

Autoencoders and latent spaces

Raoul Grouls, 11 juni 2024

Motivation, part 1

Motivation for autoencoders

An autoencoder learns a “latent space” with efficient encodings of unlabeled data. Some applications:

- Dimensionality reduction
- Anomaly detection
- Denoising
- Data Compression

Recap

Data

$$X = \{\vec{x}_1, \dots, \vec{x}_n \mid \vec{x} \in \mathbb{R}^d\}$$

$$y = \{y_1, \dots, y_n \mid y \in \{0,1\}\}$$

(Non)linearity

$$f(X) = WX + b \quad \sigma(X) = \max(0, X)$$

Predict

$$\hat{y} = f_n \circ \sigma \circ f_{n-1} \circ \dots \circ \sigma \circ f_1$$

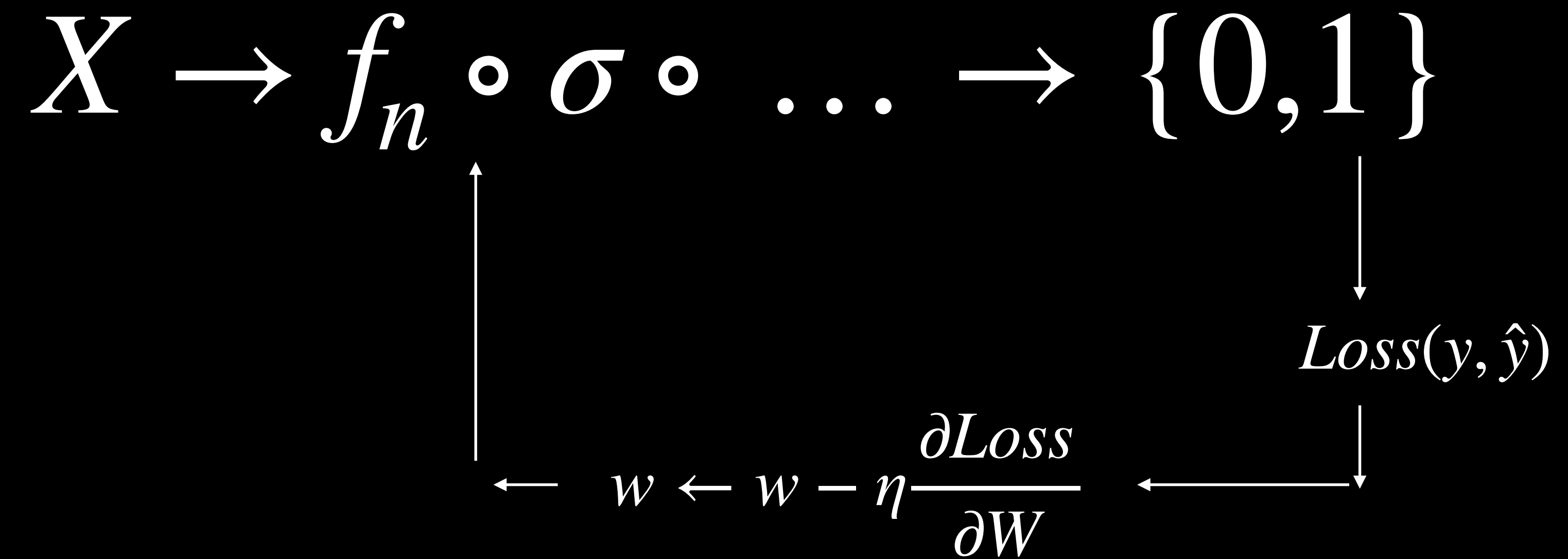
Loss

$$Loss(y, \hat{y})$$

Optimize

$$w \leftarrow w - \eta \frac{\partial Loss}{\partial W}$$

Recap



Contrast with supervised learning

Latent space

We are going to learn what a latent space is by looking at the differences with the supervised learning we have been studying the last 5 lessons.

This strategy should remind you of a deep learning technique...

Contrast with supervised learning

Latent space

A latent space, also known as a feature space or hidden space, refers to a vectorspace \mathbb{R}^d where the data's features are represented.

It is just a different name for what we have been using the last 5 lessons.

For autoencoders, the dimensionality is typically much lower than that of the input.

Contrast with supervised learning

Encoder - decoder

- Let's start with writing the mapping $X \rightarrow \{0,1\}$ a bit more verbose:

$$X \rightarrow \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2} \rightarrow \dots \rightarrow \mathbb{R}^{d_m} \rightarrow \dots \mathbb{R}^{d_n} \rightarrow \{0,1\}$$

- Now, instead of mapping to some label $\{0,1\}$, the idea is to map the input X back to itself. Let's split the network conceptually into an encoder-decoder architecture:

- An encoder $e = f_m \circ \sigma \circ f_{m-1} \circ \dots \circ \sigma \circ f_1$ that maps

$$e: X \rightarrow \mathbb{R}^{d_m}$$

- A decoder $d = f_n \circ \sigma \circ f_{n-1} \circ \dots \circ \sigma \circ f_{m+1}$ that reconstructs input:

$$d: \mathbb{R}^{d_m} \rightarrow X$$

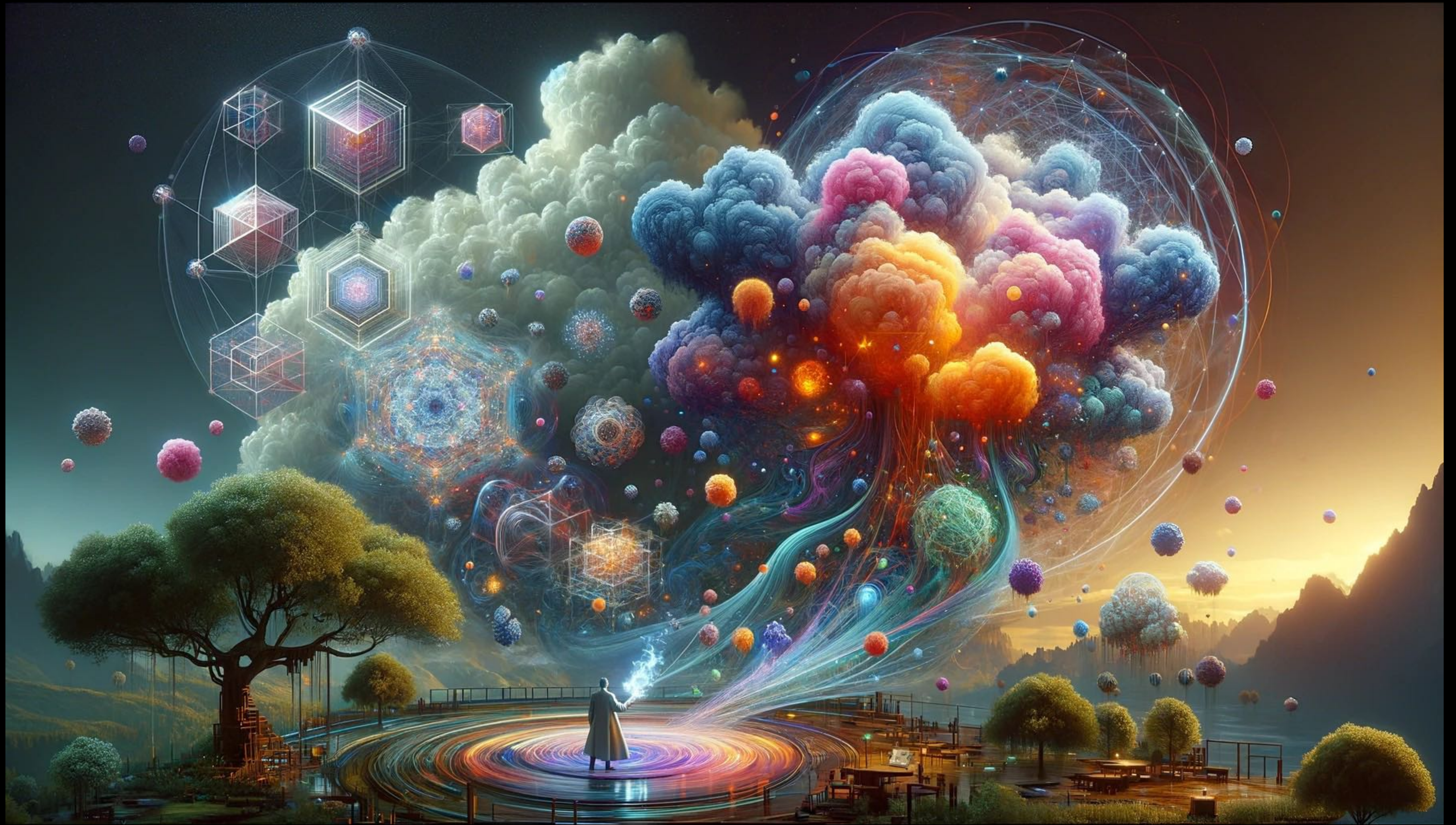
Contrast with supervised learning

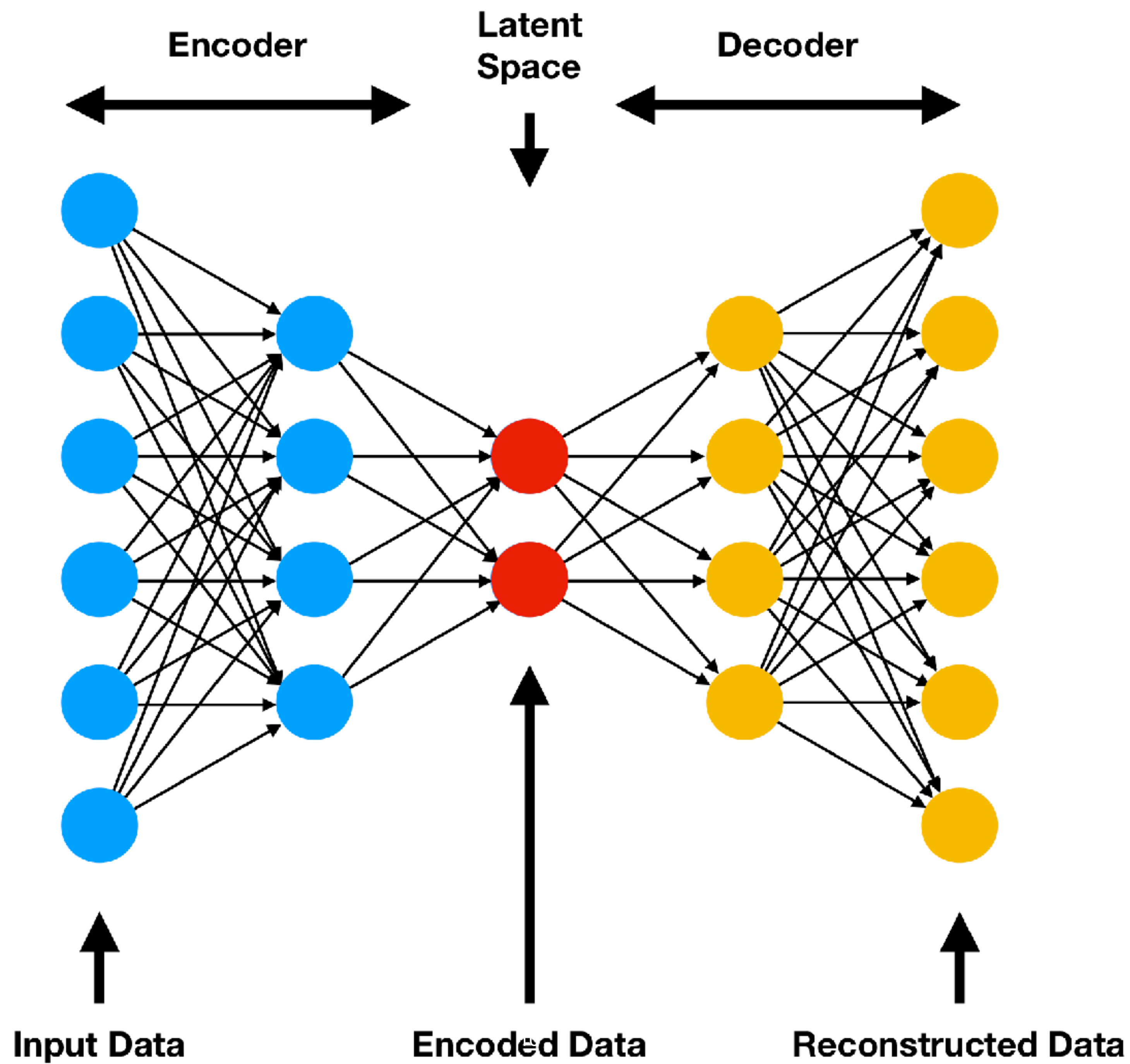
Reducing dimensionality

An autoencoder is a network $AE(x) = d(e(x))$, which gives us:

$$AE: X \rightarrow \mathbb{R}^d \rightarrow X$$

The encoder maps input space X to latent space, that typically involves a reduction in dimensionality: $\dim(Z) < \dim(X)$, and then maps back to the original.





Contrast with supervised learning

Minimize reconstruction error

Instead of minimizing the error between \hat{y} and y , the goal is to minimize the reconstruction error between $d(e(x))$ and x

Contrast with supervised learning

Minimize reconstruction error

Why do we do this? We already have X , so isn't it pointless to predict X ?

Well, it's not the output we are after, but it is actually what happens in the latent space what we find to be valuable!

Key differences with supervised learning

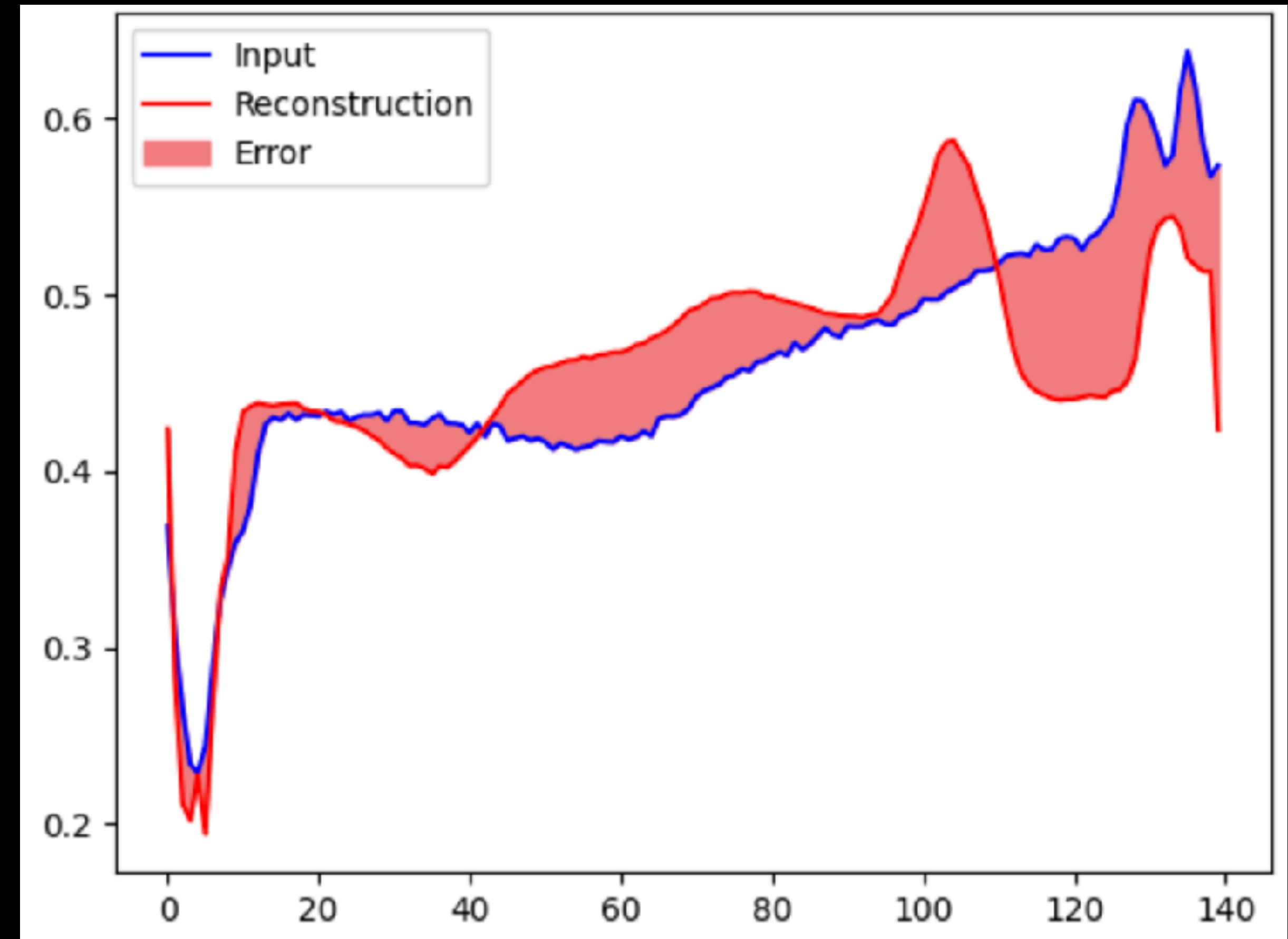
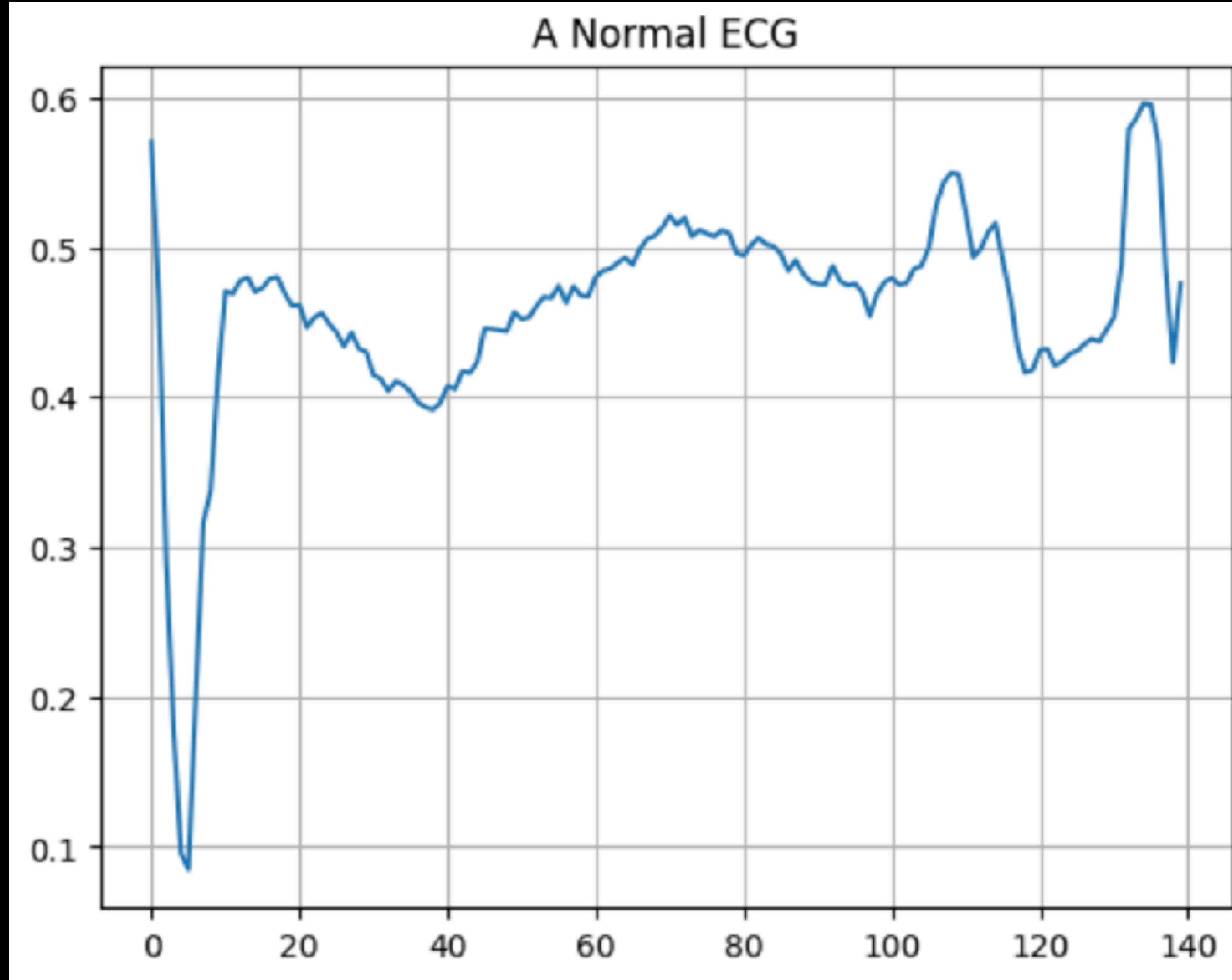
- By restricting the dimensionality of Z , we force the model to learn to be as efficient as possible and make summaries.
- We don't need external labels
- We don't focus on accuracy per se, but on useful summarisation (in terms of our endgoal). We don't want a perfect reconstruction, but a latent space that captures the essence!
- Generative AI explores the latent space as a source of creativity
- Often we just want the encoder or decoder, instead of using the full model for inference.

Motivation, part 2

motivation for autoencoders

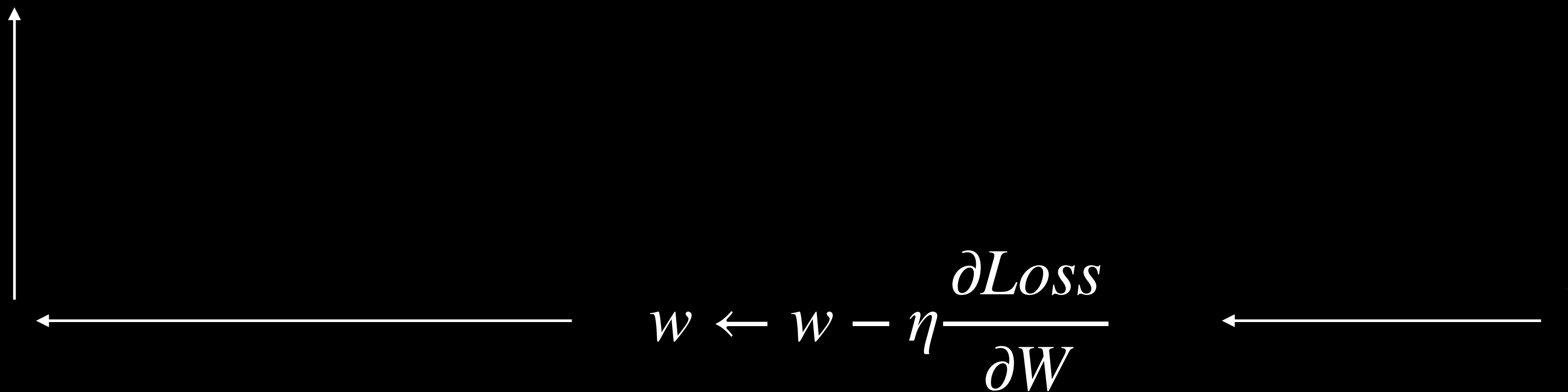
- Dimensionality reduction (encoder) : Capture the most significant features, making it easier to visualise and process data.
- Data Compression (encoder): the latent space is compressed, so we can use that in itself.
- Anomaly detection (encoder-decoder): By learning the “normal” pattern of data, the reconstruction error will be bigger with anomalies even though (more precise, exactly because of this) the network hasn't been trained with labels of anomalies.
- Denoising (encoder-decoder): the latent space is smaller, so has to be more efficient and will remove noise

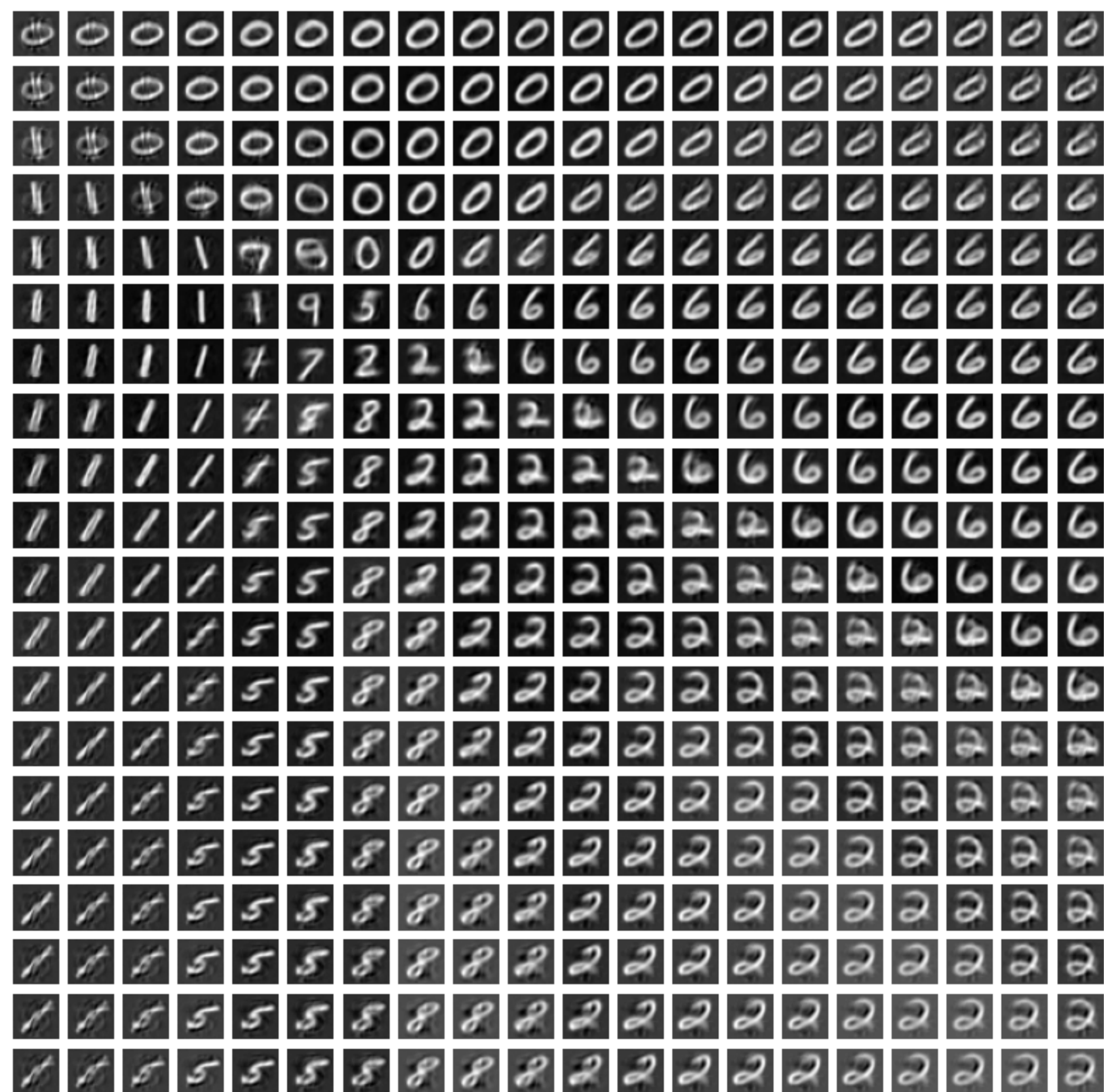
Anomaly detection

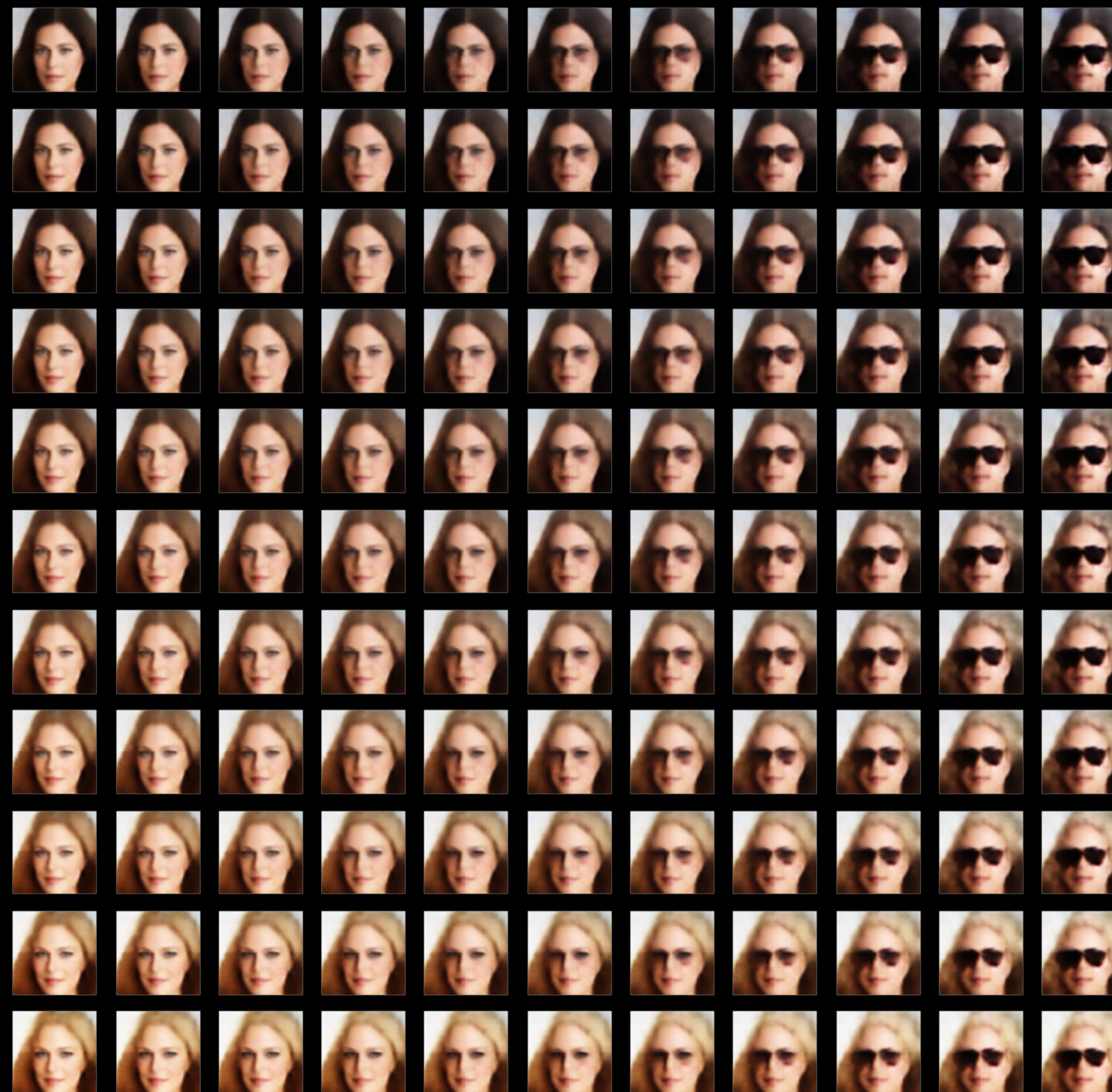


Autoencoder

$$X \rightarrow f_m \circ \sigma \circ \dots \rightarrow Z \rightarrow f_n \circ \sigma \circ \dots \rightarrow \hat{X} \rightarrow Loss(X, \hat{X})$$







Unsupervised Classification

- Map your unlabeled training data to Z
- Map the new, unlabeled input to the latent space Z
- Find the k items in your trainingsdata that are closest in Z

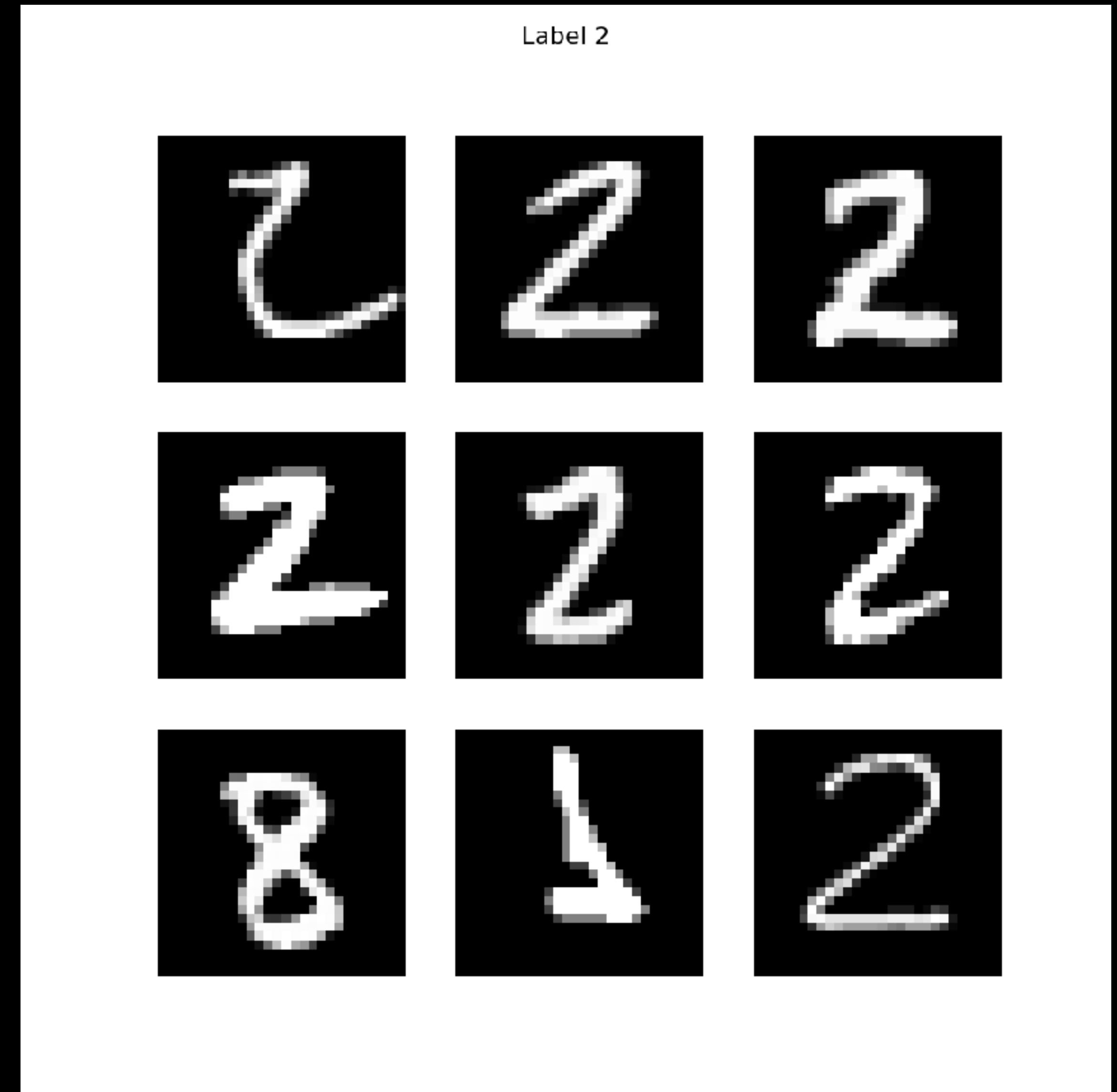


Fig: the 9 items closest to the new input

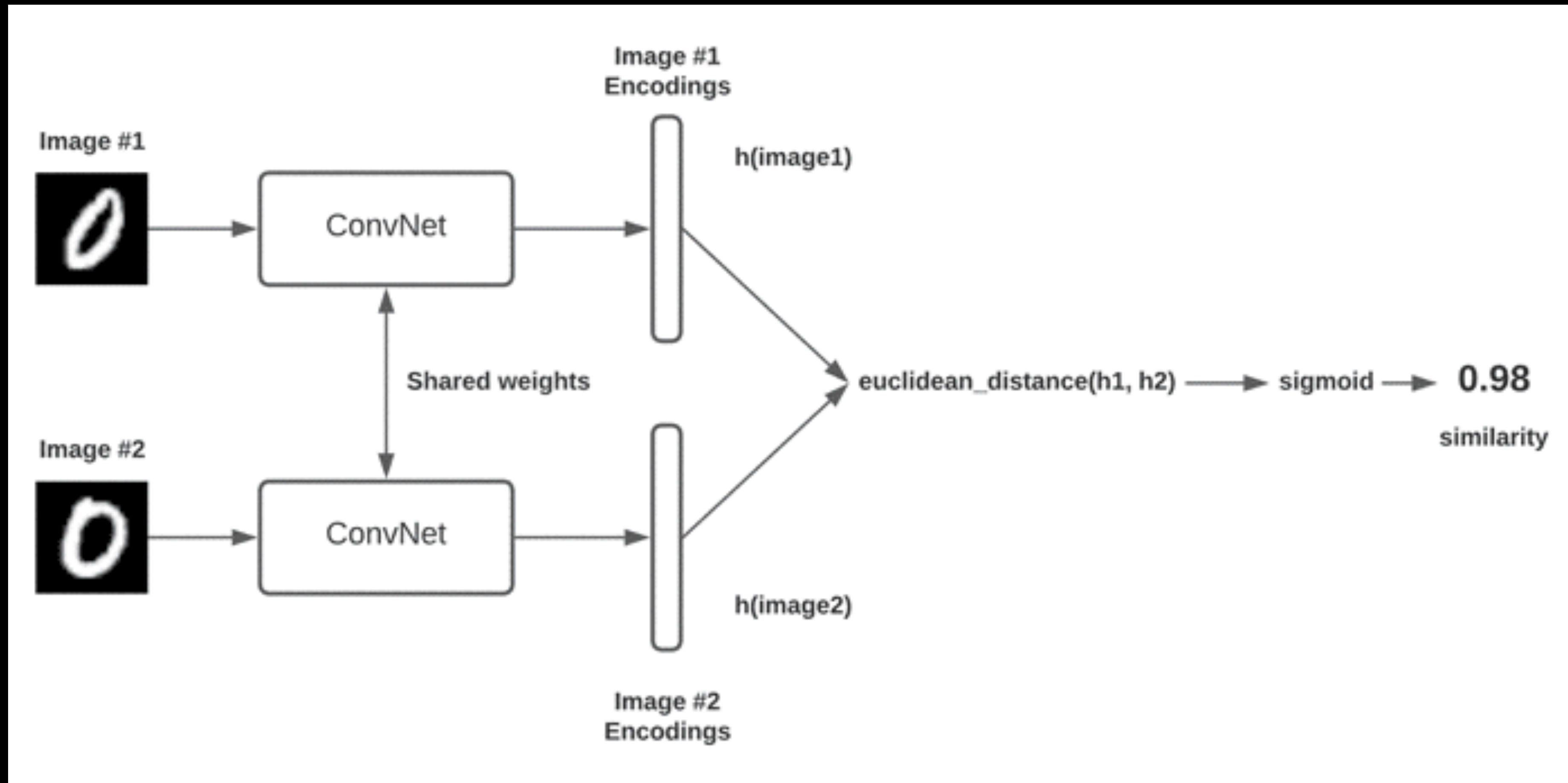
Siamese networks

Semisupervised

- $X = \{x_1, \dots, x_j \mid x \in \mathbb{R}^D\}$
- A labeling function $g: X \times X \rightarrow \{0, 1\}$ defined as $g(x_i, x_j) = \begin{cases} 1 & \text{if } x_i \sim x_j \\ 0 & \text{if } x_i \neq x_j \end{cases}$
- An encoder $f: x \rightarrow Z$ with $Z \subset \mathbb{R}^d$ and $d < D$
- A distance function $s(z_i, z_j)$, eg euclidian distance
- A loss function $Loss(s(z_i, z_j), y)$ that requires the distance to be close if the label is 1.

Siamese networks

Semisupervised



Joint-Embedding Predictive Architecture

Cognitive learning theories have suggested that a driving mechanism behind representation learning in biological systems is

- the adaptation of an internal model to predict sensory input responses

See: Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture, Assran et al. (2023)

Joint-Embedding Predictive Architecture

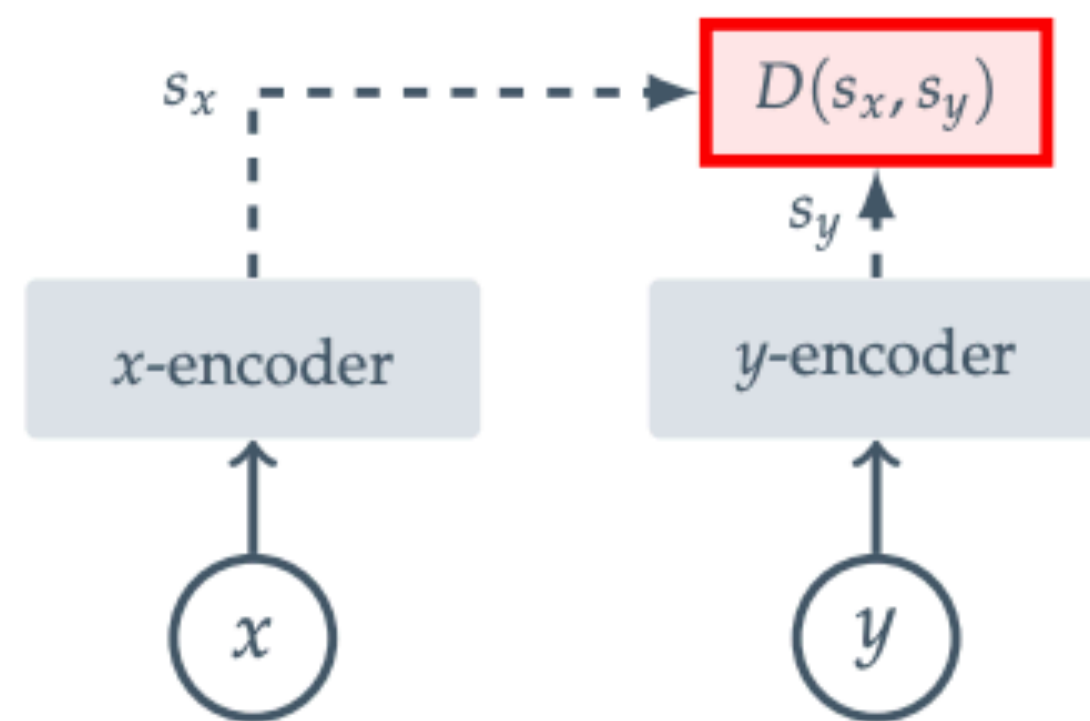
Compared to generative methods that predict in pixel/token space, I-JEPA predicts directly in embedding-space.

This means it predicts the representation, not the details!

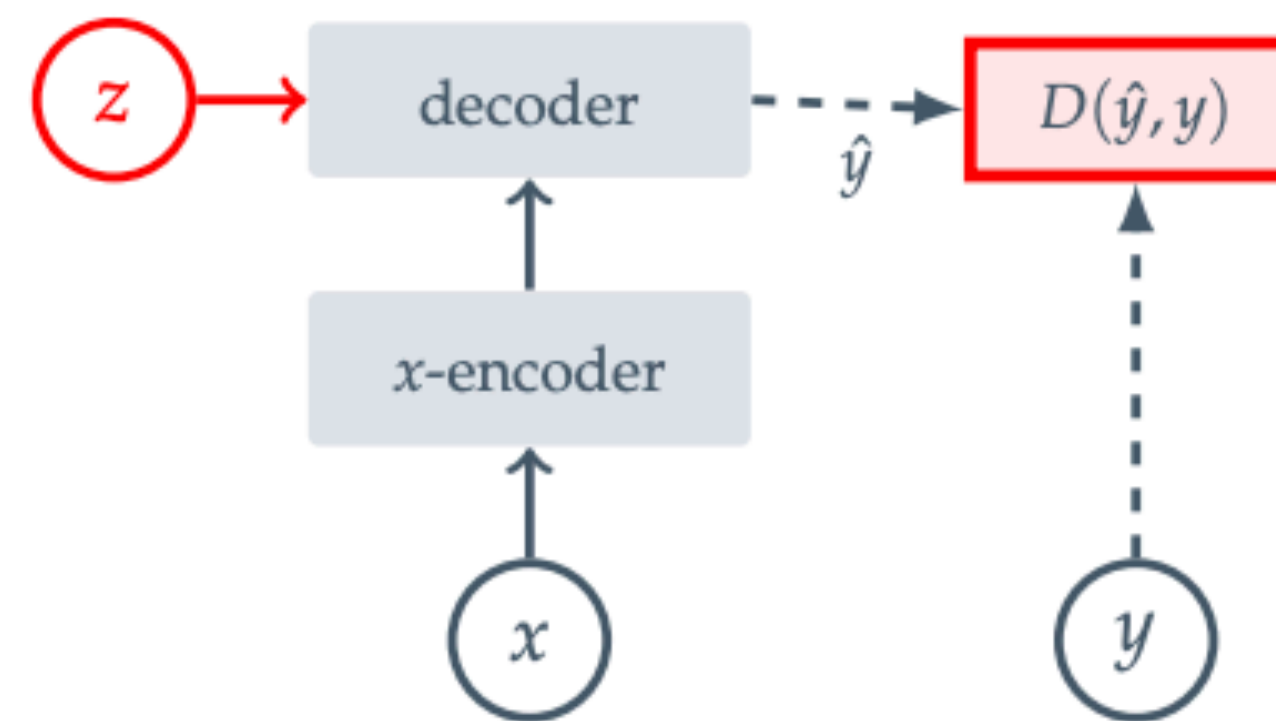
Compare this to planning a journey on the level of all the muscle contractions, versus planning it with a simplified model of the route (go left after the big green tree).



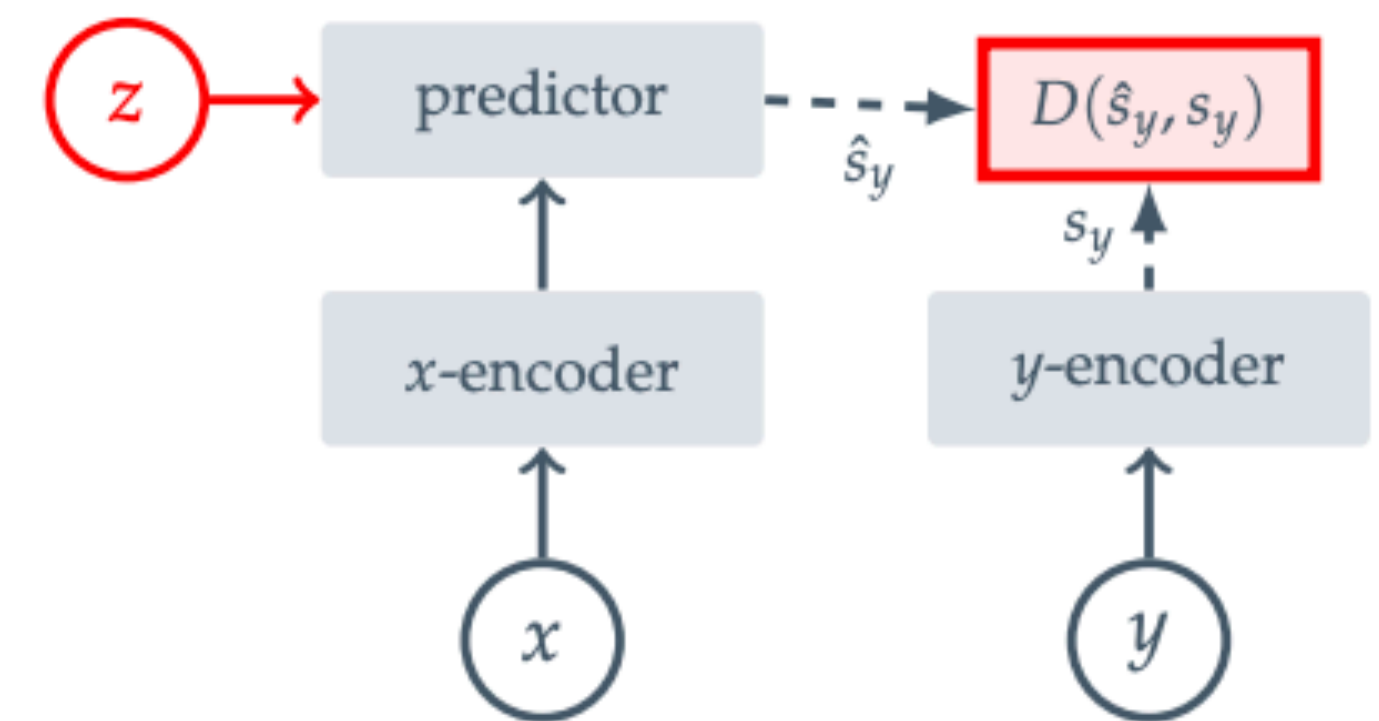
Joint-Embedding Predictive Architecture



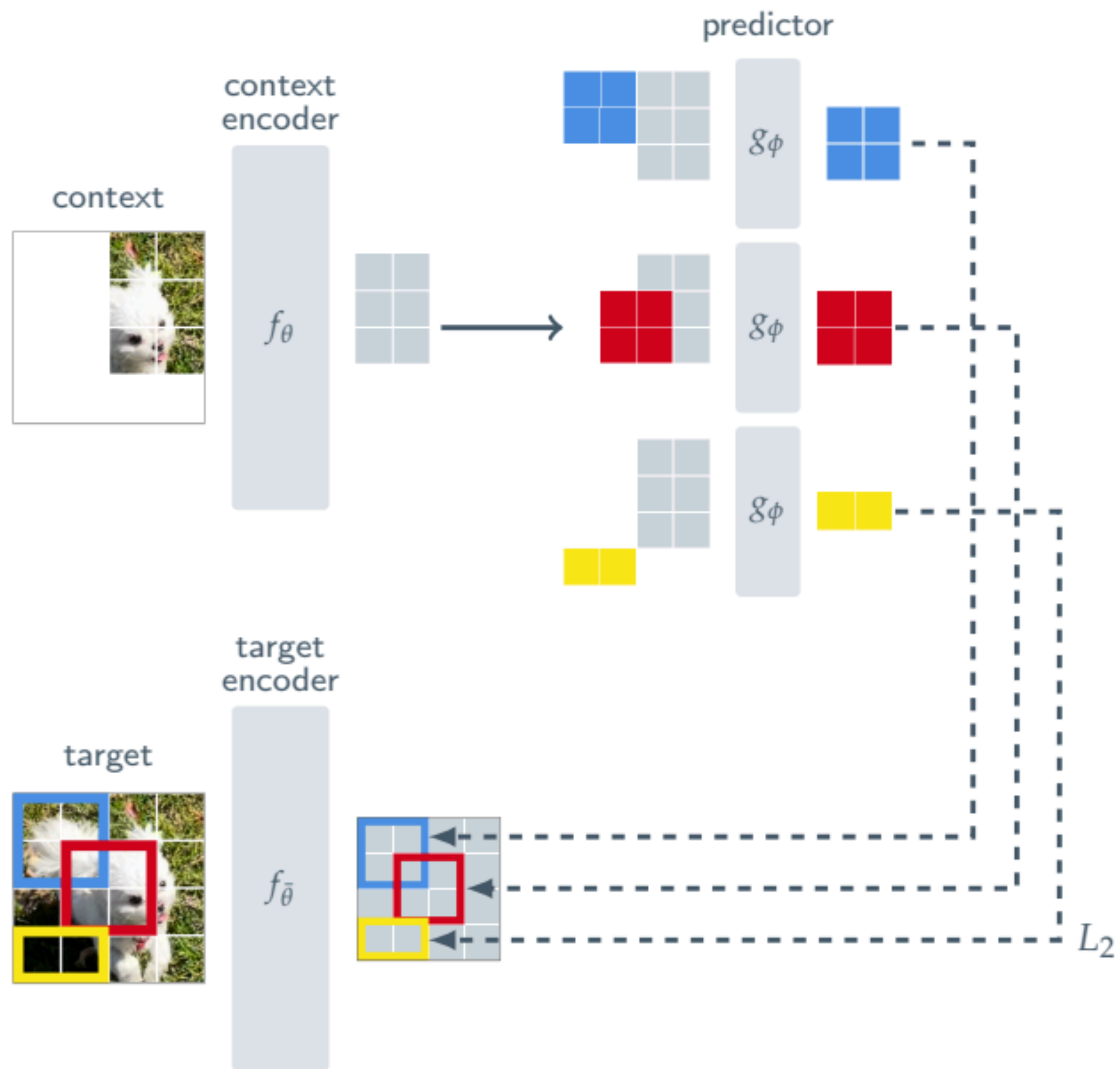
(a) Joint-Embedding Architecture



(b) Generative Architecture

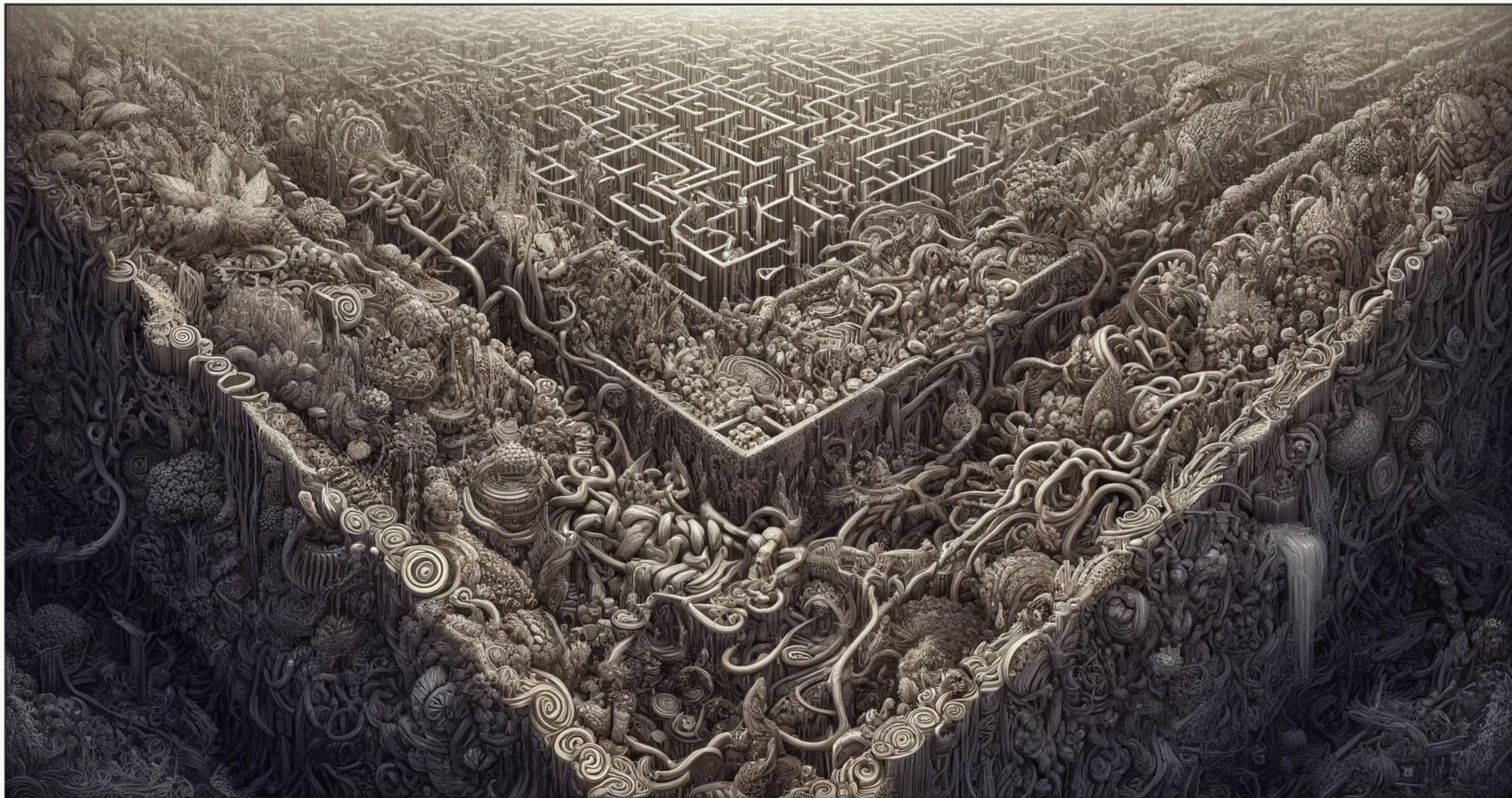


(c) Joint-Embedding Predictive Architecture



Joint-Embedding Predictive Architecture

Exactly because the input is “damaged”, the model is forced to reduce focus on details and take a higher level of abstraction.



Lounen mncldh' - 0MICE | RTSUrel

Geineahioeemethod |



To/tonle level

