and Controllable Generation



Z-Space with Noise Vectors

Original output

$g(v_1)$



Controlled output

$g(v_1+d)$

Summary

- To control output features, you need to find directions in the *Z*-space
- To modify your output, you move around in the *Z*-space

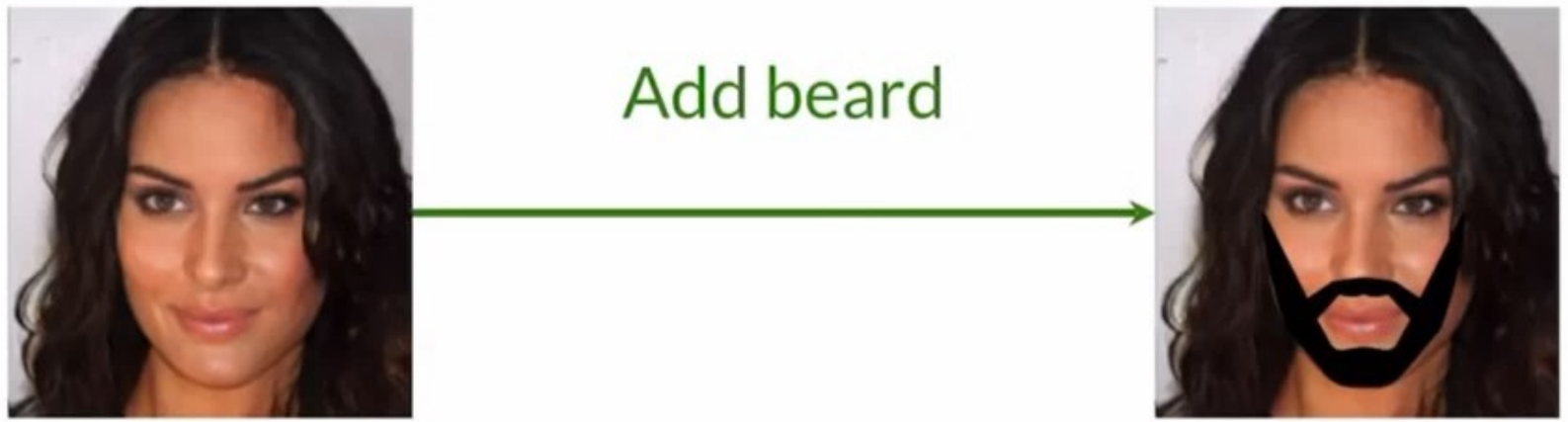
Challenges with Controllable Generation

Outline

- Output feature correlation
- Z-space entanglement

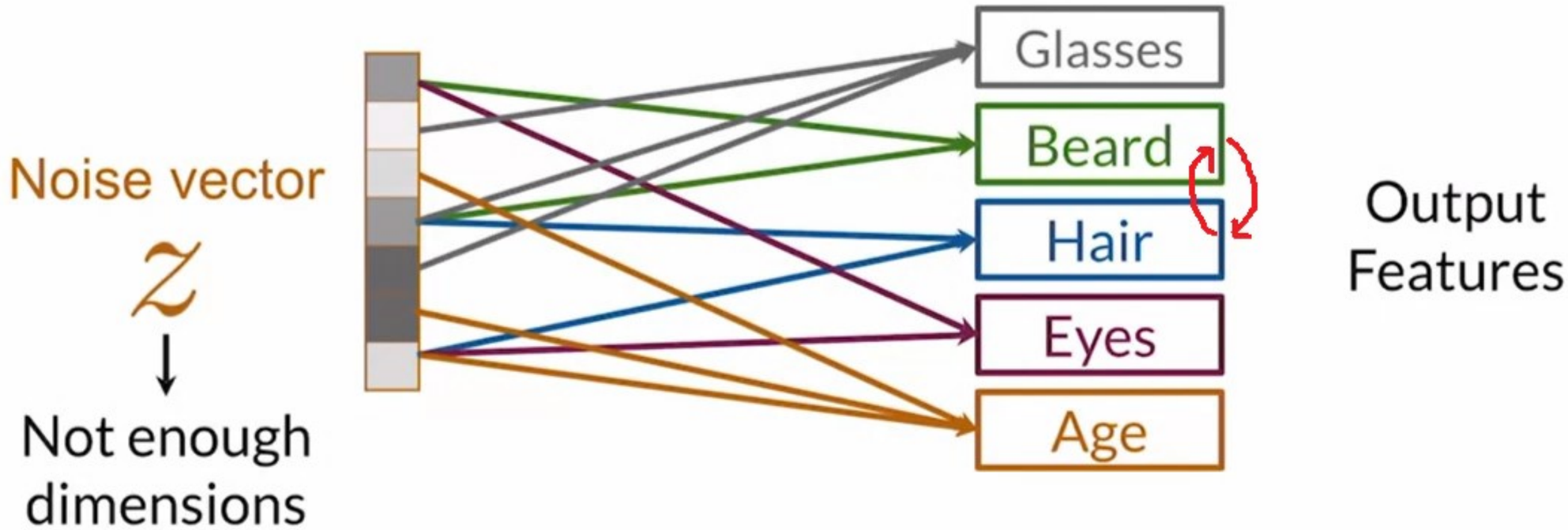
Feature Correlation

Uncorrelated Features



Correlated Features

Z-Space Entanglement



Changing beard will end up changing other features.

It is not possible to control single output features

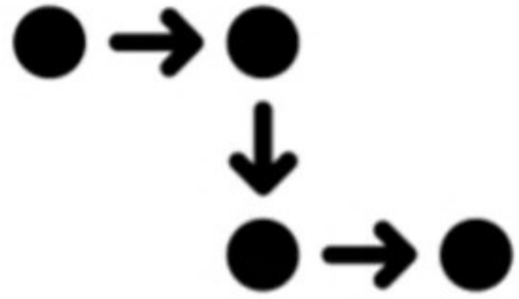
Summary

- When trying to control one feature, others that are correlated change
- *Z*-space entanglement makes controllability difficult, if not impossible
- Entanglement happens when *z* does not have enough dimensions

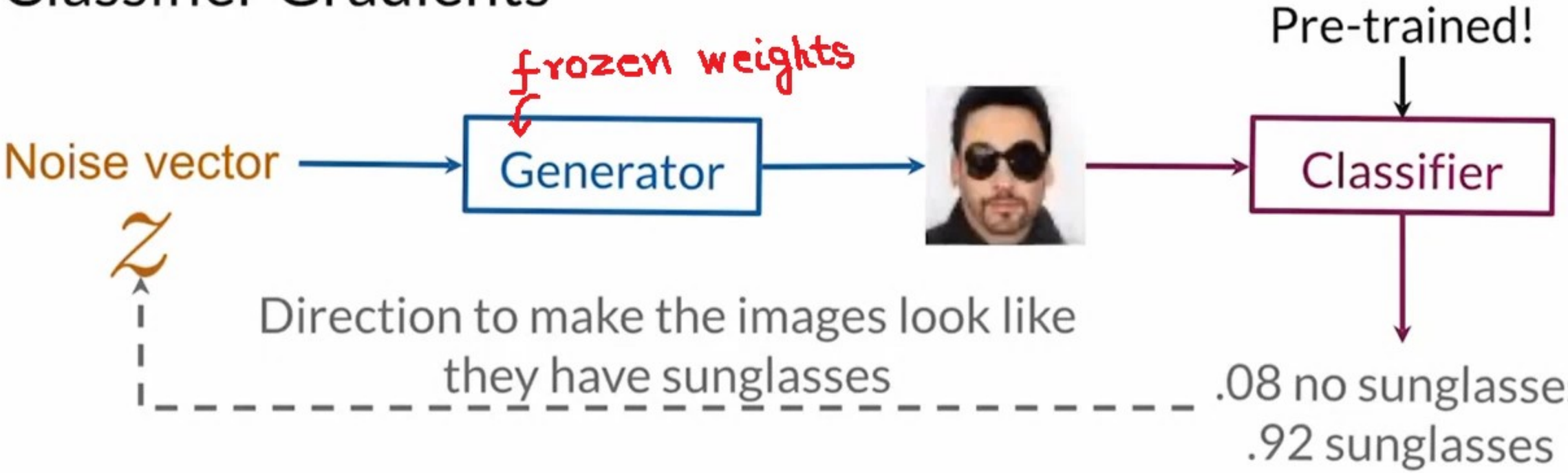
Classifier Gradients

Outline

- How to use classifiers to find directions in the Z-space
- Requirements to use this method



Classifier Gradients



Modify **just** the **noise vector** until the feature emerges

Summary

- Classifiers can be used to find directions in the Z-space
- To find directions, the updates are done just to the noise vector

Disentanglement

Outline

- What a disentangled *Z*-space means
- Ways to encourage disentangled *Z*-spaces

Disentangled Z-Space

$$v_1 = [\ 1, 2, 3, \ldots\ ]$$

$$v_2 = [\ 5, 6, 7, \ldots\ ]$$

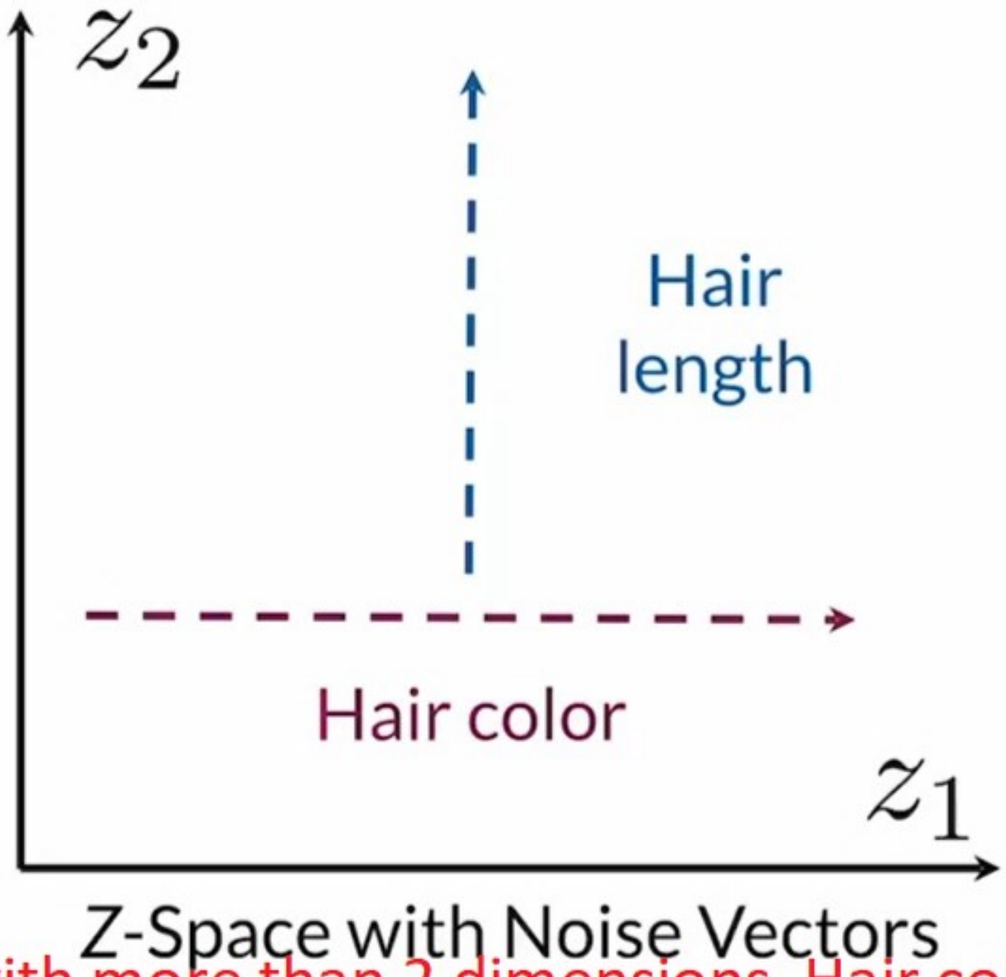
z_1 z_2

Hair color

Hair length

Latent factors of variation

The noise vectors are of higher dimensions. For simplicity we are using 2 dimensional Z-space here.



Z-Space with Noise Vectors

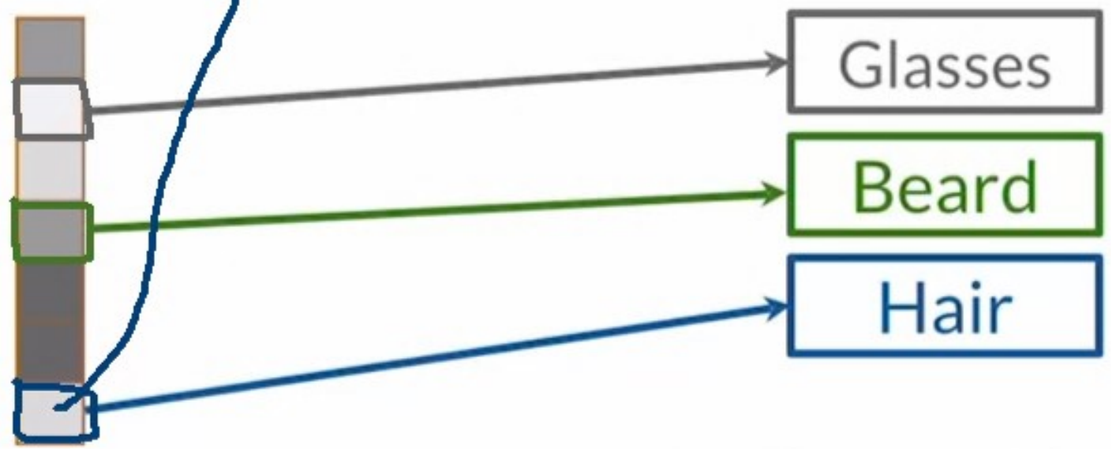
To control 2 features, we need a noise vector with more than 2 dimensions. Hair color and hair length are disentagnled but others might not be...

Disentangled Z-Space

Changing the single dimensiton will change single feature only

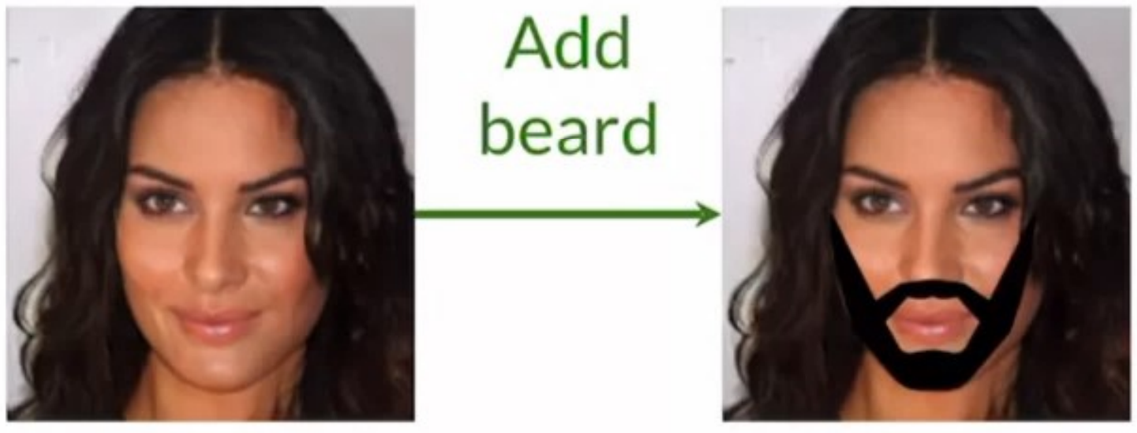
Noise vector

z



Output Features

Changes to one feature don't affect the others



Encourage Disentanglement: Supervision

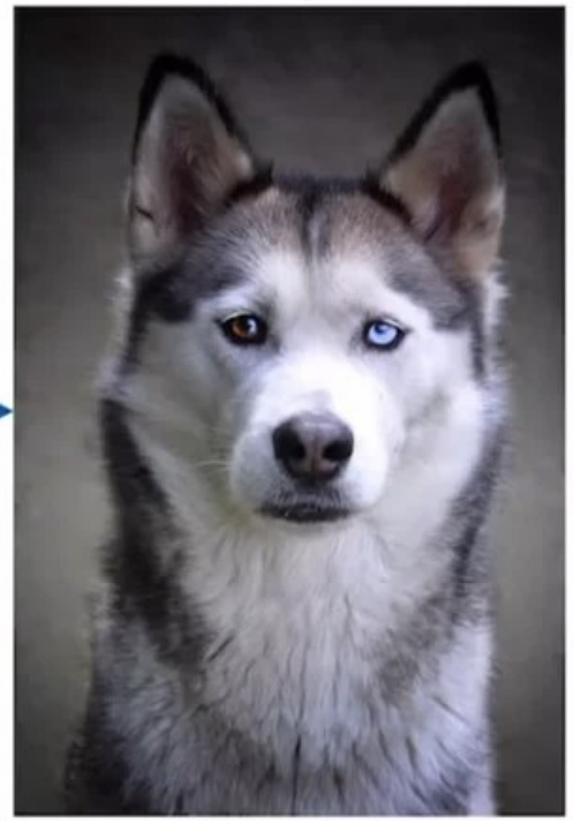
Noise vector

Husky

Difficult for continuous classes!

Generator

Output



Class (one-hot) vector

0 1 0 0 0 0

Husky

Classes are already emebedded in the noise vector.

Encourage Disentanglement: Loss Function

$$v_1 = [\ 1, 2, 3, \ldots\ ]$$

z_1 z_2

$$v_2 = [\ 5, 6, 7, \ldots\ ]$$

Output feature #1

Output feature #2

$$L_{\text{new}} = L_{\text{original}} + \text{reg}_d$$

Original loss

Regularization

Can be any loss function
(e.g. BCE, W-Loss)

Summary

- Disentangled *Z*-spaces let you control individual features by corresponding *z* values directly to them
- There are supervised and unsupervised methods to achieve disentanglement