TÜBİTAK

ODTÜ
METU

INFORMATICS INSTITUTE

EURO^2

Fundamental Concepts of Generative Machine Learning

Erdem Akagündüz, PhD
Graduate School of Informatics, METU, Türkiye

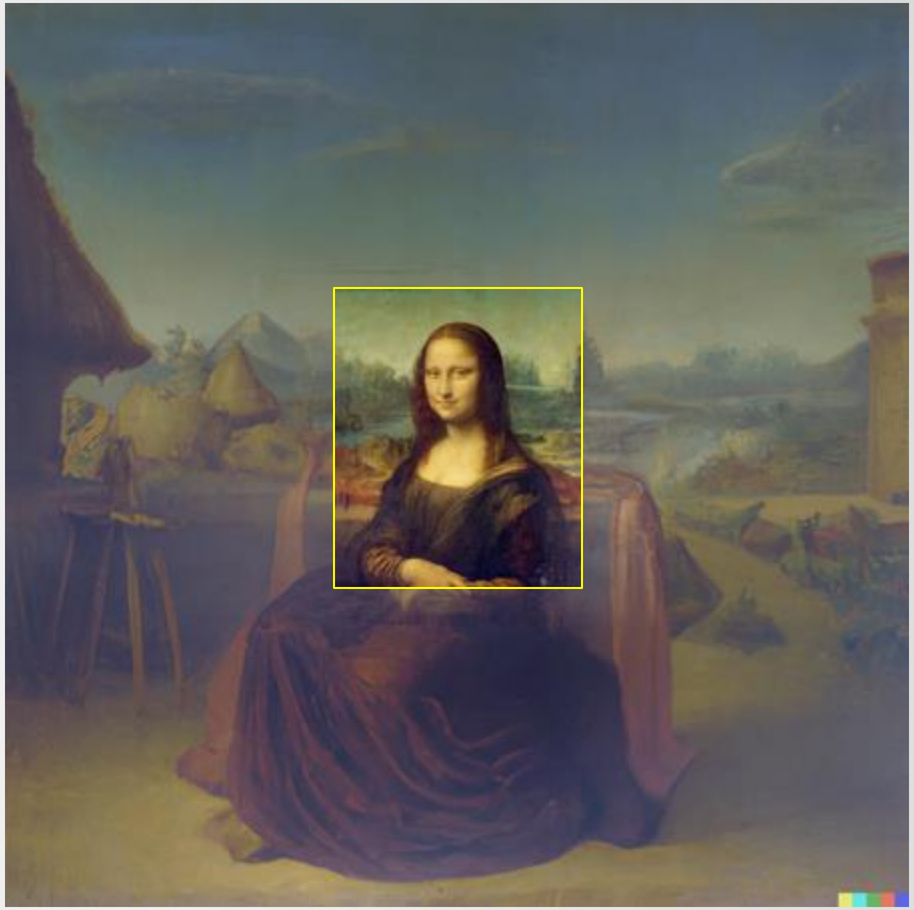
ncc@ulakbim.gov.tr

Lesson 2: Latent Spaces

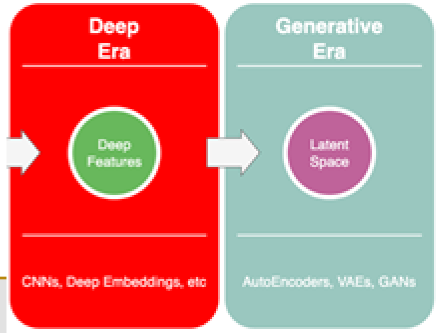
Welcome to Part II: "Latent Spaces"

This part includes two subsections:

- (The Curse of) Dimensionality, Deep Features vs. Latent Spaces
- **Latent Space properties: Continuity, Entanglement, etc**

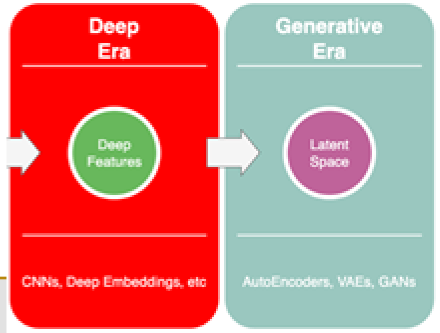


Deep Features vs Latent Space



- The difference between deep features learned by a deep encoder and a latent space of a generative model lies in their origin, representation, and usage within generative modeling.
- Let's delve into a comparative overview of deep features learned by an encoder and the latent space of a generative model, highlighting their differences and the significance of these distinctions in the context of generative modeling:

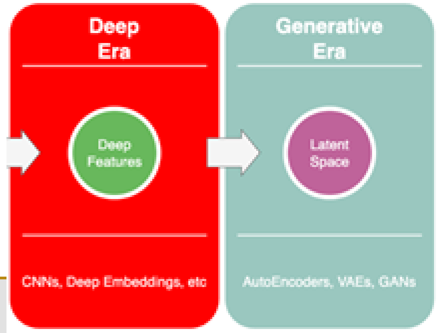
Deep Features vs Latent Space



- Representation vs. Distribution Modelling
  - **Deep Features:** Deep features learned by an encoder are representations of data that capture high-level patterns and discriminative information. **They focus on encoding the salient features necessary for classification or other downstream tasks.**
  - **Latent Space**: The latent space of a generative model is a lower-dimensional representation of the data that aims to capture the underlying structure and variations. **It focuses on modeling the probability distribution of the data in the latent space for generation purposes.**

- While deep features focus on capturing discriminative information for specific tasks, the latent space in generative models aims to learn a compressed, structured representation that can be used for data synthesis and generation.

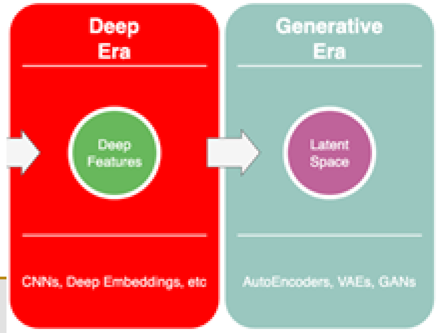
Deep Features vs Latent Space



- Supervised vs. Unsupervised Learning:
  - **Deep Features:** Learning deep features typically requires labeled data for supervision. **Models are trained using labeled examples to optimize the features for the specific task at hand, such as image classification.**
  - **Latent Space:** Generative models learn the latent space in an unsupervised manner, often using unlabeled data. **The models capture the underlying patterns and variations present in the data distribution without relying on explicit labels.**

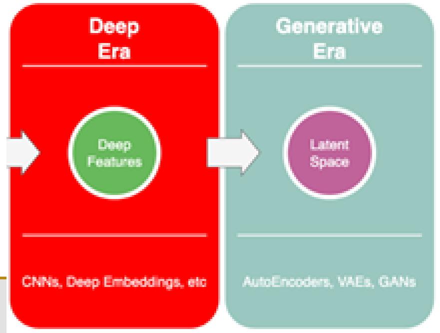
- Unsupervised learning of the latent space allows generative models to capture the inherent structure of the data without the need for labeled examples.

Deep Features vs Latent Space



- Discrimination vs. Generation:
  - **Deep Features**: Deep features focus on capturing discriminative information to distinguish between different classes or categories in the data. **They emphasize what differentiates one data point from another.**
  - **Latent Space**: The latent space of a generative model emphasizes the generation of new, realistic samples that resemble the original data distribution. **It encapsulates the factors of variation present in the data, allowing for the synthesis of diverse and novel instances.**
- The focus on generation in the latent space enables generative models to go beyond discrimination and create new data samples. This is crucial for tasks like image synthesis, data augmentation, and generating novel instances in various creative applications.

Deep Features vs Latent Space



- Fixed vs. Controllable Representations:
  - **Deep Features**: Deep features learned by an encoder are fixed representations that do not offer explicit control over specific attributes or characteristics of the data. **They are optimized for the specific task and lack explicit manipulability.**
  - **Latent Space:** The latent space of a generative model offers the potential for explicit control and manipulation of specific attributes or factors of variation. **By exploring different regions or interpolating between latent vectors, specific attributes can be modified or combined to generate desired outputs.**

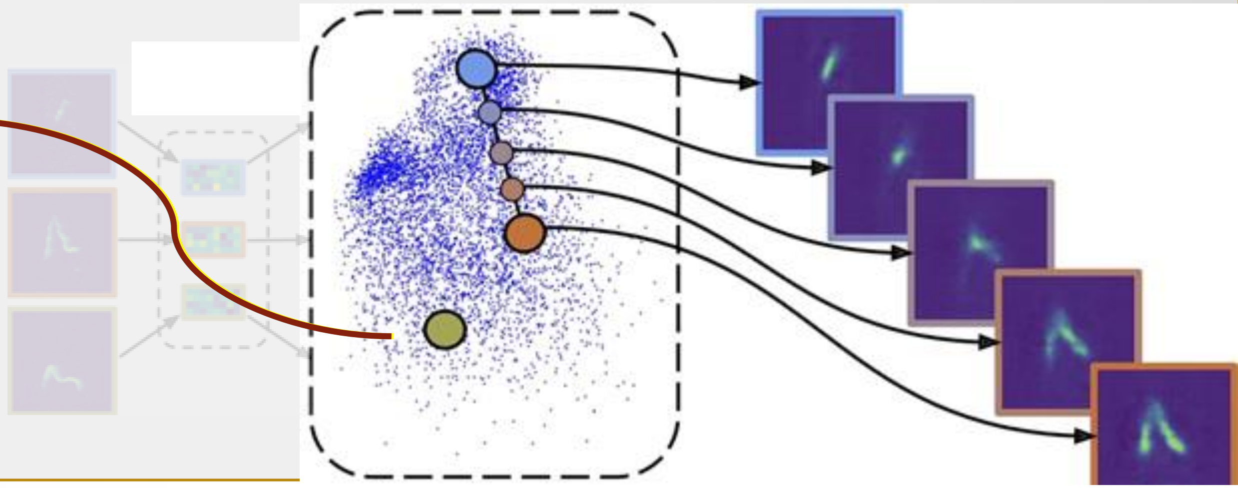
- The controllability of the latent space provides flexibility in generative modeling, allowing users to manipulate specific attributes or create variations in the generated outputs, particularly useful in applications like image editing, style transfer, and interactive content creation.

Previous slide from today!

Sampling from a Vector Space

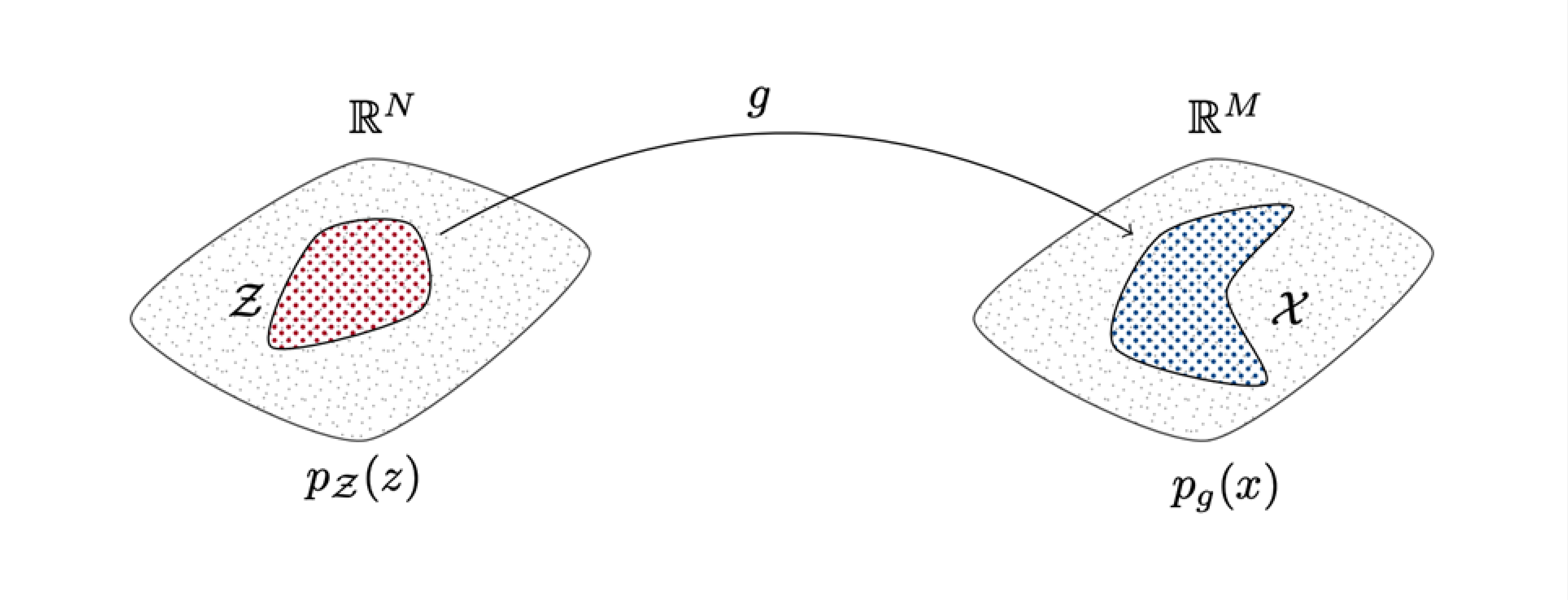
- In generative modeling, we require a intermediate vector space that is both **continuous** and **representative of the underlying data distribution** for effective sampling from a generator.
- A continuous vector space enables smooth interpolation and exploration of the data, allowing for **seamless transitions** and generation of new samples.

The Latent Space

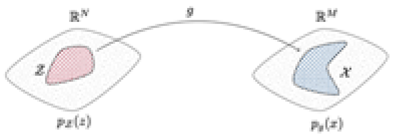


The Latent Space

- A generative model g is a function that maps
  - a latent space $\mathcal{Z} \subseteq \mathbb{R}^N$ → on a target space $\mathcal{X} \subseteq \mathbb{R}^M$

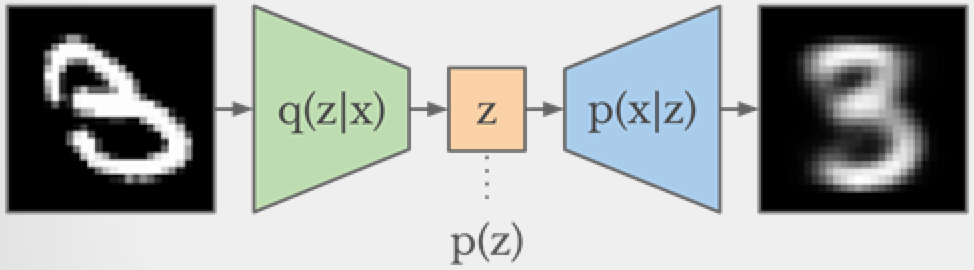


a latent space $\boldsymbol{z} \subseteq \mathbb{R}^N \rightarrow$ on a target space $\boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^M$



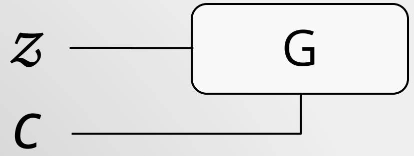
The Latent Space

- Early generative models primarily focused on explicit feature engineering and handcrafted representations.
- Latent space was not explicitly defined, and generation relied on manually designed algorithms or statistical models.
- Autoencoders, (not generative models but will evolve in to generative VAEs), played a significant role in developing the concept of the latent space.

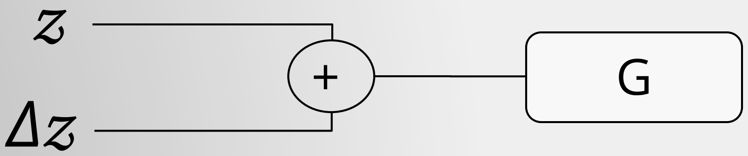


Controllable Generation

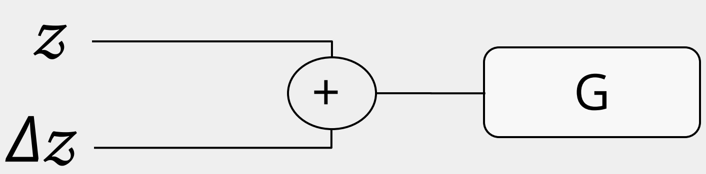
- Controllability vs Conditioning
  - Conditioning involves providing explicit information or constraints during generation, guiding the model to produce output aligned with specified conditions.

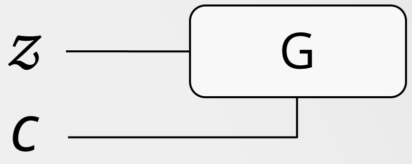


  - Controllability emphasizes user-driven customization, allowing manipulation of specific features in the generated output.



Controllable Generation

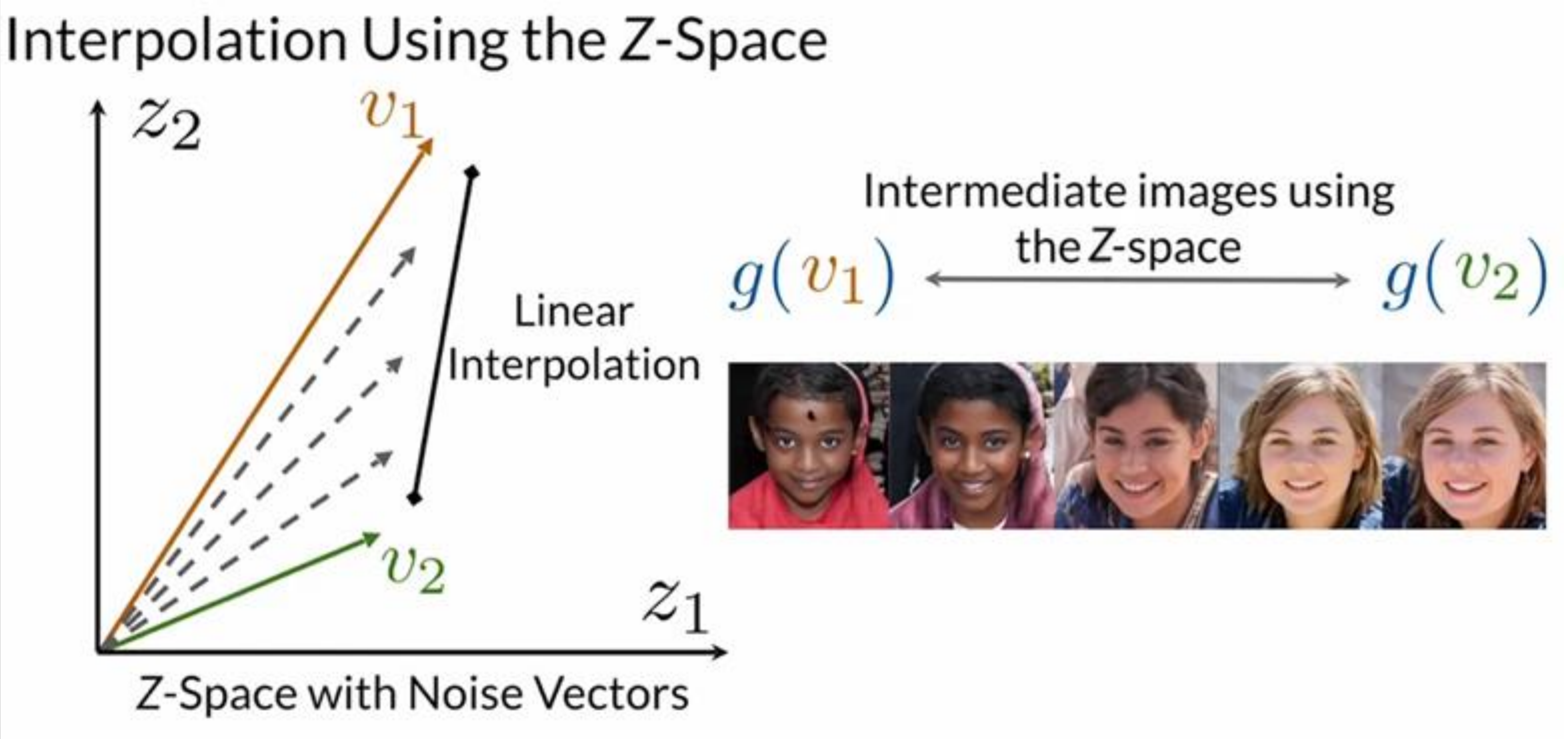




Controllable	Conditional
Examples with the **features that you want**	Examples from *the classes you want*
Training dataset **doesn't need to be labeled**	Training dataset *needs to be labeled*
Manipulate the z vector input	*Append a class vector* to the input

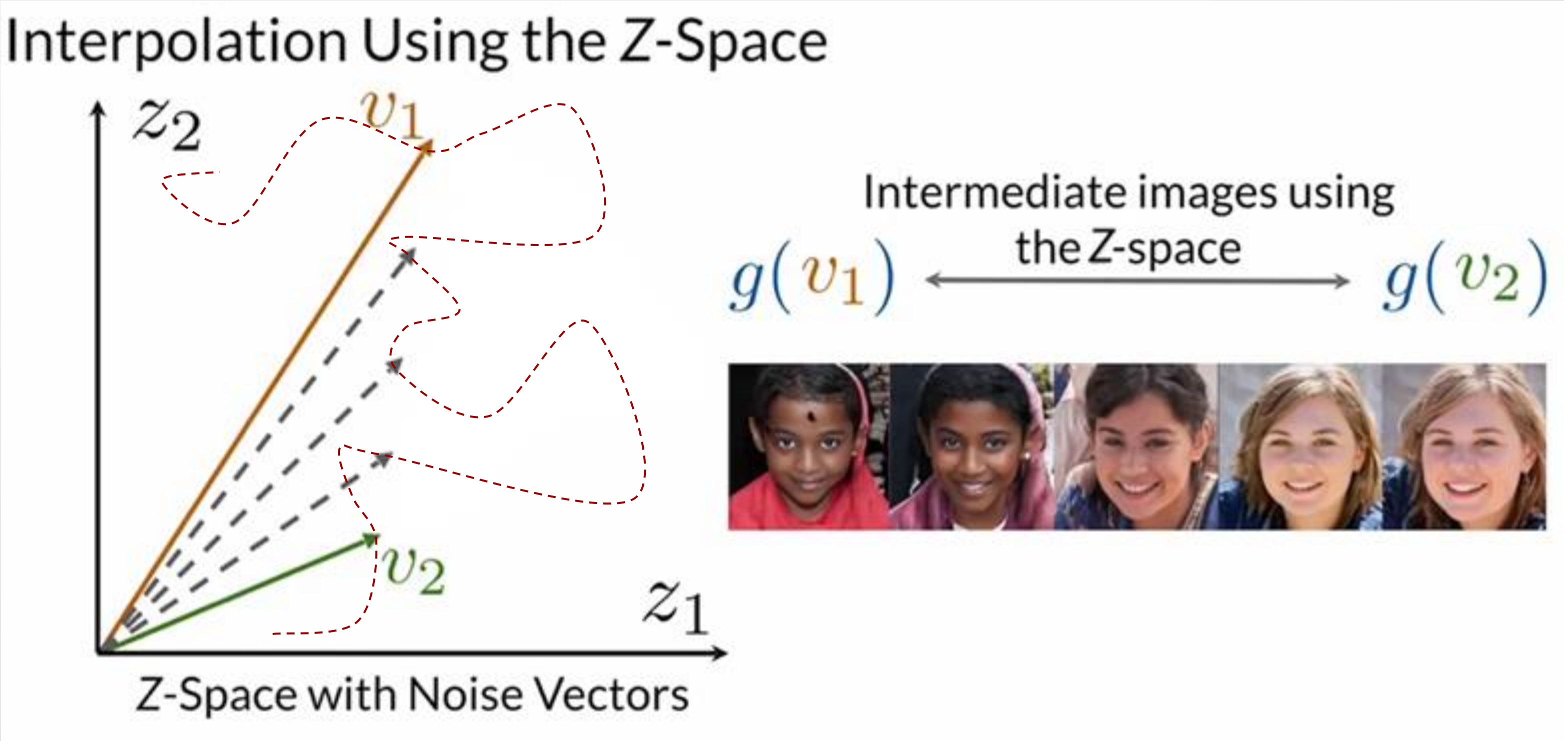
Controllable Generation

- Desired case is we can interpolate the z-space with perceptually smooth changes in the target space.



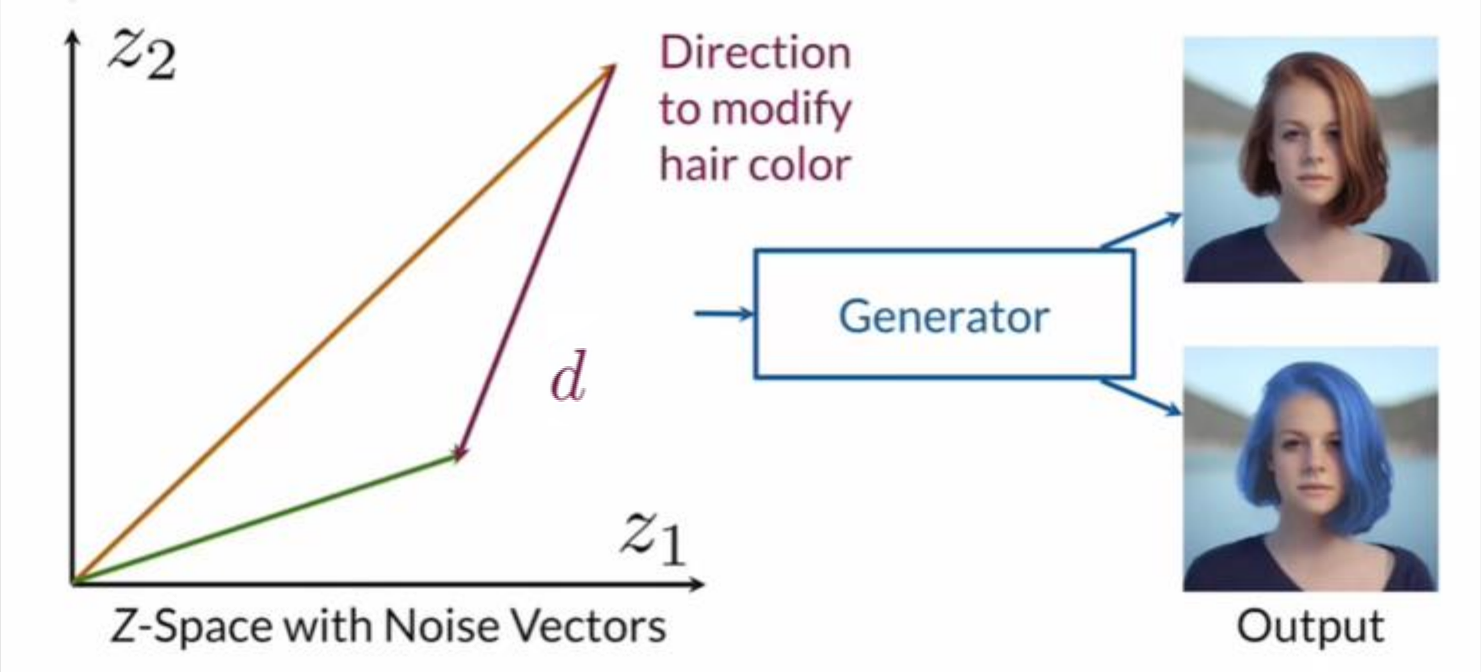
Controllable Generation

- However the reality is more like (if you do not specifically design it) this:



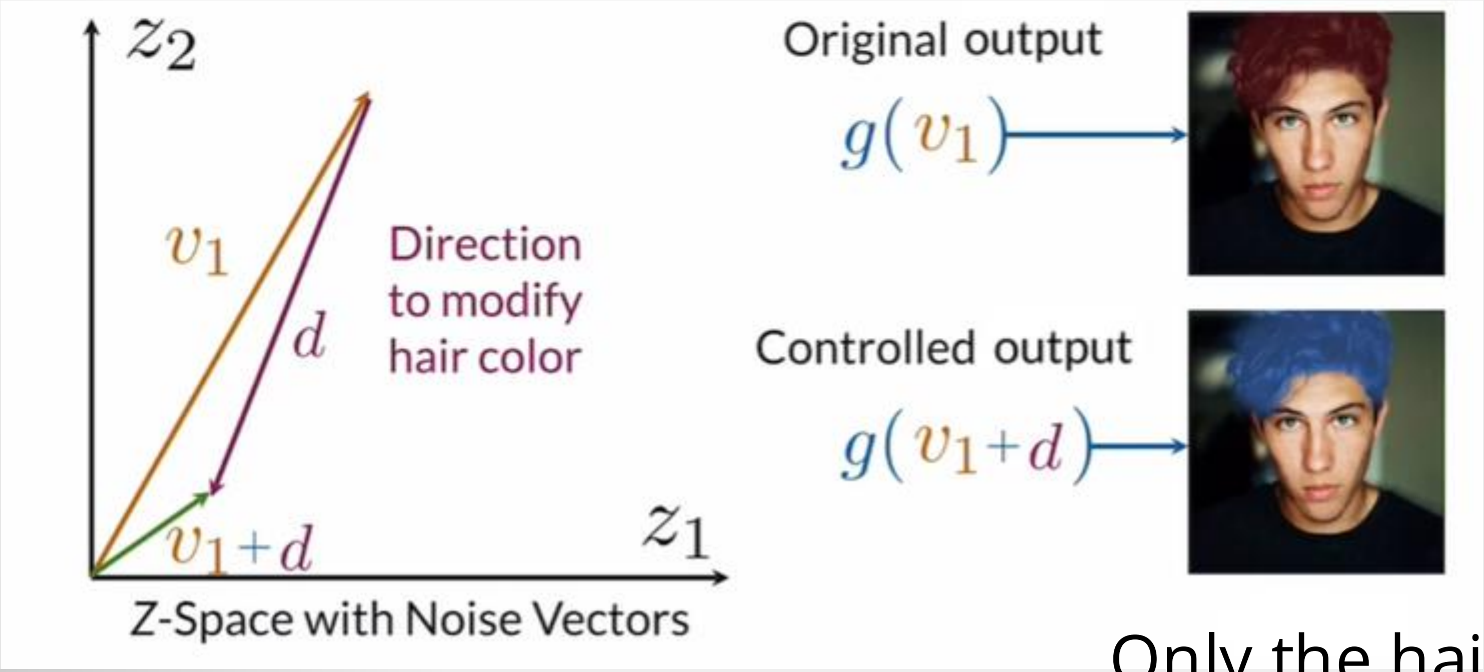
Controllable Generation (desired case)

- In controlling the generation process, our goal is to find “desired” directions.



Controllable Generation (desired case)

- In controlling the generation process, our goal is to find “desired” directions.



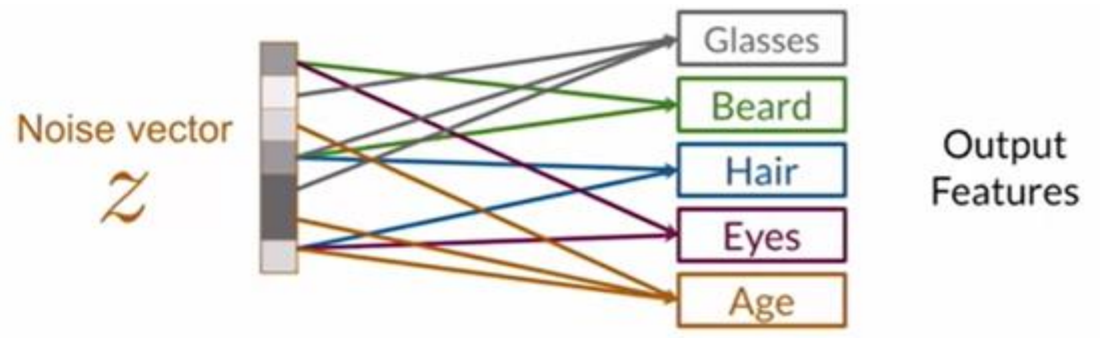
Only the hair!

Feature Correlation

- The fundamental reason linear interpolation in the latent space is not feasible is because the features are correlated (maybe for the best)

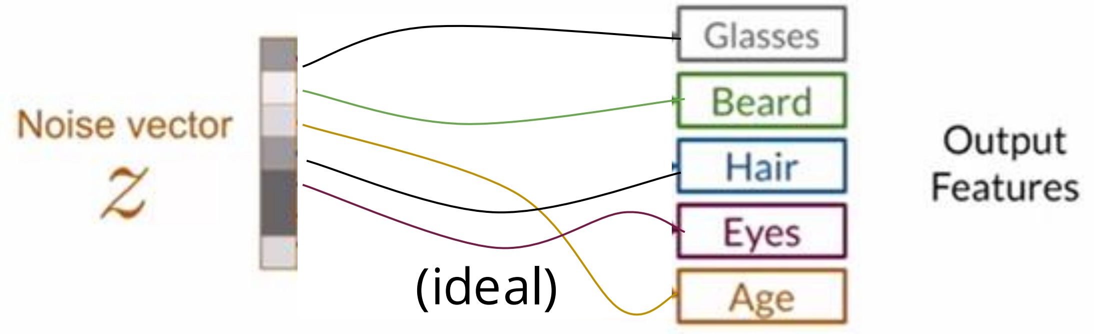
slide by coursera

Entanglement



- Entanglement refers to the intricate interdependence of different dimensions or features within the latent space.
- In simpler terms, changes in one dimension impact and influence other dimensions, creating complex relationships.
- The degree of entanglement directly affects the model's capacity to capture and represent diverse and complex patterns in the data.
- Understanding entanglement is crucial for improving the interpretability and controllability of generated outputs.

Entanglement



- Entanglement refers to the intricate interdependence of different dimensions or features within the latent space.
- In simpler terms, changes in one dimension impact and influence other dimensions, creating complex relationships.
- The degree of entanglement directly affects the model's capacity to capture and represent diverse and complex patterns in the data.
- Understanding entanglement is crucial for improving the interpretability and controllability of generated outputs.

Next lecture:

- **PART III: Auto-Encoding**

- **Autoencoders and Dimensionality Reduction**
- Variational Inference and VAEs
- Conclusions

Thanks

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia