Machine Learning I
MATH60629A

Modern Generative Models
— Week #11

Rapid evolution of (image) generation capability

P(image)

https://twitter.com/goodfellow_ian/status/1084973596236144640/photo/1

Conditional Generation

text

A photo of a raccoon
wearing an astronaut helmet,
looking out of the window at night.

$P(\text{image} \mid \text{text})$

image

https://imagen.research.google/

Why generate images?

- It used to be a tougher question to answer
- To use wherever images are used (visualizations, video games, presentations, ads, etc…)
- Human-in-the-loop design

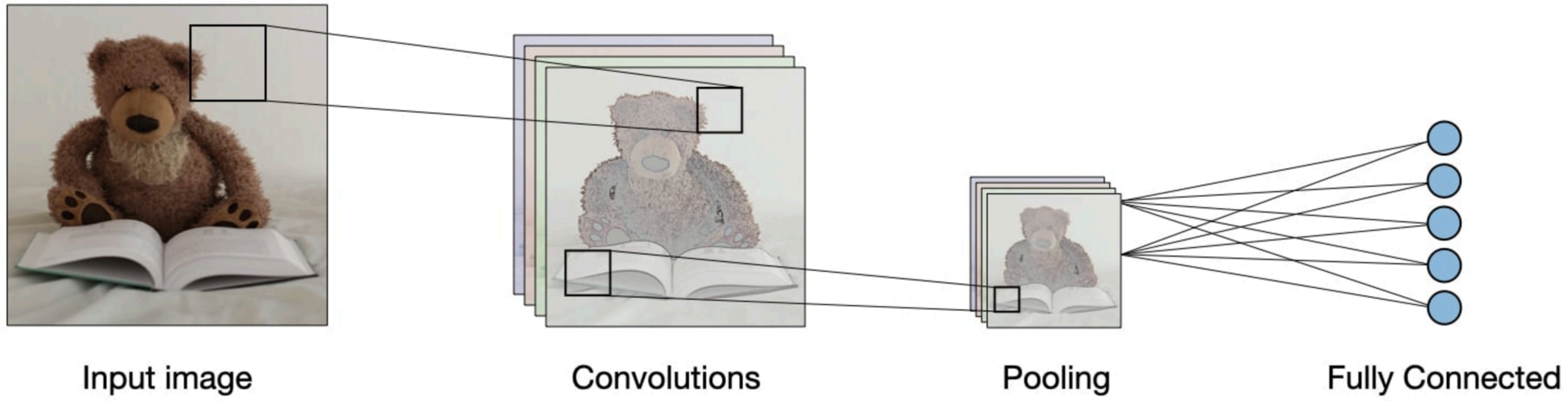
Today's Plan*

- "Predict" an image
- Generative Models
  - Images (P(x)):
    - Frameworks: Variational auto-encoders (VAEs), Generative Adversarial Networks (GANs)
  - Images conditioned on text (P(x | y)) :
    - Dall-E 2, Imagen

* Like last week, the slides are mostly from David Berger

Convolution Neural Network (CNN) — Recall

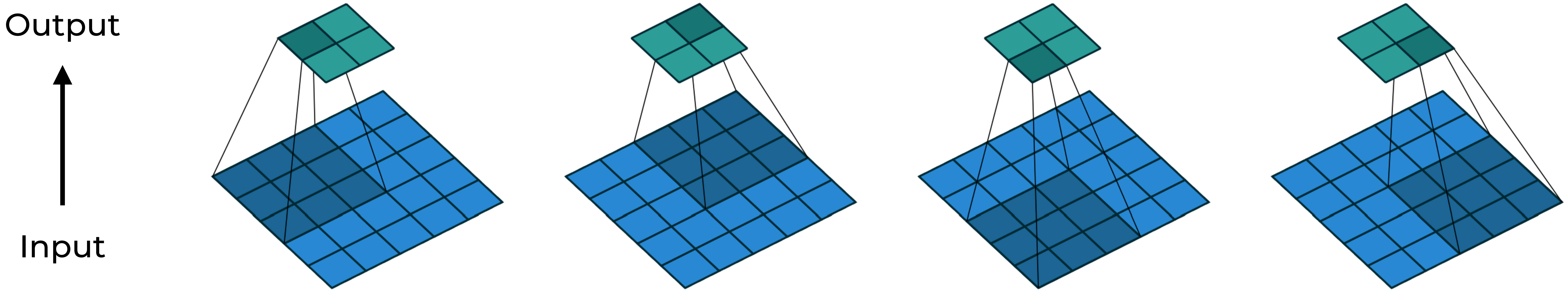
- Séries of layers (blocs): convolutions + pooling
  - Each layer reduces the dimensionality of the representation



https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

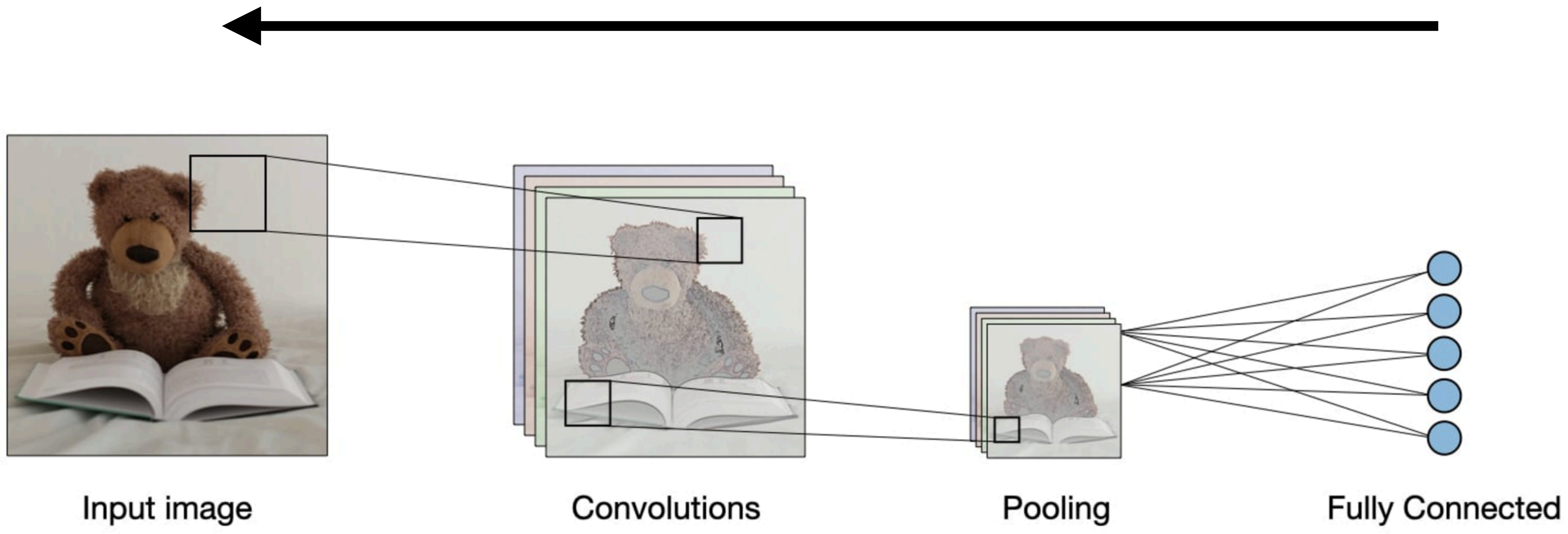
Convolution

- **Input: 5 x 5**
- **Filter (kernel): 3 x 3. Stride 2.**
- **Output: 2 x 2**



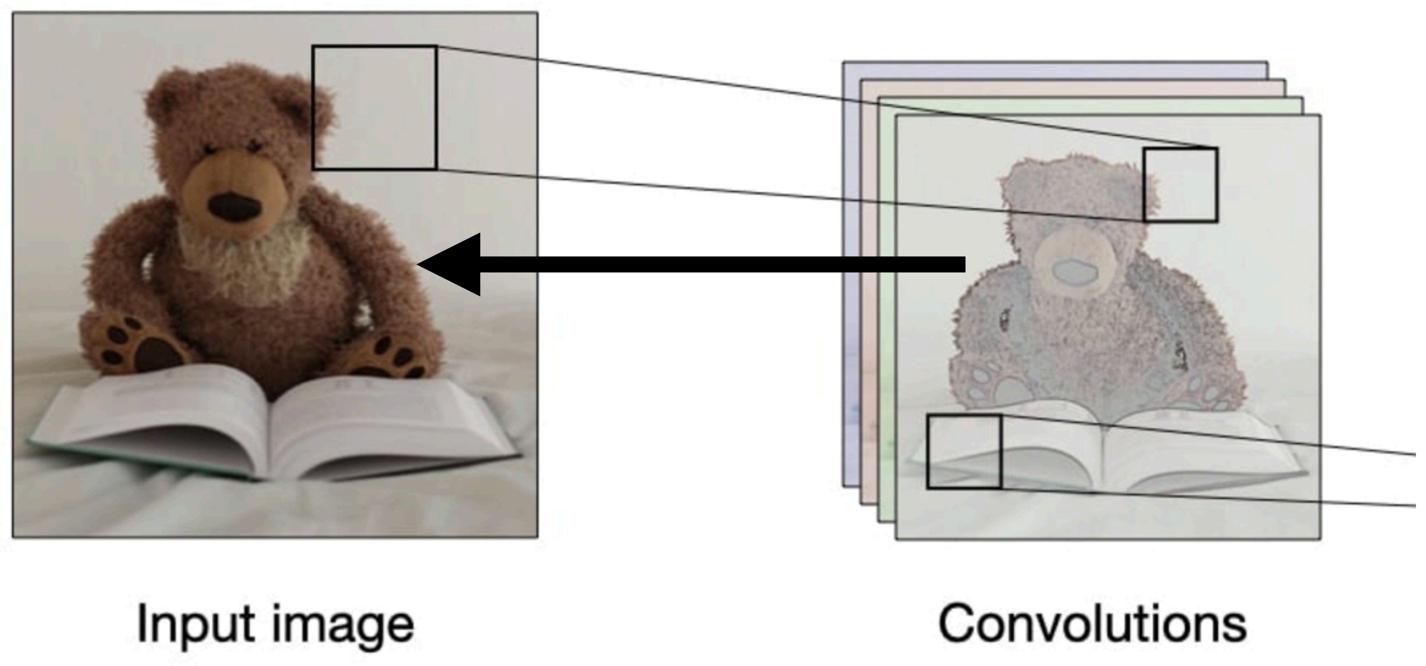
"Reverse" a convolution network

- **Each layers increases the dimensionality of the incoming representation**



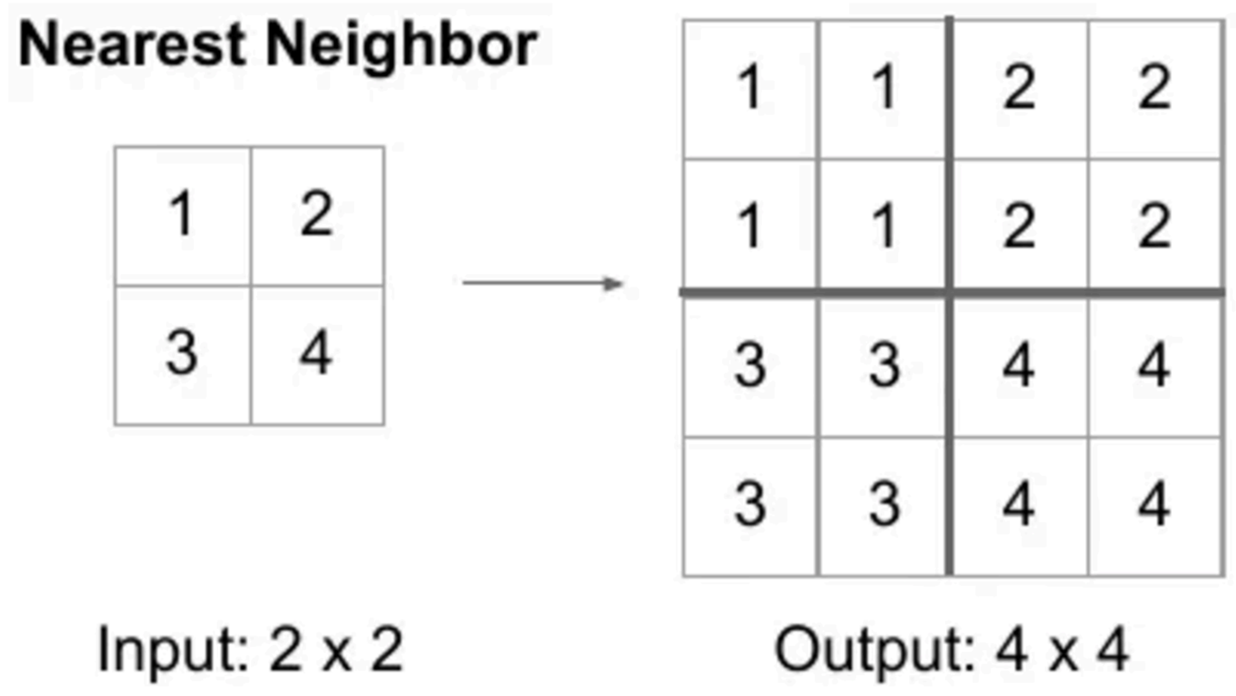
- **(Note: this is in fact the same operation as done by backdrop in a CNN.)**

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks



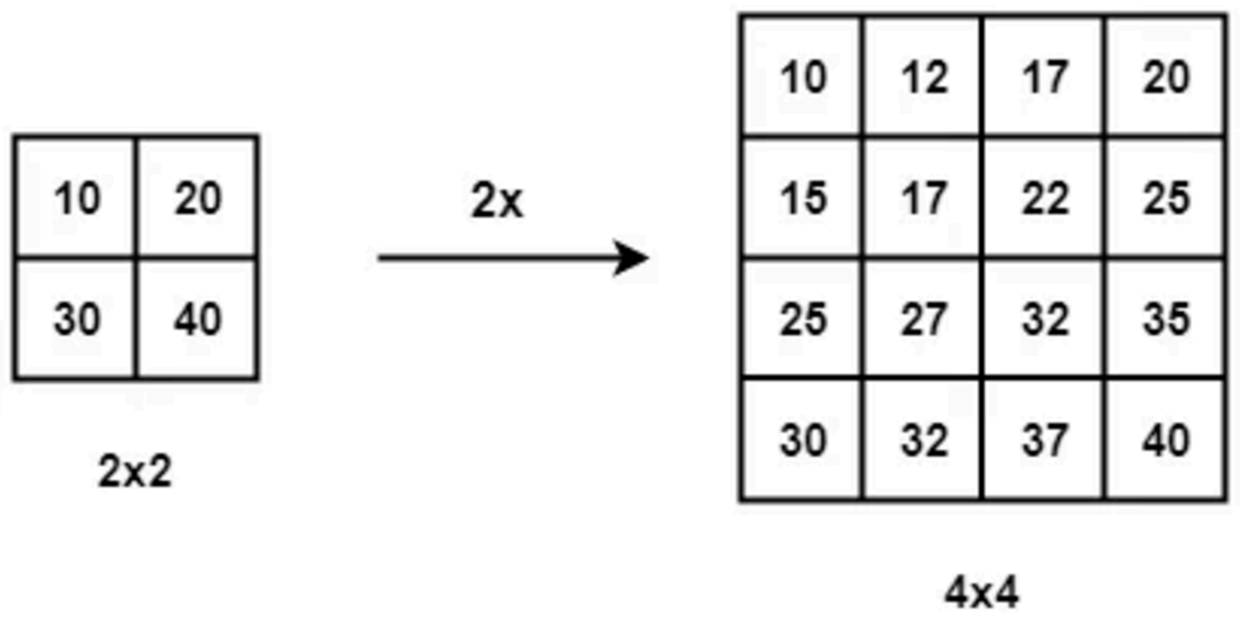
Interpolation

1. Repetition



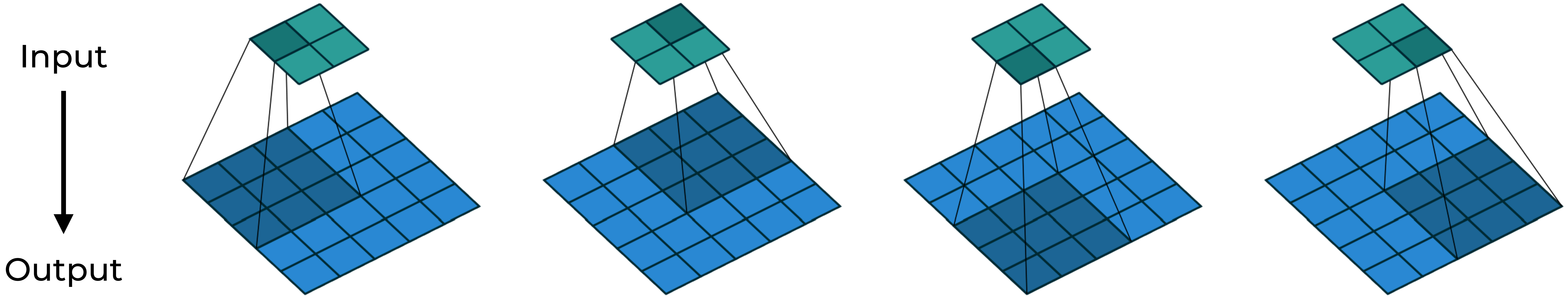
- No parameters to learn

2. (Bi-)linear interpolation



Transposed Convolution

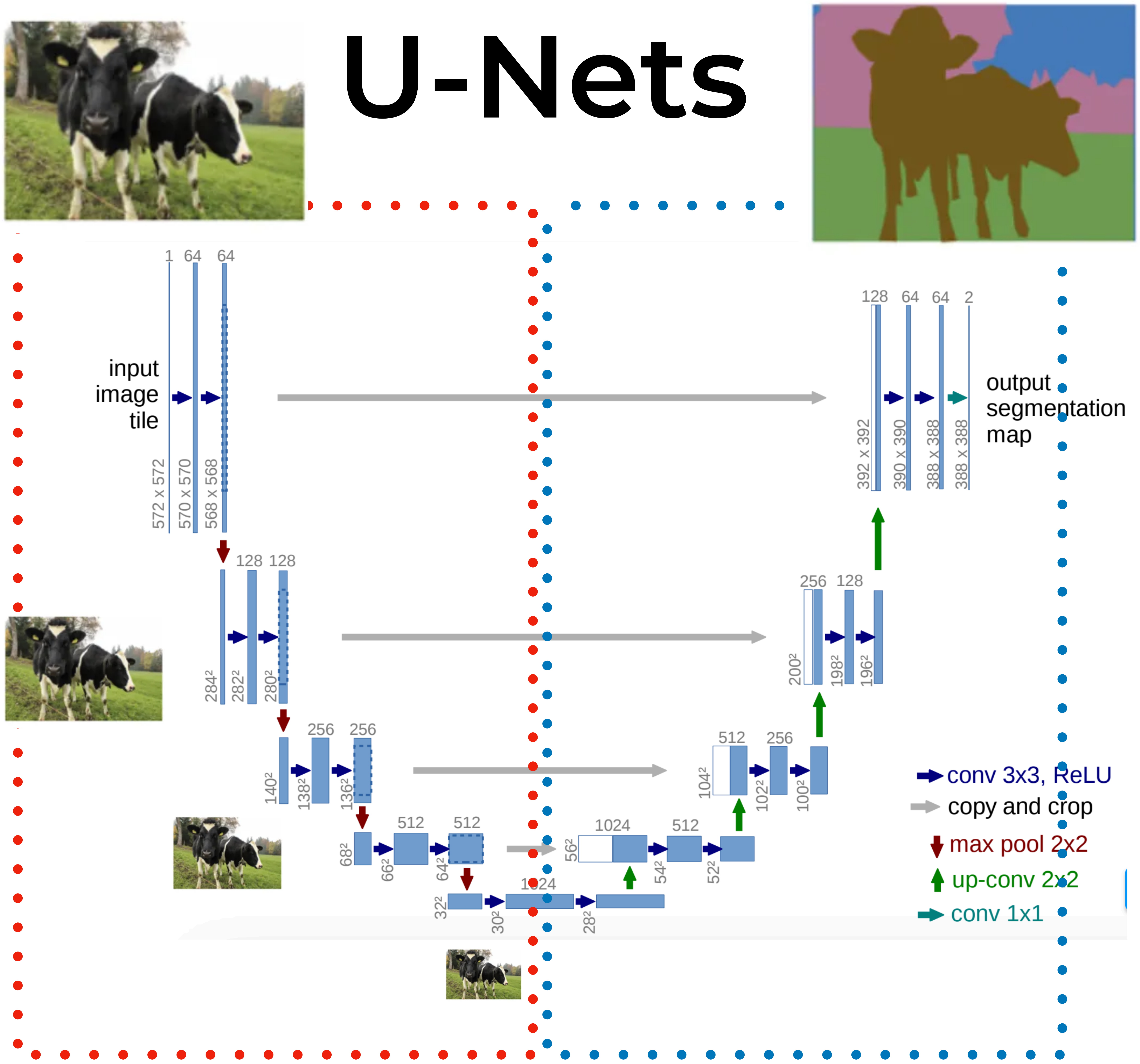
- Useful to generate images or increase the resolution of an image (like in movies)
- Input: 2 x 2
- Filter: 3 x 3. Stride 1.
- Output: 3 x 3



- If we write down a convolution as a matrix operation (by vectorizing the image), then the reverse operation is multiplying by the transpose (hence the name)

https://arxiv.org/pdf/1603.07285

U-Nets



- Encoder-Decoder Architecture
  - Encoder: Obtain a représentation of the image (classic CNN without the classification output)
  - Decoder: From the representation, obtain an image
  - Connections between the encoder and decoder allow for precise localization
- up-conv ->transposed convolution (for example)
- Proposed for segmentation
- Has become standard for image generation

U-Nets: https://arxiv.org/pdf/1505.04597

https://wiki.cloudfactory.com/docs/mp-wiki/key-principles-of-computer-vision/upsampling-and-downsampling-techniques-in-machine-learning

Generative models

- A U-Net can obtain a representation from an image, but it does not have a probabilistic interpretation
- A generative model is a method for parametrizing $P(x)$ — unsupervised

Why are generative models (often) probabilistic?

- Allows different types of evaluation
  - For example, the probability of an image according to the model: $P_\theta(x_{\text{new}})$
- Can obtain samples $x \sim P_\theta(x)$
- Quantifies uncertainty
- It has been popular recently to parametrize distributions with neural networks (think of a softmax layer) — these are not always proper generative models

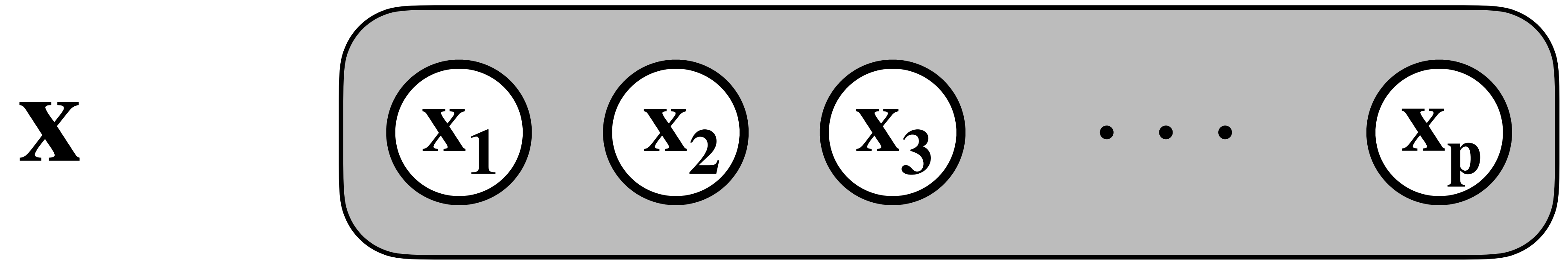
Auto-Encoder (AE)

Auto-encoder - Recall

For an AE with a single hidden layer:

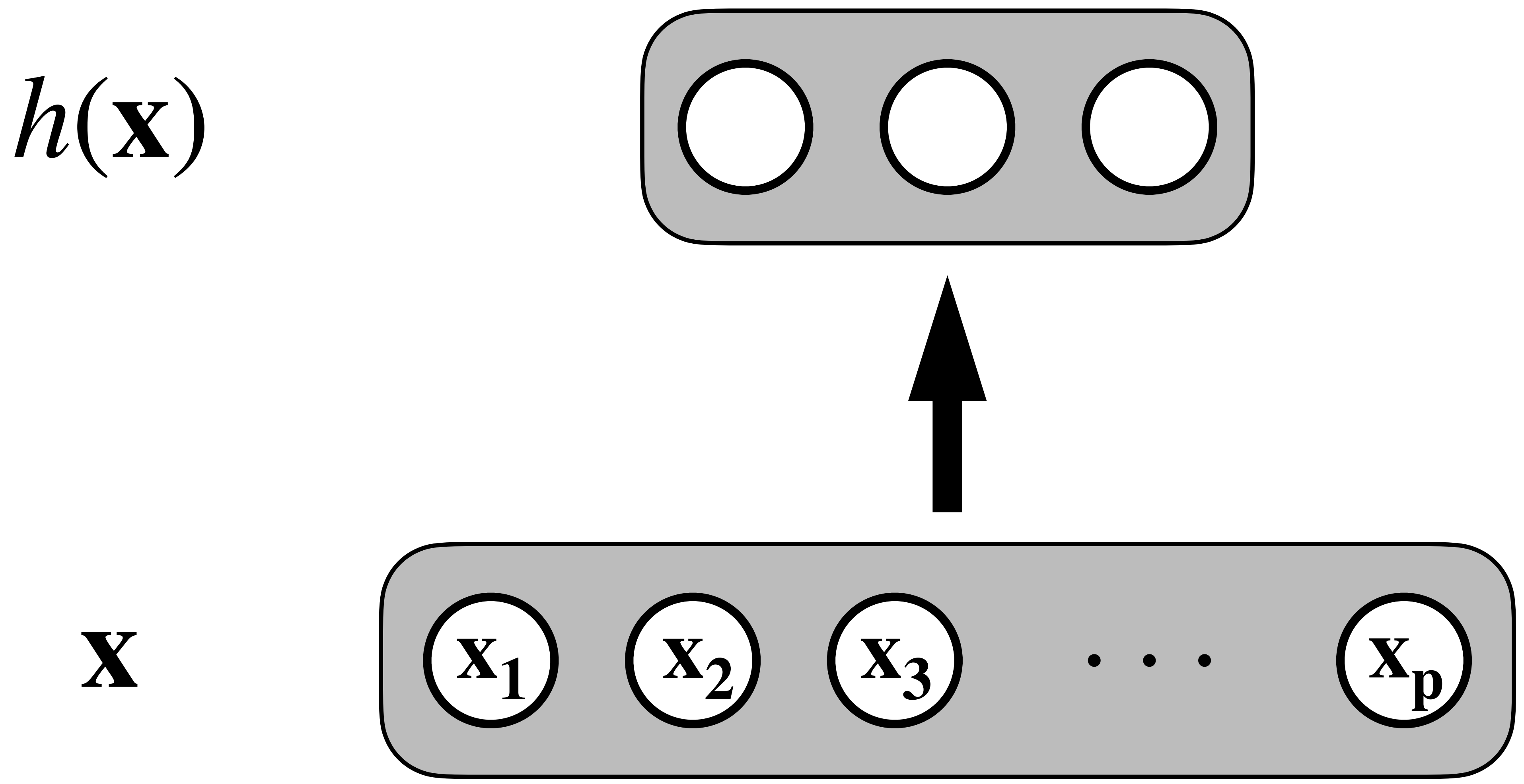
Auto-encoder - Recall

For an AE with a single hidden layer:



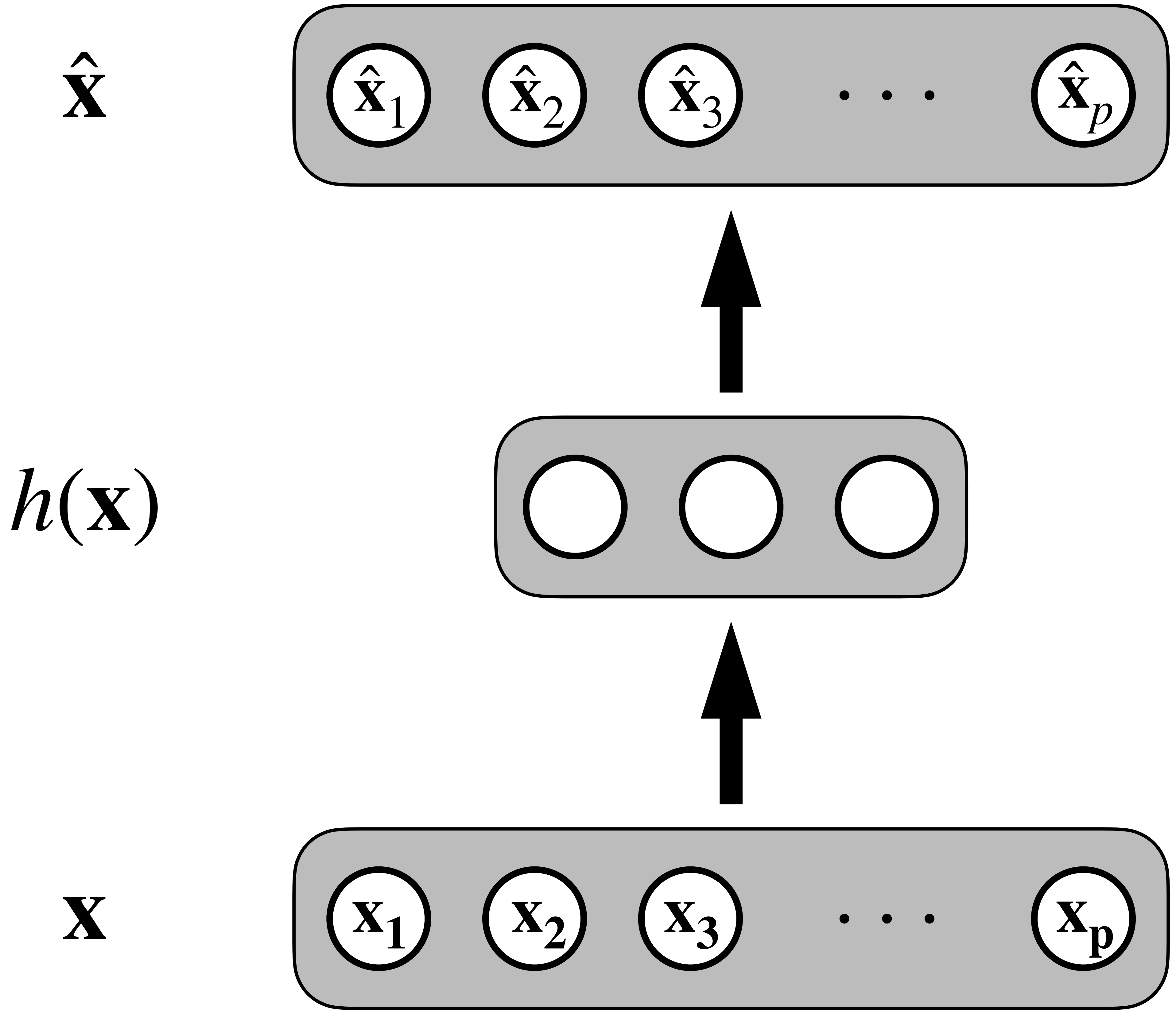
Auto-encoder - Recall

For an AE with a single hidden layer:



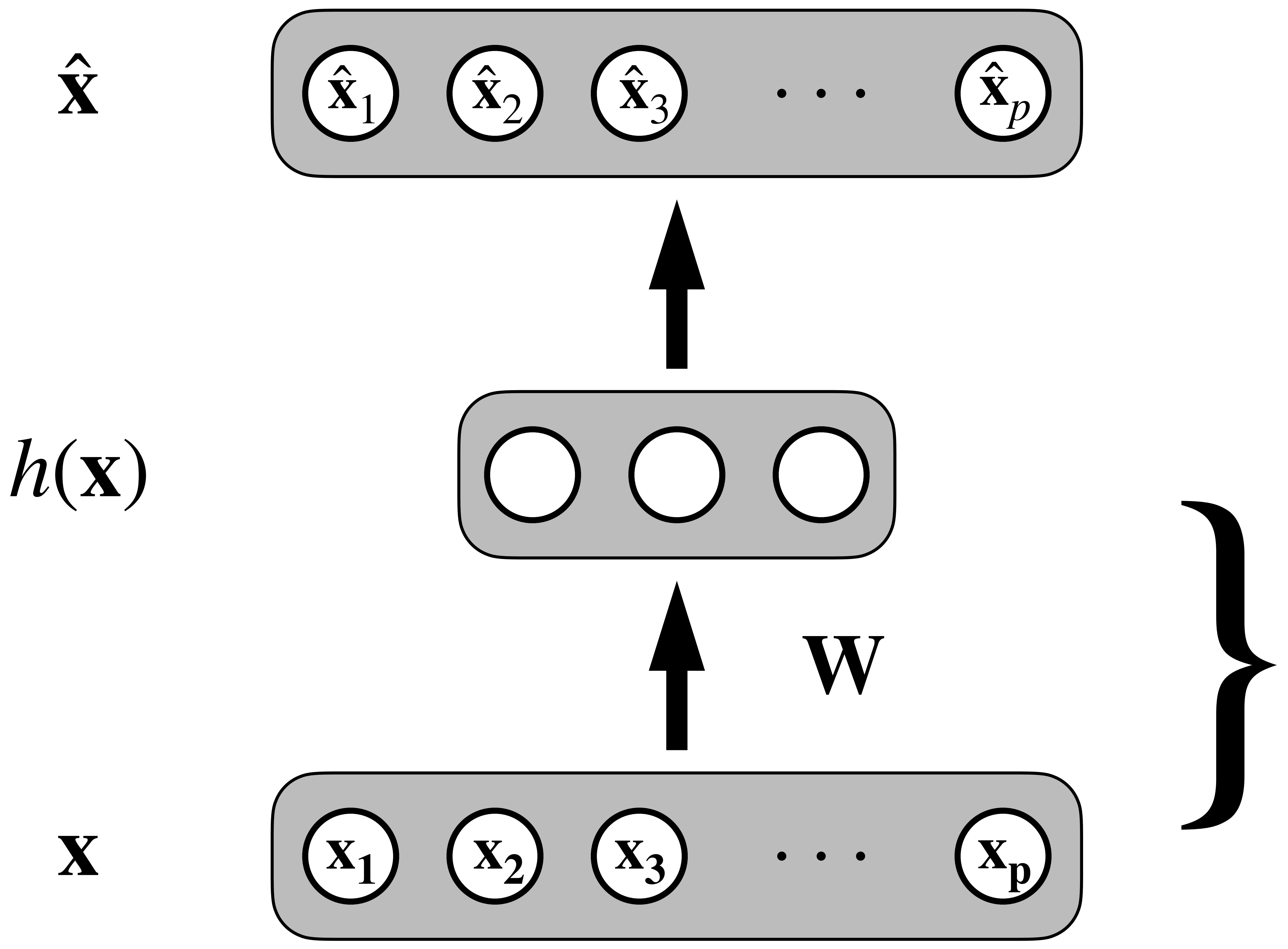
Auto-encoder - Recall

For an AE with a single hidden layer:



Auto-encoder - Recall

For an AE with a single hidden layer:

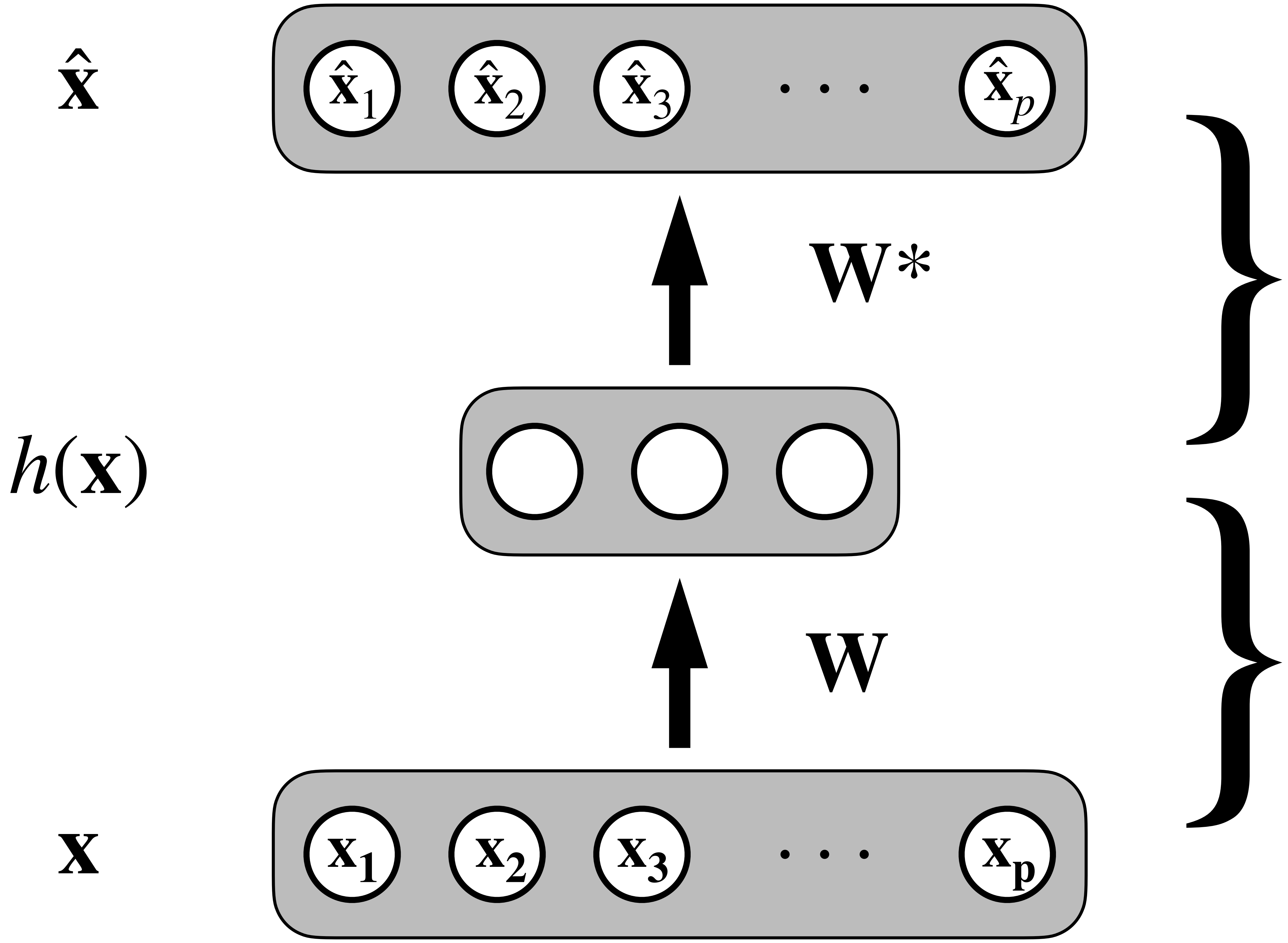


Encoder

$$h(\mathbf{x}) = g(\mathbf{a}(\mathbf{x}))$$
$$= g(\mathbf{W}\mathbf{x} + \mathbf{b})$$

Auto-encoder - Recall

For an AE with a single hidden layer:



Decoder

$$\begin{aligned}\hat{\mathbf{x}} &= o(\hat{\mathbf{a}}(\mathbf{x})) \\ &= o(\mathbf{W}^* h(\mathbf{x}) + \mathbf{b}^*)\end{aligned}$$

Encoder

$$\begin{aligned}h(\mathbf{x}) &= g(\mathbf{a}(\mathbf{x})) \\ &= g(\mathbf{W}\mathbf{x} + \mathbf{b})\end{aligned}$$

Variational Auto-encoder (VAE)

* Understanding these models precisely requires concepts that are beyond our class. Here, we aim for familiarization with the terms and an intuitive understanding.

Variational Auto-Encoder - Motivation

Idea:

- The data are generated conditioned on a random variable (Z):

$$\mathbb{P}_{\theta}\left(\mathbf{x} \mid \mathbf{z}\right)$$

You can think of Z has an embedding.
VAEs will learn a prior P(z) and
a posterior P(z|x) over it.

Variational Auto-Encoder - Motivation

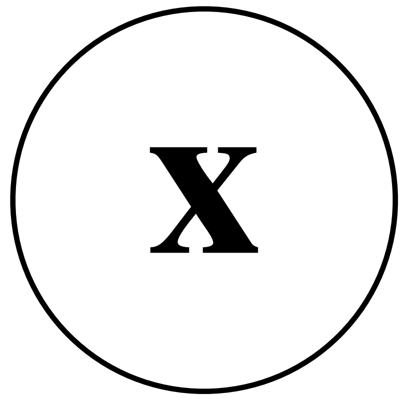
Idea:

- The data are generated conditioned on a random variable (Z):

$$\mathbb{P}_\theta\left(\mathbf{x} \mid \mathbf{z}\right)$$

You can think of Z has an embedding.
VAEs will learn a prior P(z) and
a posterior P(z|x) over it.

<u>Graphical representation</u>



Variational Auto-Encoder - Motivation

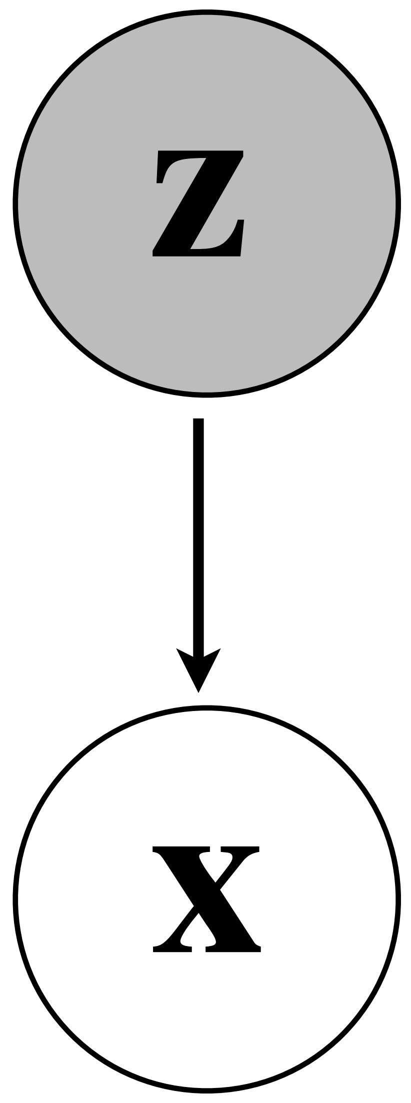
Idea:

- The data are generated conditioned on a random variable (Z):

$$\mathbb{P}_\theta \left(\mathbf{x} \mid \mathbf{z} \right)$$

You can think of Z has an embedding.
VAEs will learn a prior P(z) and
a posterior P(z|x) over it.

Graphical representation



Variational Auto-Encoder - Motivation

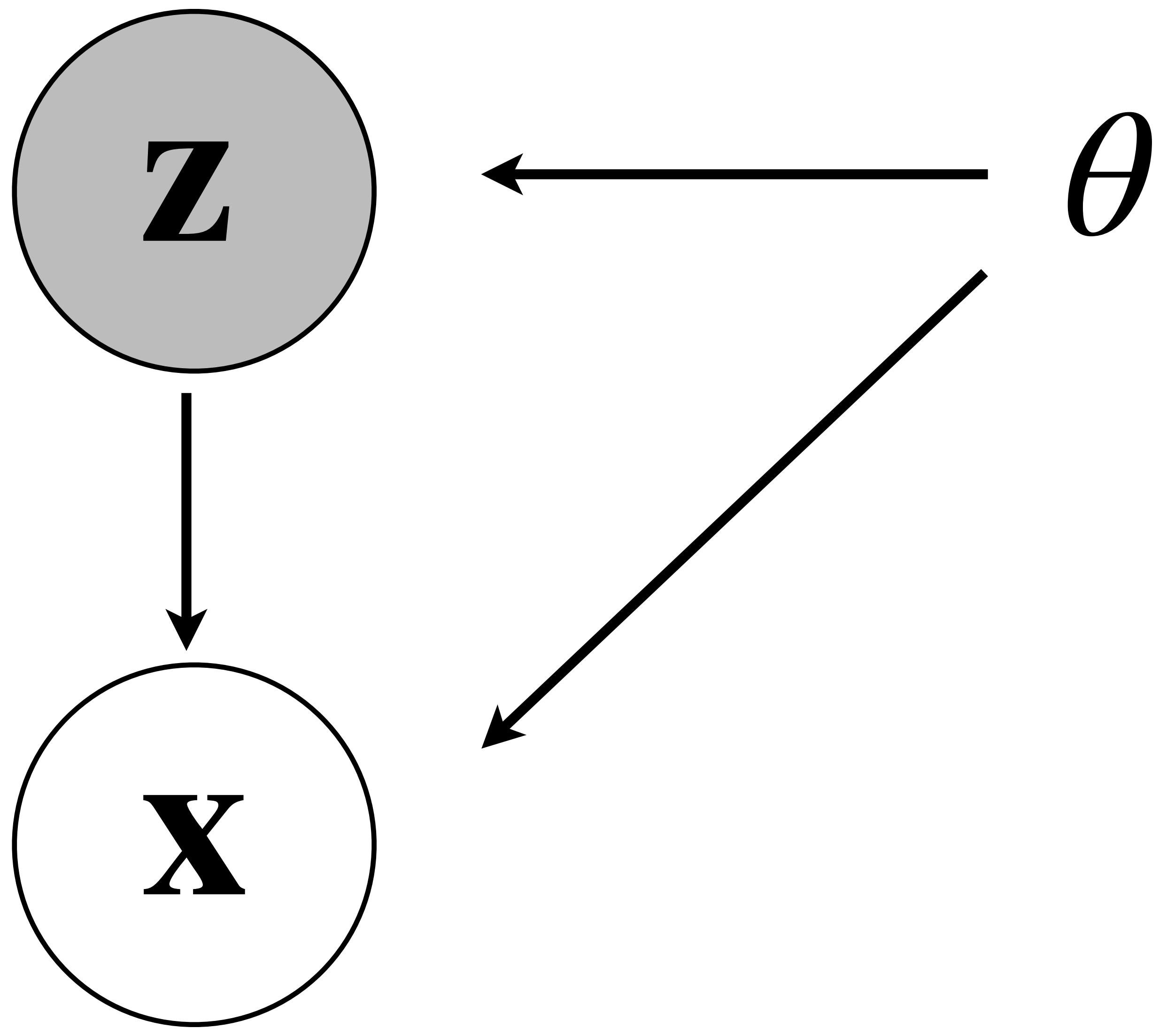
Idea:

- The data are generated conditioned on a random variable (Z):

$$\mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z})$$

You can think of Z has an embedding.
VAEs will learn a prior P(z) and
a posterior P(z|x) over it.

<u>Graphical representation</u>



Variational Auto-Encoder - Motivation

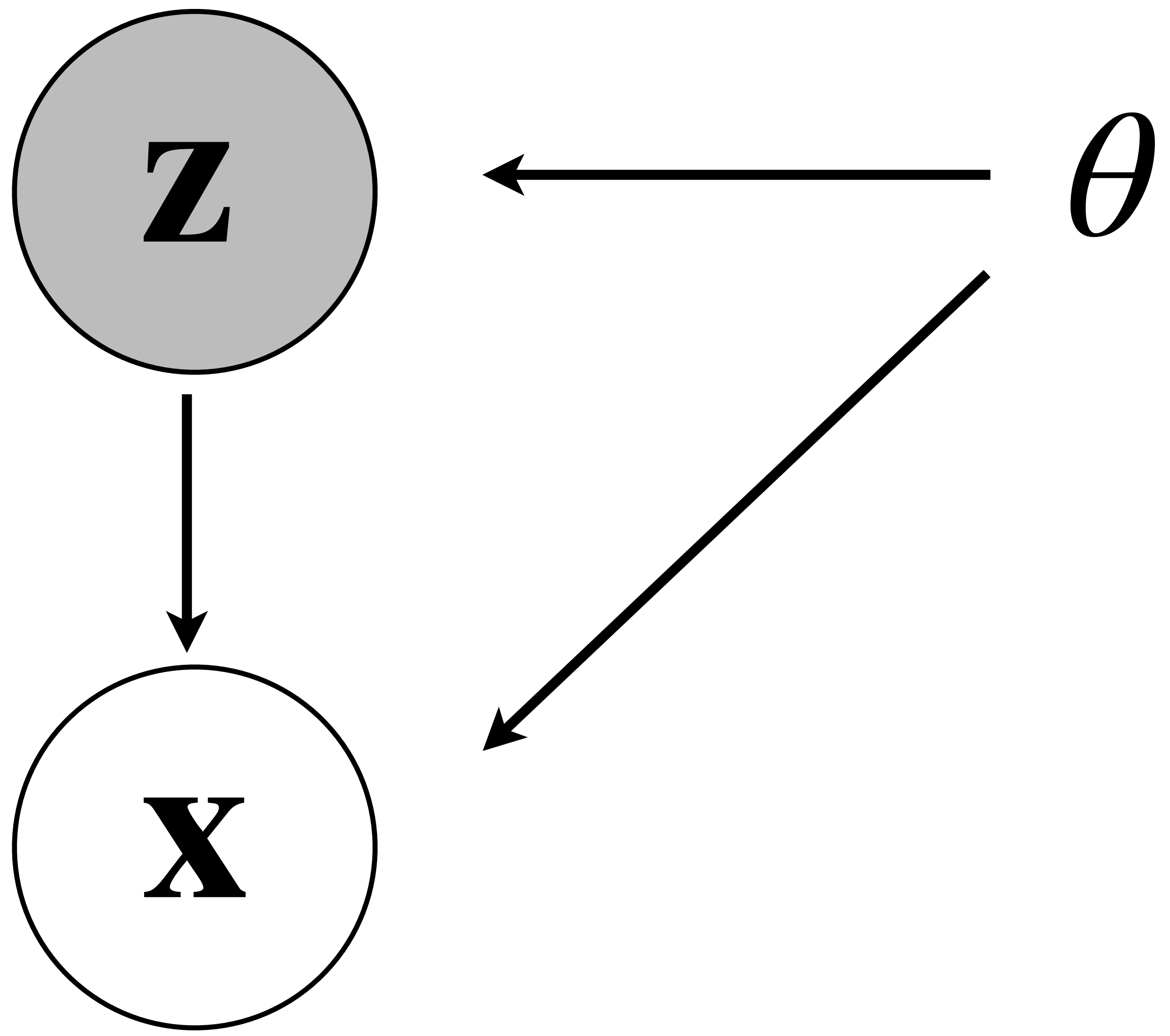
Idea:

- The data are generated conditioned on a random variable (Z):

$$\mathbb{P}_{\theta}\left(\mathbf{x} \mid \mathbf{z}\right)$$

You can think of Z has an embedding.
VAEs will learn a prior P(z) and
a posterior P(z|x) over it.

Graphical representation



Question:

- How do we learn such a distribution?

Variational Auto-Encoder - Motivation

Variational Auto-Encoder - Motivation

Partial answer:

- A "good" model should maximize the likelihood of the data P(x)
- Bayes, provides us with two possibilities:

Variational Auto-Encoder - Motivation

Partial answer:

- A “good” model should maximize the likelihood of the data P(x)
- Bayes, provides us with two possibilities:

$$\mathbb{P}_{\theta}\left(\mathbf{x}\right) = \int_{\mathbf{z}} \mathbb{P}_{\theta}\left(\mathbf{z} \mid \mathbf{x}\right) \mathbb{P}_{\theta}\left(\mathbf{x}\right) \mathrm{d}\mathbf{z}$$

Variational Auto-Encoder - Motivation

Partial answer:

- A "good" model should maximize the likelihood of the data P(x)
- Bayes, provides us with two possibilities:

$$\mathbb{P}_{\theta}(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_{\theta}(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_{\theta}(\mathbf{x}) \mathrm{d}\mathbf{z}$$

Problem:

Variational Auto-Encoder - Motivation

Partial answer:

- A “good” model should maximize the likelihood of the data P(x)
- Bayes, provides us with two possibilities:

$$\mathbb{P}_{\theta}(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_{\theta}(\mathbf{z} \mid \mathbf{x})\mathbb{P}_{\theta}(\mathbf{x})\mathrm{d}\mathbf{z}$$

Problem:

- The posterior can be intractable.

Variational Auto-Encoder - Motivation

Partial answer:

- A “good” model should maximize the likelihood of the data P(x)
- Bayes, provides us with two possibilities:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \boxed{\mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x})} \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

Problem:

- The posterior can be intractable.

Variational Auto-Encoder - Motivation

Partial answer:

- A “good” model should maximize the likelihood of the data P(x)
- Bayes, provides us with two possibilities:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \boxed{\mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x})} \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z}) \mathbb{P}_\theta(\mathbf{z}) d\mathbf{z}$$

Problem:

- The posterior can be intractable.

Variational Auto-Encoder - Motivation

Partial answer:

- A "good" model should maximize the likelihood of the data P(x)
- Bayes, provides us with two possibilities:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \boxed{\mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x})} \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z}) \mathbb{P}_\theta(\mathbf{z}) d\mathbf{z}$$

Problem:

- The posterior can be intractable.
- The integral is intractable.

Variational Auto-Encoder - Motivation

Partial answer:

- A “good” model should maximize the likelihood of the data P(x)
- Bayes, provides us with two possibilities:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z}) \mathbb{P}_\theta(\mathbf{z}) d\mathbf{z}$$

Problem:

- The posterior can be intractable.
- The integral is intractable.

Variational Auto-Encoder - Motivation

Let's explore the first problem:

- The posterior can be intractable:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) \mathrm{d}\mathbf{z}$$

Variational Auto-Encoder - Motivation

Let's explore the first problem:

- The posterior can be intractable:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) \mathrm{d}\mathbf{z}$$

Idea:

- We could approximate the real posterior to make it tractable

Variational Auto-Encoder - Motivation

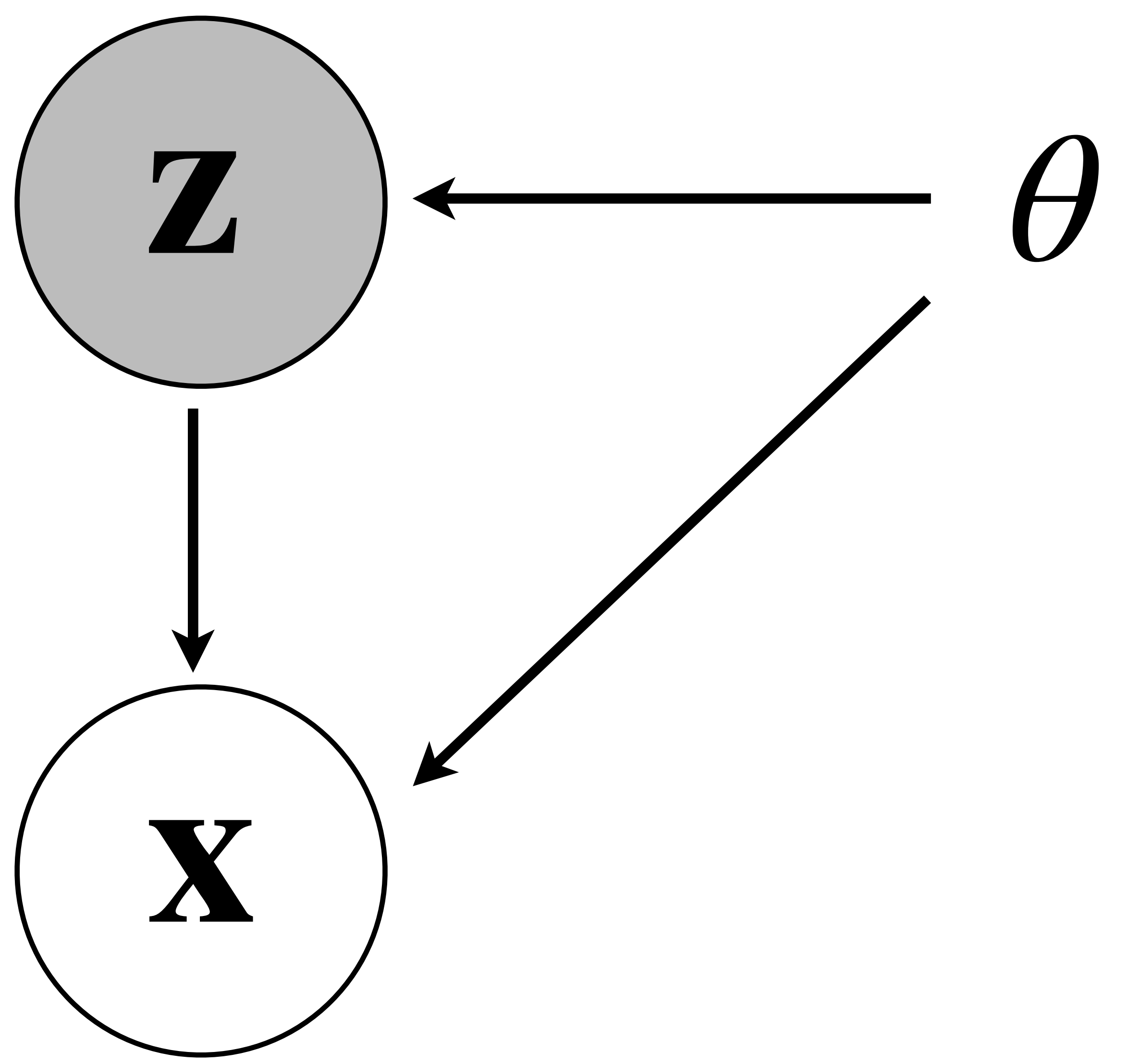
Let's explore the first problem:

- The posterior can be intractable:

$$\mathbb{P}_{\theta}\left(\mathbf{x}\right) = \int_{\mathbf{z}} \mathbb{P}_{\theta}\left(\mathbf{z} \mid \mathbf{x}\right) \mathbb{P}_{\theta}\left(\mathbf{x}\right) \mathrm{d}\mathbf{z}$$

Idea:

- We could approximate the real posterior to make it tractable



Variational Auto-Encoder - Motivation

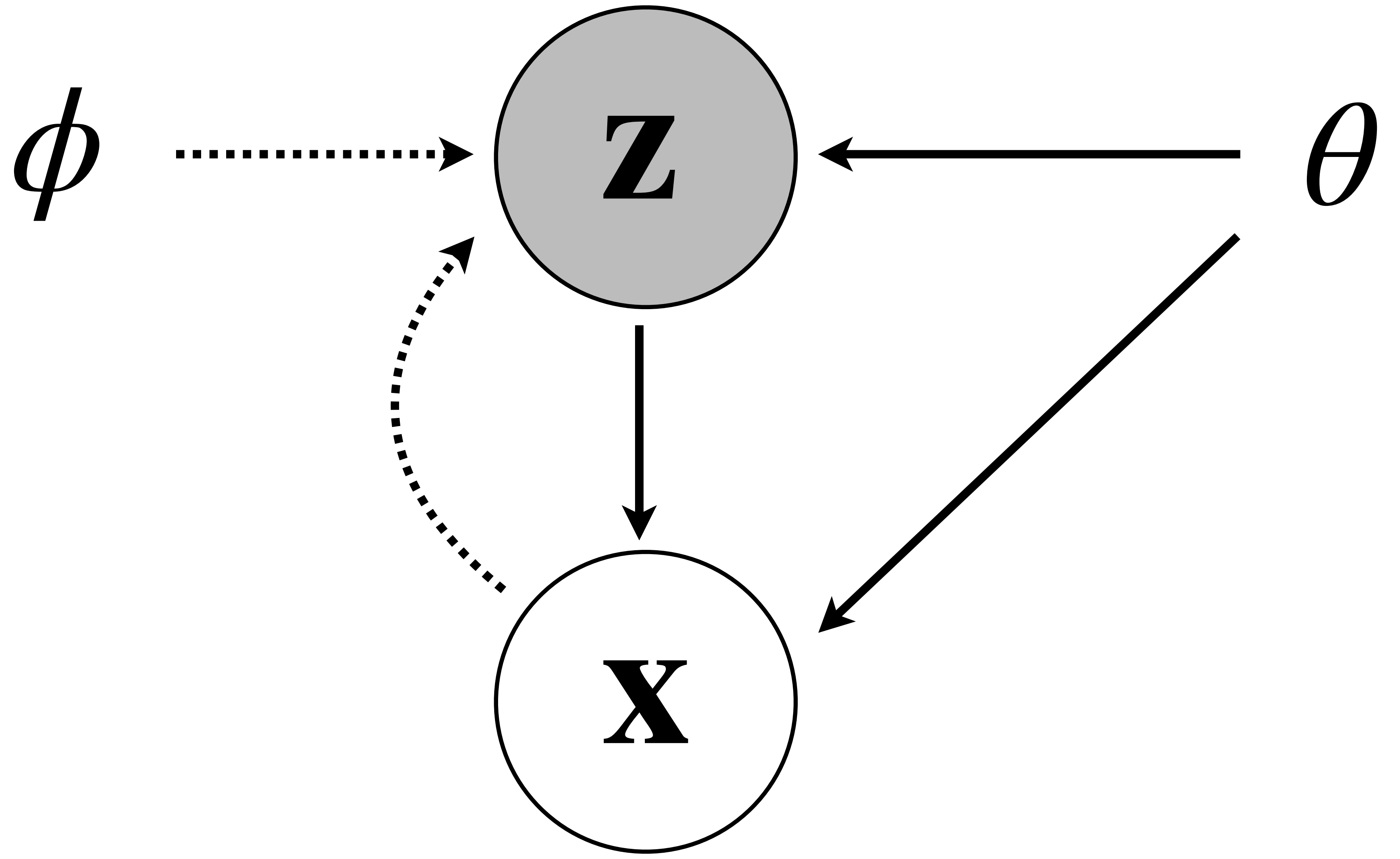
Let's explore the first problem:

- The posterior can be intractable:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

Idea:

- We could approximate the real posterior to make it tractable



Variational Auto-Encoder - Motivation

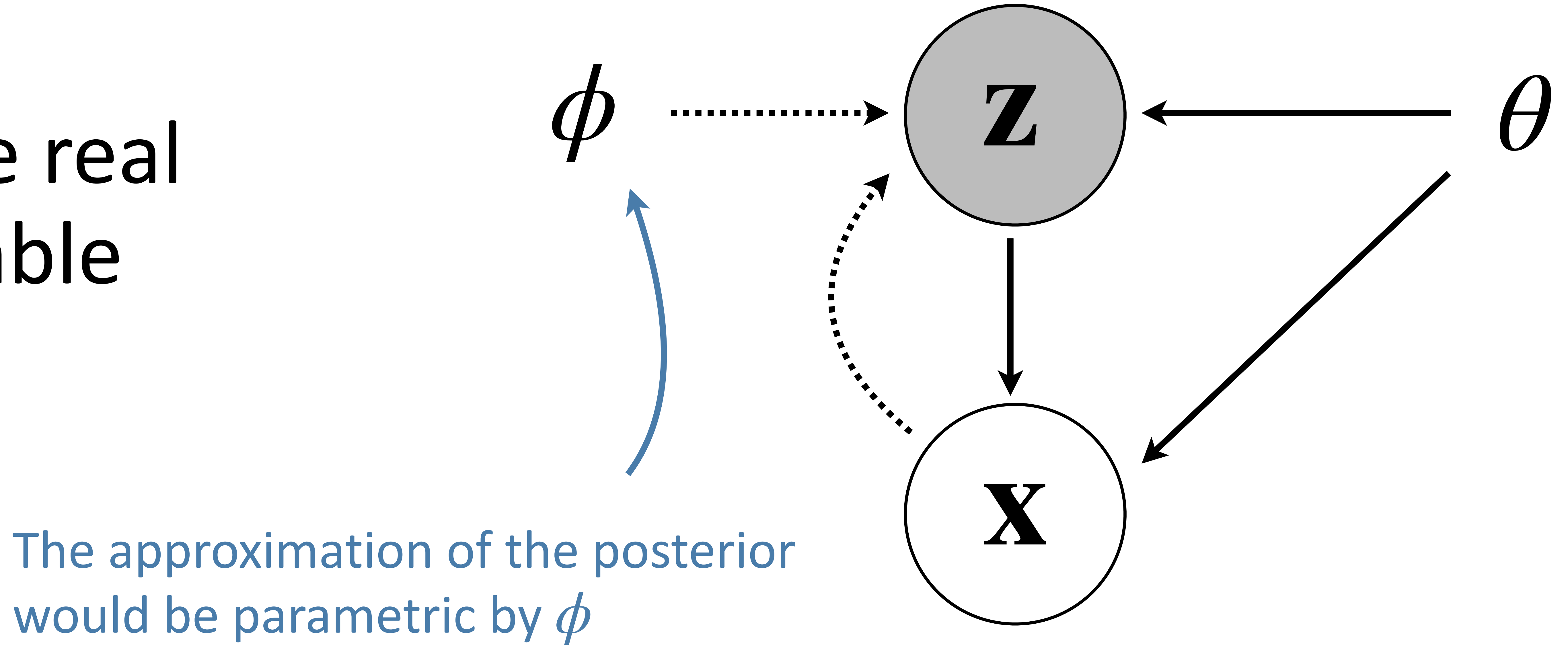
Let's explore the first problem:

- The posterior can be intractable:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

Idea:

- We could approximate the real posterior to make it tractable



Variational Auto-Encoder - Formalism

We can obtain a bound on the log-likelihood of $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \;\geq -\,\mathsf{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \;||\; q_\phi(\mathbf{z})\} \;+\; \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Variational Auto-Encoder - Formalism

We can obtain a bound on the log-likelihood of $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \;\geq -\,\mathsf{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \;||\; q_\phi(\mathbf{z})\} \;+\; \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Where:

Variational Auto-Encoder - Formalism

We can obtain a bound on the log-likelihood of $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \;\geq -\,\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \;||\; q_\phi(\mathbf{z})\} \;+\; \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Where:

- We're trying to the distance between the posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ and the prior $q_\phi(\mathbf{z})$.

Variational Auto-Encoder - Formalism

We can obtain a bound on the log-likelihood of $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\boxed{\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \mid\mid q_\phi(\mathbf{z})\}} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Where:

- We're trying to the distance between the posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ and the prior $q_\phi(\mathbf{z})$.

Variational Auto-Encoder - Formalism

We can obtain a bound on the log-likelihood of $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\boxed{\mathsf{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \;||\; q_\phi(\mathbf{z})\}} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Where:

- We're trying to the distance between the posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ and the prior $q_\phi(\mathbf{z})$.
- While maximizing the conditional log-likelihood of $\mathbf{x}$.

Variational Auto-Encoder - Formalism

We can obtain a bound on the log-likelihood of $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \mid\mid q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Where:

- We're trying to the distance between the posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ and the prior $q_\phi(\mathbf{z})$.
- While maximizing the conditional log-likelihood of $\mathbf{x}$.

Variational Auto-Encoder - Formalism

We can obtain a bound on the log-likelihood of $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\text{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \mid\mid q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Where:

- We're trying to the distance between the posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ and the prior $q_\phi(\mathbf{z})$.
- While maximizing the conditional log-likelihood of $\mathbf{x}$.

Question:

Variational Auto-Encoder - Formalism

We can obtain a bound on the log-likelihood of $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \mid\mid q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Where:

- We're trying to the distance between the posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ and the prior $q_\phi(\mathbf{z})$.
- While maximizing the conditional log-likelihood of $\mathbf{x}$.

Question:

- How do we do this in practice?

Variational Auto-Encoder - Formalism

Variational Auto-Encoder - Formalism

$$\log p_\theta(\mathbf{x}) \;\; \geq - \mathsf{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \;||\; q_\phi(\mathbf{z})\} \;\; + \;\; \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Variational Auto-Encoder - Formalism

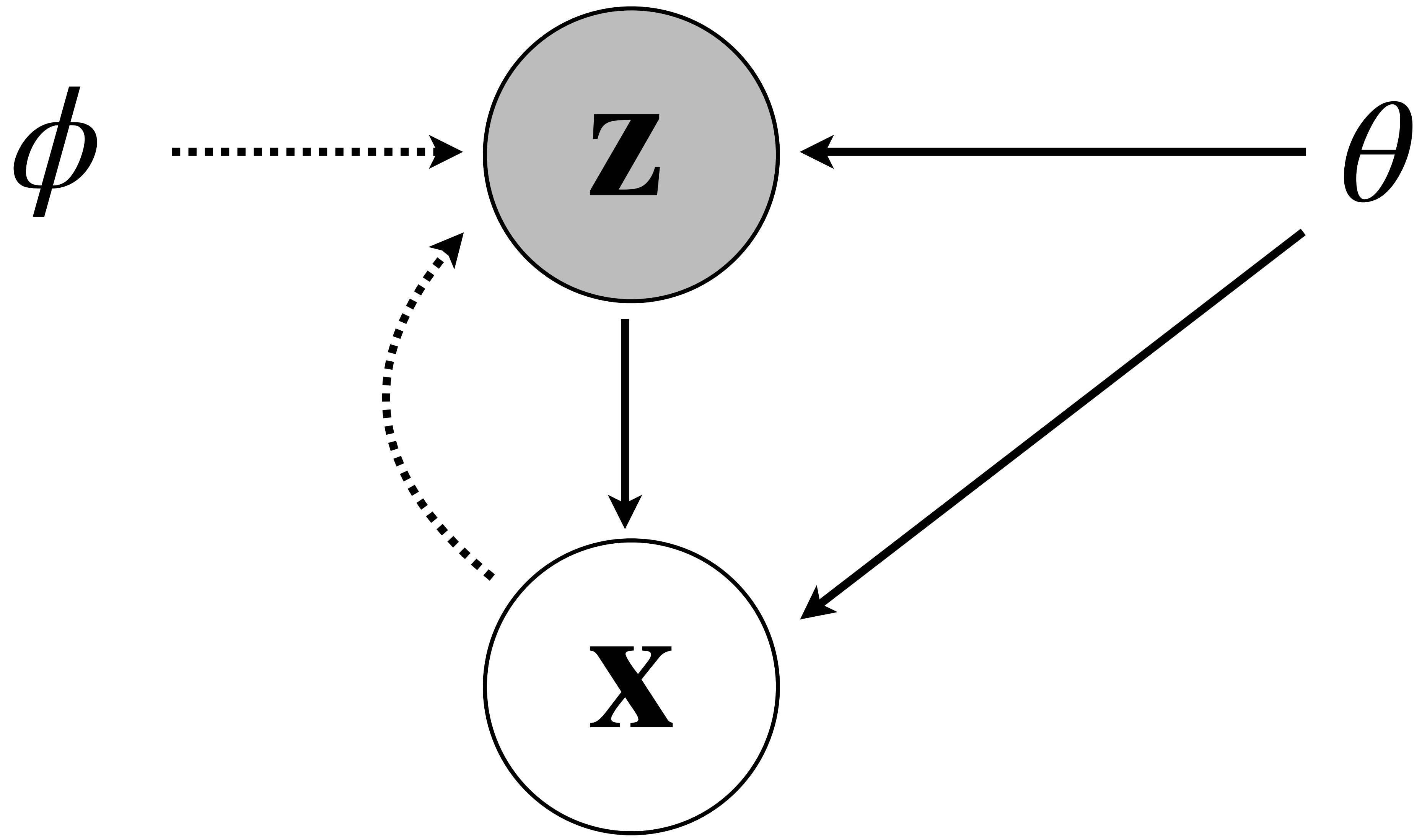
$$\log p_\theta(\mathbf{x}) \ \geq - \mathsf{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \mid\mid q_\phi(\mathbf{z})\} \ + \ \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Graphical view:

Variational Auto-Encoder - Formalism

$$\log p_\theta(\mathbf{x}) \;\geq -\,\mathsf{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \;||\; q_\phi(\mathbf{z})\} \;+\; \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

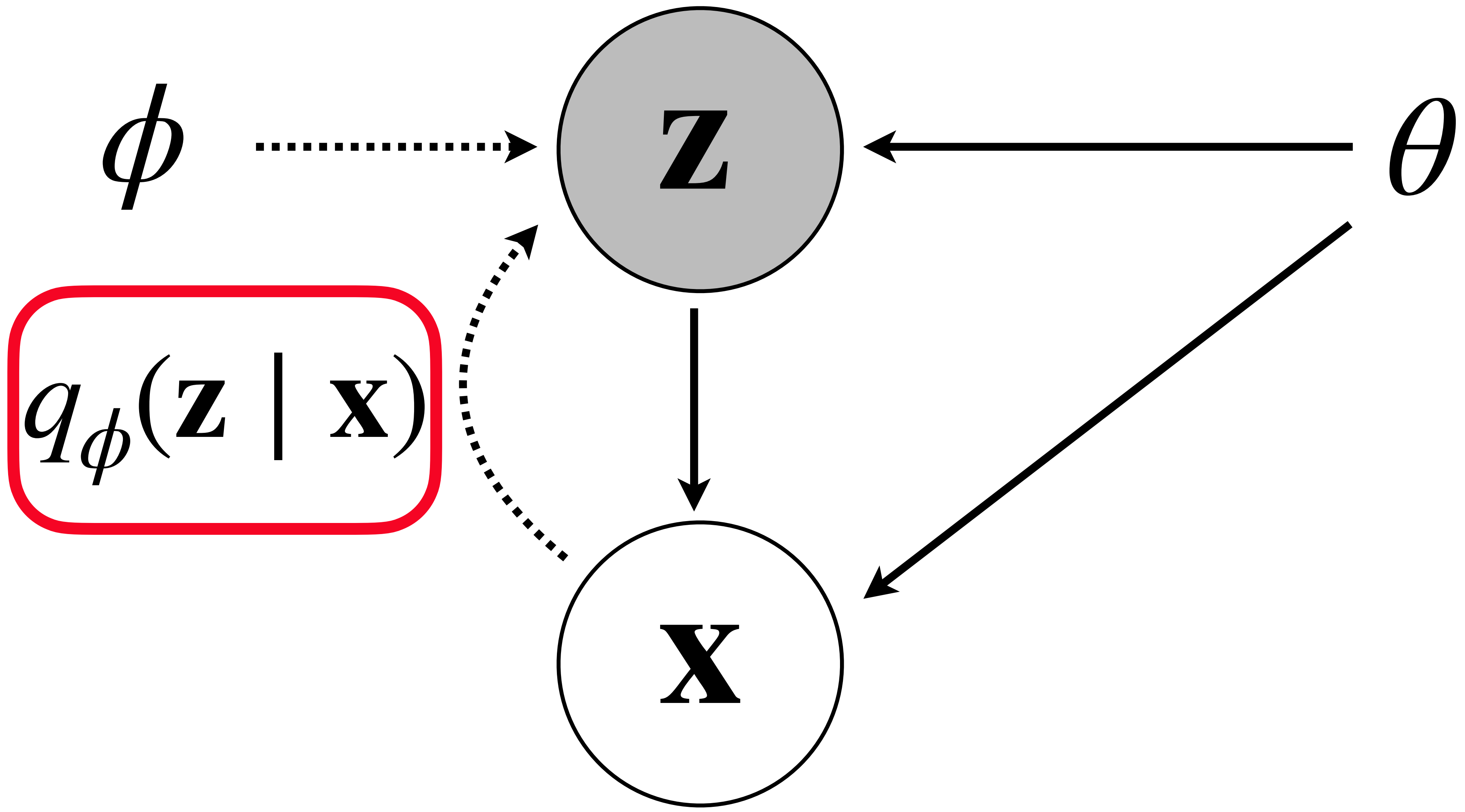
Graphical view:



Variational Auto-Encoder - Formalism

$$\log p_\theta(\mathbf{x}) \;\geq -\,\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} \;+\; \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

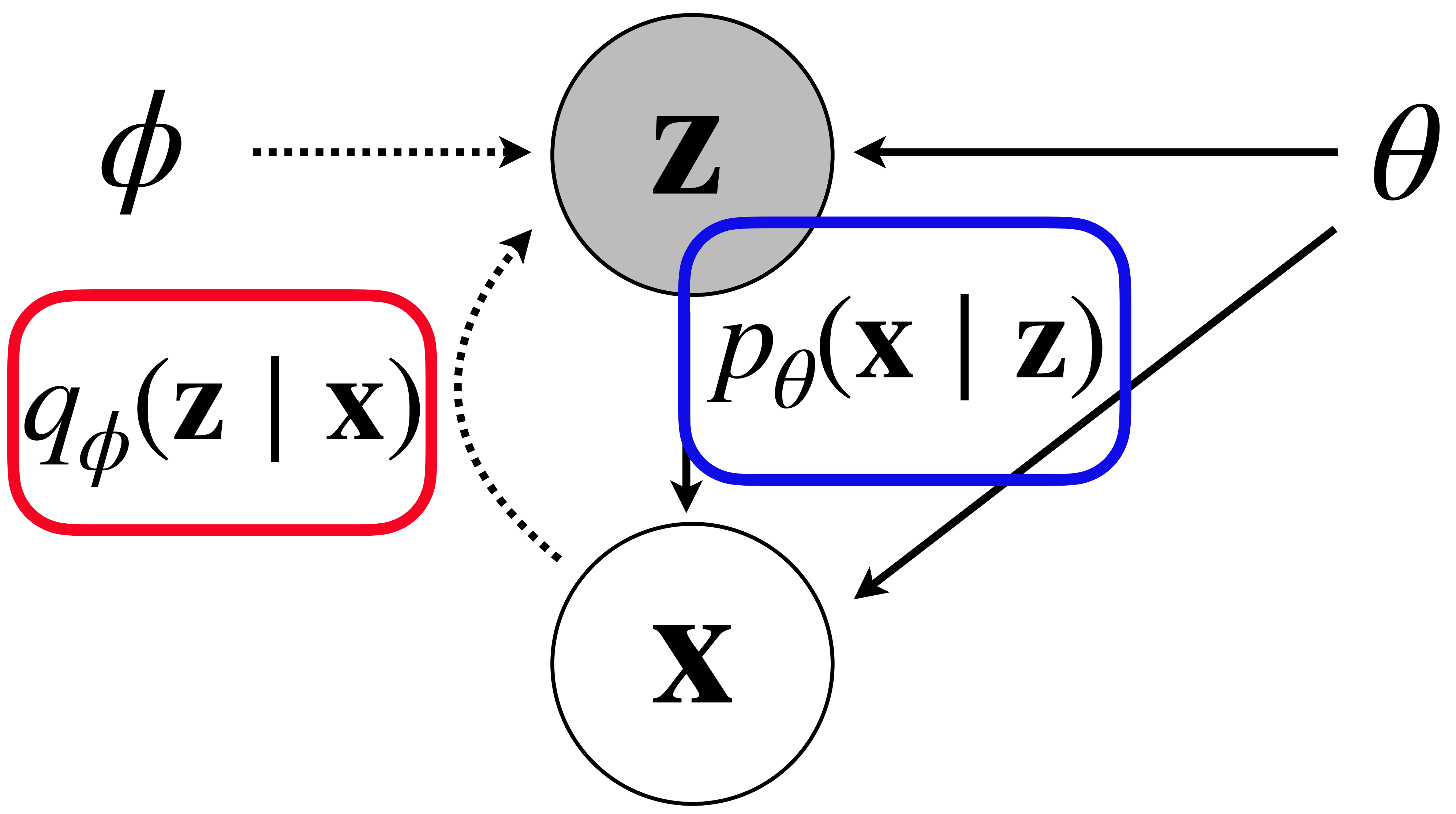
Graphical view:



Variational Auto-Encoder - Formalism

$$\log p_\theta(\mathbf{x}) \geq -\text{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \mid\mid q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

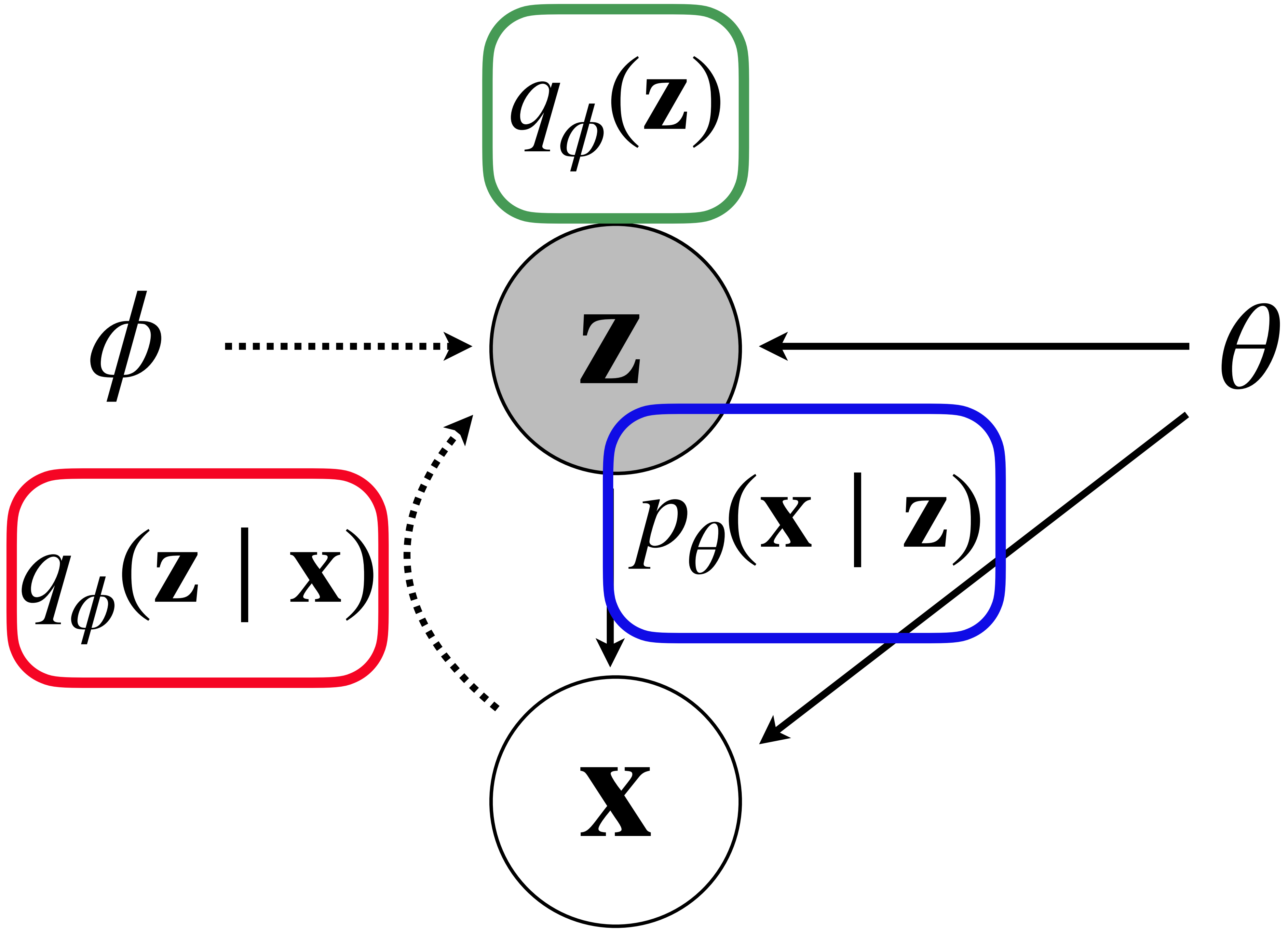
Graphical view:



Variational Auto-Encoder - Formalism

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

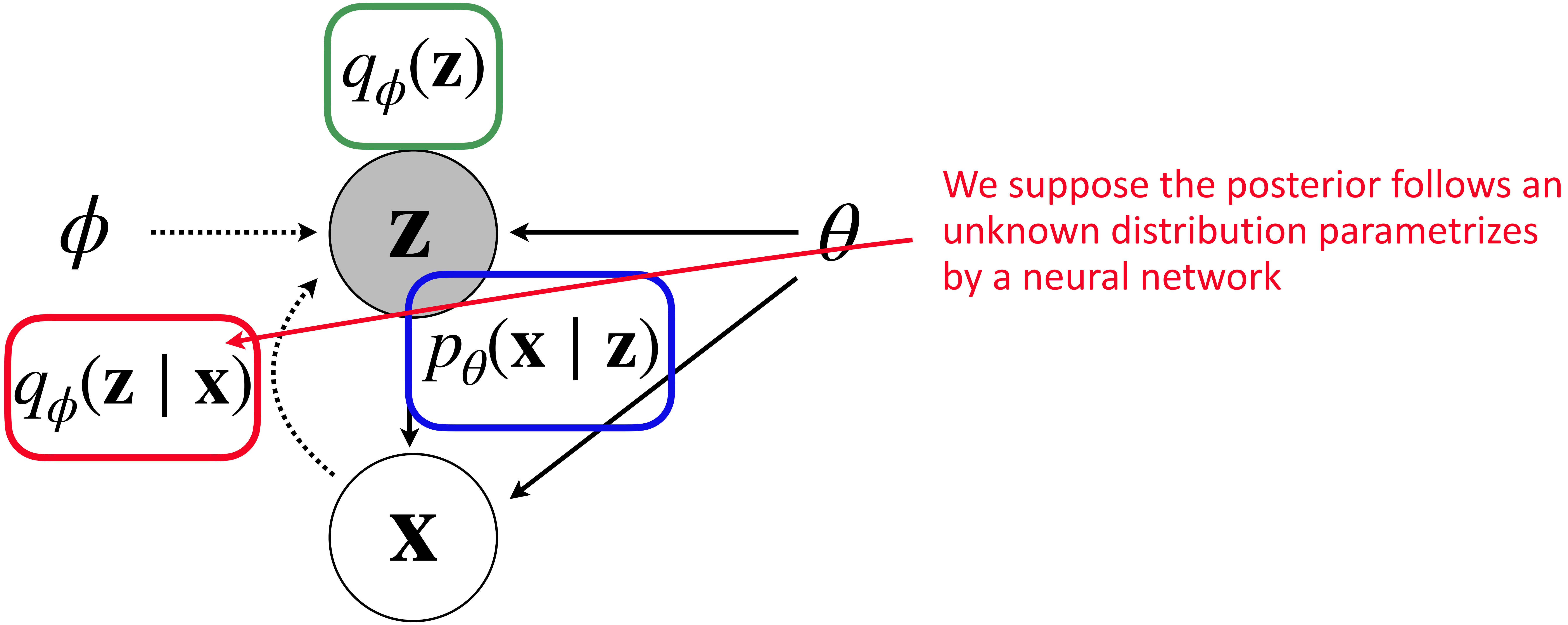
Graphical view:



Variational Auto-Encoder - Formalism

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

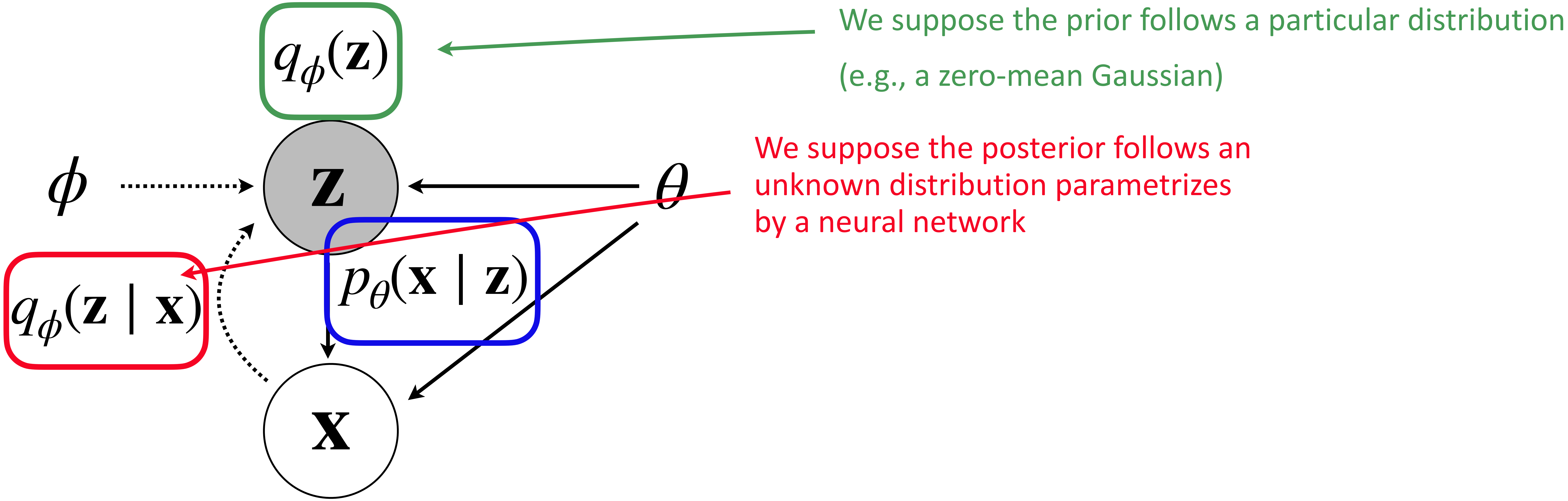
Graphical view:



Variational Auto-Encoder - Formalism

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Graphical view:



Variational Auto-encoders

Variational Auto-encoders

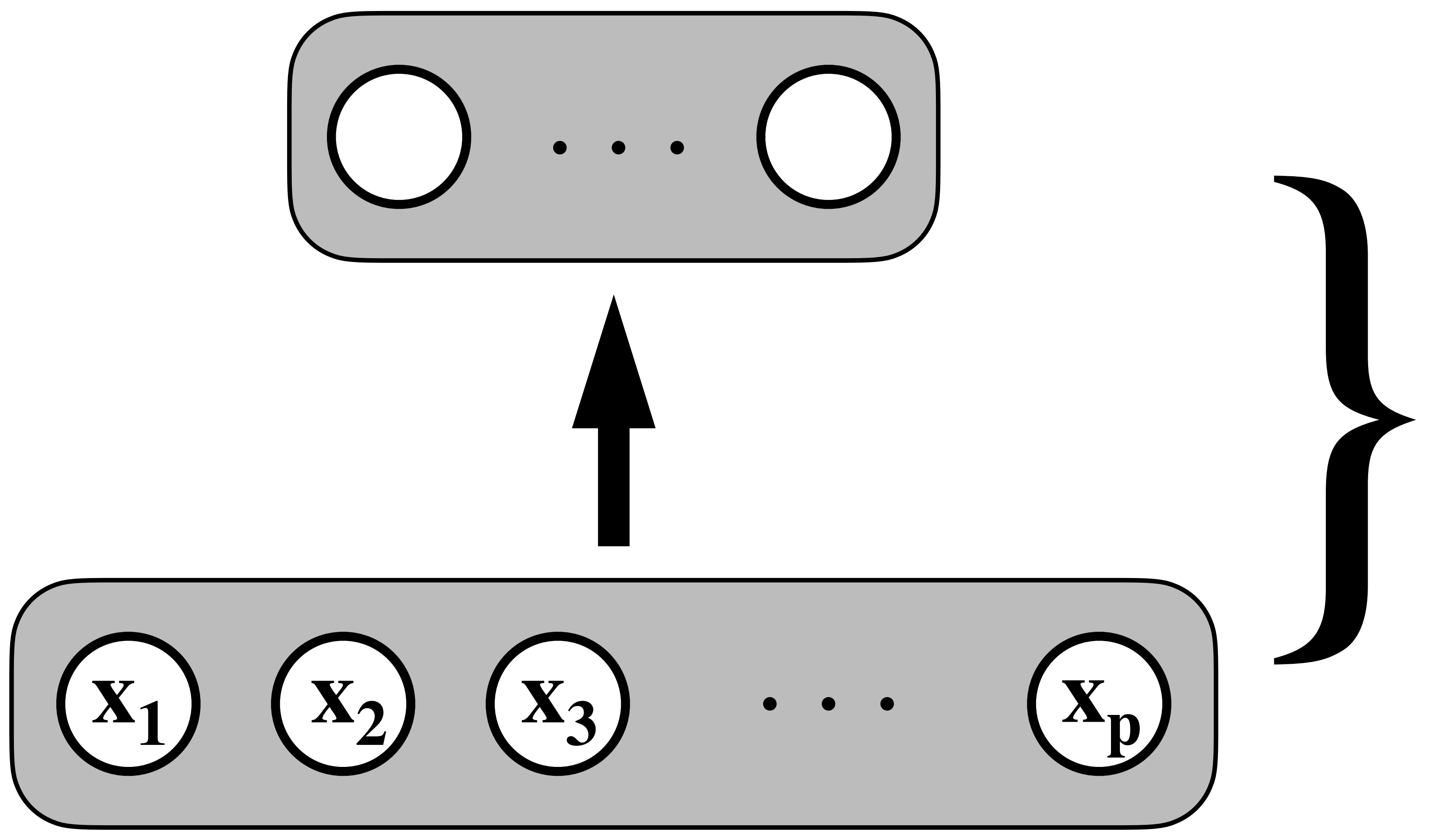
We can estimate these parameters using an auto encoder:

Variational Auto-encoders

We can estimate these parameters using an auto encoder:

Variational Auto-encoders

We can estimate these parameters using an auto encoder:



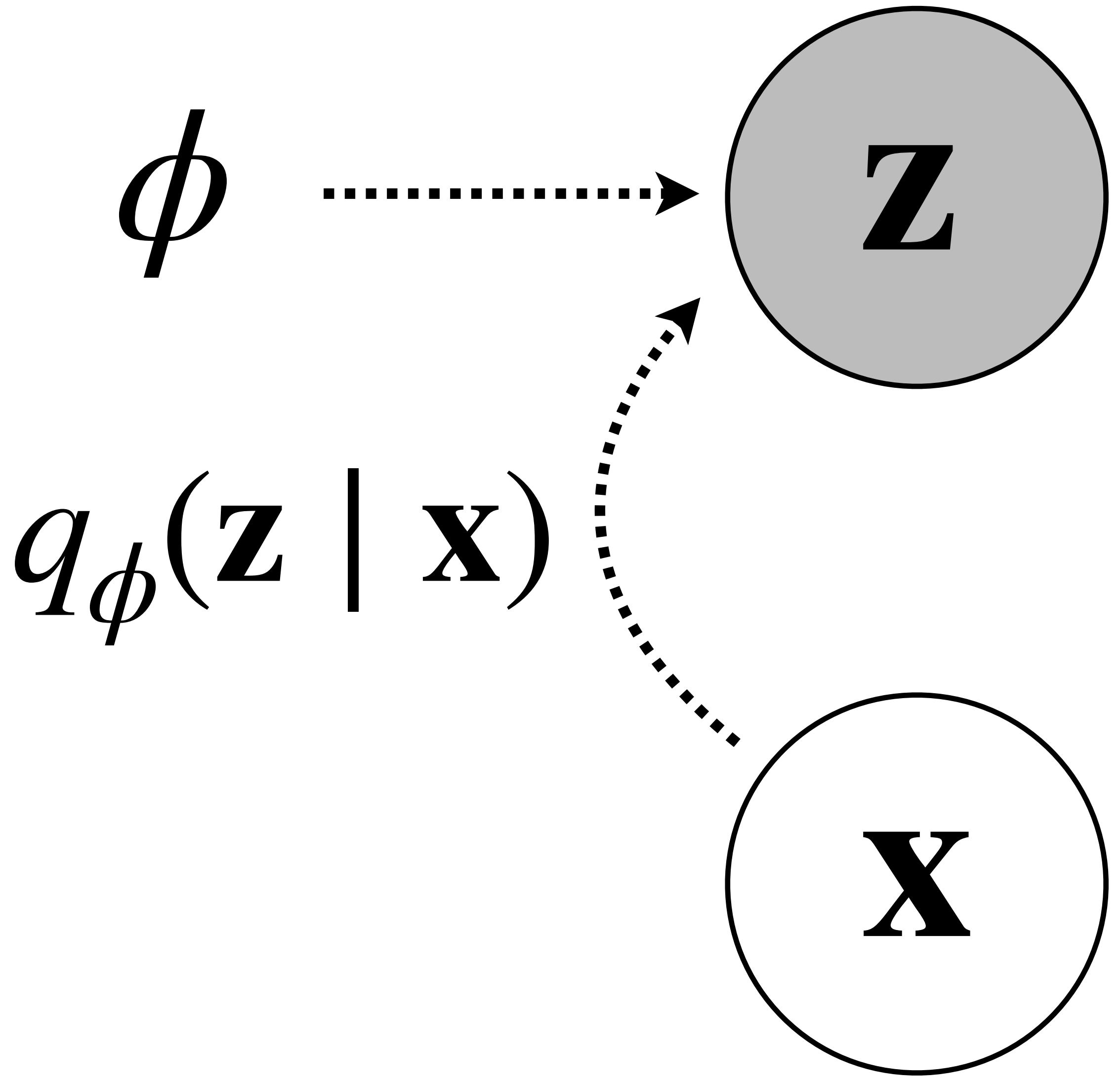
Encodeur

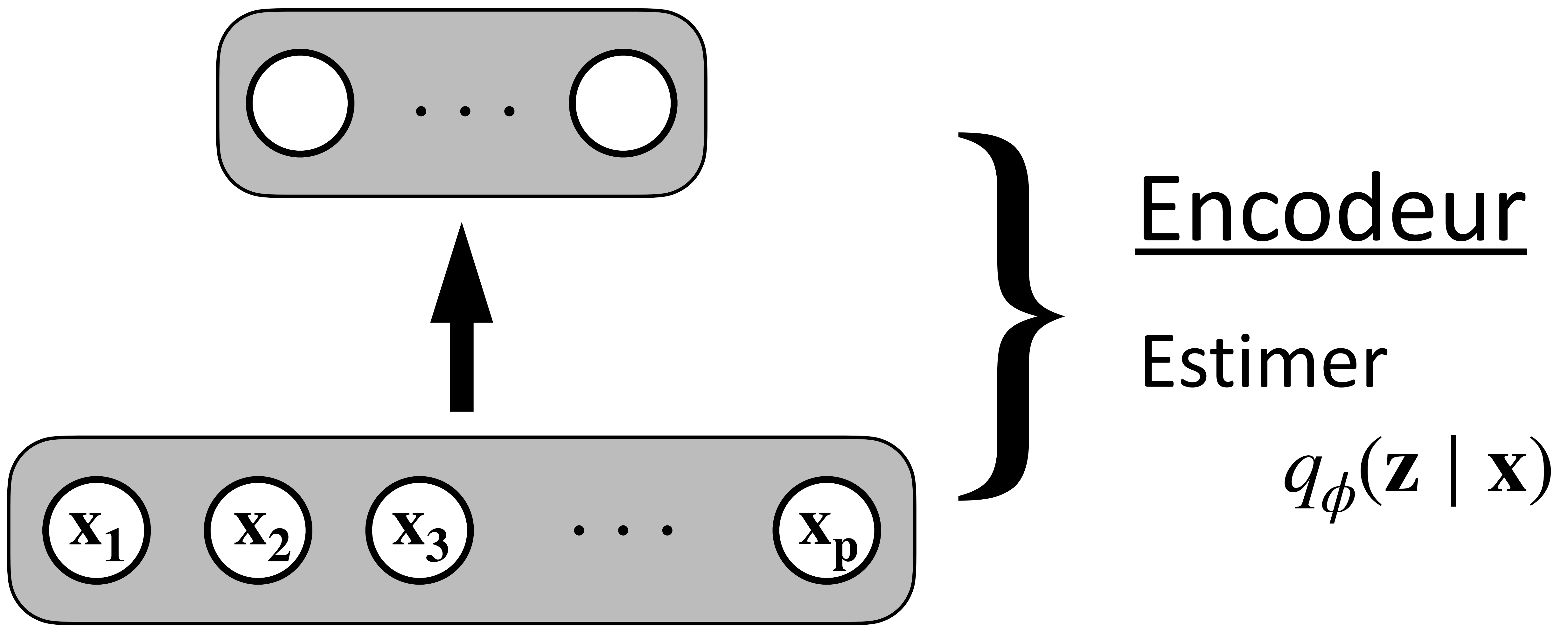
Estimer

$q_\phi(\mathbf{z} \mid \mathbf{x})$

Variational Auto-encoders

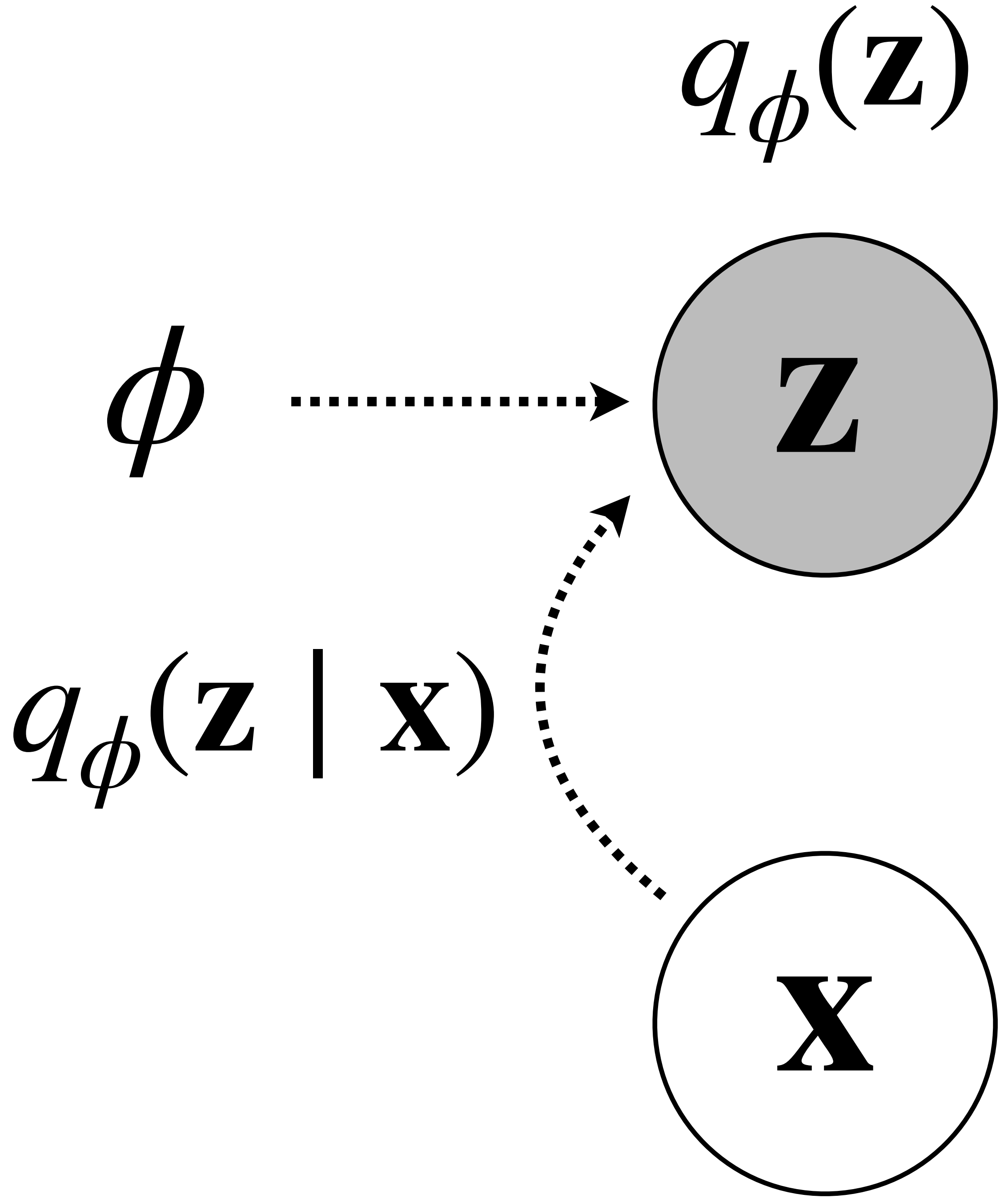
We can estimate these parameters using an auto encoder:

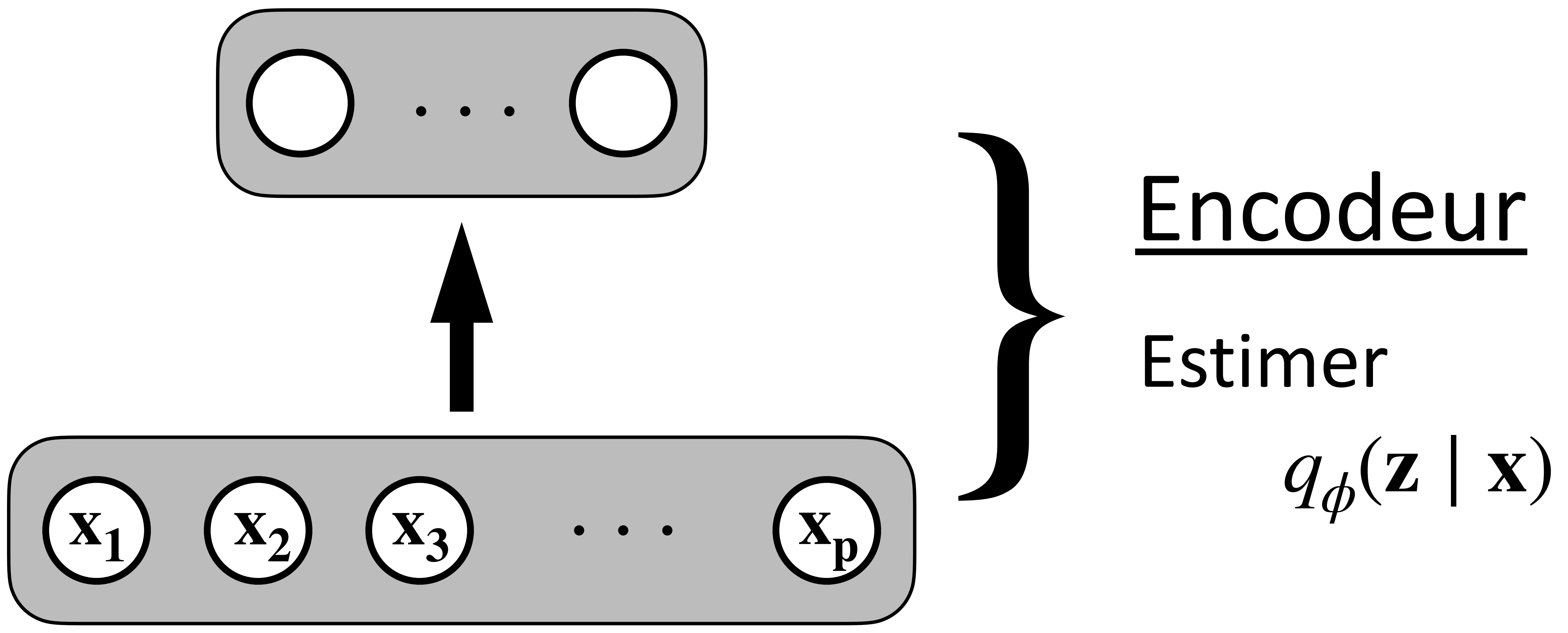




Variational Auto-encoders

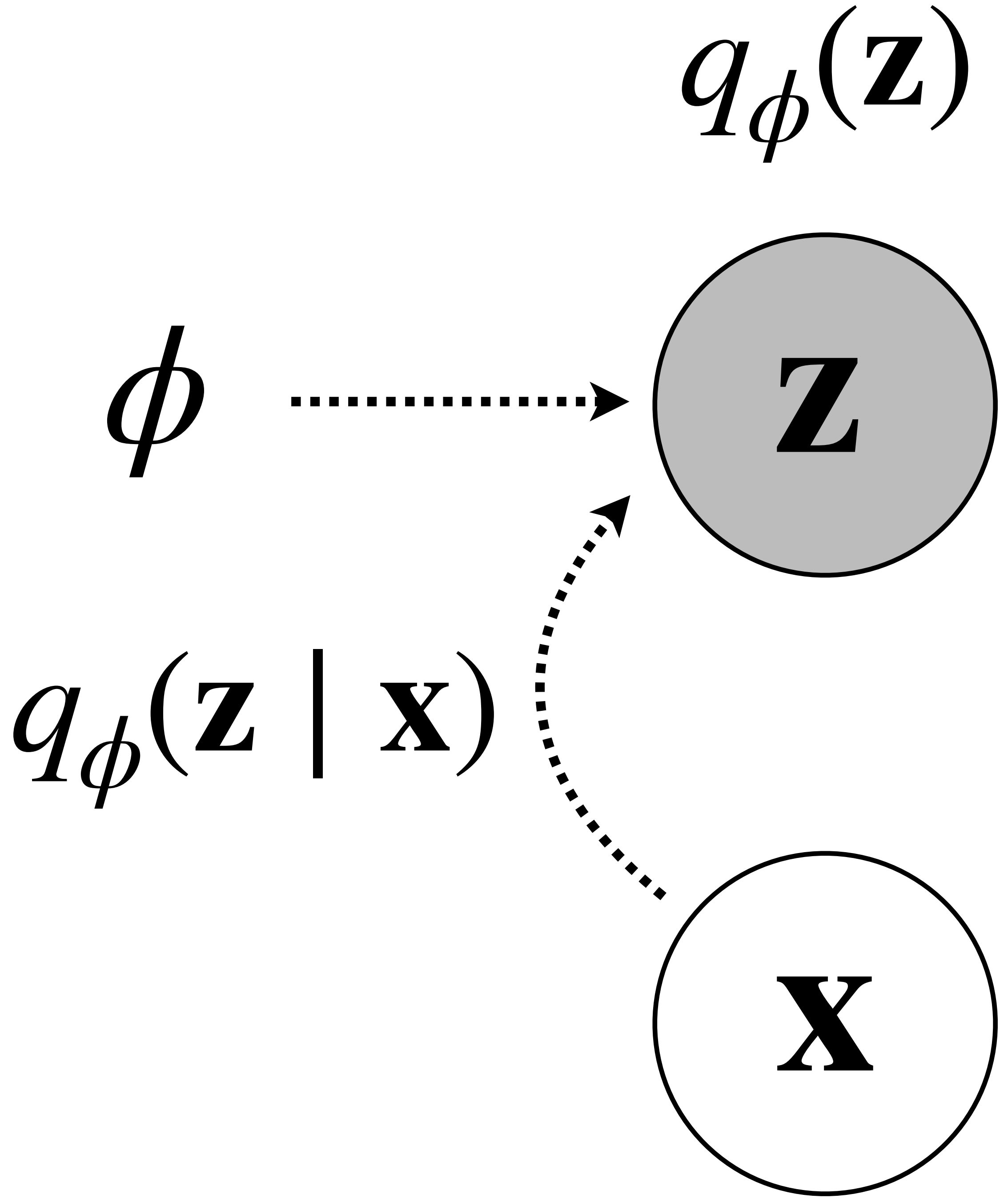
We can estimate these parameters using an auto encoder:

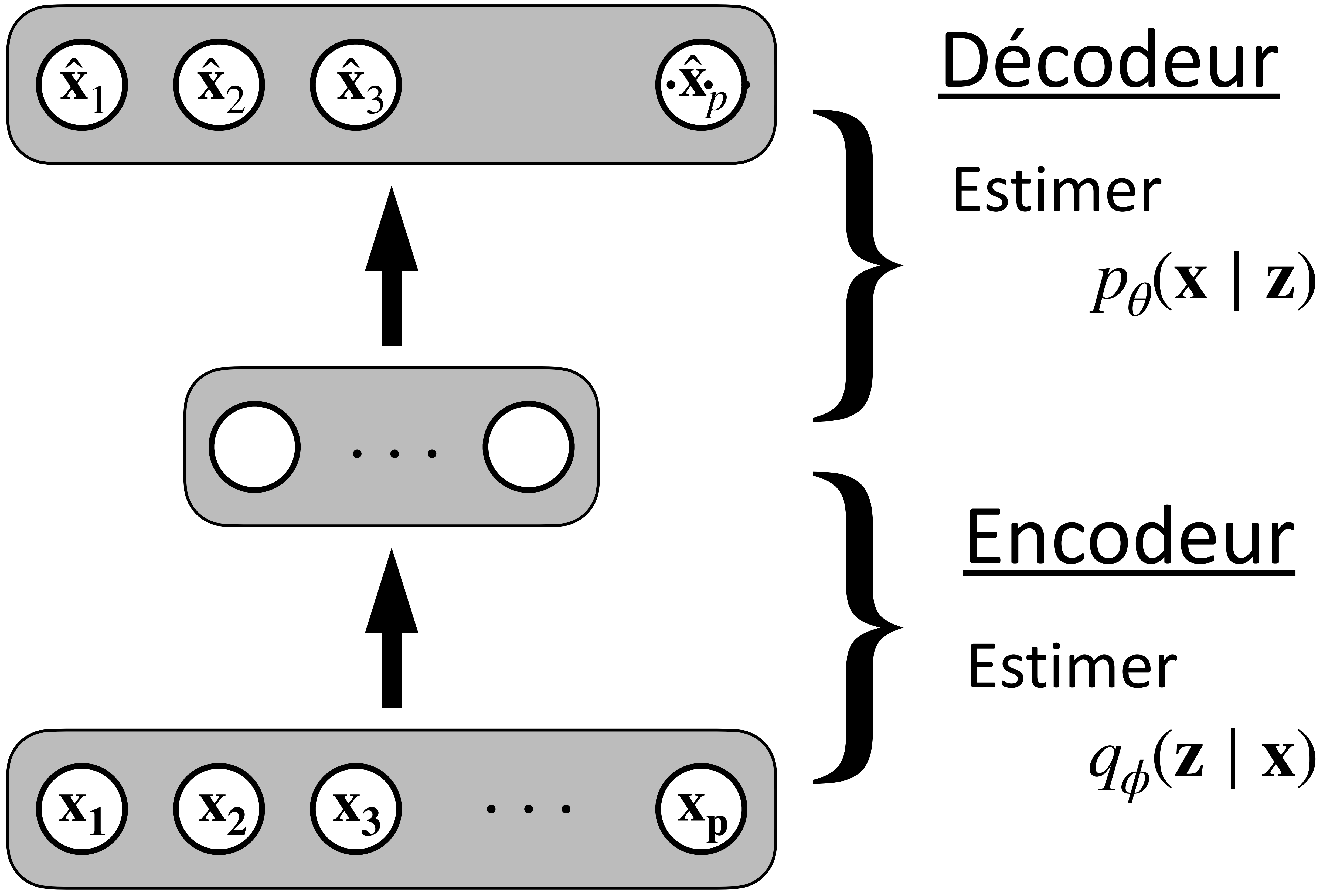




Variational Auto-encoders

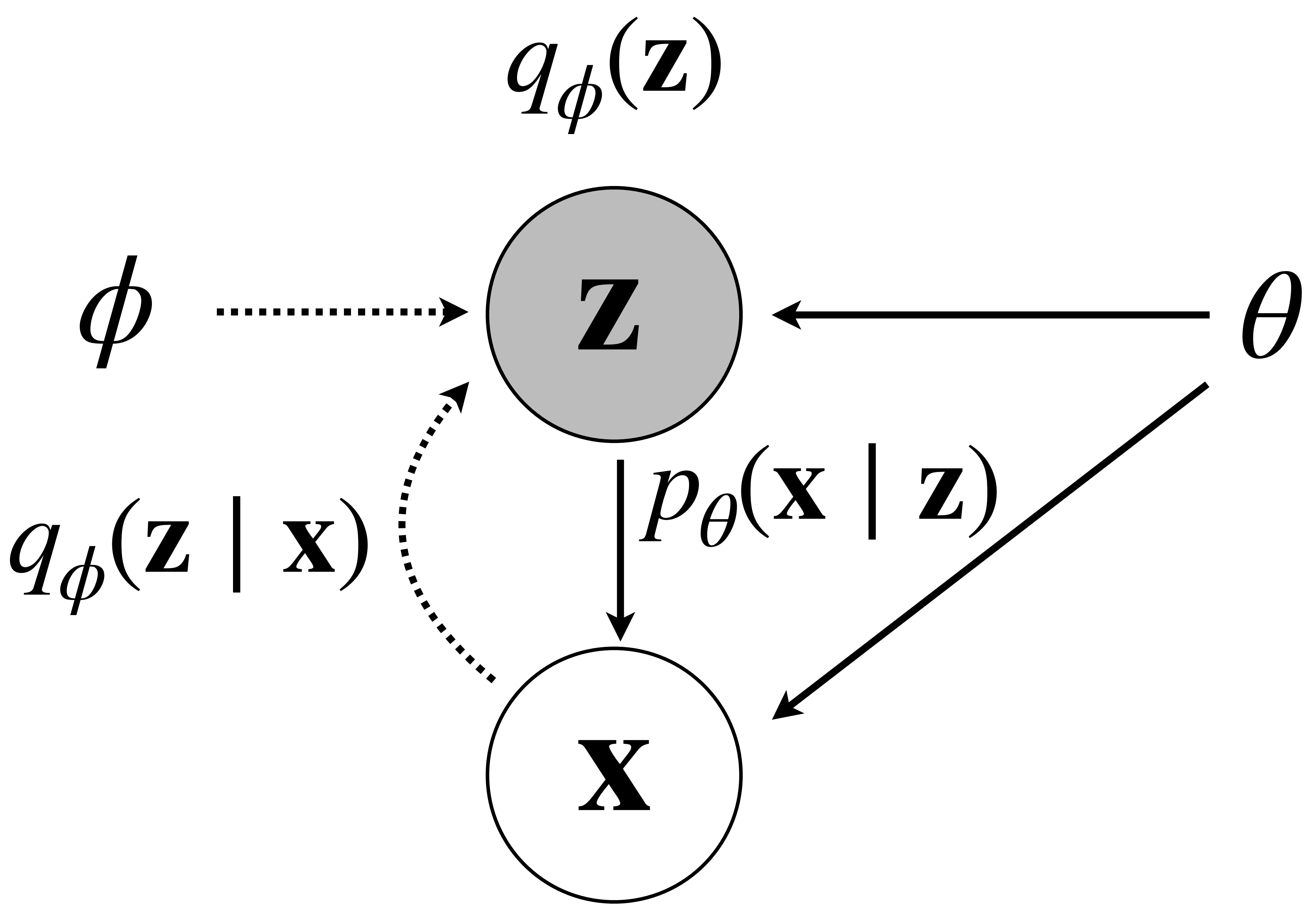
We can estimate these parameters using an auto encoder:

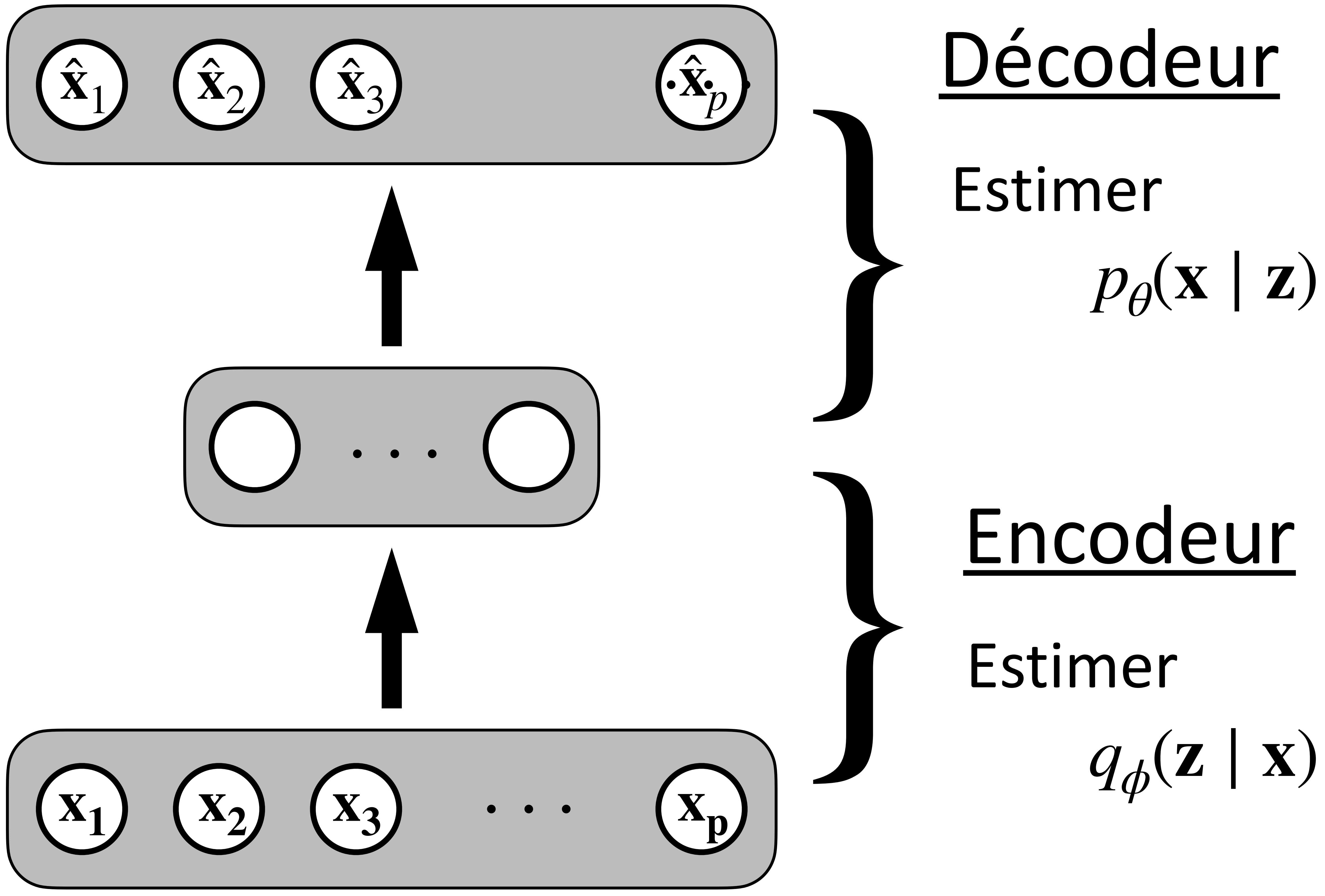




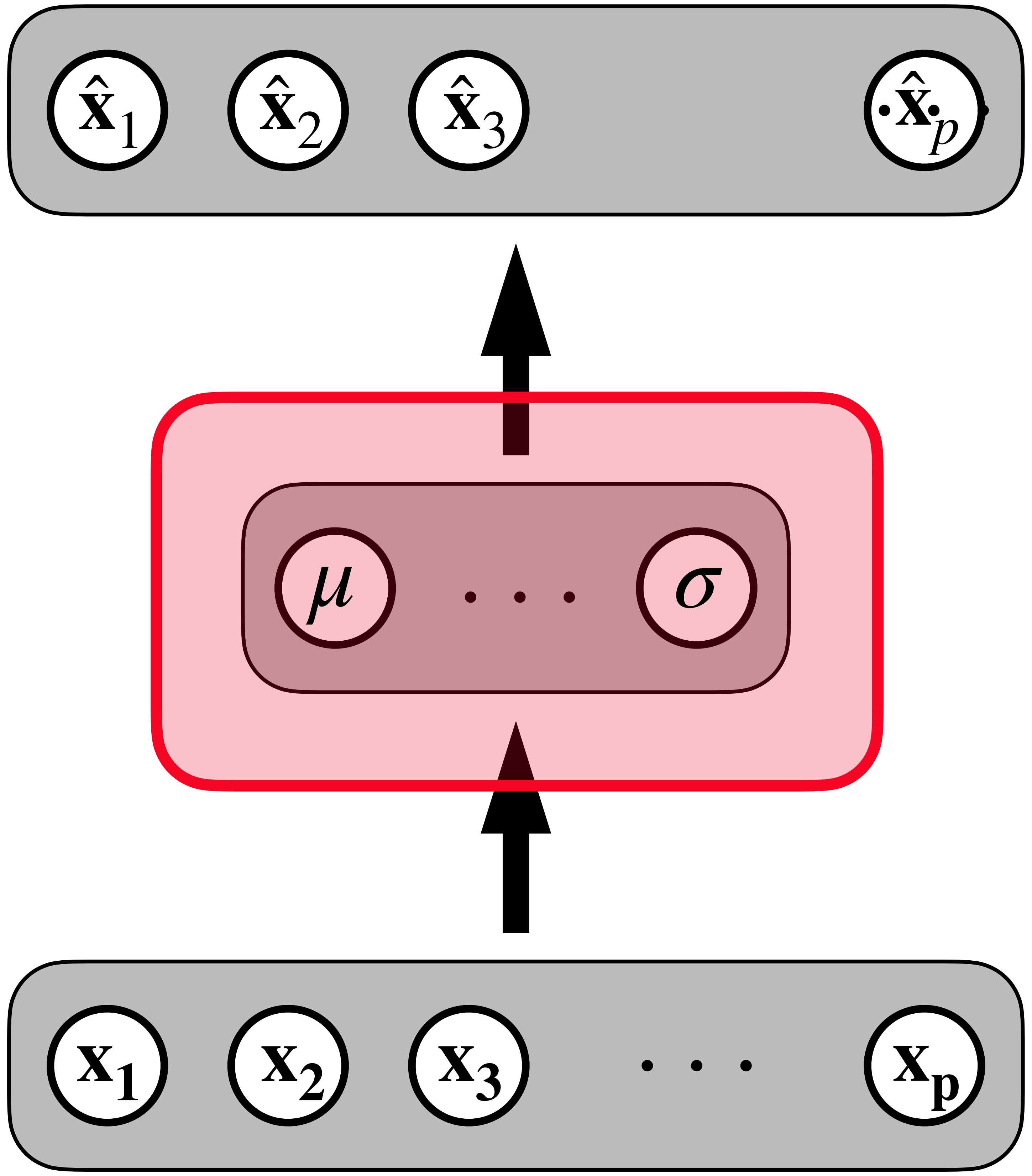
Variational Auto-encoders

We can estimate these parameters using an auto encoder:





Variational Auto-encoders



Decode

Estimate

$p_\theta(\mathbf{x} \mid \mathbf{z})$

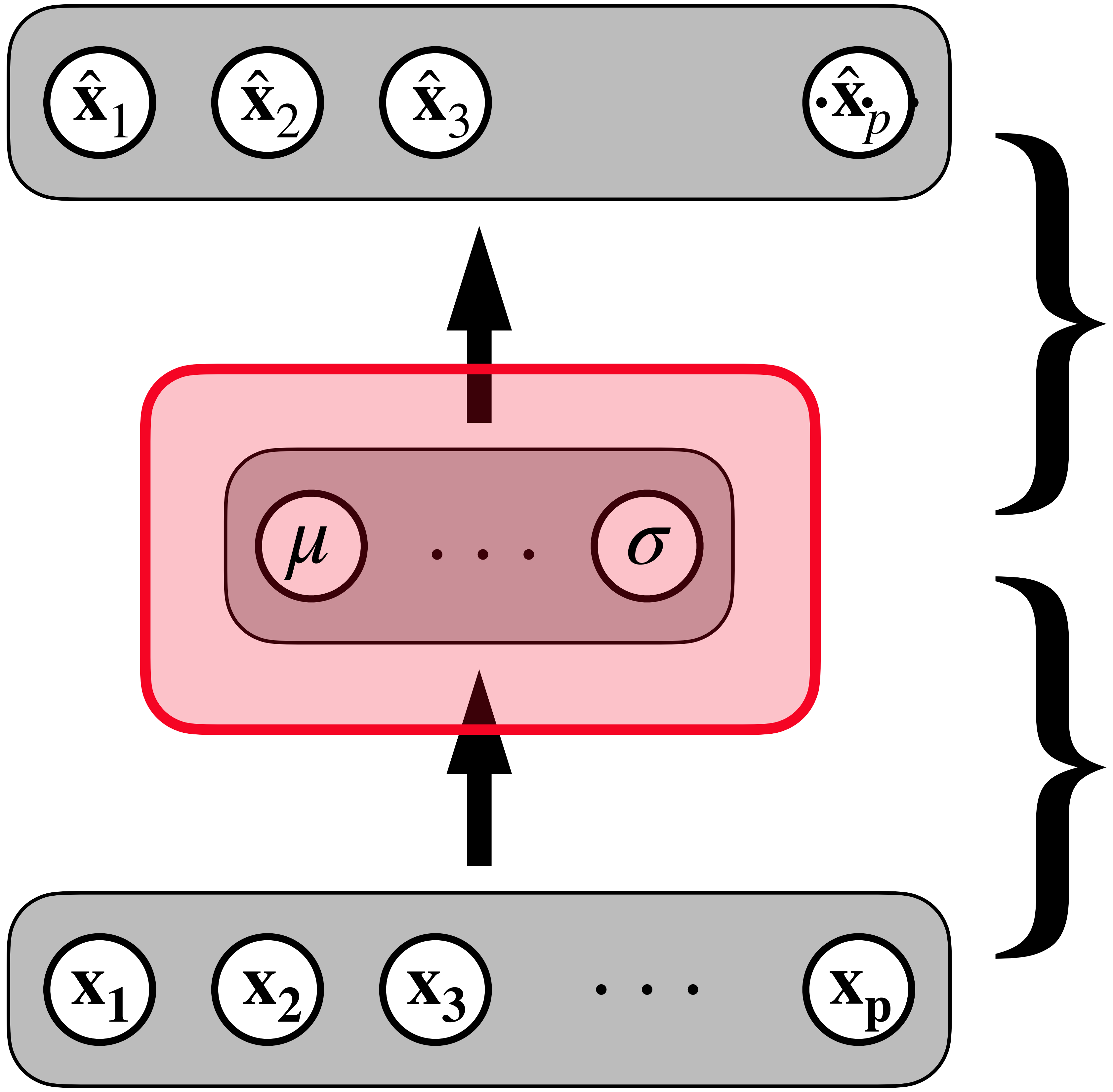
Encoder

Estimate

$q_\phi(\mathbf{z} \mid \mathbf{x})$

Variational Auto-encoders

Reparametrize the hidden layer:



<u>Decode</u>

Estimate

$p_\theta(\mathbf{x} \mid \mathbf{z})$

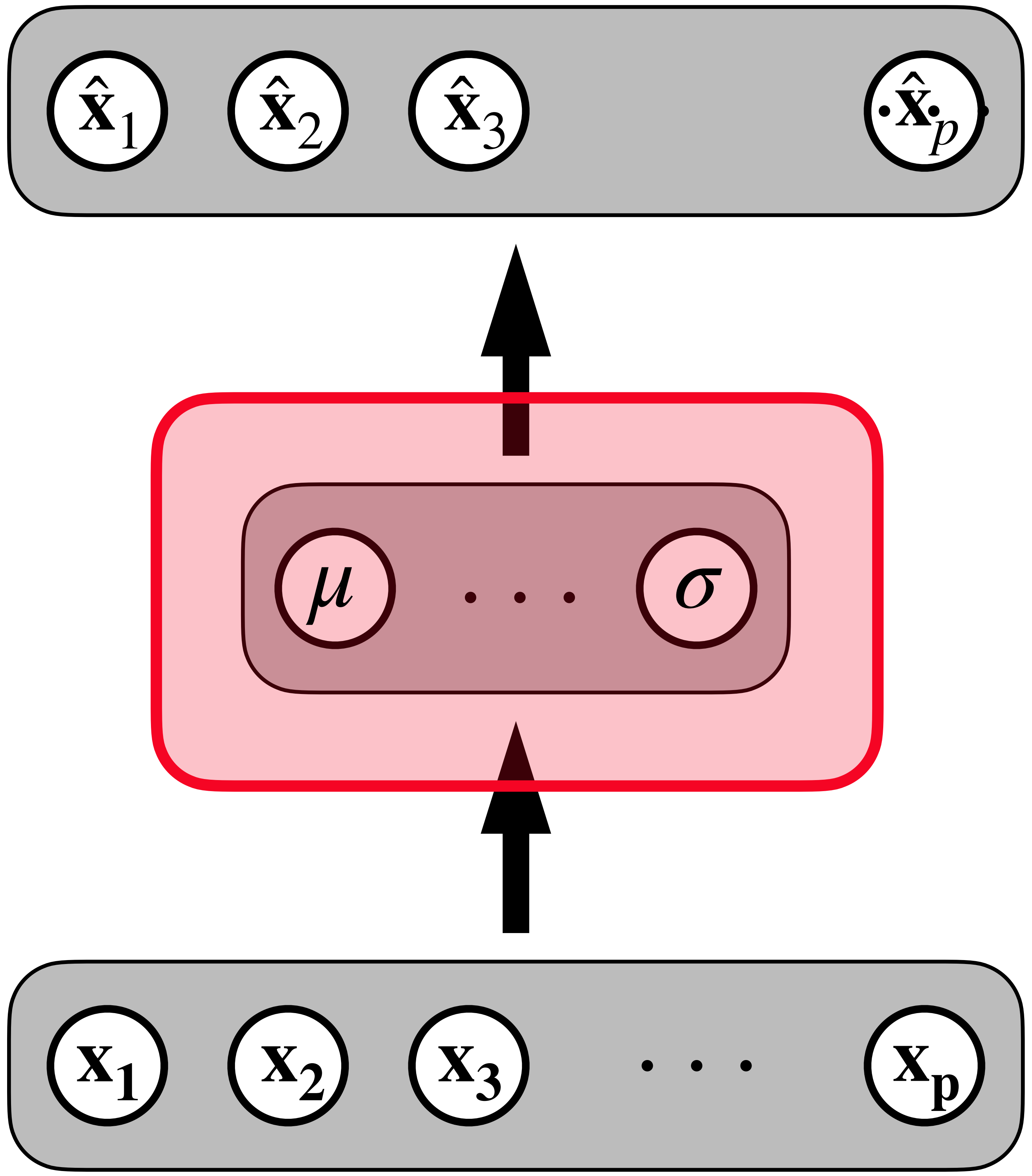
<u>Encoder</u>

Estimate

$q_\phi(\mathbf{z} \mid \mathbf{x})$

Variational Auto-encoders

Reparametrize the hidden layer:



<u>Decode</u>

Estimate

$p_\theta(\mathbf{x} \mid \mathbf{z})$

<u>Encoder</u>

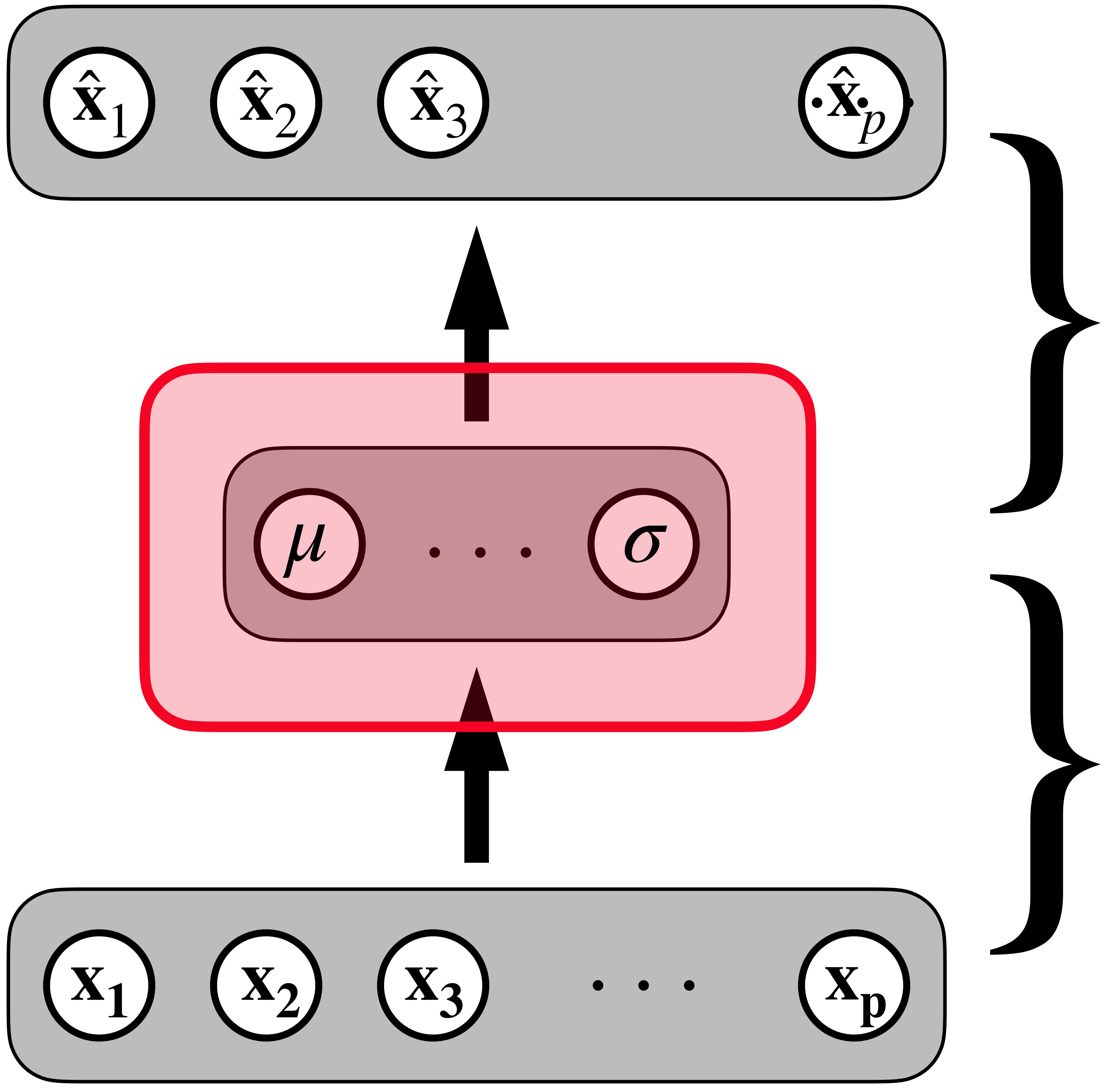
Estimate

$q_\phi(\mathbf{z} \mid \mathbf{x})$

Suppose that $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$.

Variational Auto-encoders

Reparametrize the hidden layer:



Decode

Estimate

$p_\theta(\mathbf{x} \mid \mathbf{z})$

Encoder

Estimate

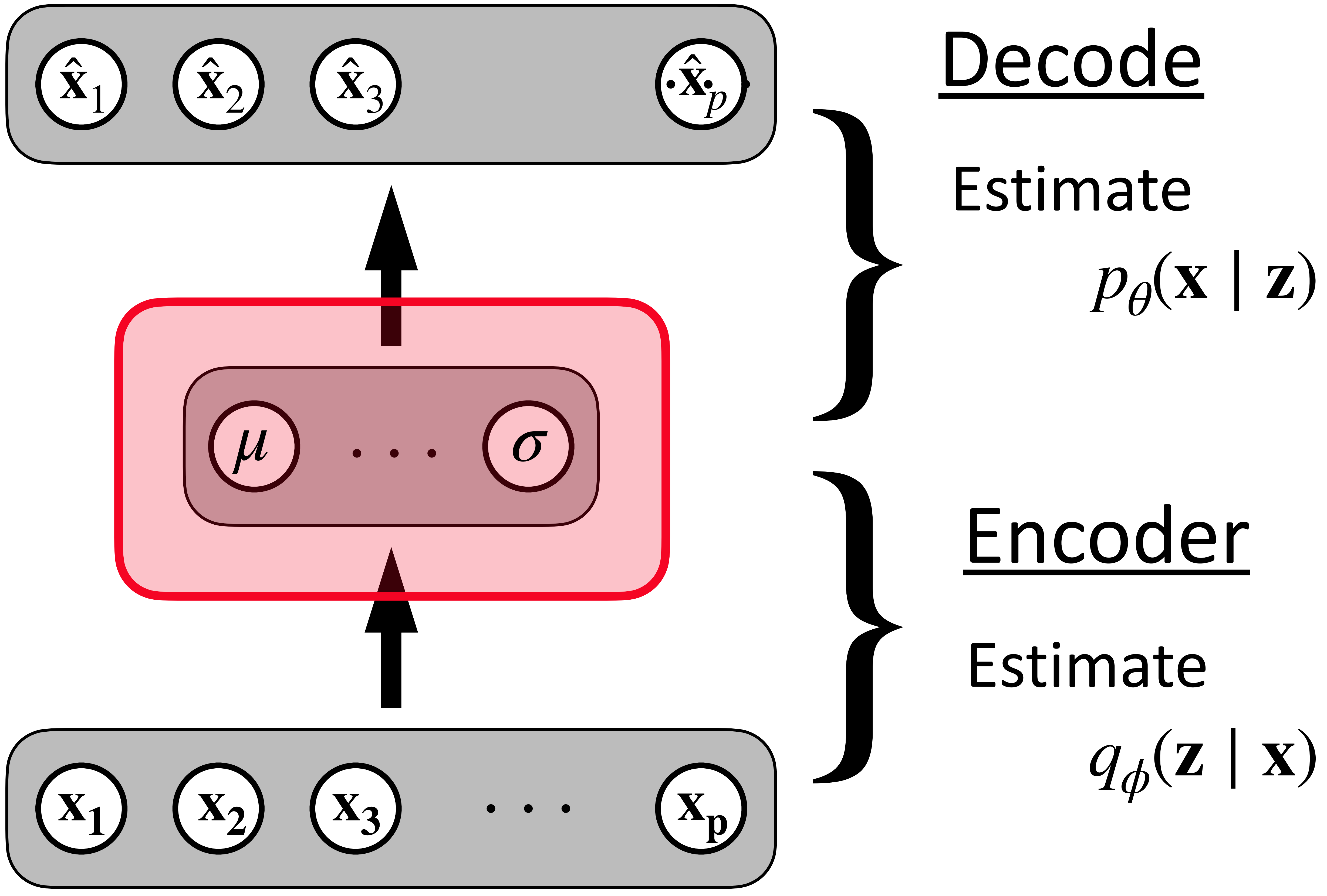
$q_\phi(\mathbf{z} \mid \mathbf{x})$

Suppose that $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$.

Then we can write $z = \mu + \sigma\epsilon$ où $\epsilon \sim \mathcal{N}(0,1)$.

Variational Auto-encoders

Reparametrize the hidden layer:



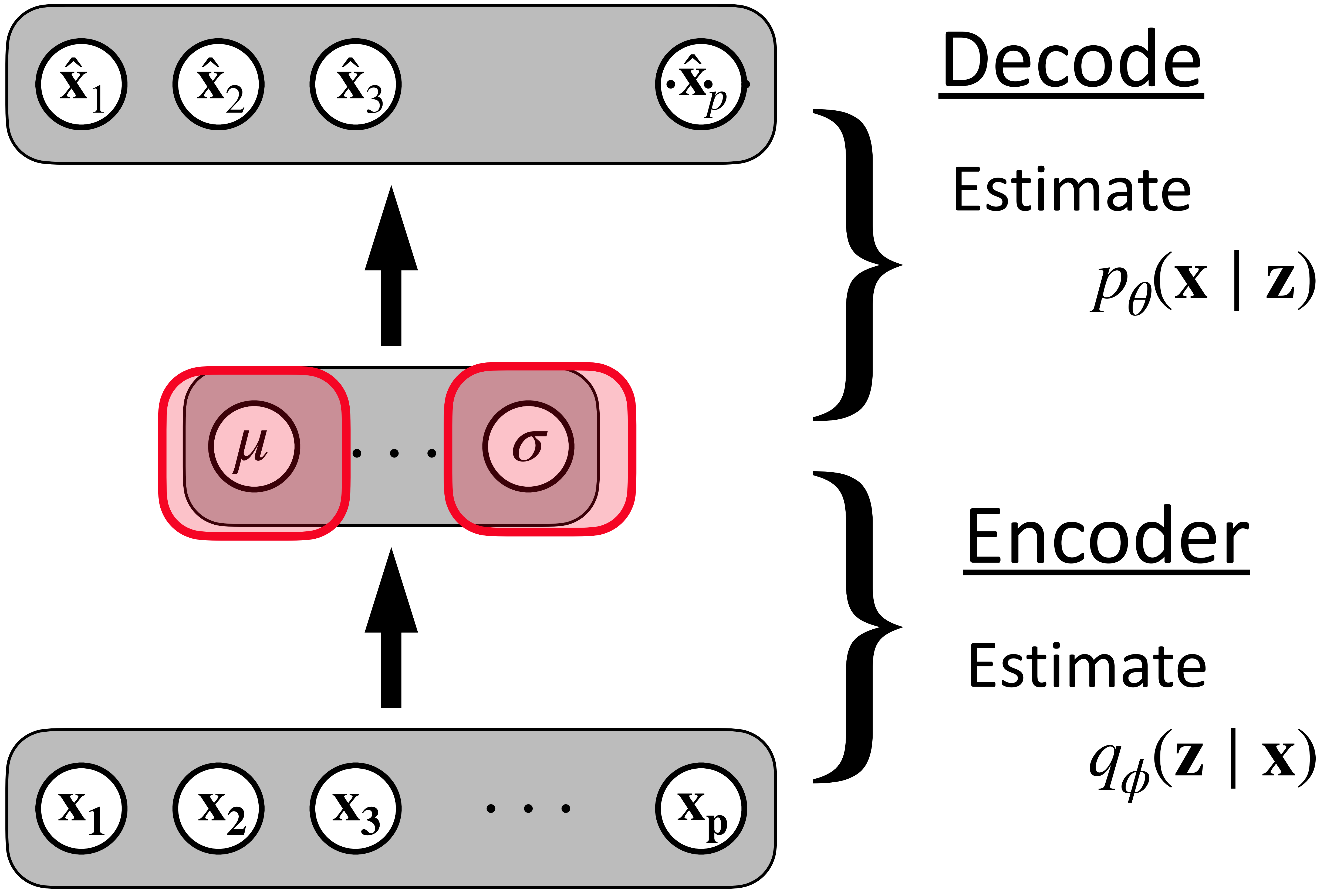
Suppose that $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$.

Then we can write $z = \mu + \sigma\epsilon$ où $\epsilon \sim \mathcal{N}(0,1)$.

Now, learning means estimating parameters μ et σ!

Variational Auto-encoders

Reparametrize the hidden layer:

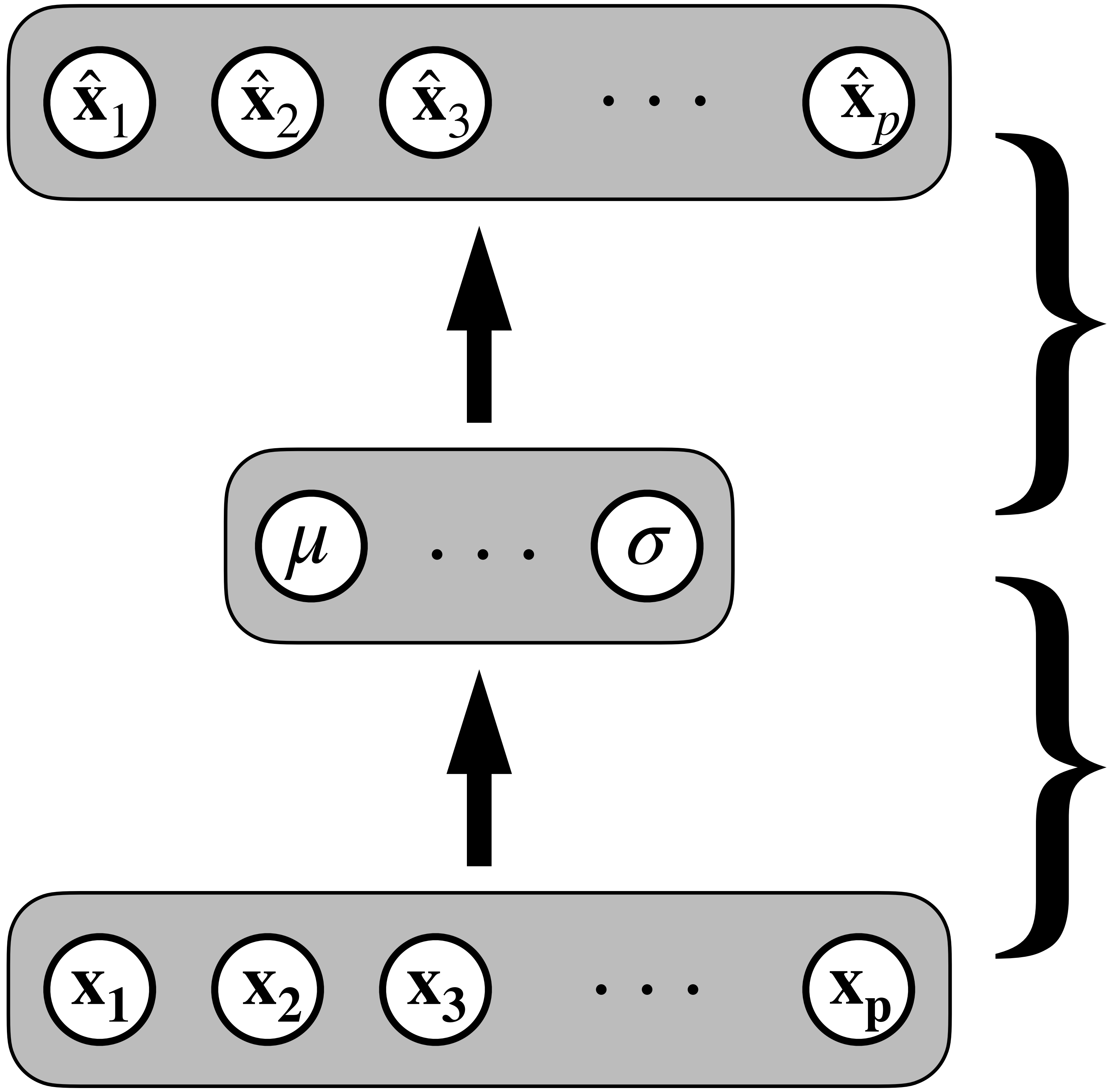


Suppose that $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$.

Then we can write $z = \mu + \sigma\epsilon$ où $\epsilon \sim \mathcal{N}(0,1)$.

Now, learning means estimating parameters μ et σ!

Variational Auto-encoders



Decoder

Estimate

$p_\theta(\mathbf{x} \mid \mathbf{z})$

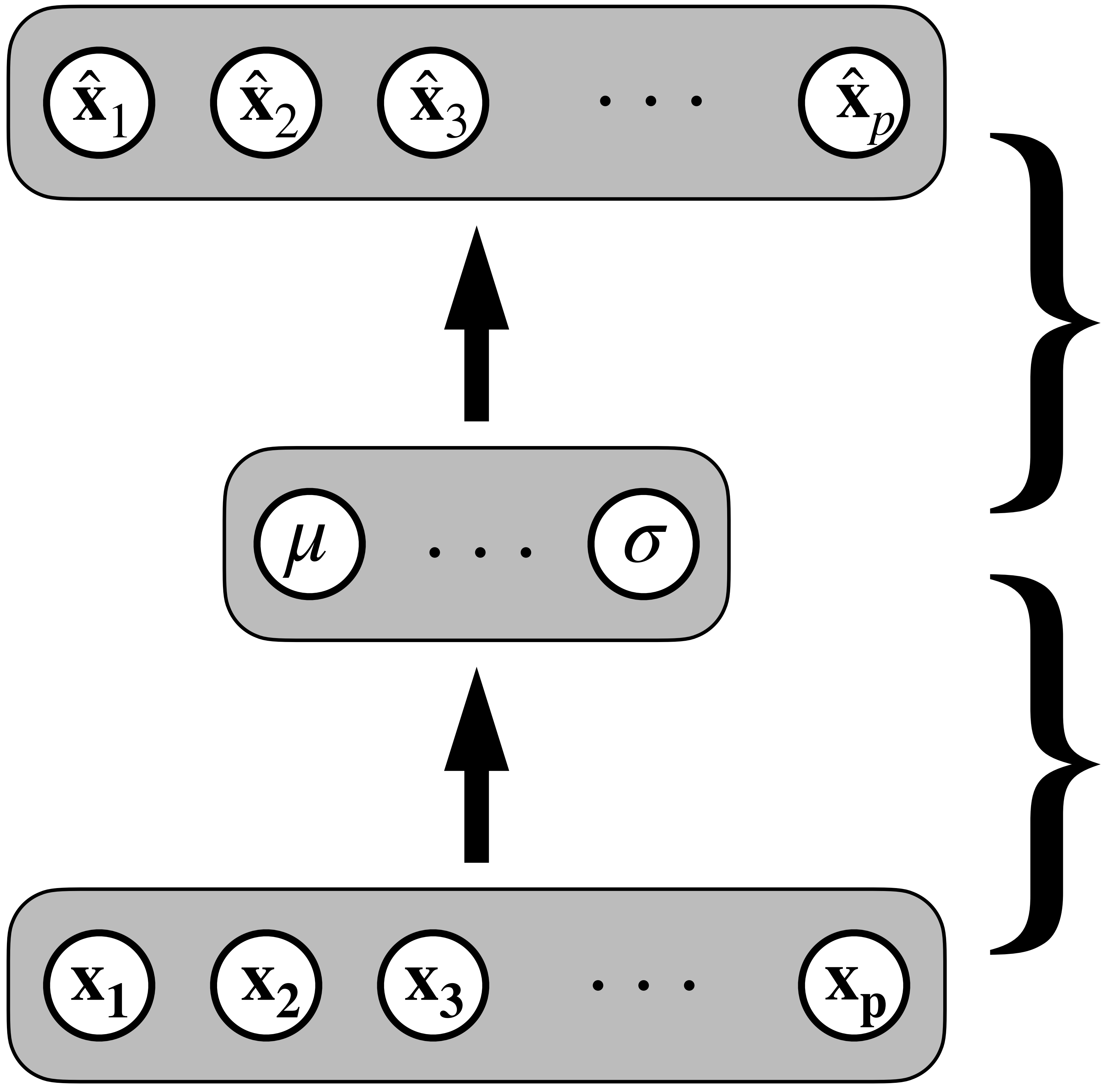
Encoder

Estimate

$q_\phi(\mathbf{z} \mid \mathbf{x})$

Variational Auto-encoders

VAEs are a way to train a generative model



Decoder

Estimate

$p_\theta(\mathbf{x} \mid \mathbf{z})$

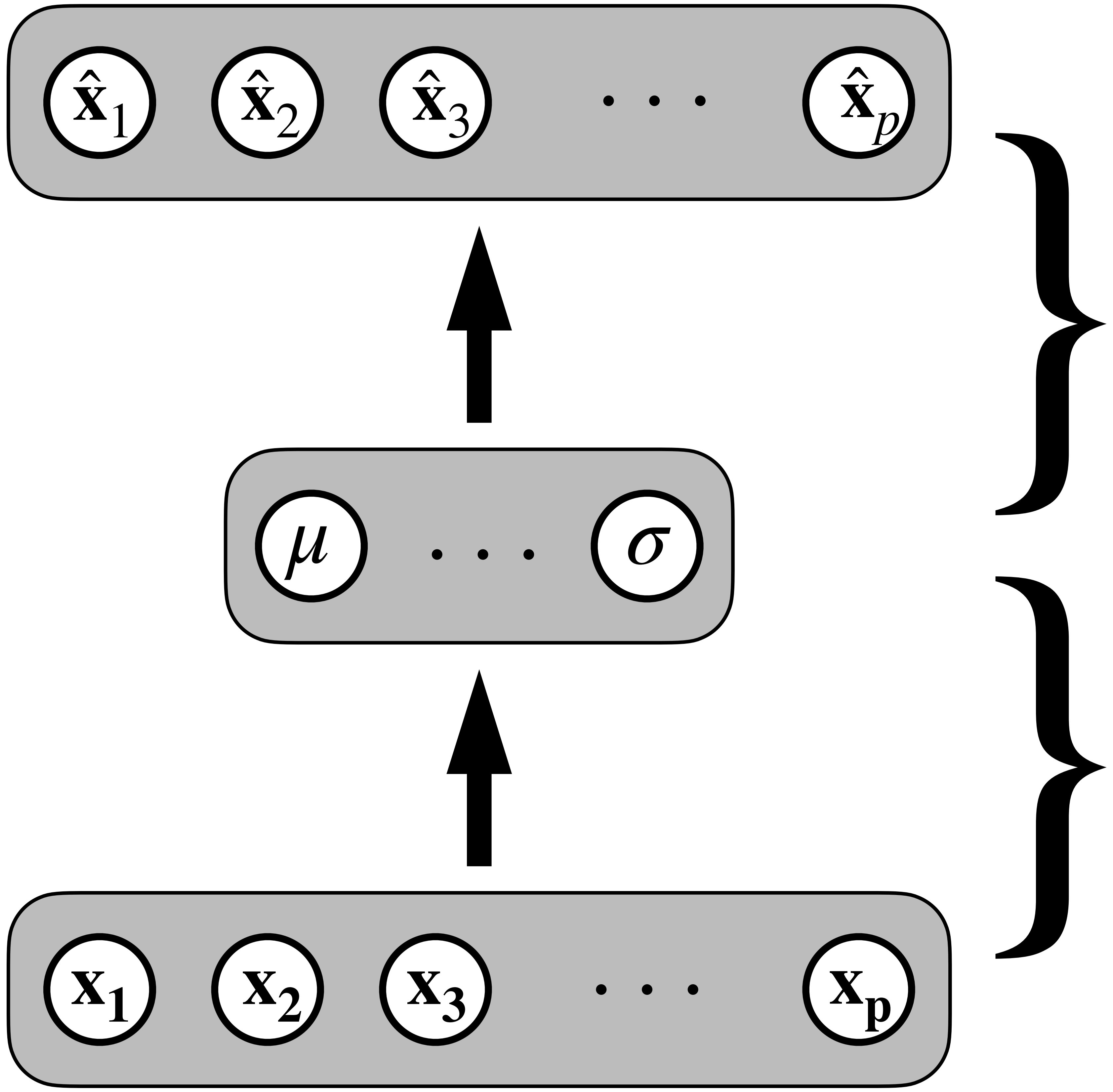
Encoder

Estimate

$q_\phi(\mathbf{z} \mid \mathbf{x})$

Variational Auto-encoders

VAEs are a way to train a generative model



Decoder

Estimate

$p_\theta(\mathbf{x} \mid \mathbf{z})$

Encoder

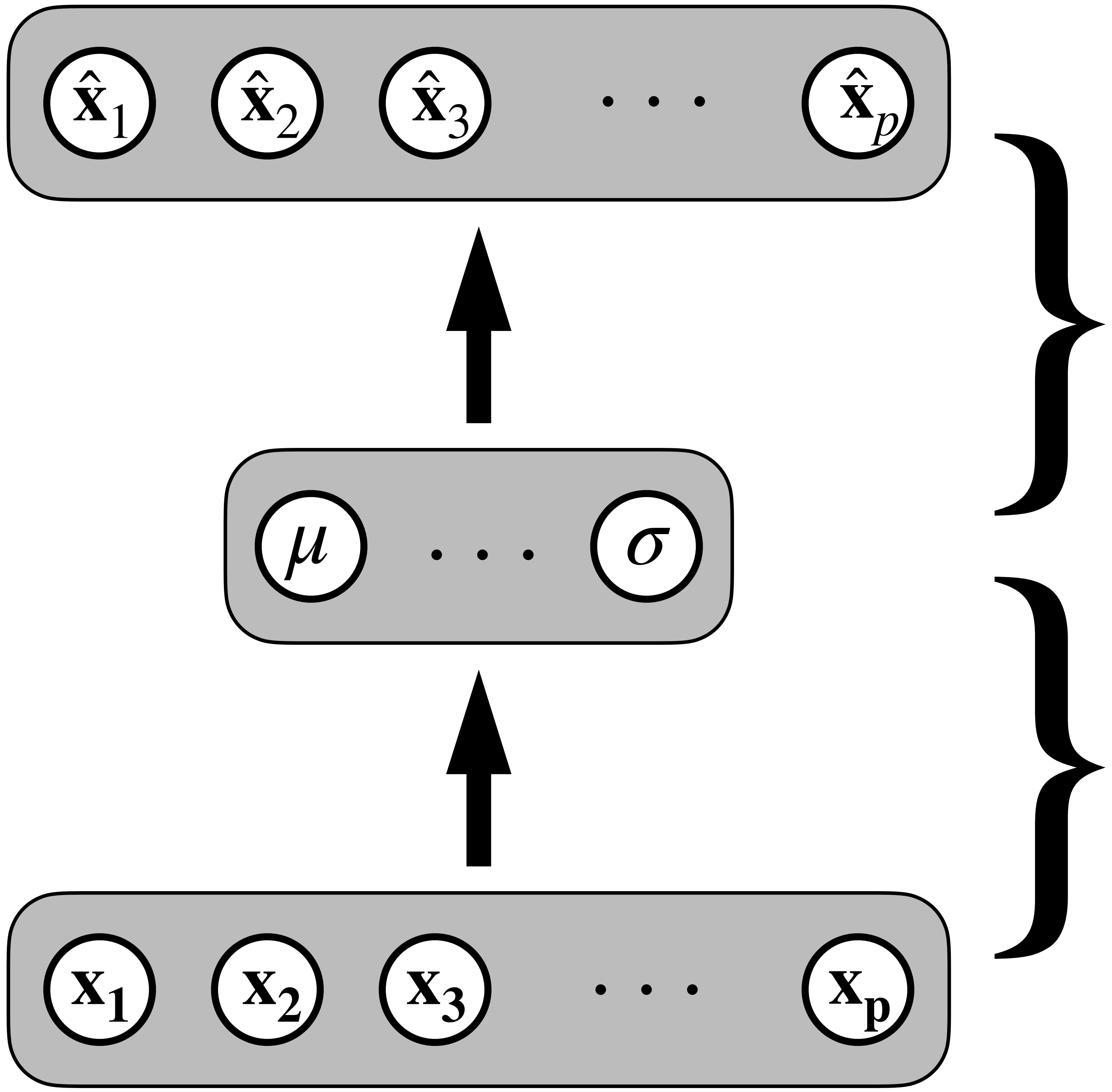
Estimate

$q_\phi(\mathbf{z} \mid \mathbf{x})$

Steps:

Variational Auto-encoders

VAEs are a way to train a generative model



Decoder

Estimate $p_\theta(\mathbf{x} \mid \mathbf{z})$

Encoder

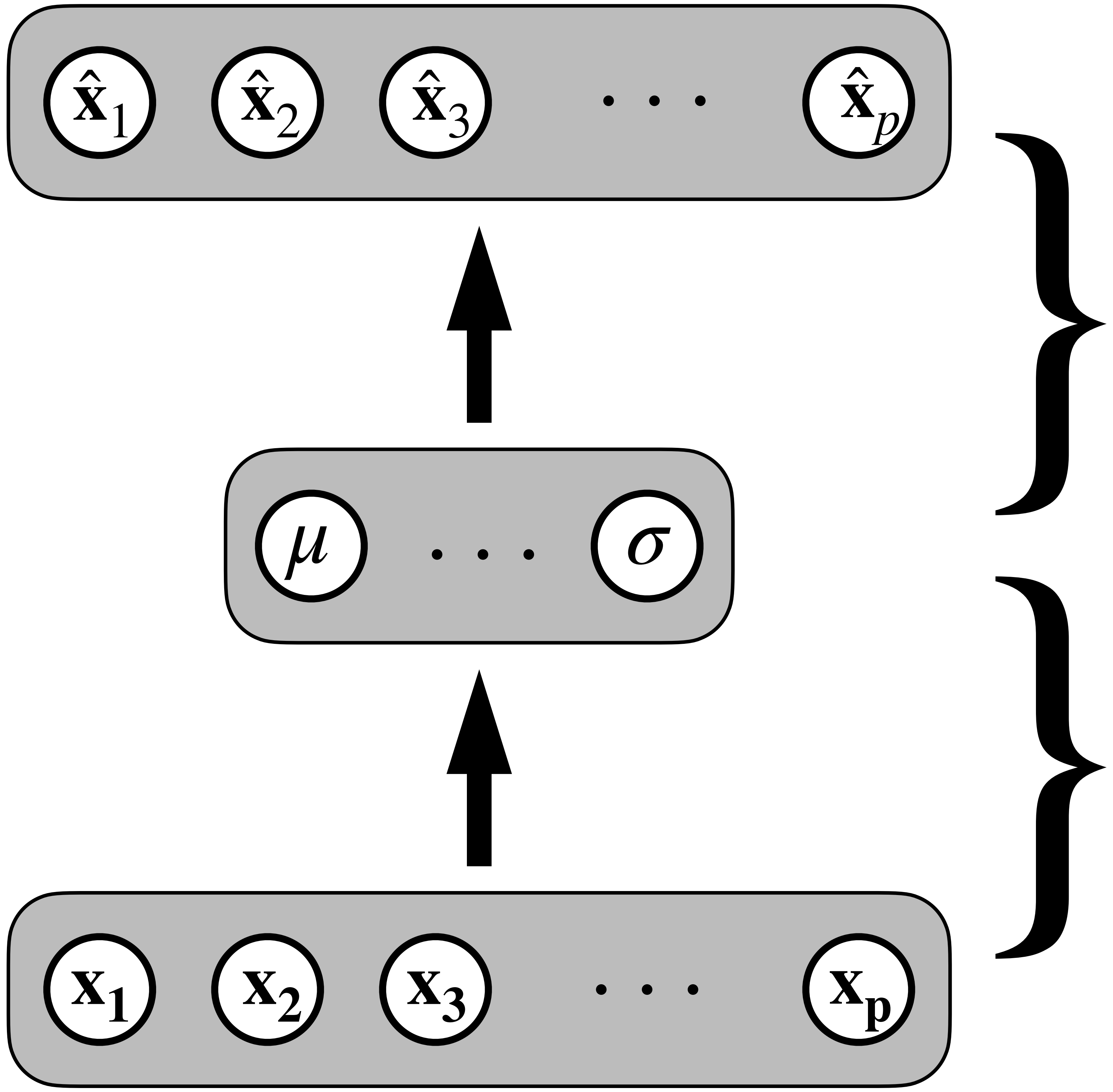
Estimate $q_\phi(\mathbf{z} \mid \mathbf{x})$

Steps:

1. Train the model to learn:

Variational Auto-encoders

VAEs are a way to train a generative model



Decoder

Estimate

$p_\theta(\mathbf{x} \mid \mathbf{z})$

Encoder

Estimate

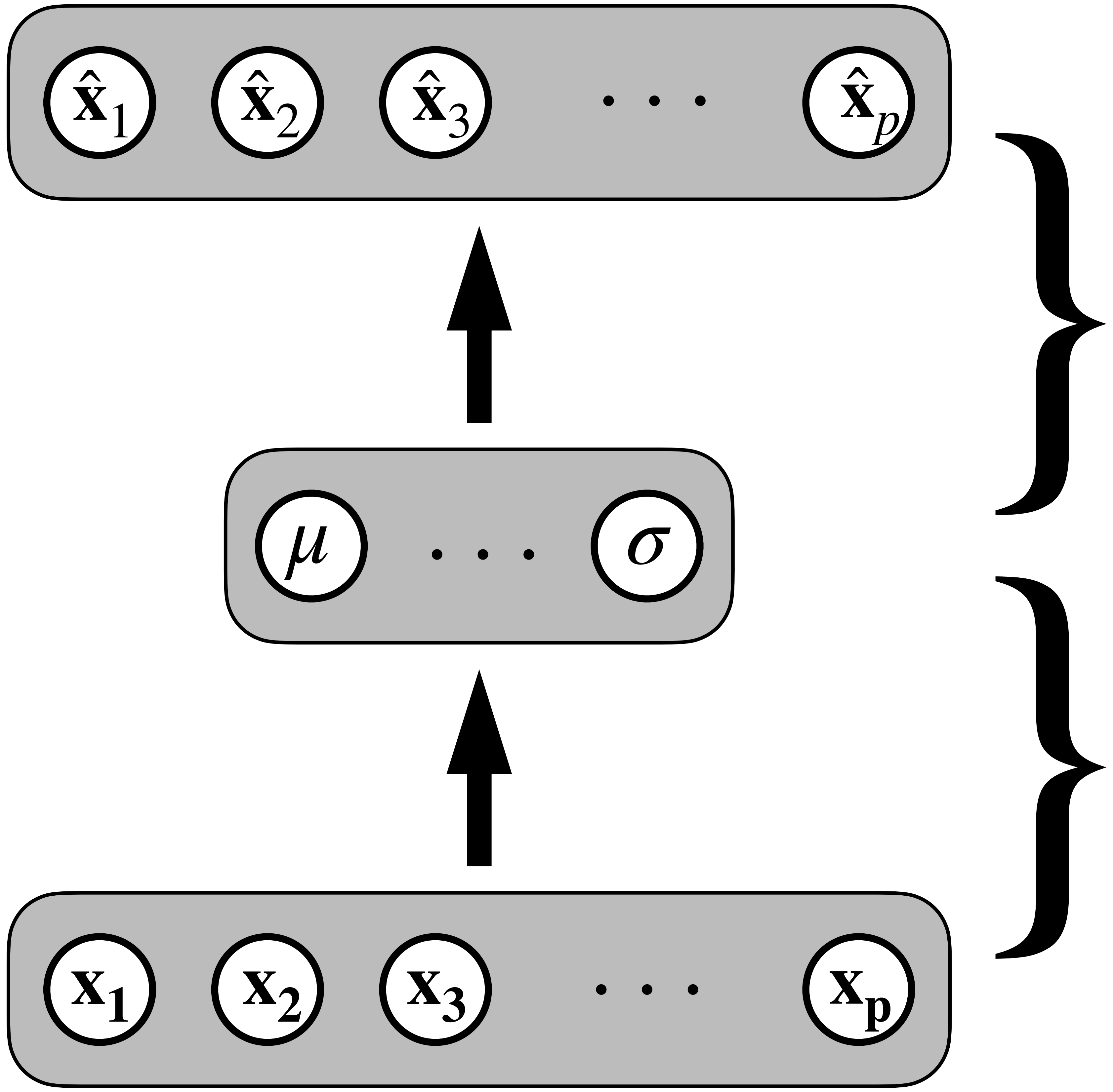
$q_\phi(\mathbf{z} \mid \mathbf{x})$

Steps:

1. Train the model to learn:
   - $p_\theta(\mathbf{x} \mid \mathbf{z})$

Variational Auto-encoders

VAEs are a way to train a generative model



Decoder

Estimate

$p_\theta(\mathbf{x} \mid \mathbf{z})$

Encoder

Estimate

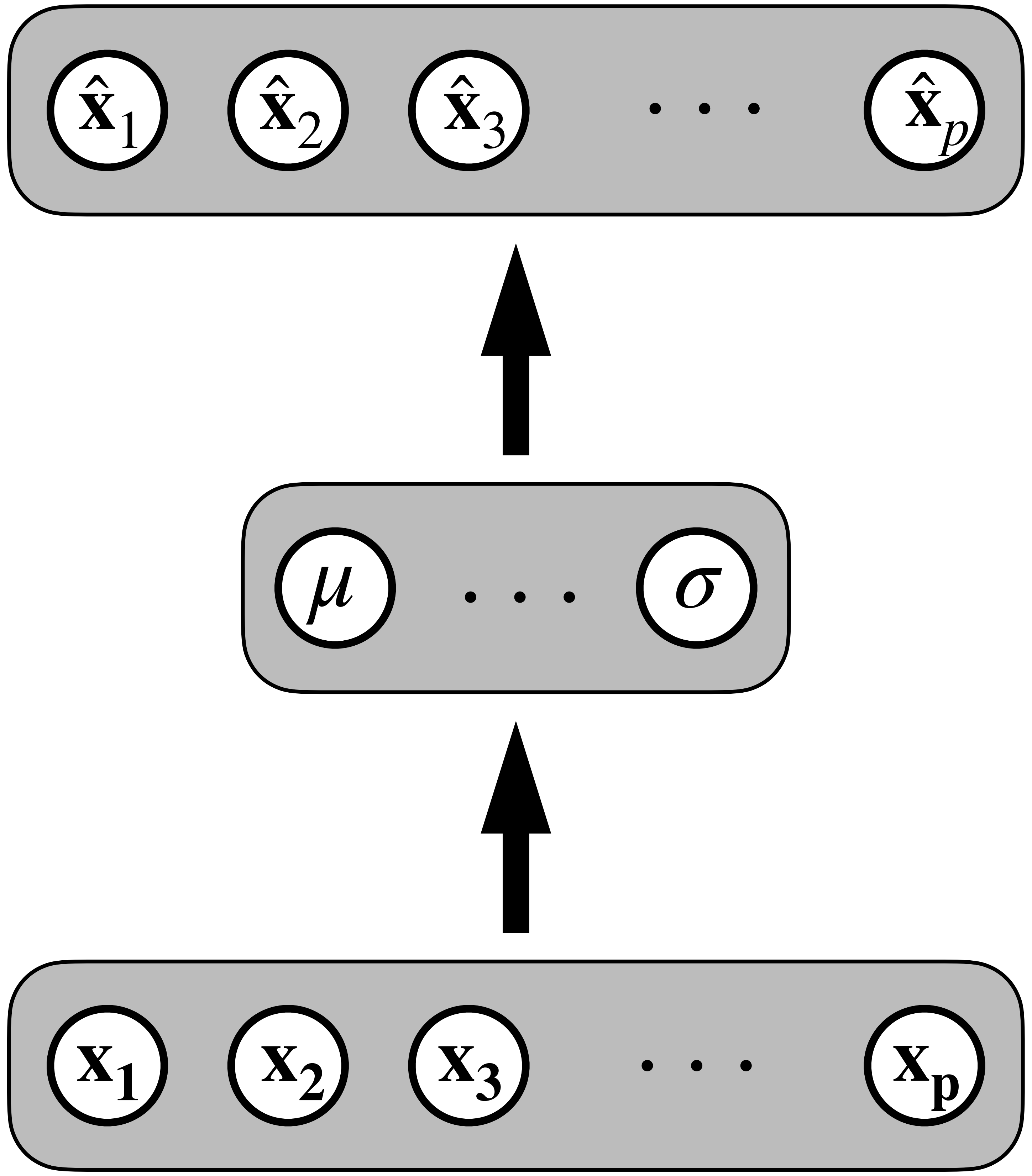
$q_\phi(\mathbf{z} \mid \mathbf{x})$

Steps:

1. Train the model to learn:
   - $p_\theta(\mathbf{x} \mid \mathbf{z})$
   - $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$

Variational Auto-encoders

VAEs are a way to train a generative model



Decoder

Estimate

$p_\theta(\mathbf{x} \mid \mathbf{z})$

Encoder

Estimate

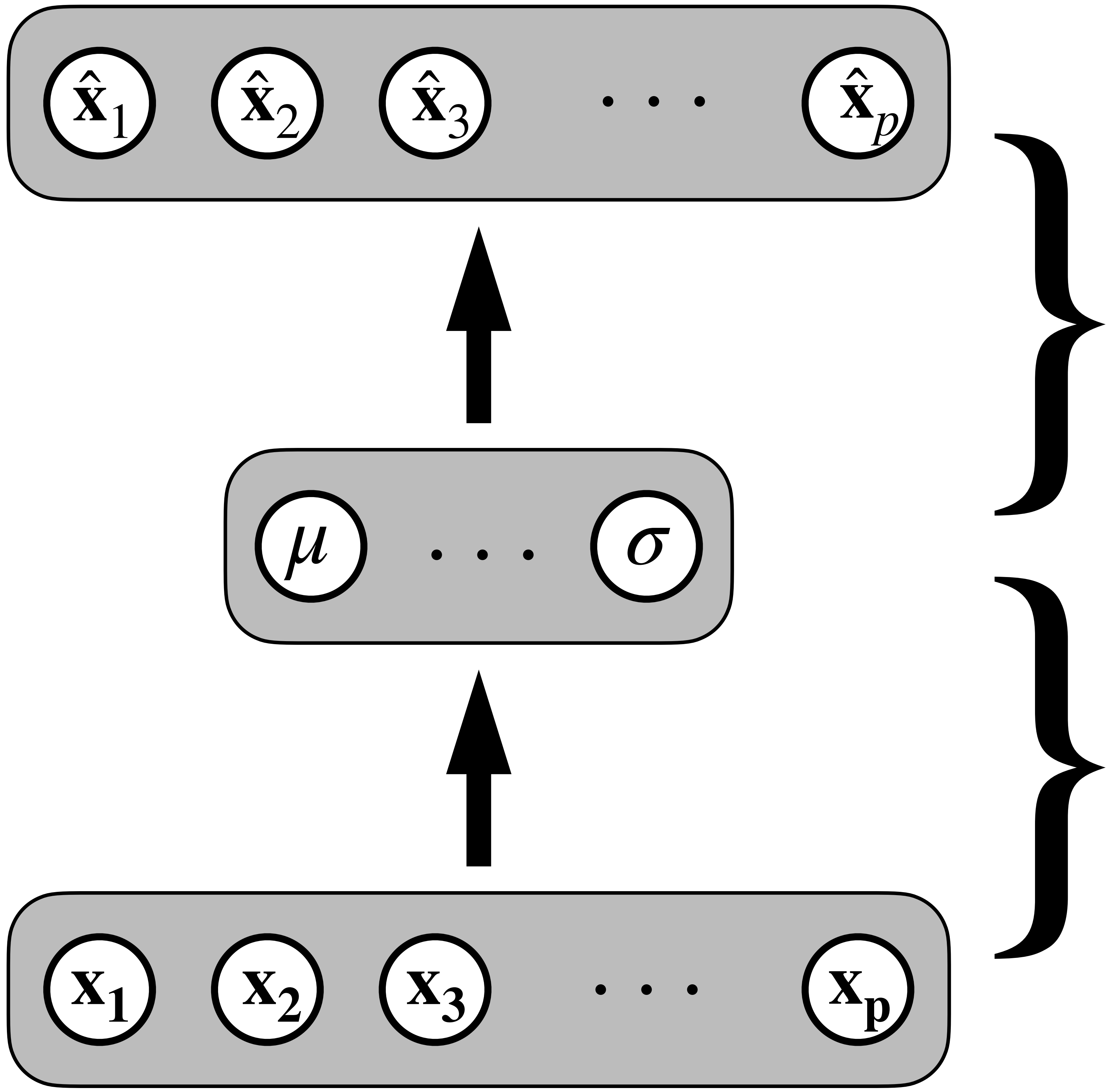
$q_\phi(\mathbf{z} \mid \mathbf{x})$

Steps:

1. Train the model to learn:
   - $p_\theta(\mathbf{x} \mid \mathbf{z})$
   - $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$
2. Generate ϵ et obtain $z \sim P(z)$.

Variational Auto-encoders

VAEs are a way to train a generative model



<u>Decoder</u>

Estimate $p_\theta(\mathbf{x} \mid \mathbf{z})$

<u>Encoder</u>

Estimate $q_\phi(\mathbf{z} \mid \mathbf{x})$

<u>Steps:</u>

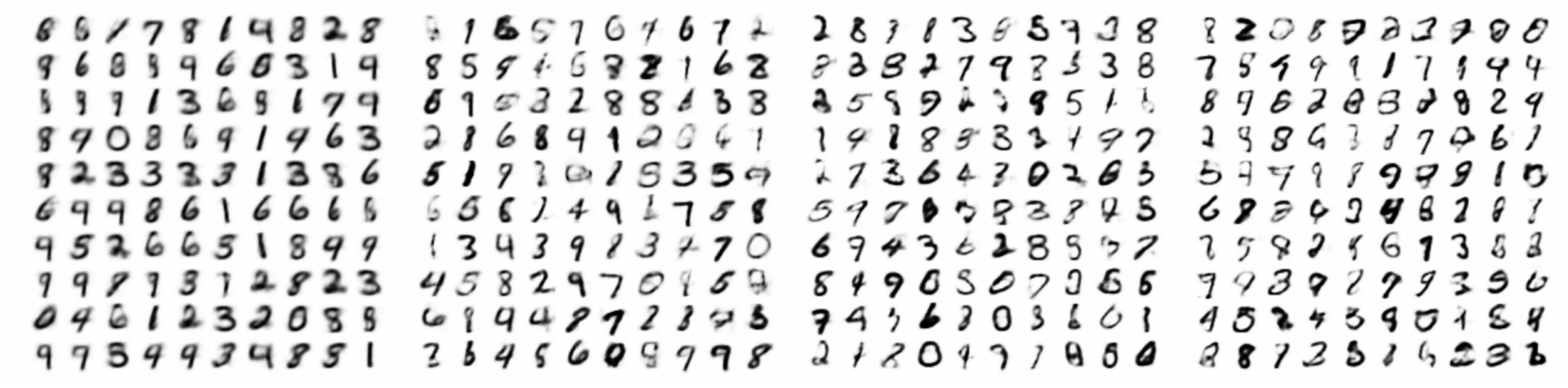
1. Train the model to learn:
   - $p_\theta(\mathbf{x} \mid \mathbf{z})$
   - $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^{\mathbf{2}})$
2. Generate ϵ et obtain $z \sim P(z)$.
3. Obtain $\mathbf{x}$ from $p_\theta(\mathbf{x} \mid \mathbf{z})$.

Variational Auto-encoders

Example: Generating numbers (range 0—9)

Variational Auto-encoders

Example: Generating numbers (range 0—9)



Multivariate Normal ($\mathbb{R}^2$)

Multivariate Normal ($\mathbb{R}^5$)

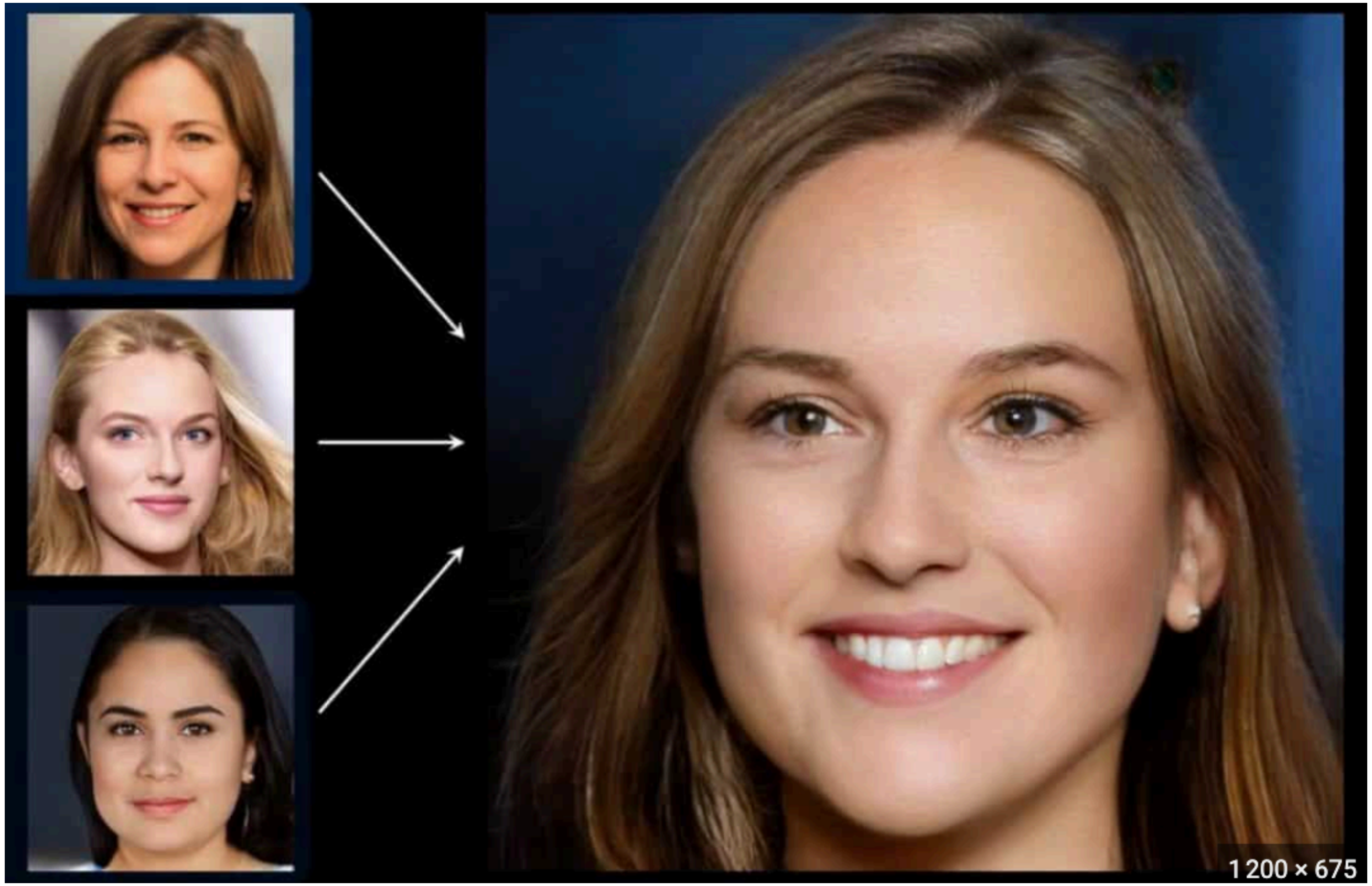
Multivariate Normal ($\mathbb{R}^{10}$)

Multivariate Normal ($\mathbb{R}^{20}$)

Generative Adversarial Networks (GANs)

GANs - Introduction

Well-known for image generation



Historical note

- Framework for learning a generative model (for example, an “inverted” CNN)
- Developed by researchers at Université de Montréal (2014)
- The first to generate high-quality complex images
- Have been (mostly?) replaced by diffusion models
  - The study of GANs has provided insights into 2-player (min-max) optimization

GANs — Intuition

GANs — Intuition

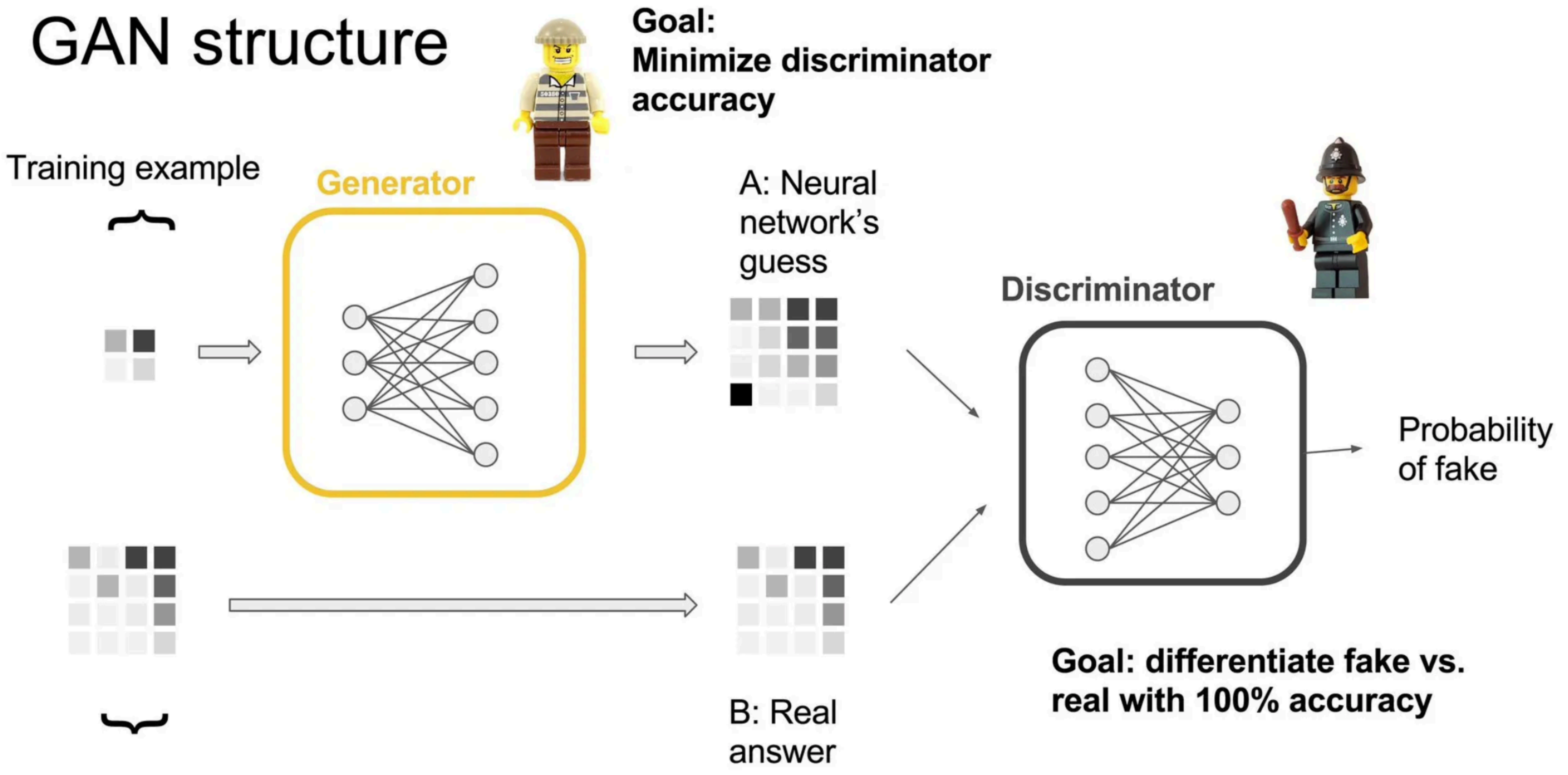
- Two players (each is a neural network)
  - Generator: The first player. It learns to generate a (good) image
  - Discriminator: The second player, learns to recognize (discriminate) good images from bad images

GANs — Intuition

- Two players (each is a neural network)
  - Generator: The first player. It learns to generate a (good) image
  - Discriminator: The second player, learns to recognize (discriminate) good images from bad images
- The game (at each round):
  - The second player receives an image. It must determine whether the image comes from the generator or from the training data
    - Depending on the response, the players update (their weights)

GANS — Introduction

Visually:



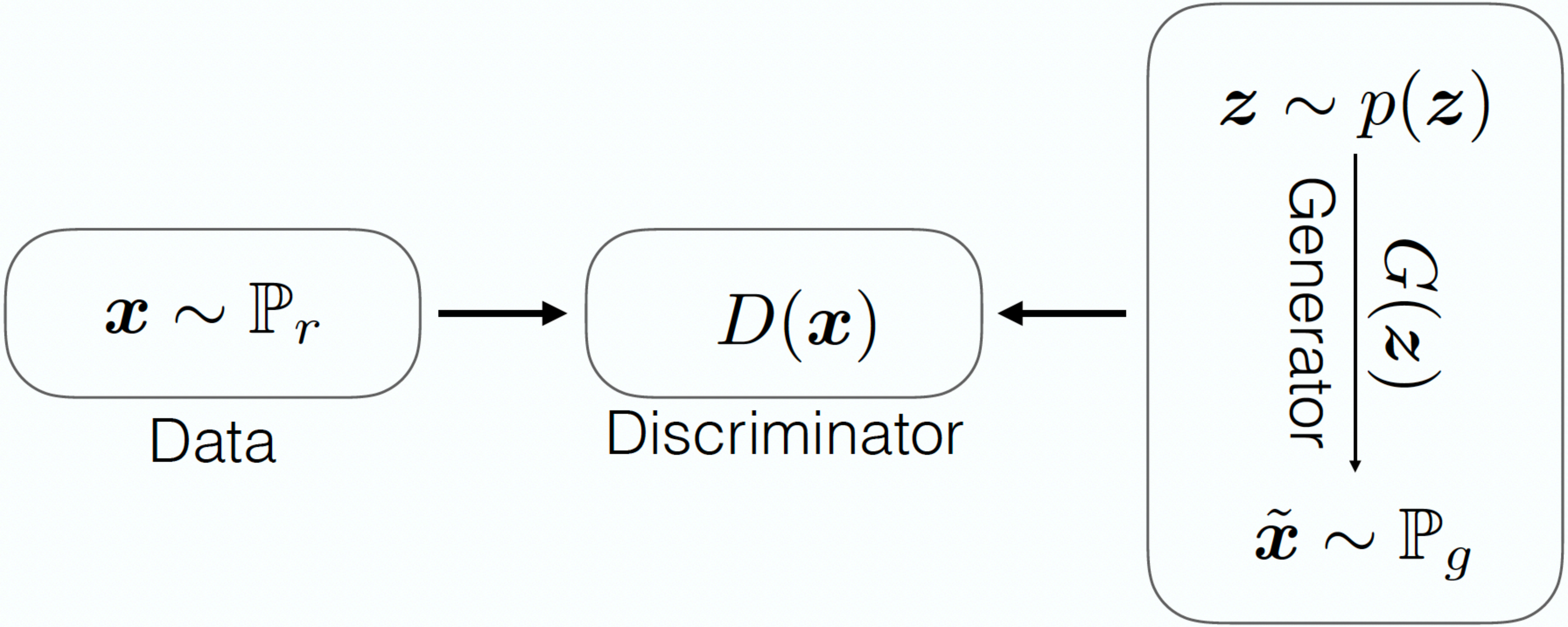
GANs - formalism

$$\boldsymbol{x} \sim \mathbb{P}_r$$

Data

GANs - formalism

GANs - formalism



GANs - objectives

GANs have two objectives (one for each player)

- The output of D is “1” for real and “0" for fake.

Objective of the discriminator

$$\max_{D} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} [\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} [\log(1 - D(\tilde{\boldsymbol{x}}))].$$

Objective of the generator

$$\max_{G} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} [\log(D(\tilde{\boldsymbol{x}}))].$$

GANs - objectives

GANs have two objectives (one for each player)

- The output of D is “1” for real and “0" for fake.

Objective of the discriminator

$$\max_{D} \underset{\boldsymbol{x}\sim\mathbb{P}_r}{\mathbb{E}}[\log(D(\boldsymbol{x}))] + \underset{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}{\mathbb{E}}[\log(1 - D(\tilde{\boldsymbol{x}}))].$$

Objective of the generator

$$\max_{G} \underset{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}{\mathbb{E}}[\log(D(\tilde{\boldsymbol{x}}))].$$

where:

$\mathbb{P}_r$ Is the distribution that “generates” the real data

GANs - objectives

GANs have two objectives (one for each player)

- The output of D is “1” for real and “0" for fake.

Objective of the discriminator

$$\max_{D} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(1 - D(\tilde{\boldsymbol{x}}))].$$

Objective of the generator

$$\max_{G} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(D(\tilde{\boldsymbol{x}}))].$$

where:

$\mathbb{P}_r$ Is the distribution that “generates” the real data

$\mathbb{P}_g$ Is the distribution generating data

$\tilde{\boldsymbol{x}} = G(\boldsymbol{z}), \quad \boldsymbol{z} \sim p(\boldsymbol{z})$ Where z follows a normal (e.g.)

GANs - formalism

The game is framed as a two players game between the discriminator and the generator

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r} [\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g} [\log(1 - D(\tilde{\boldsymbol{x}}))].$$

GANs - formalism

The game is framed as a two players game between the discriminator and the generator

$$\min_{G} \max_{D} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} [\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} [\log(1 - D(\tilde{\boldsymbol{x}}))].$$

where:

$\mathbb{P}_r$ Is the distribution that "generates" the real data

GANs - formalism

The game is framed as a two players game between the discriminator and the generator

$$\min_{G} \max_{D} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(1 - D(\tilde{\boldsymbol{x}}))].$$

where:

$\mathbb{P}_r$ Is the distribution that “generates” the real data

$\mathbb{P}_g$ Is the distribution generating data

$\tilde{\boldsymbol{x}} = G(\boldsymbol{z}), \quad \boldsymbol{z} \sim p(\boldsymbol{z})$ Where z follows a normal (e.g.)

Training GANs in practice alternates between training D and G

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**
  for k steps **do**
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

GANs - Varia


Monet ⟲ Photos
Zebras ⟲ Horses
Summer ⟲ Winter
Monet → photo
zebra → horse
summer → winter
photo → Monet
horse → zebra
winter → summer
Photograph
→
Monet
Van Gogh
Cezanne
Ukiyo-e

DALL-E 2
(As an example of using a diffusion model)

https://arxiv.org/pdf/2204.06125v1

DALL-E

an espresso machine that makes coffee from human souls, artstation

panda mad scientist mixing sparkling chemicals, artstation

a corgi’s head depicted as an explosion of a nebula

a dolphin in an astronaut suit on saturn, artstation

a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese

a teddy bear on a skateboard in times square

DALL-E - Overview

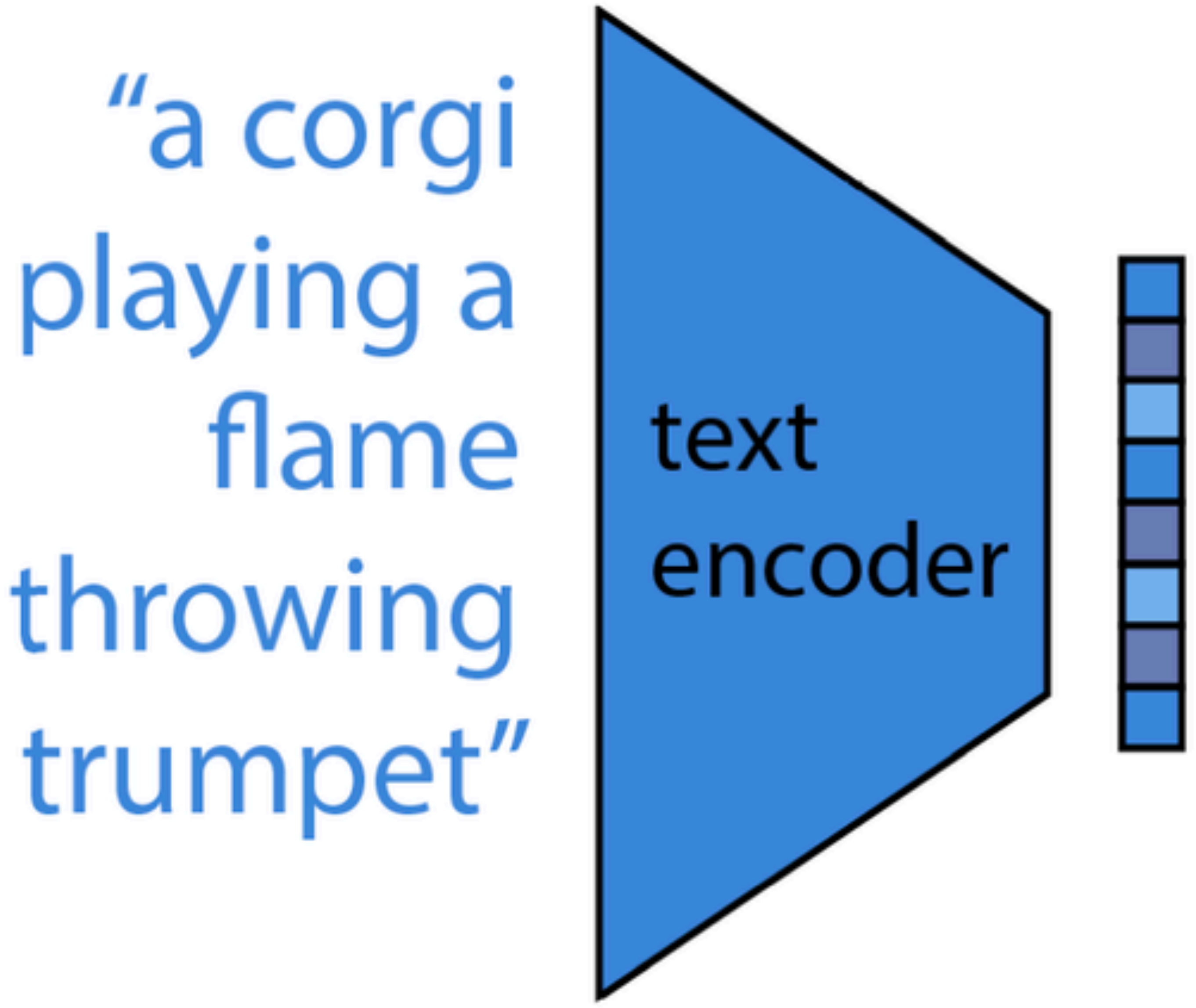
"a corgi
playing a
flame
throwing
trumpet"

DALL-E - Overview

“a corgi
playing a
flame
throwing
trumpet”

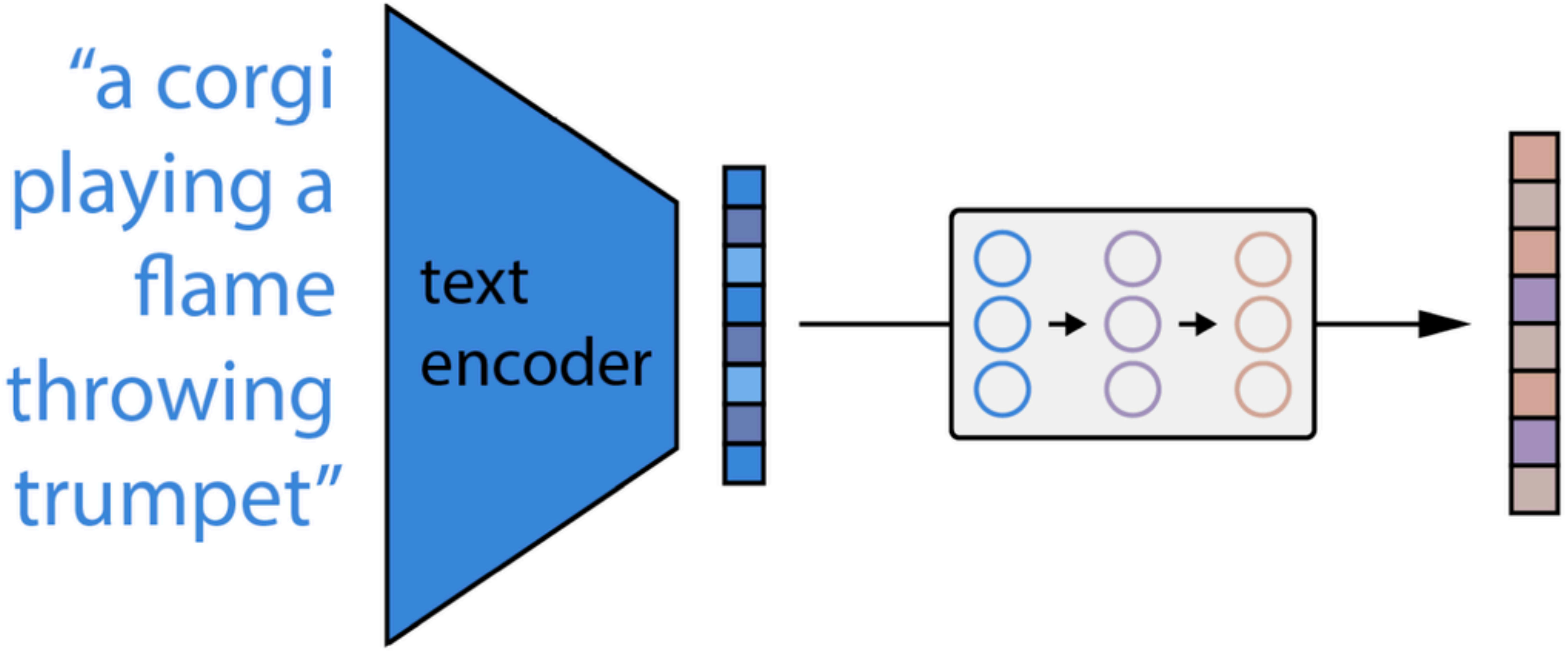
1. Input, a sentence (*prompt*) of the image we want to create

DALL-E - Overview



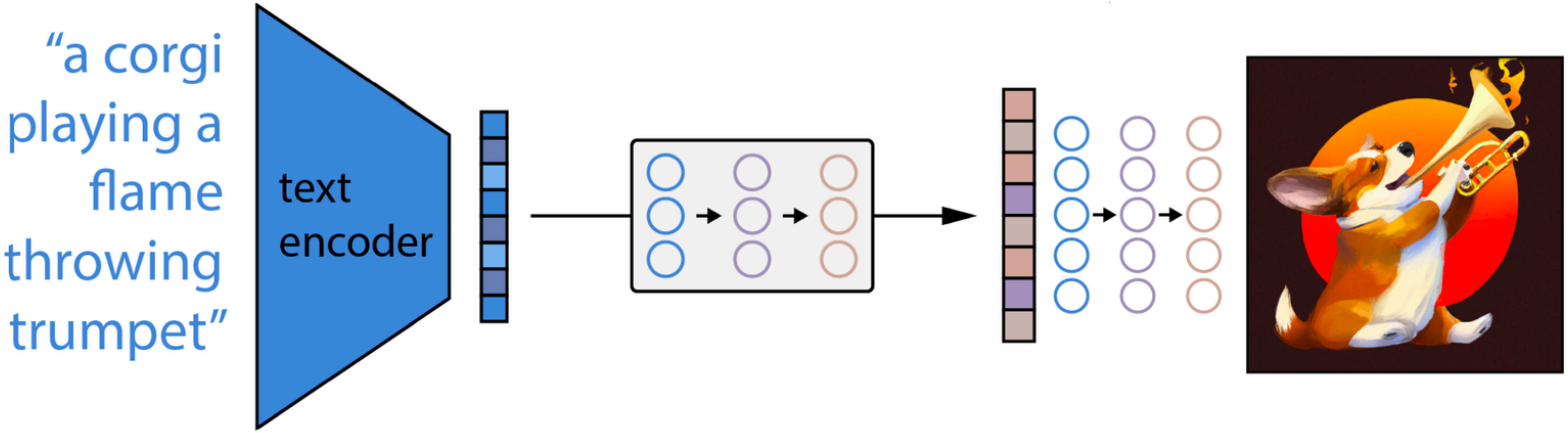
1. Input, a sentence (*prompt*) of the image we want to create
2. This prompt is encoded in a (latent) representation

DALL-E - Overview



1. Input, a sentence (*prompt*) of the image we want to create
2. This prompt is encoded in a (latent) representation
3. Transform this representation in an image representation (image space) — not shown

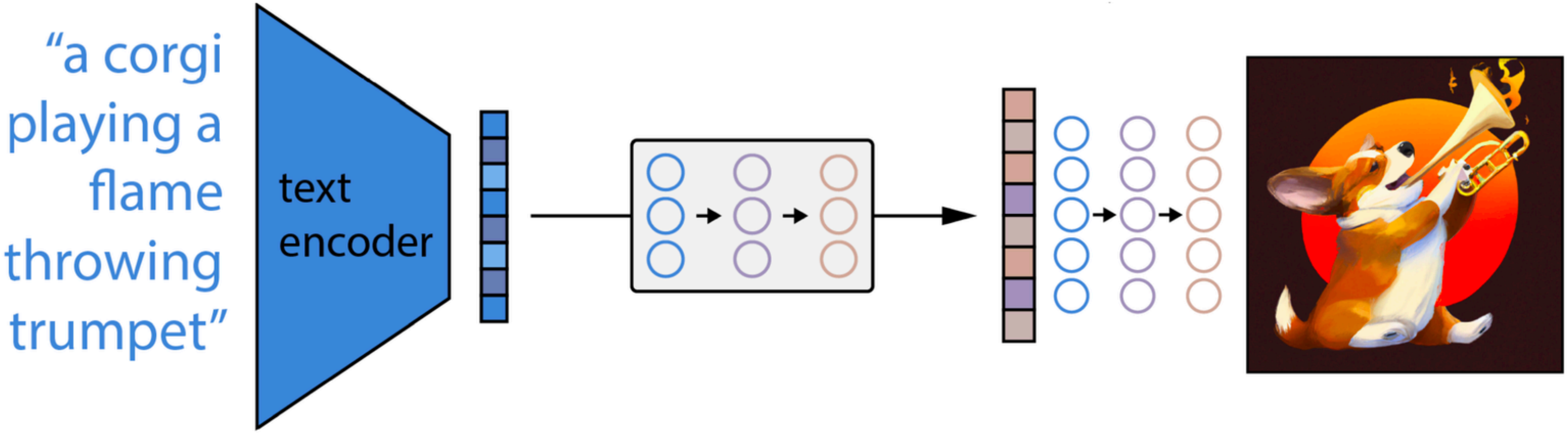
DALL-E - Overview



1. Input, a sentence (*prompt*) of the image we want to create
2. This prompt is encoded in a (latent) representation
3. Transform this representation in an image representation (image space) — not shown
4. Decode the image representation into an actual image

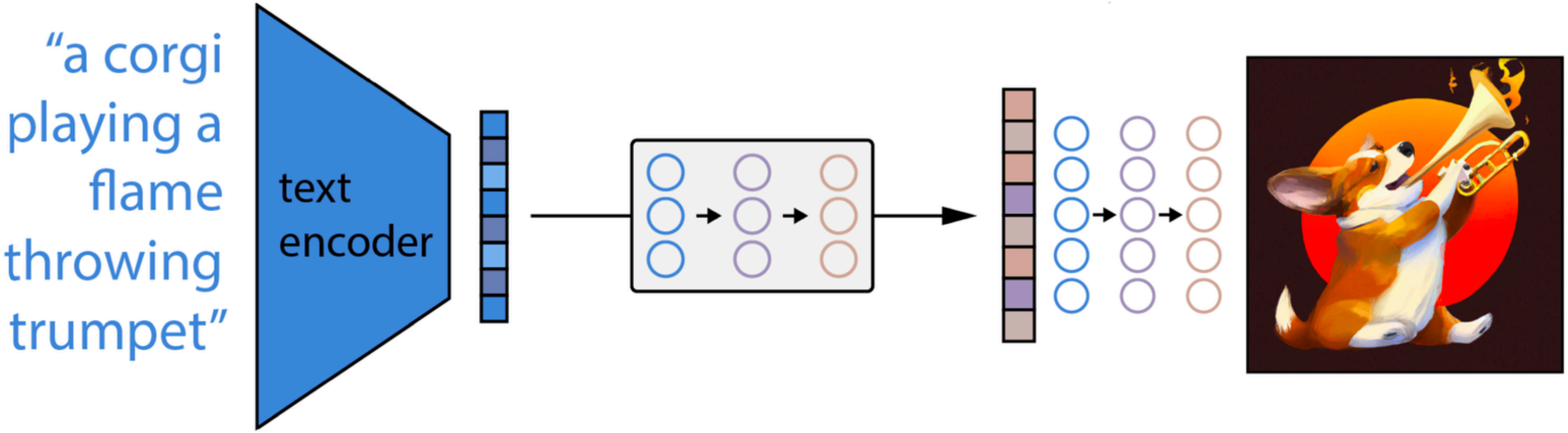
DALL-E - Overview

Each step uses specific techniques:



DALL-E - Overview

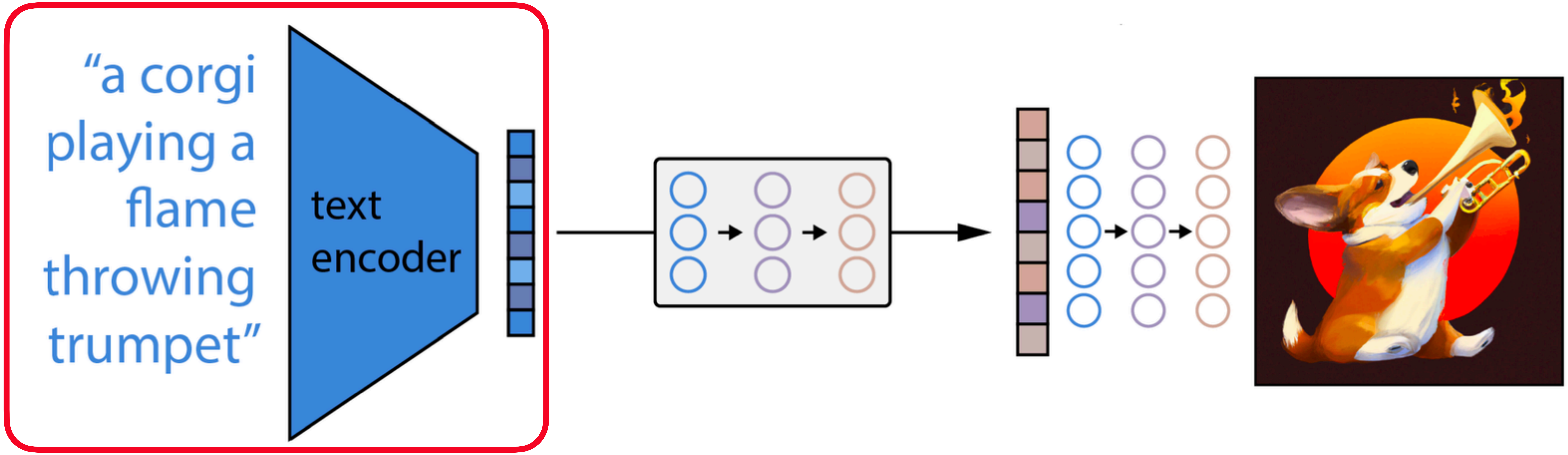
Each step uses specific techniques:



1. Contrastive Language-Image Pre-training (CLIP)

DALL-E - Overview

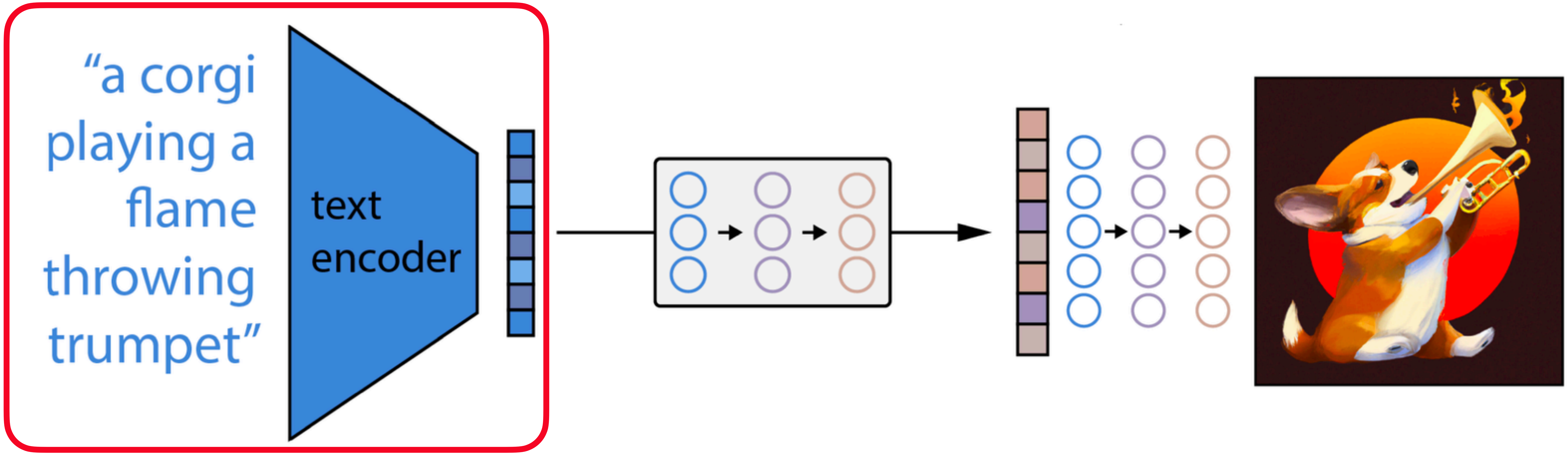
Each step uses specific techniques:



1. Contrastive Language-Image Pre-training (CLIP)

DALL-E - Overview

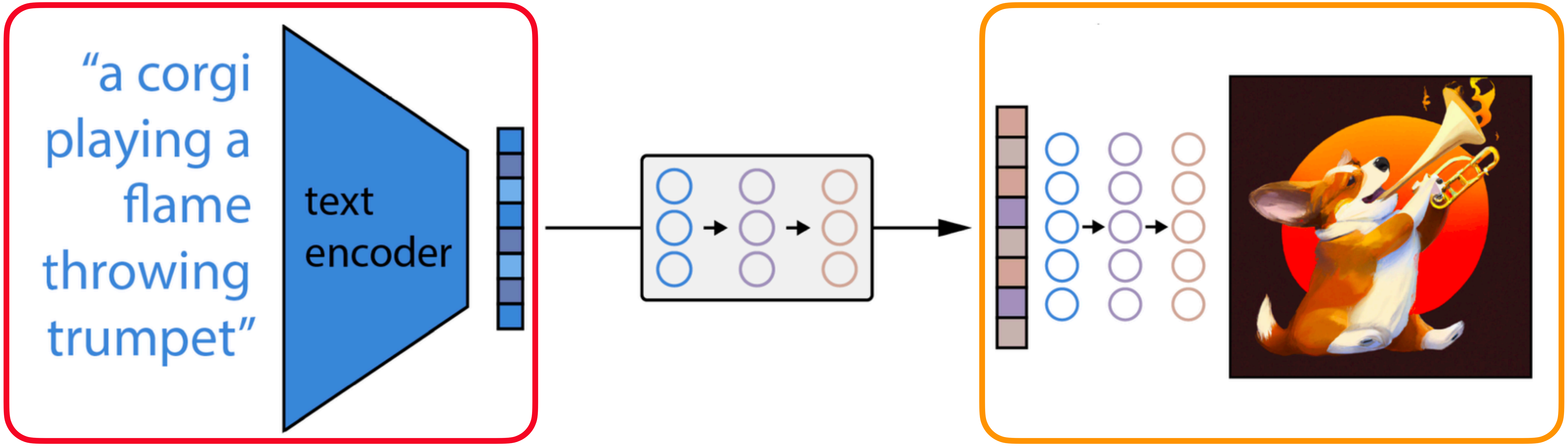
Each step uses specific techniques:



1. Contrastive Language-Image Pre-training (CLIP)
2. Generation of an image using a diffusion model

DALL-E - Overview

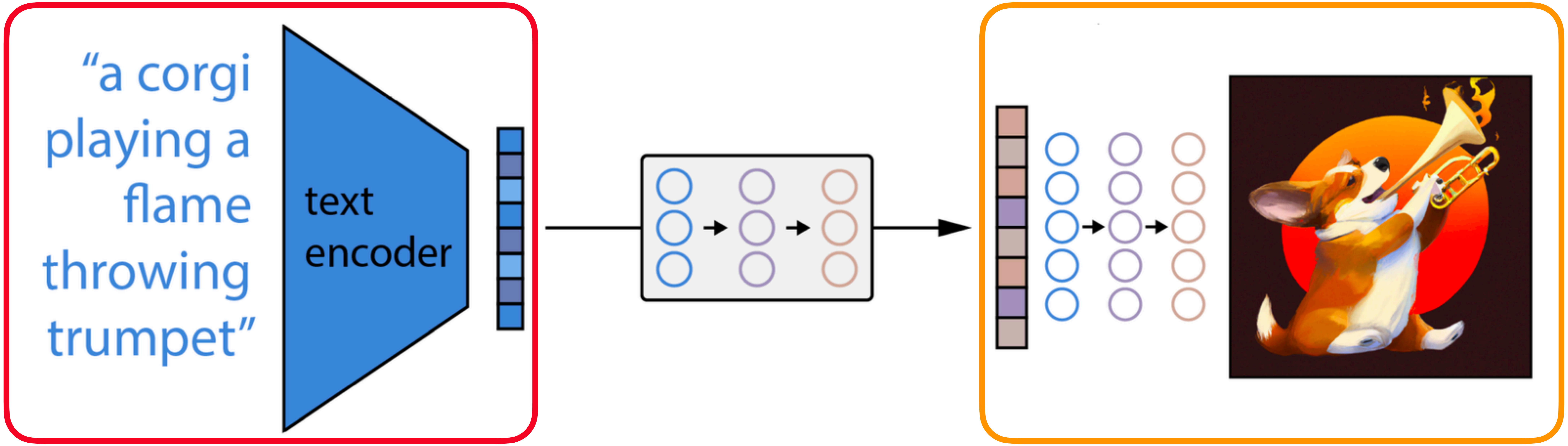
Each step uses specific techniques:



1. Contrastive Language-Image Pre-training (CLIP)
2. Generation of an image using a diffusion model

DALL-E - Overview

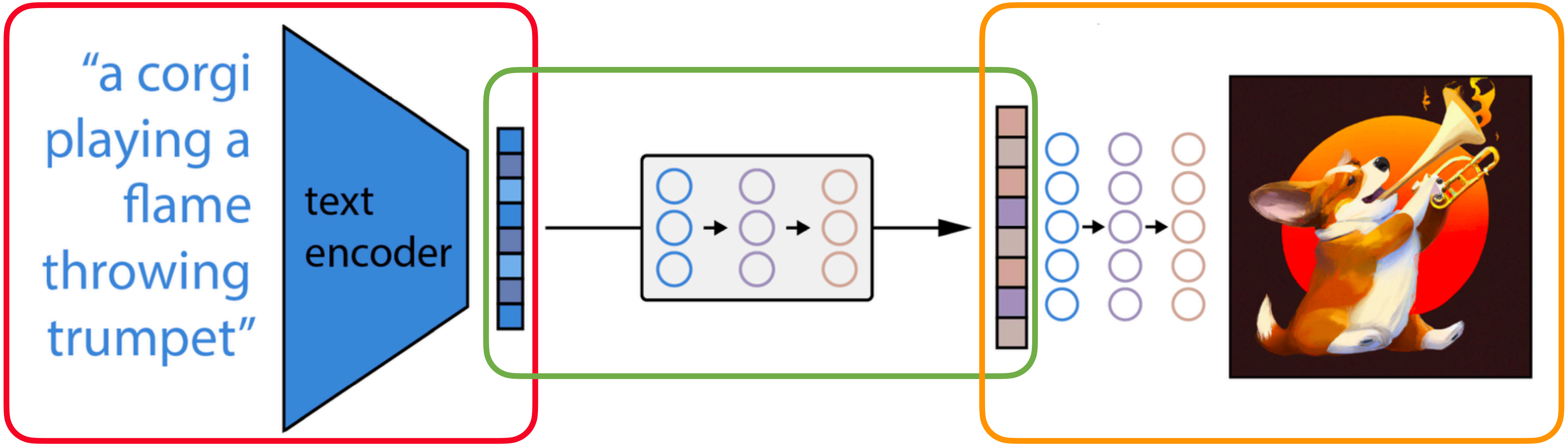
Each step uses specific techniques:



1. Contrastive Language-Image Pre-training (CLIP)
2. Generation of an image using a diffusion model
3. Learn the latent representations of text and images

DALL-E - Overview

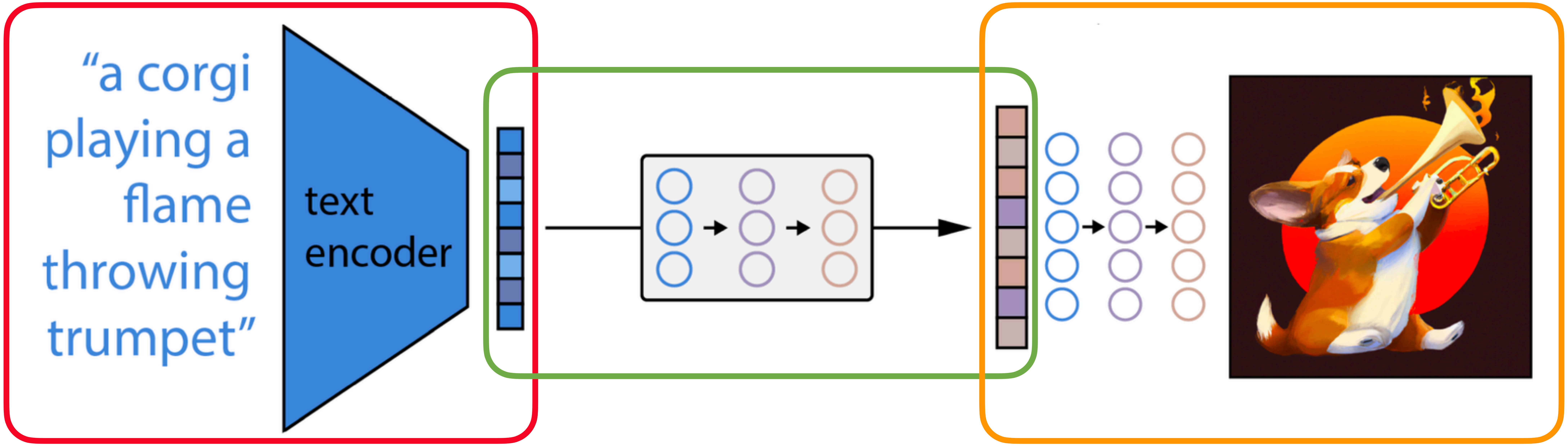
Each step uses specific techniques:



1. Contrastive Language-Image Pre-training (CLIP)
2. Generation of an image using a diffusion model
3. Learn the latent representations of text and images

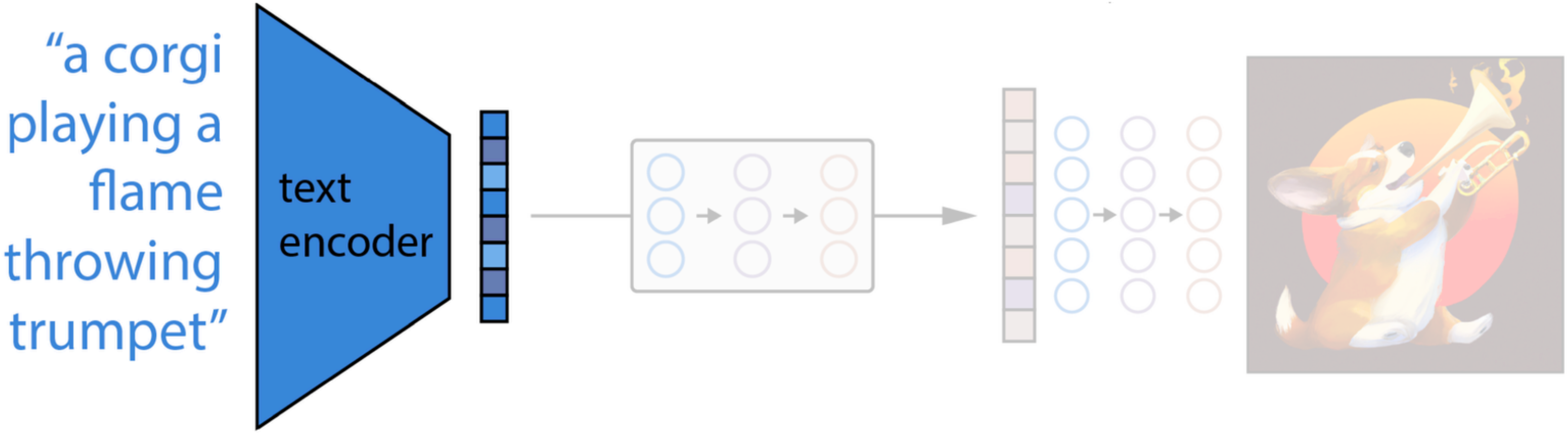
DALL-E - Overview

Each step uses specific techniques:

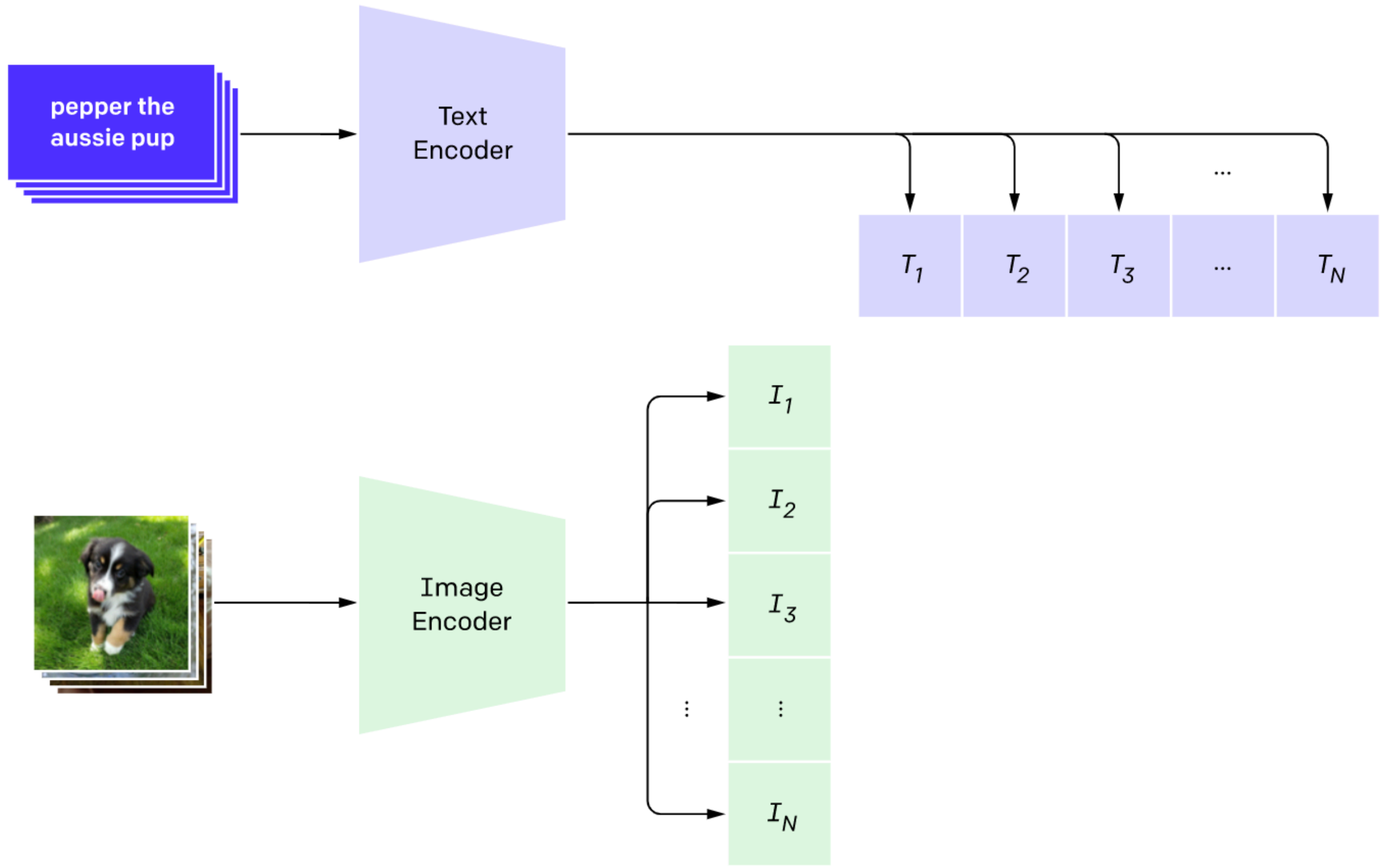


1. Contrastive Language-Image Pre-training (CLIP)
2. Generation of an image using a diffusion model
3. Learn the latent representations of text and images
4. Wrap-it up!

DALL-E - CLIP

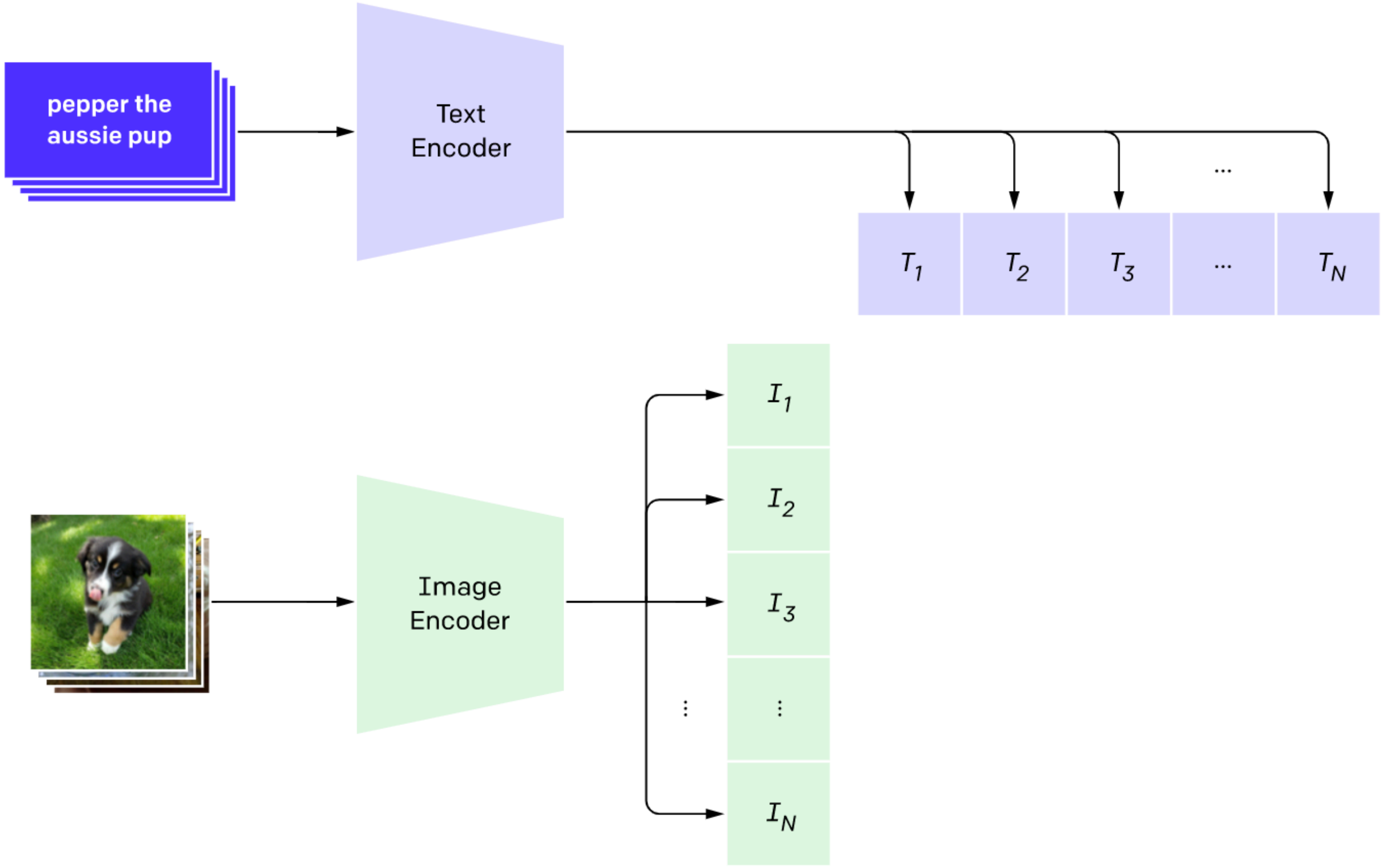


DALL-E - CLIP



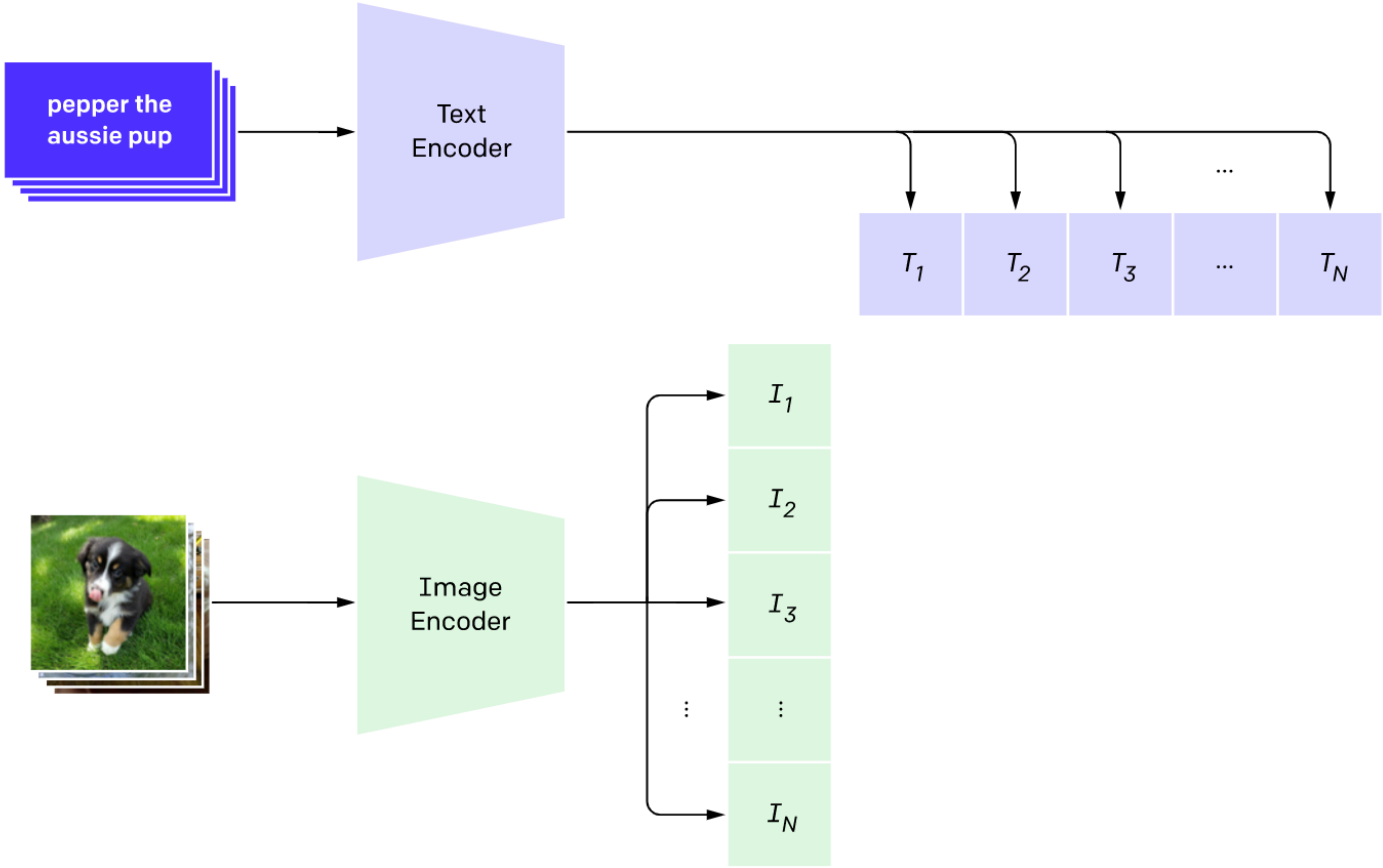
pepper the aussie pup
Text Encoder
T_1 T_2 T_3 ... T_N
Image Encoder
I_1 I_2 I_3 ⋮ I_N

DALL-E - CLIP



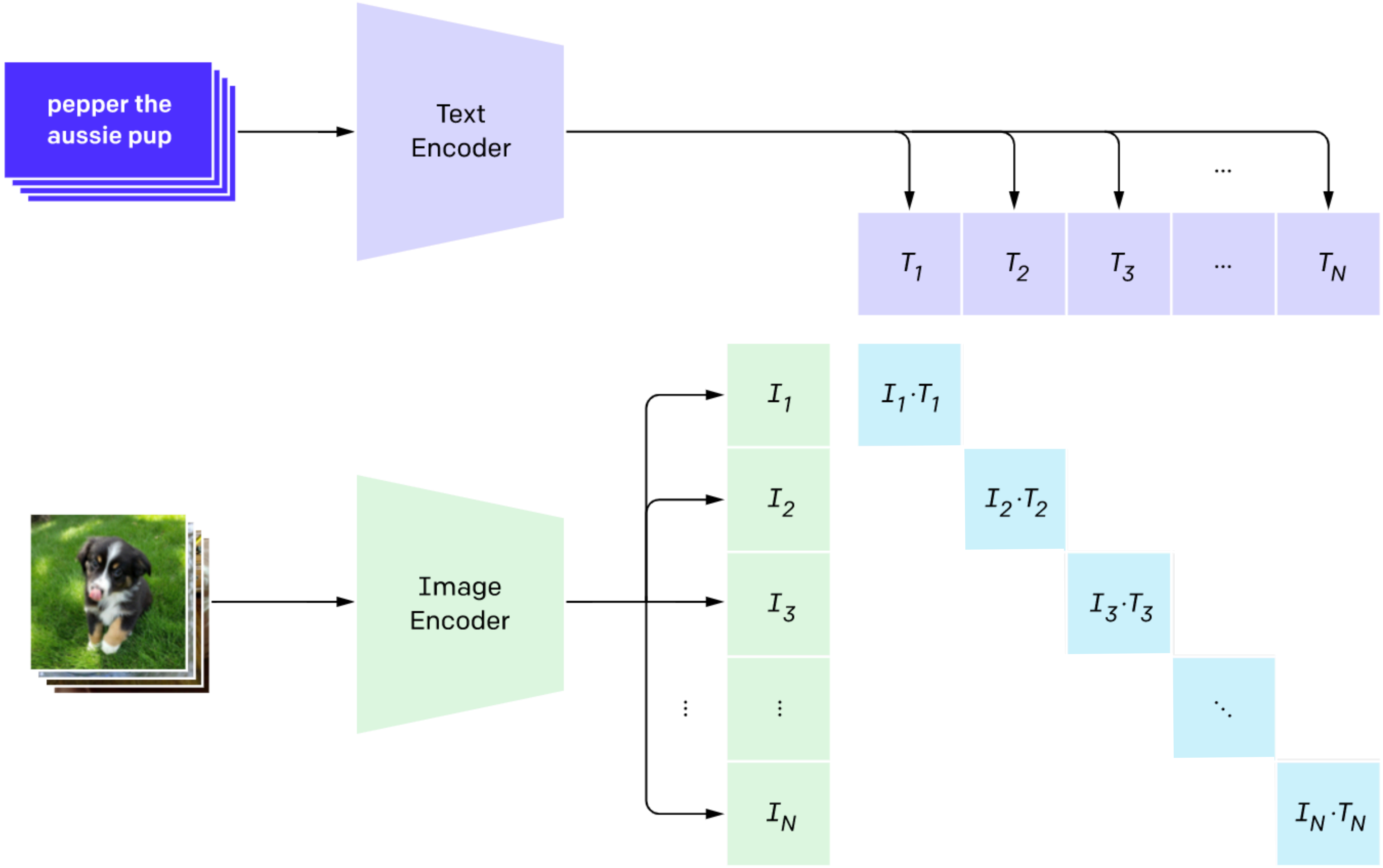
1. Learn latent representations of text T_i and images I_i

DALL-E - CLIP



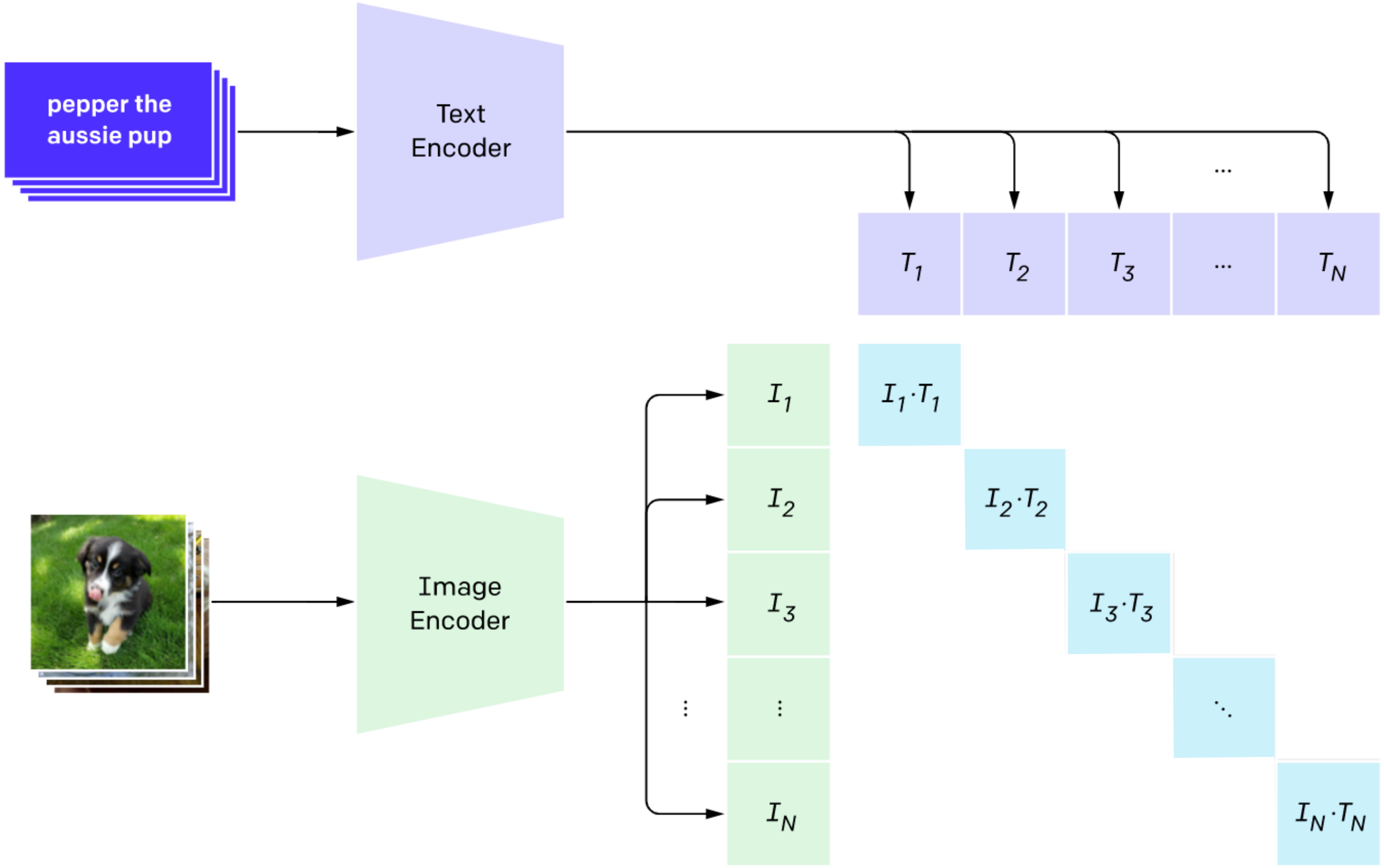
1. Learn latent representations of text T_i and images I_i
   - From a batch of size N

DALL-E - CLIP



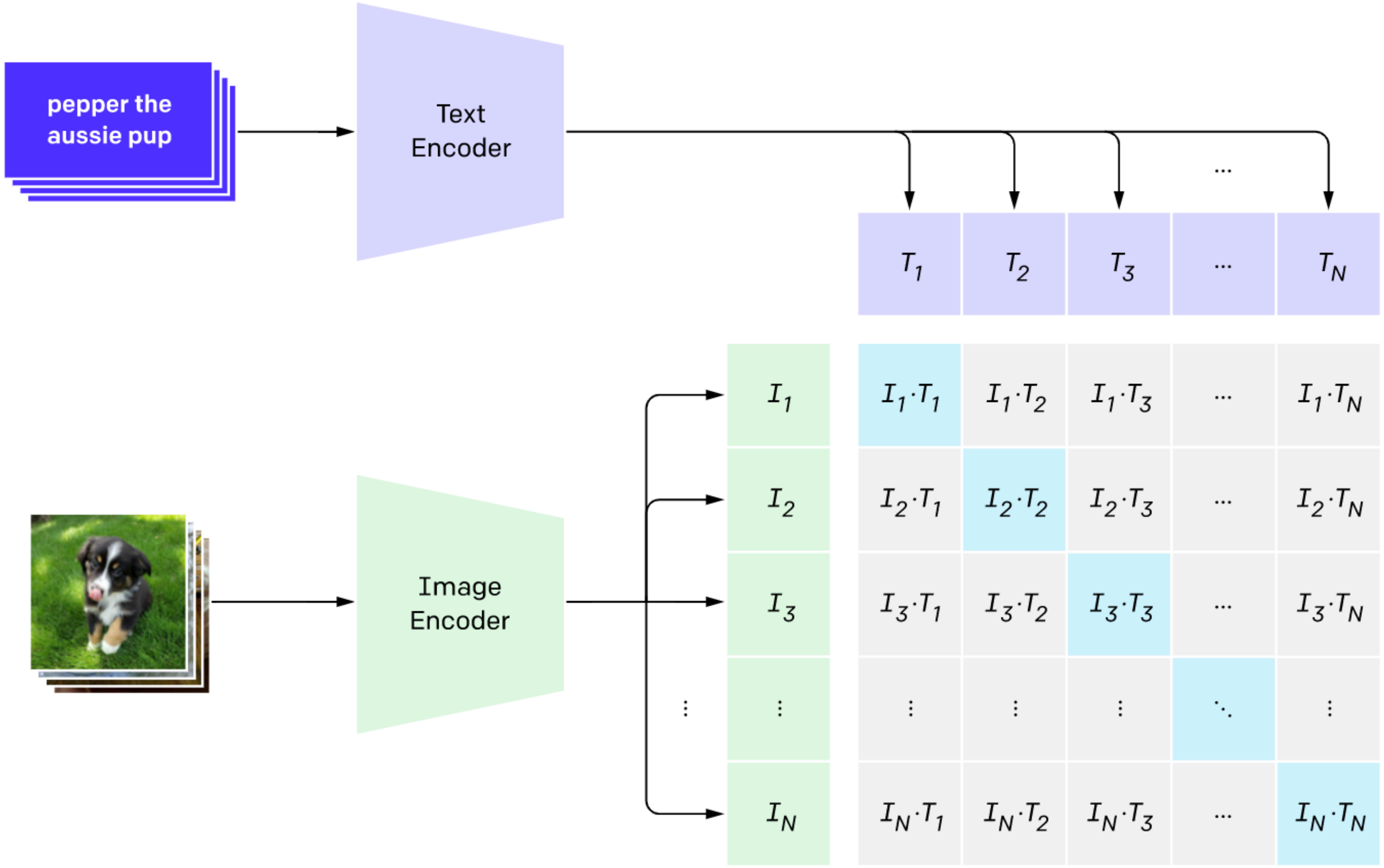
1. Learn latent representations of text T_i and images I_i
   - From a batch of size N

DALL-E - CLIP



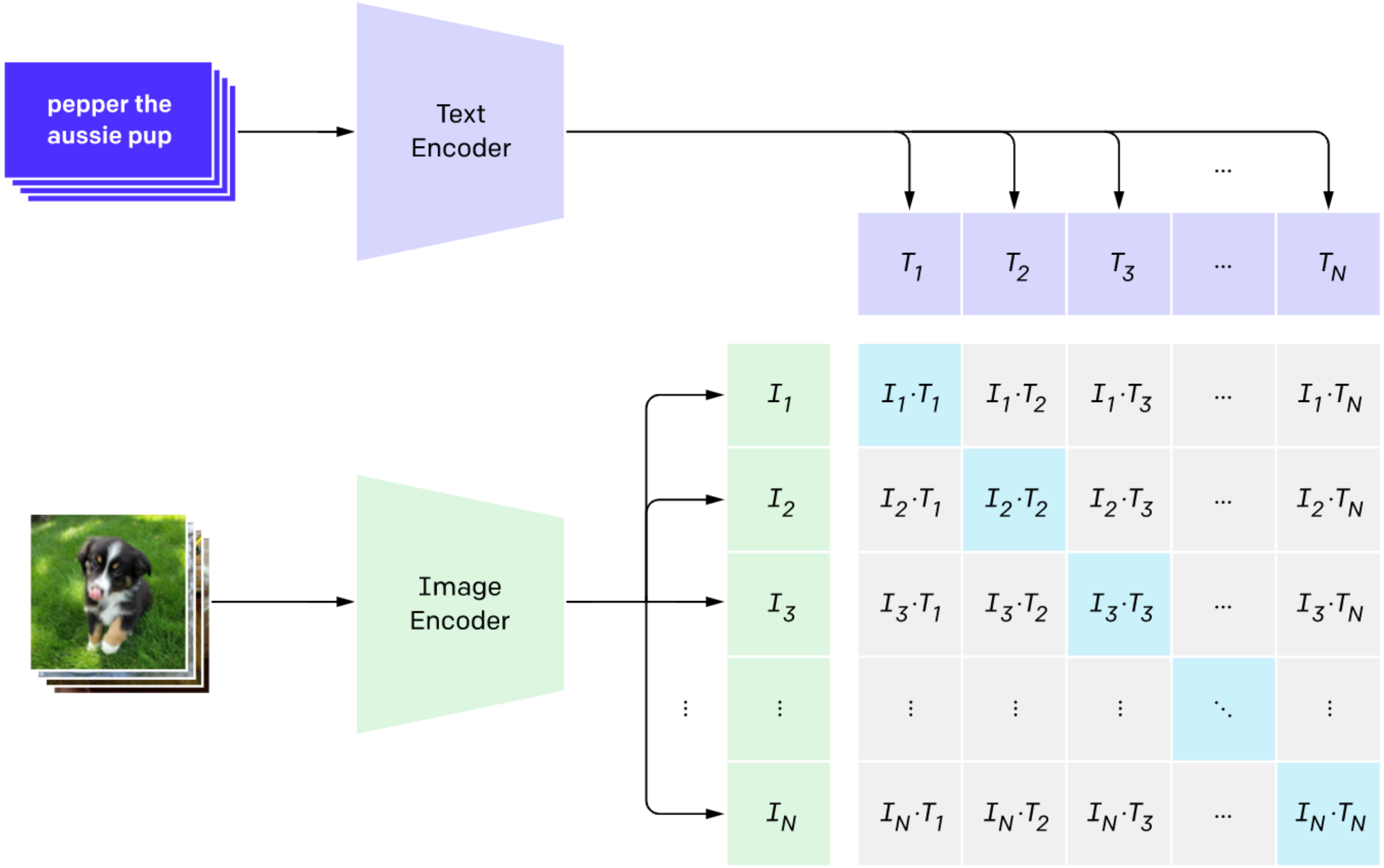
1. Learn latent representations of text T_i and images I_i
   - From a batch of size N
2. Similarity measure

DALL-E - CLIP



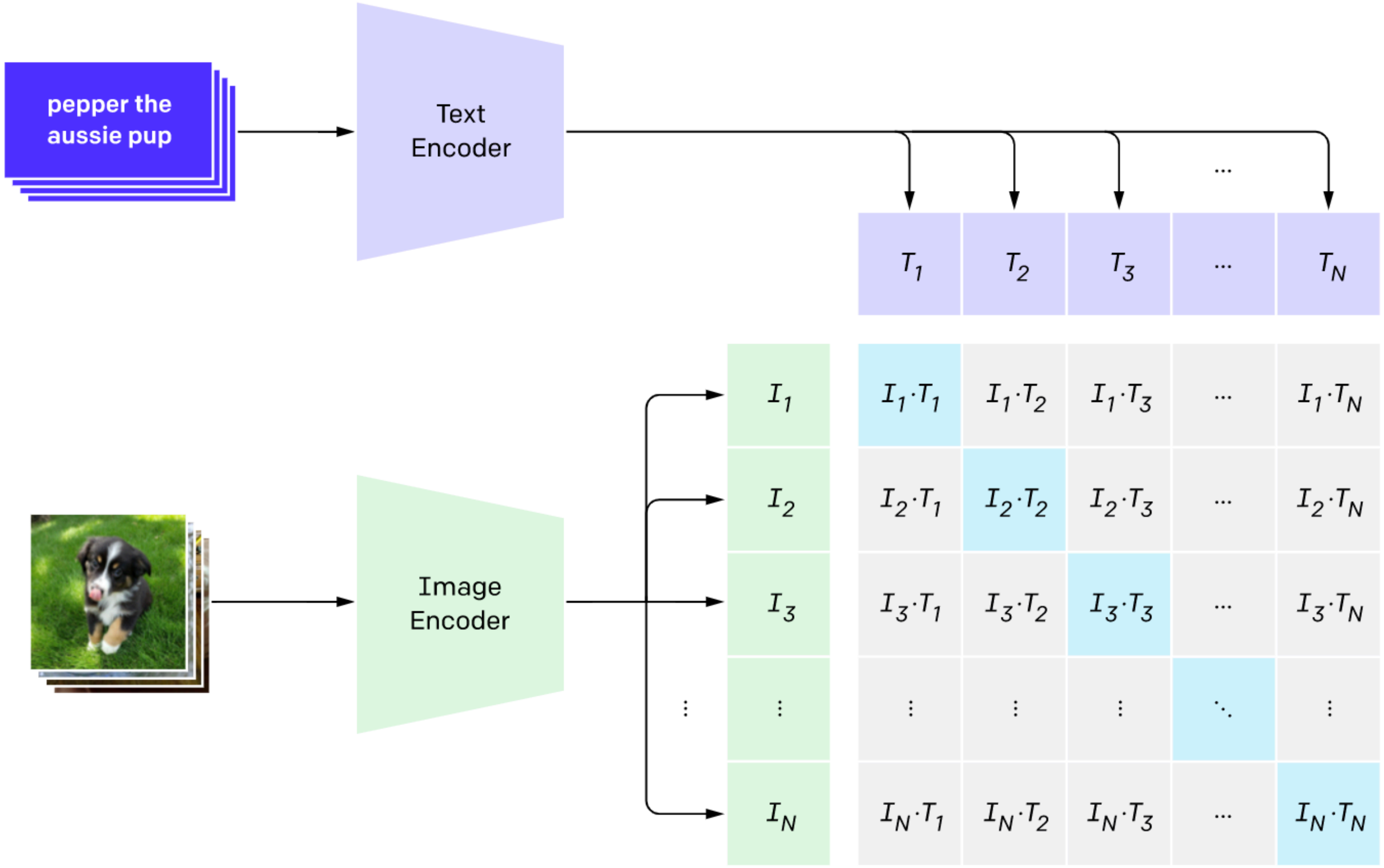
1. Learn latent representations of text T_i and images I_i
   - From a batch of size N
2. Similarity measure

DALL-E - CLIP



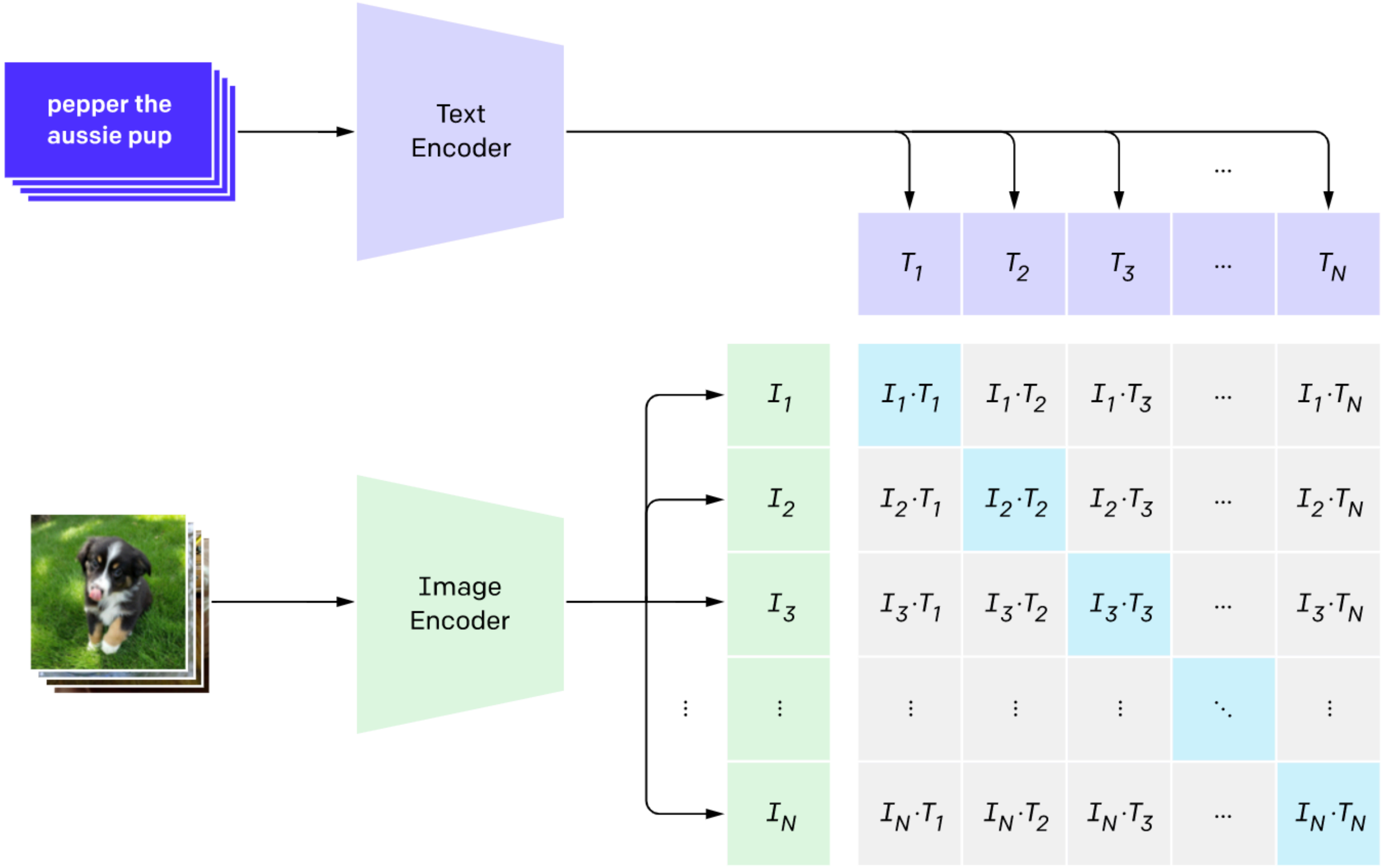
1. Learn latent representations of text T_i and images I_i
   - From a batch of size N
2. Similarity measure
   - $\max T_i I_i$

DALL-E - CLIP



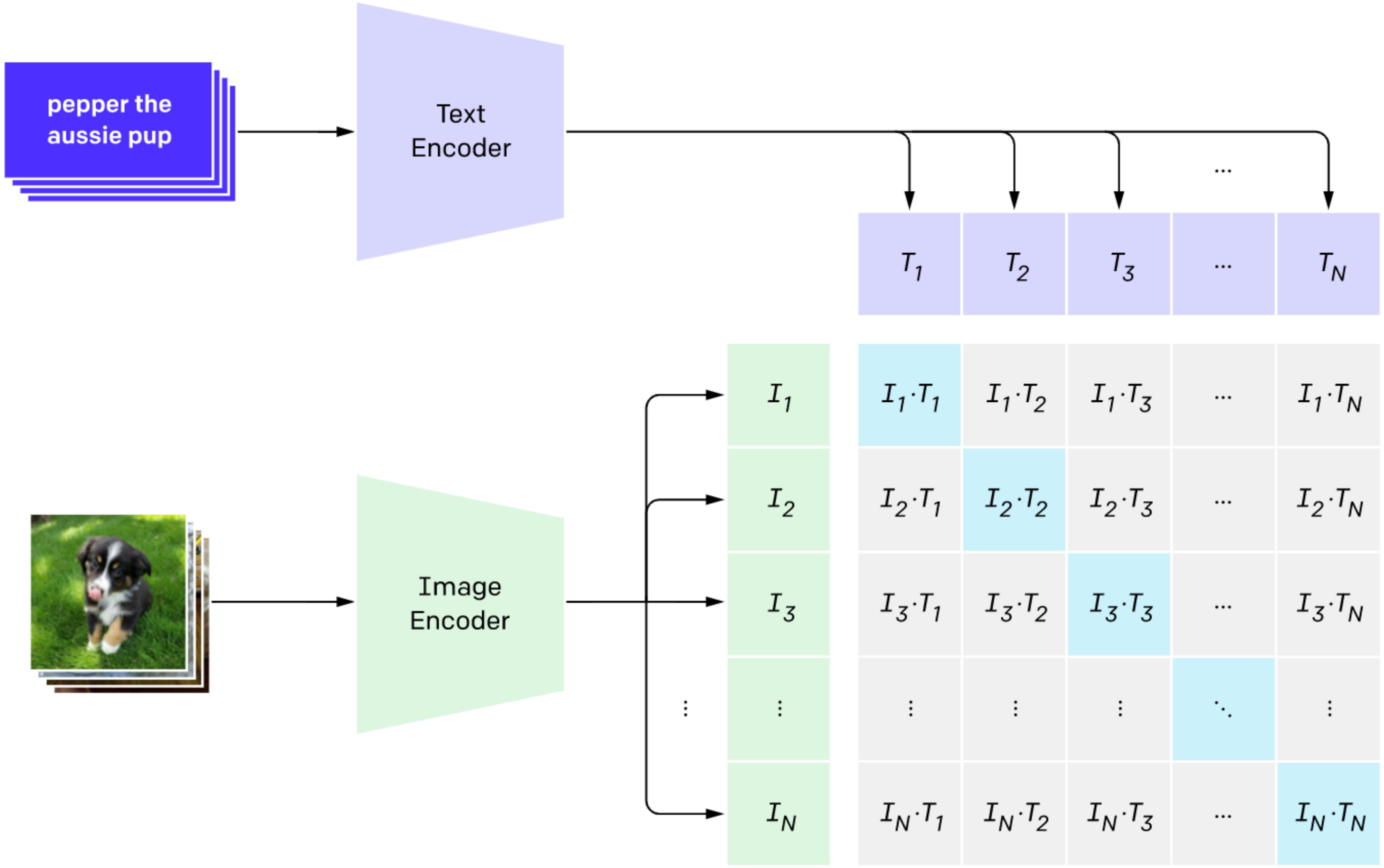
1. Learn latent representations of text T_i and images I_i
   - From a batch of size N
2. Similarity measure
   - $\max T_i I_i$
3. Dissimilarity measure

DALL-E - CLIP



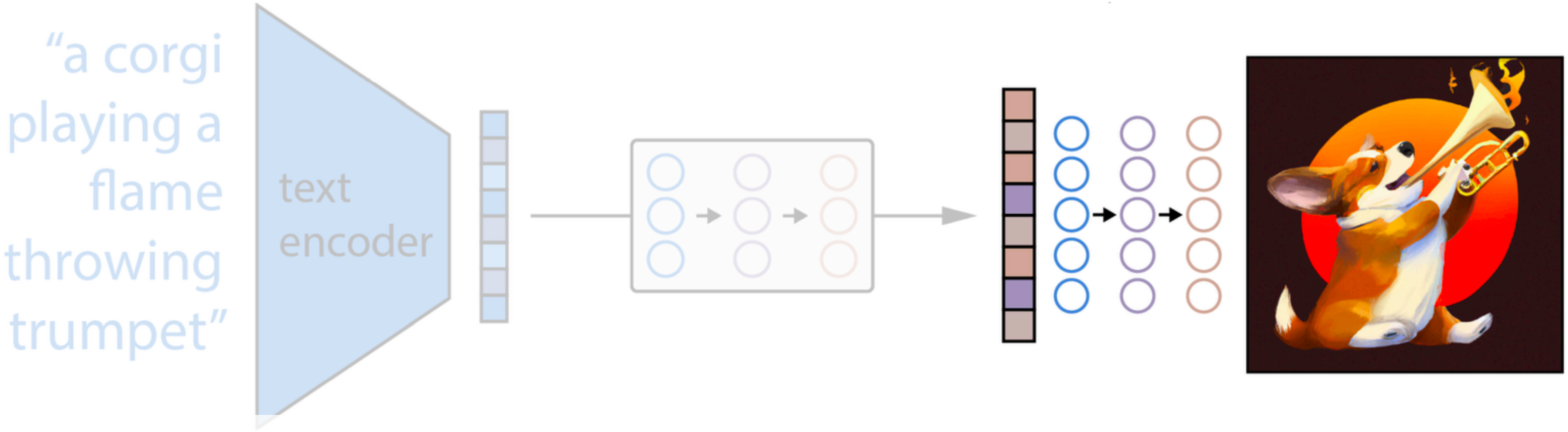
1. Learn latent representations of text T_i and images I_i
   - From a batch of size N
2. Similarity measure
   - $\max T_i I_i$
3. Dissimilarity measure
   - $\min T_i I_j \quad \forall j \neq i$

DALL-E - CLIP

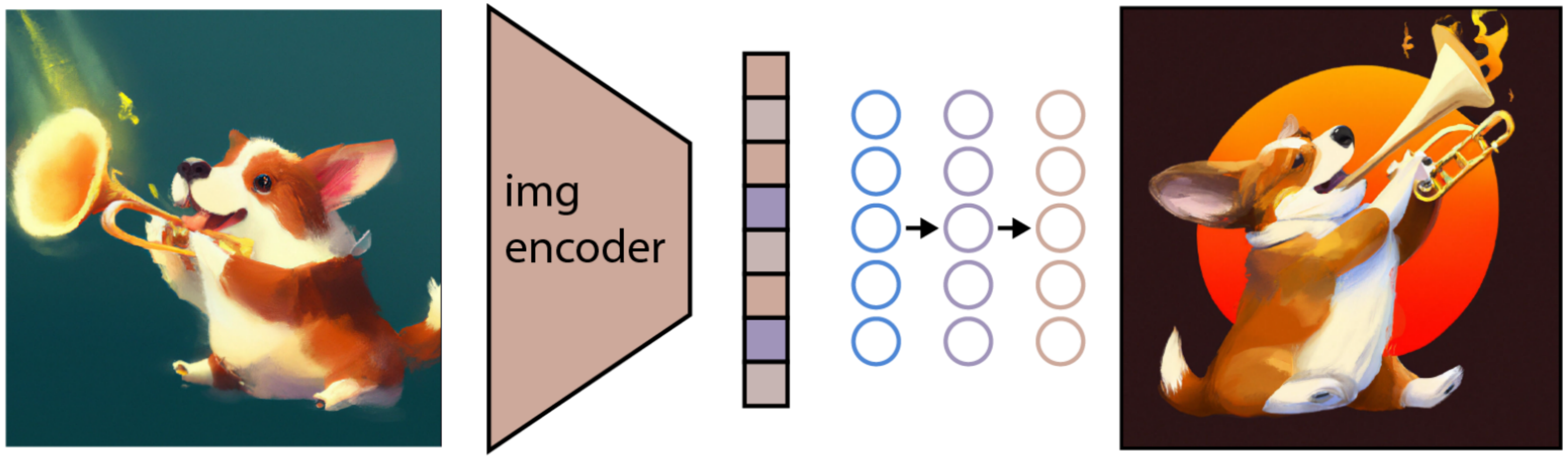


1. Learn latent representations of text T_i and images I_i
   - From a batch of size N
2. Similarity measure
   - $\max T_i I_i$
3. Dissimilarity measure
   - $\min T_i I_j \quad \forall j \neq i$
4. Maximize the similarities and minimize the dissimilarities

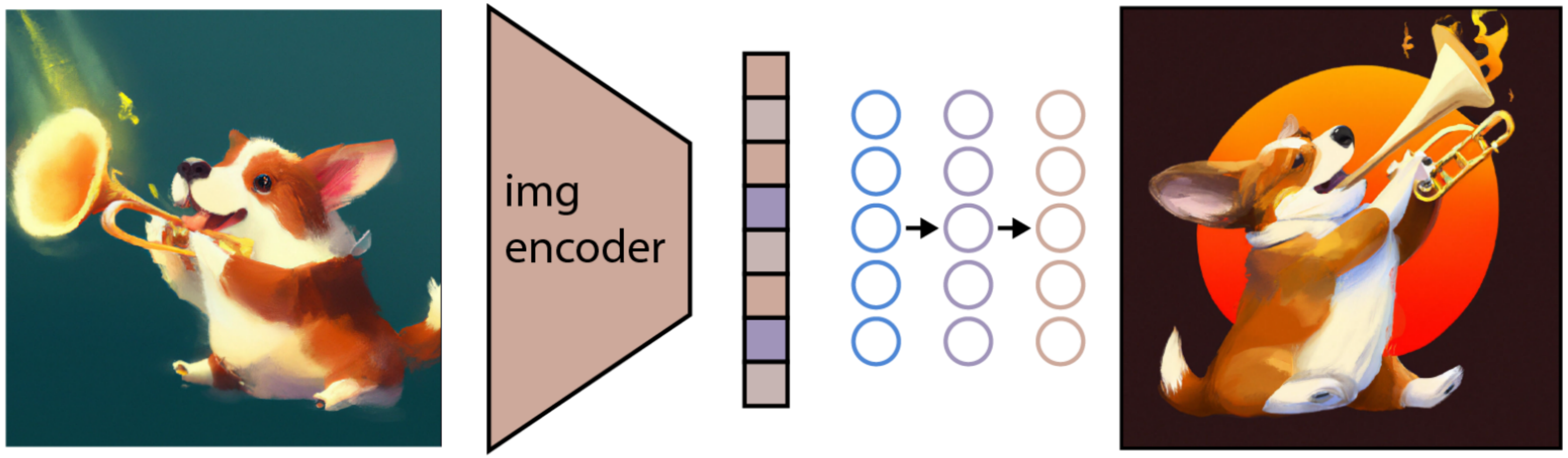
DALL-E - Diffusion Models



DALL-E - Diffusion Models

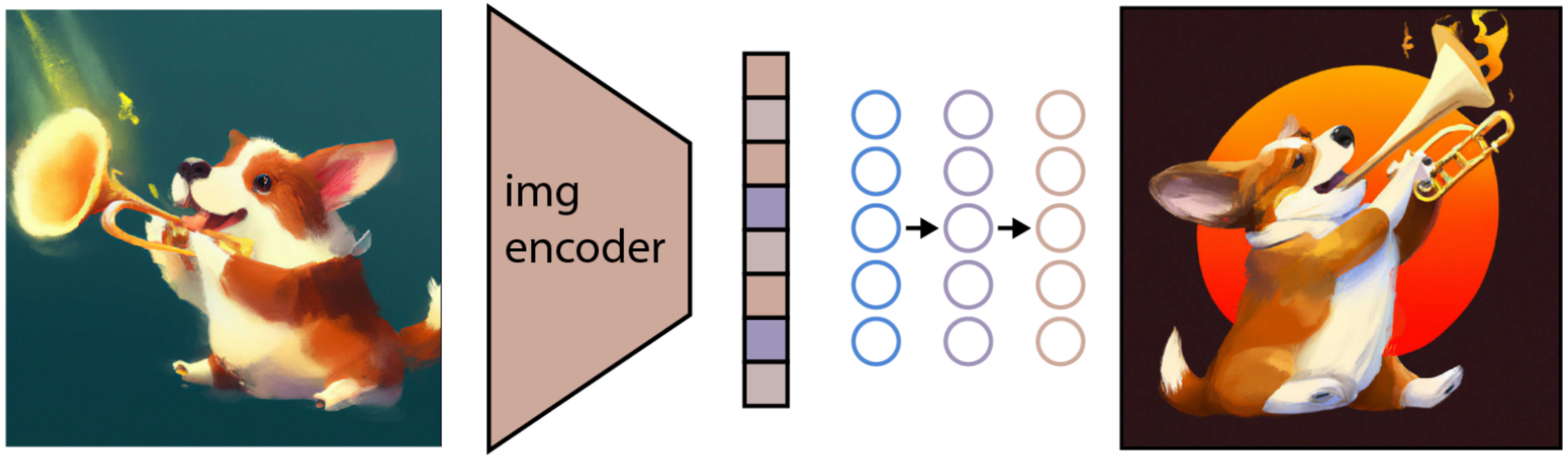


DALL-E - Diffusion Models



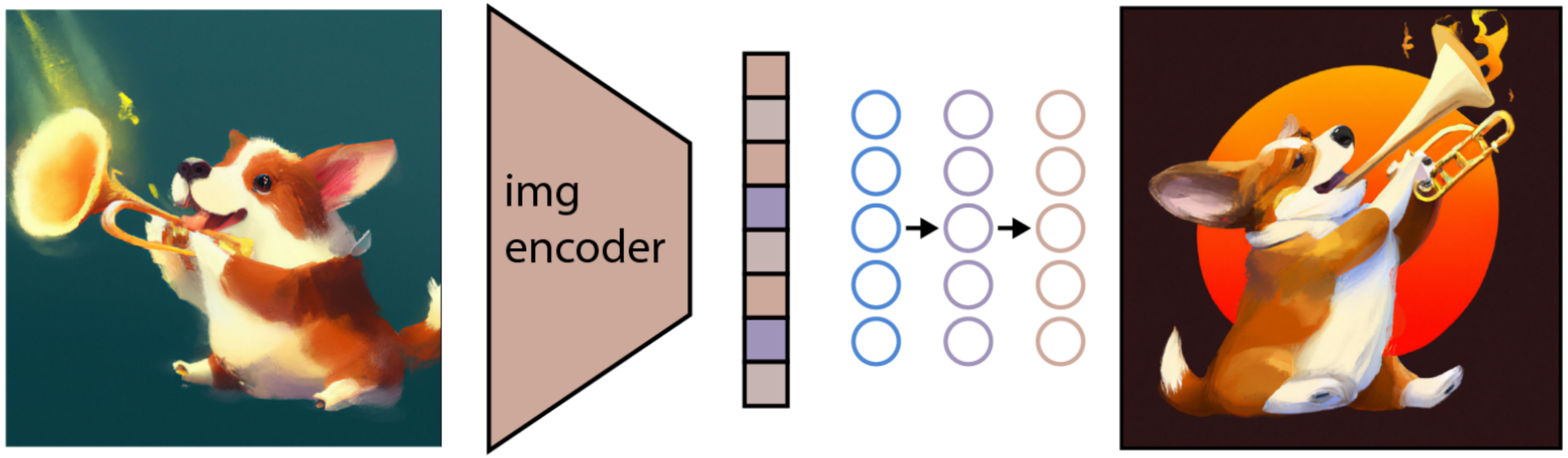
1. Contrastive Language-Image Pre-training (CLIP)

DALL-E - Diffusion Models



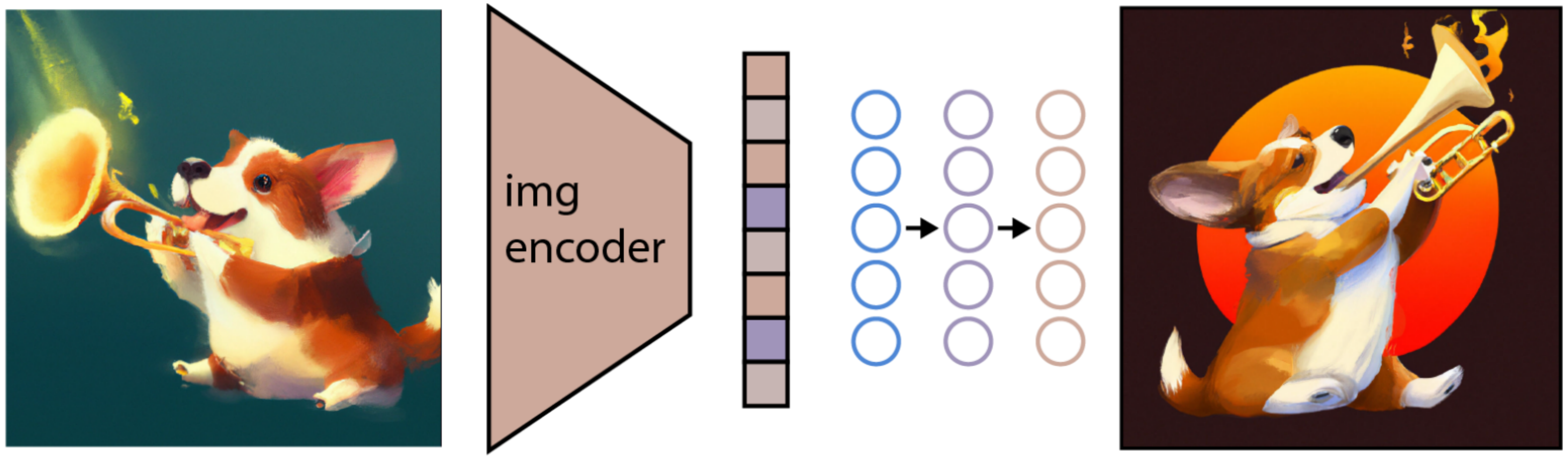
1. Contrastive Language-Image Pre-training (CLIP)
2. Generation of an image using a diffusion model

DALL-E - Diffusion Models



1. Contrastive Language-Image Pre-training (CLIP)
2. Generation of an image using a diffusion model
3. Learn the latent representations of text and images

DALL-E - Diffusion Models

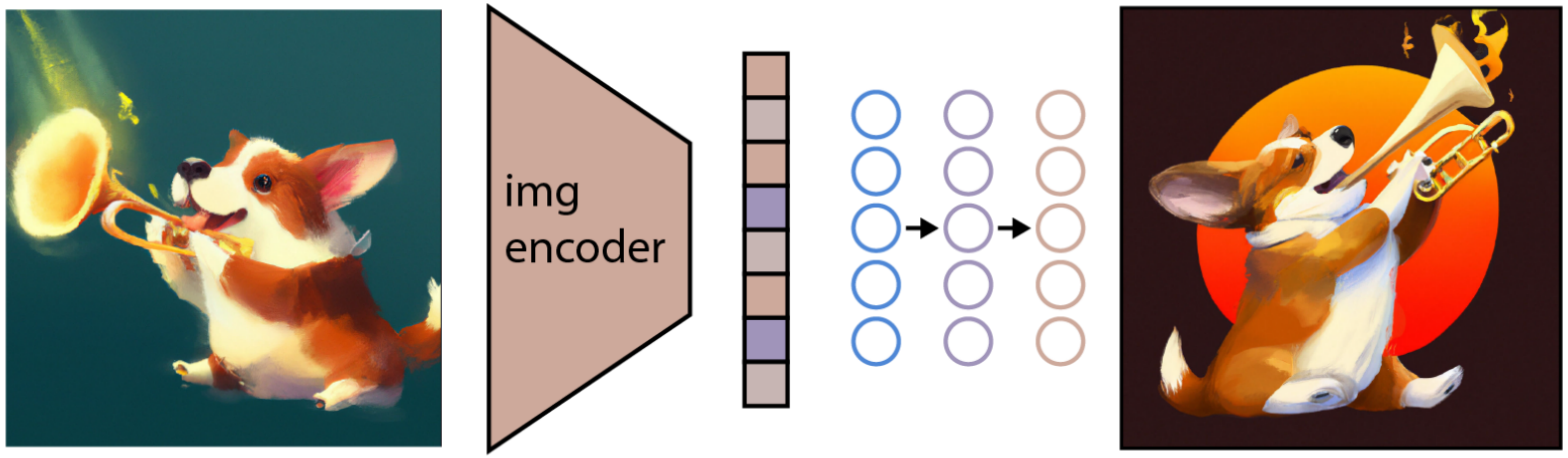


1. Contrastive Language-Image Pre-training (CLIP)
2. Generation of an image using a diffusion model
3. Learn the latent representations of text and images
4. Wrap-it up!

DALL-E - Diffusion Model

Idea:

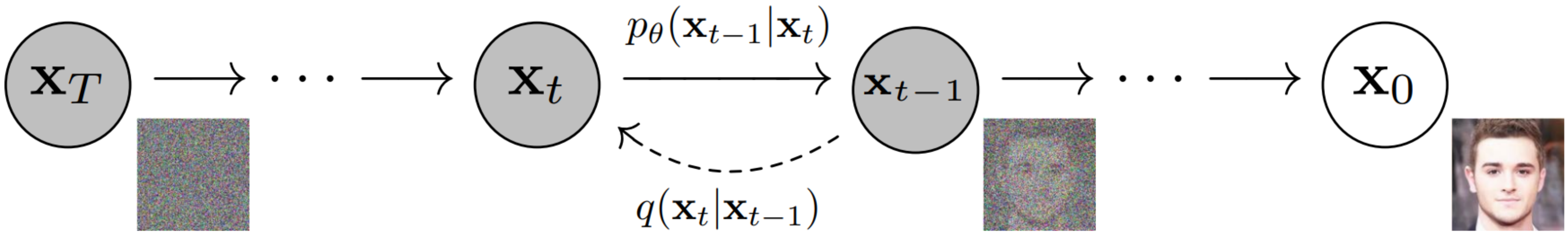
- From an image (left), generate other similar ones
- This is where diffusion models are used



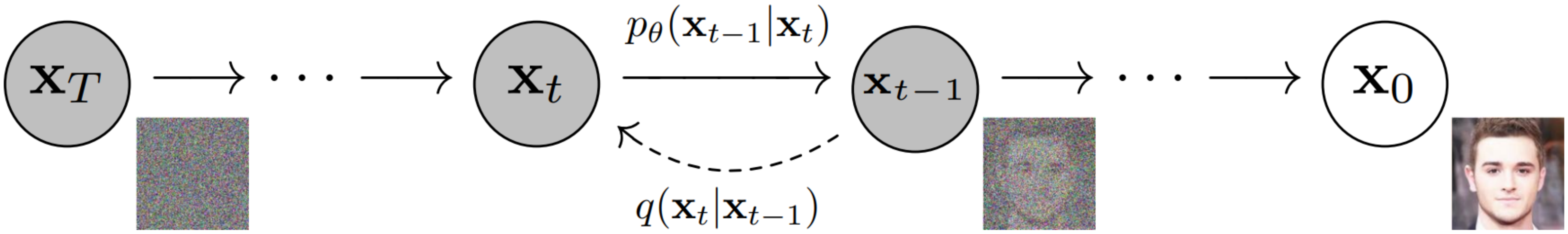
Diffusion Models

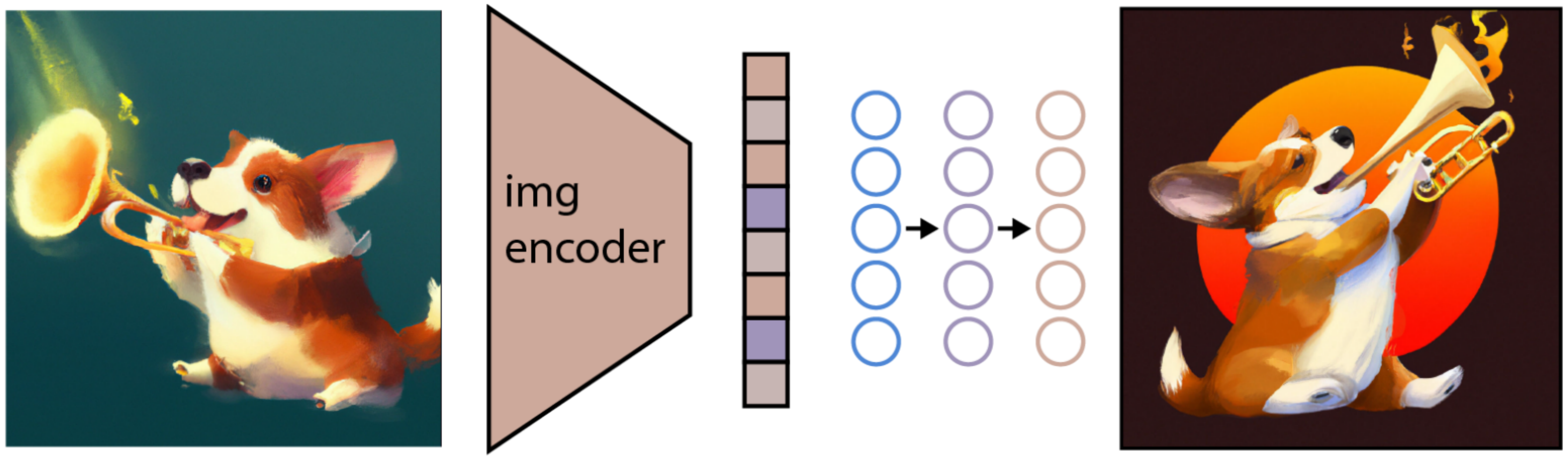
Idea:

- Add noise incrementally to an image until it is pure white noise
- Danoise the image to obtain the original image
- If we know the noise mechanism, starting from white noise, we can then generate an image



DALL-E - Diffusion Model

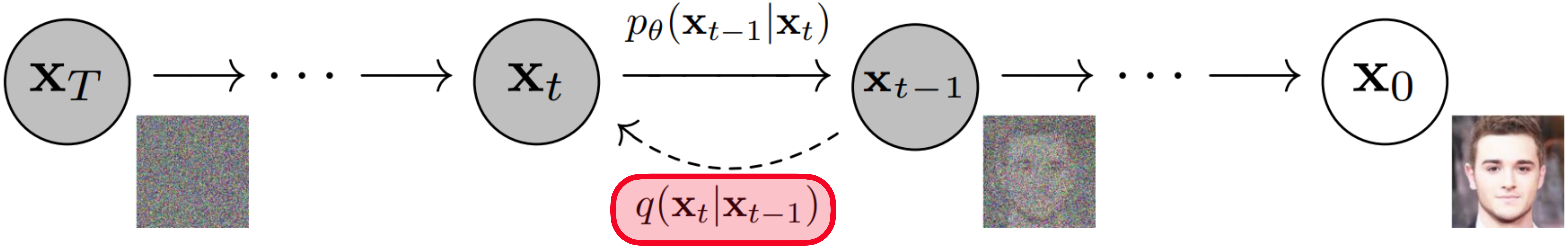


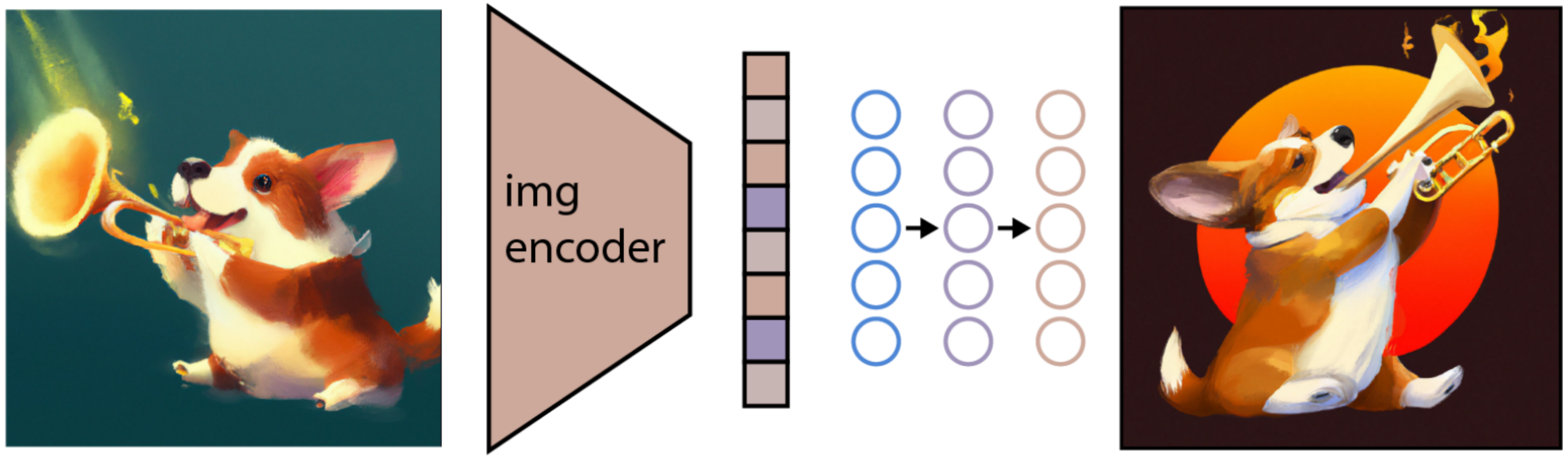


DALL-E - Diffusion Model

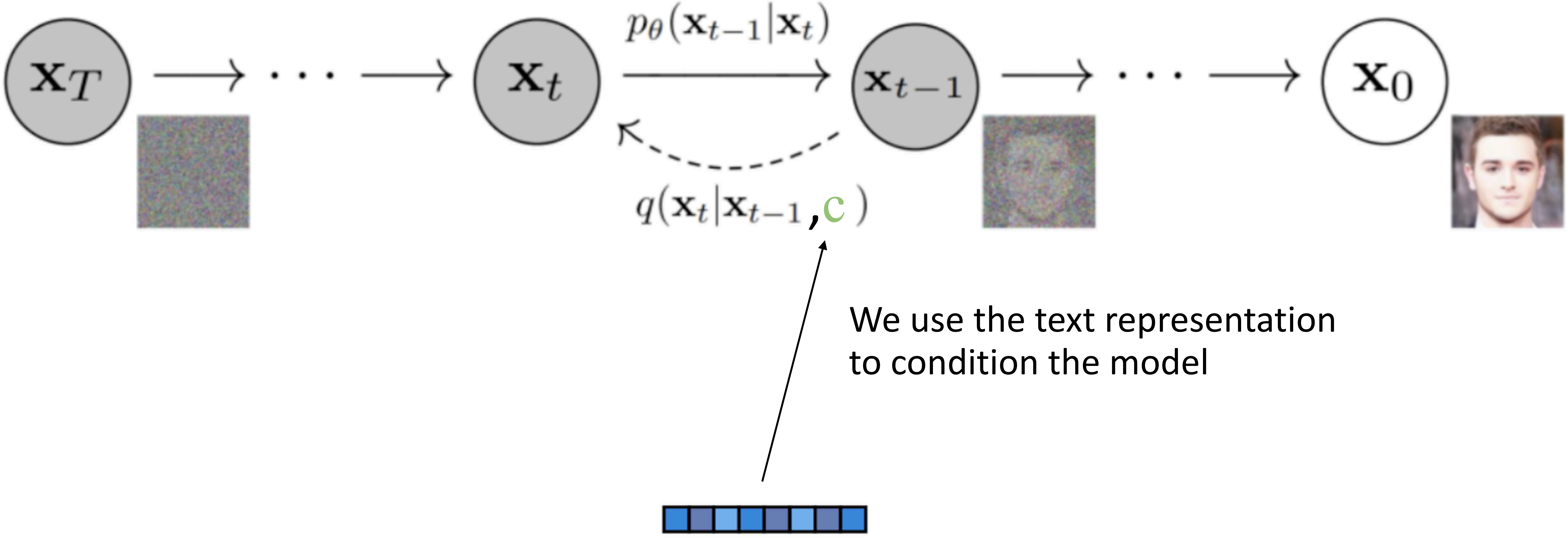
For text-to-image generation, we add information from the text

During the diffusion process



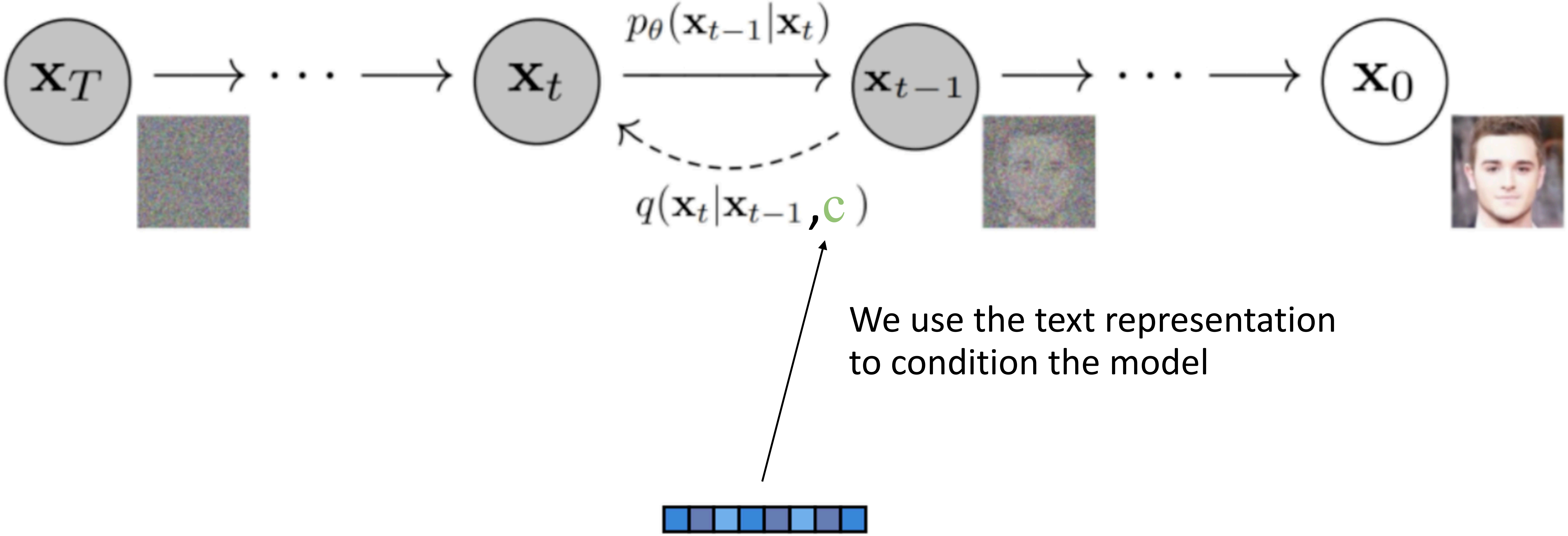


DALL-E



$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_T$
$\mathbf{x}_t$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$
$q(\mathbf{x}_t|\mathbf{x}_{t-1}, c)$
We use the text representation
to condition the model

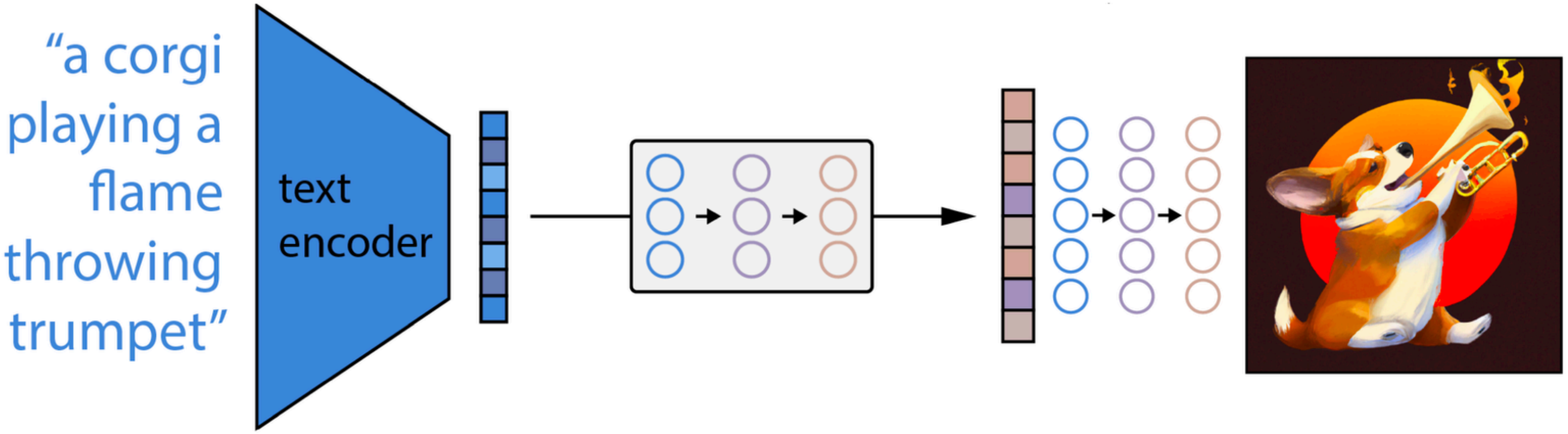
DALL-E



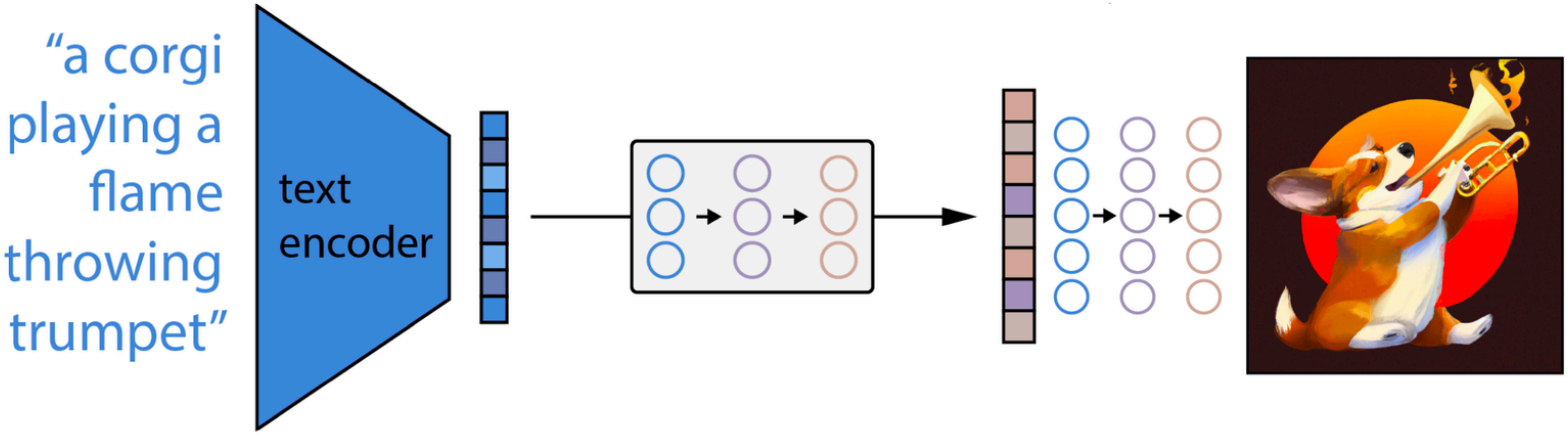
We use the text representation to condition the model

“Face face of a man with red hair”

DALL-E - Wrap-it up!

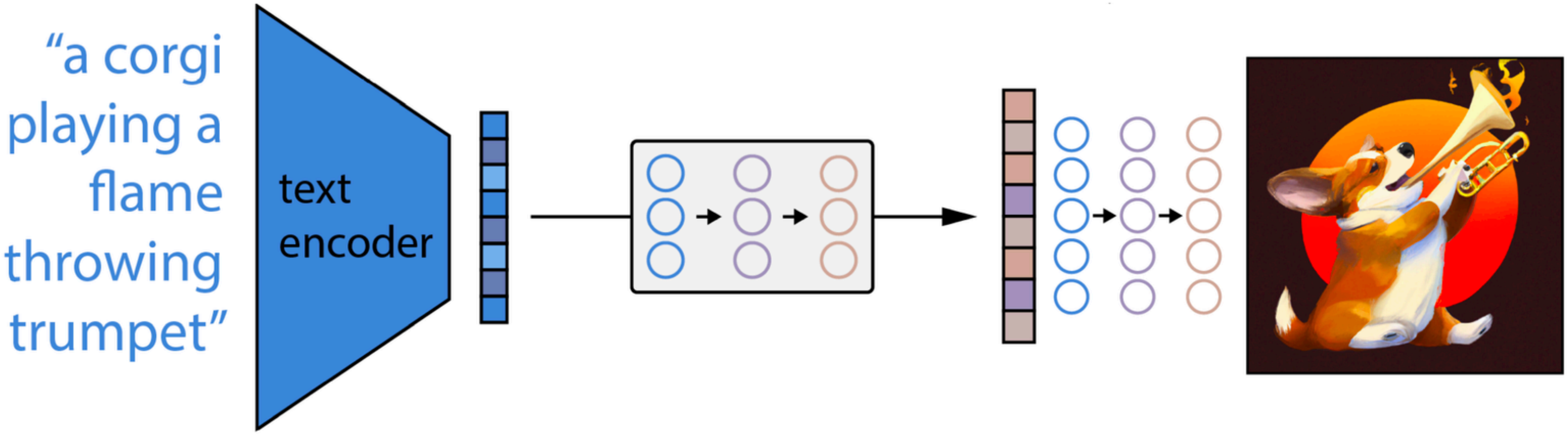


DALL-E - Wrap-it up!



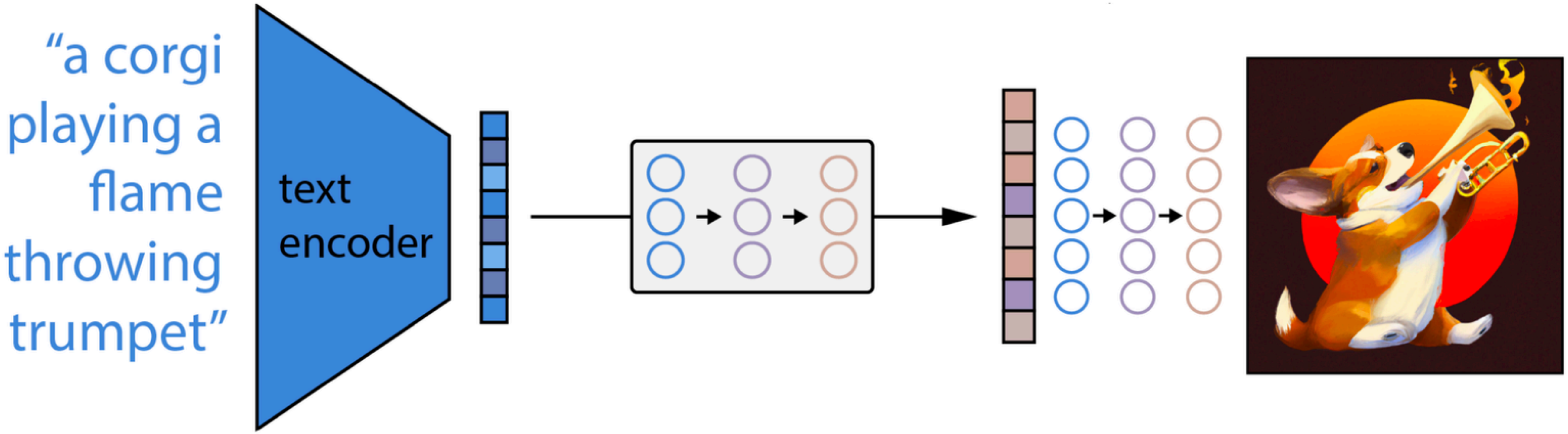
1. Contrastive Language-Image Pre-training (CLIP)

DALL-E - Wrap-it up!



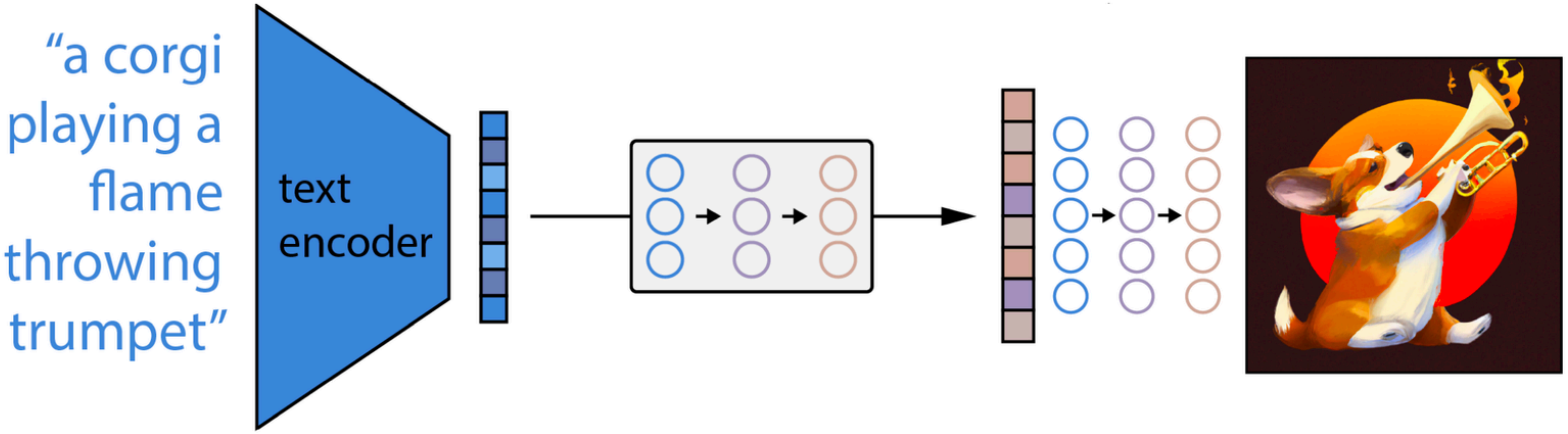
1. Contrastive Language-Image Pre-training (CLIP)
2. Generation of an image using a diffusion model

DALL-E - Wrap-it up!



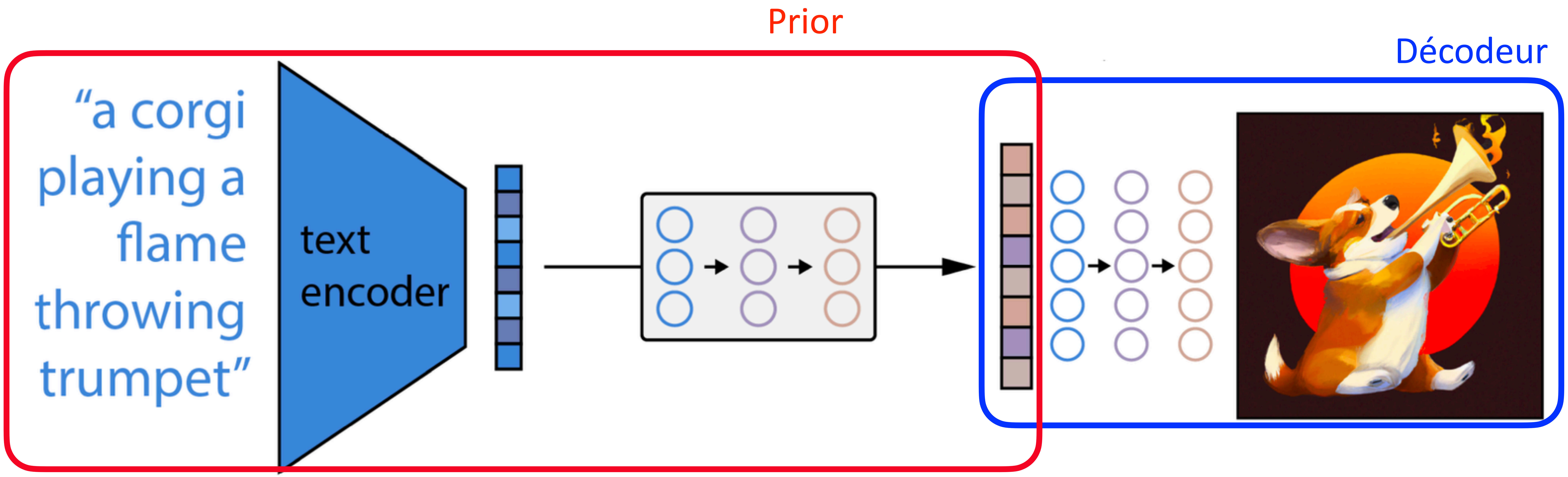
1. Contrastive Language-Image Pre-training (CLIP)
2. Generation of an image using a diffusion model
3. Learn the latent representations of text and images

DALL-E - Wrap-it up!



1. Contrastive Language-Image Pre-training (CLIP)
2. Generation of an image using a diffusion model
3. Learn the latent representations of text and images
4. Wrap-it up!

DALL-E - Wrap-it up!



Idea:

- Given (*x*, *y*) a tuple of an image *x* and text *y*.
- Given the representation of an image z.
- The distribution of the image given the text is:

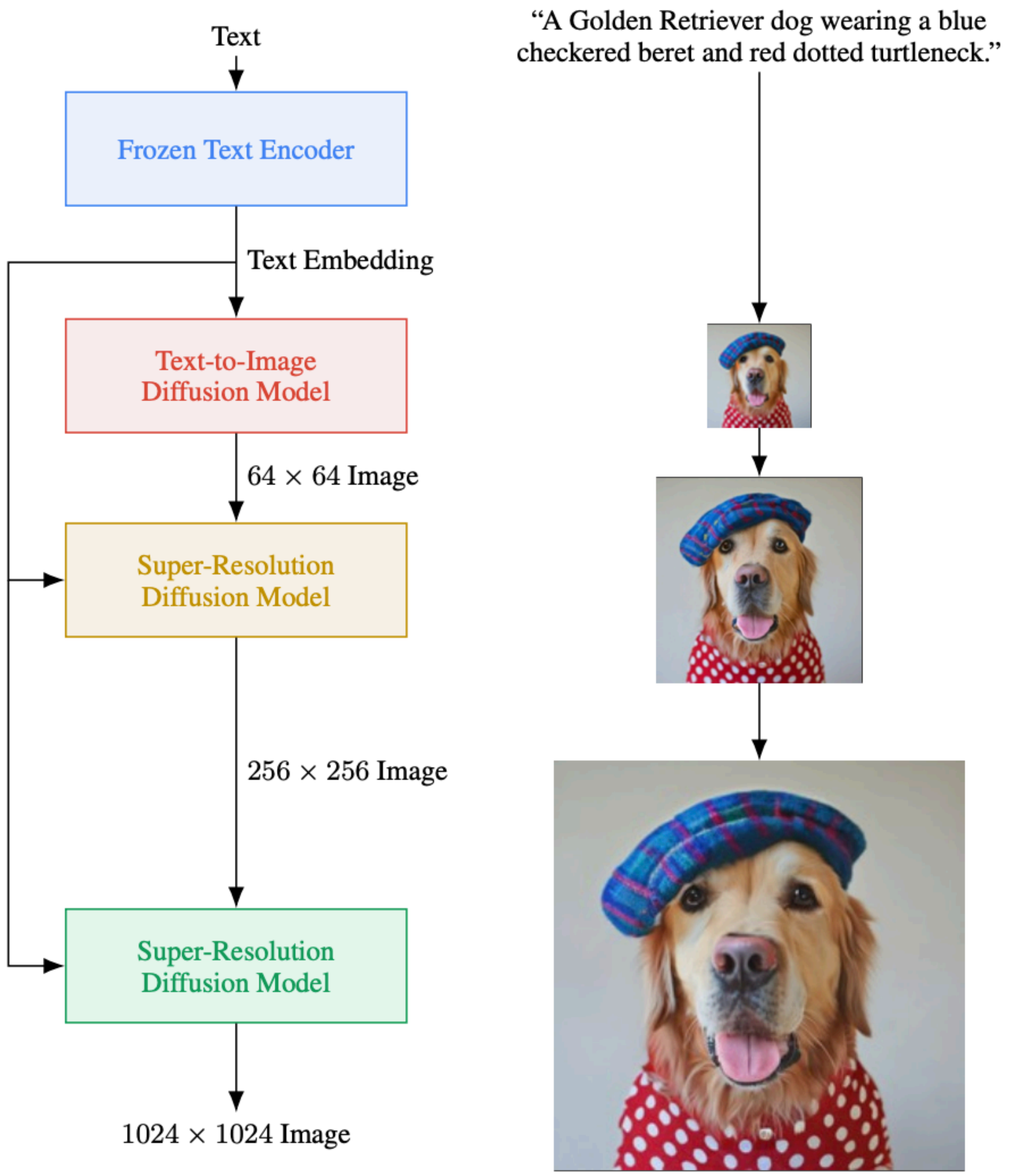
$$P(x|y) = P(x, z_i|y) = P(x|z_i, y)P(z_i|y)$$

Imagen

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

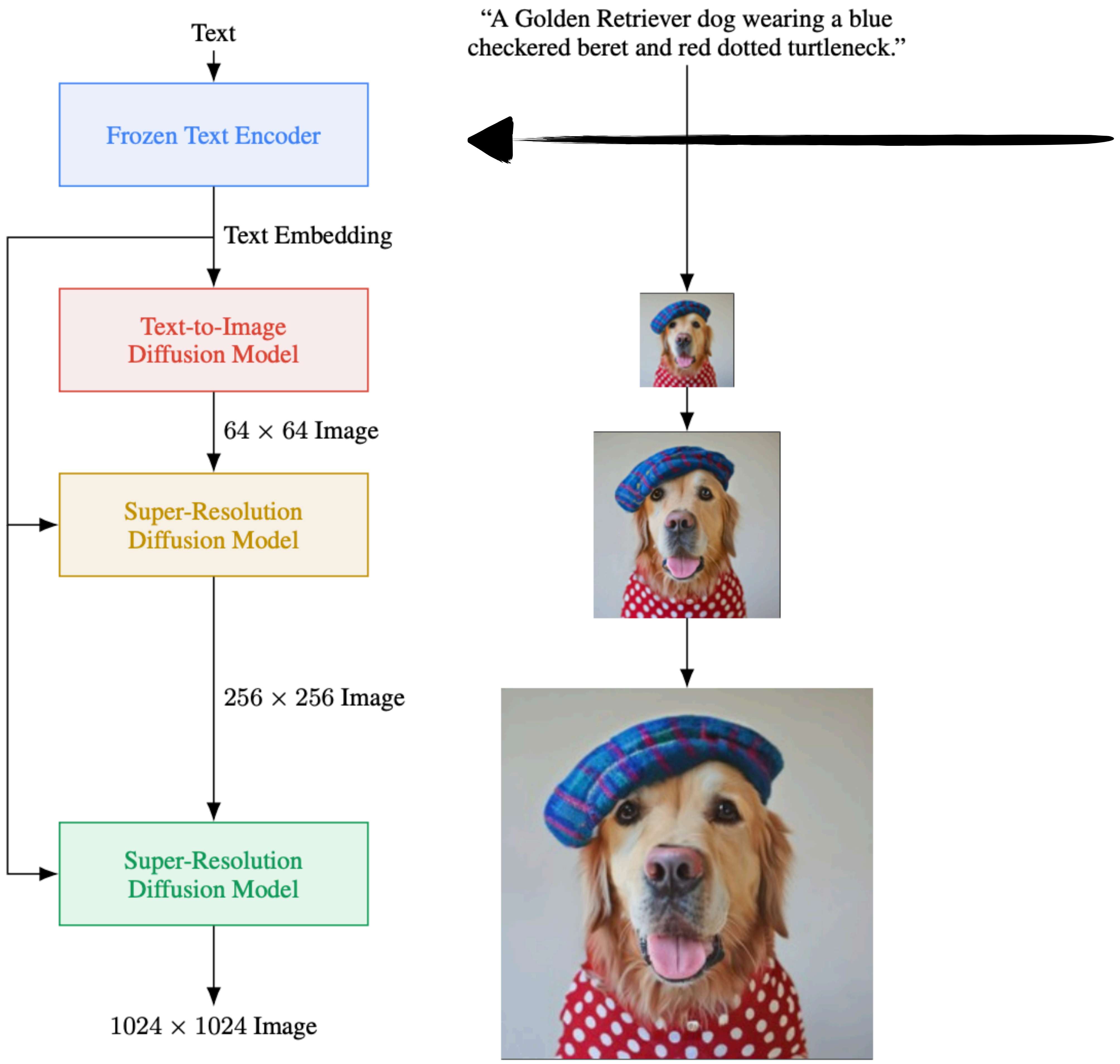
https://imagen.research.google/

Architecture



https://arxiv.org/pdf/2205.11487

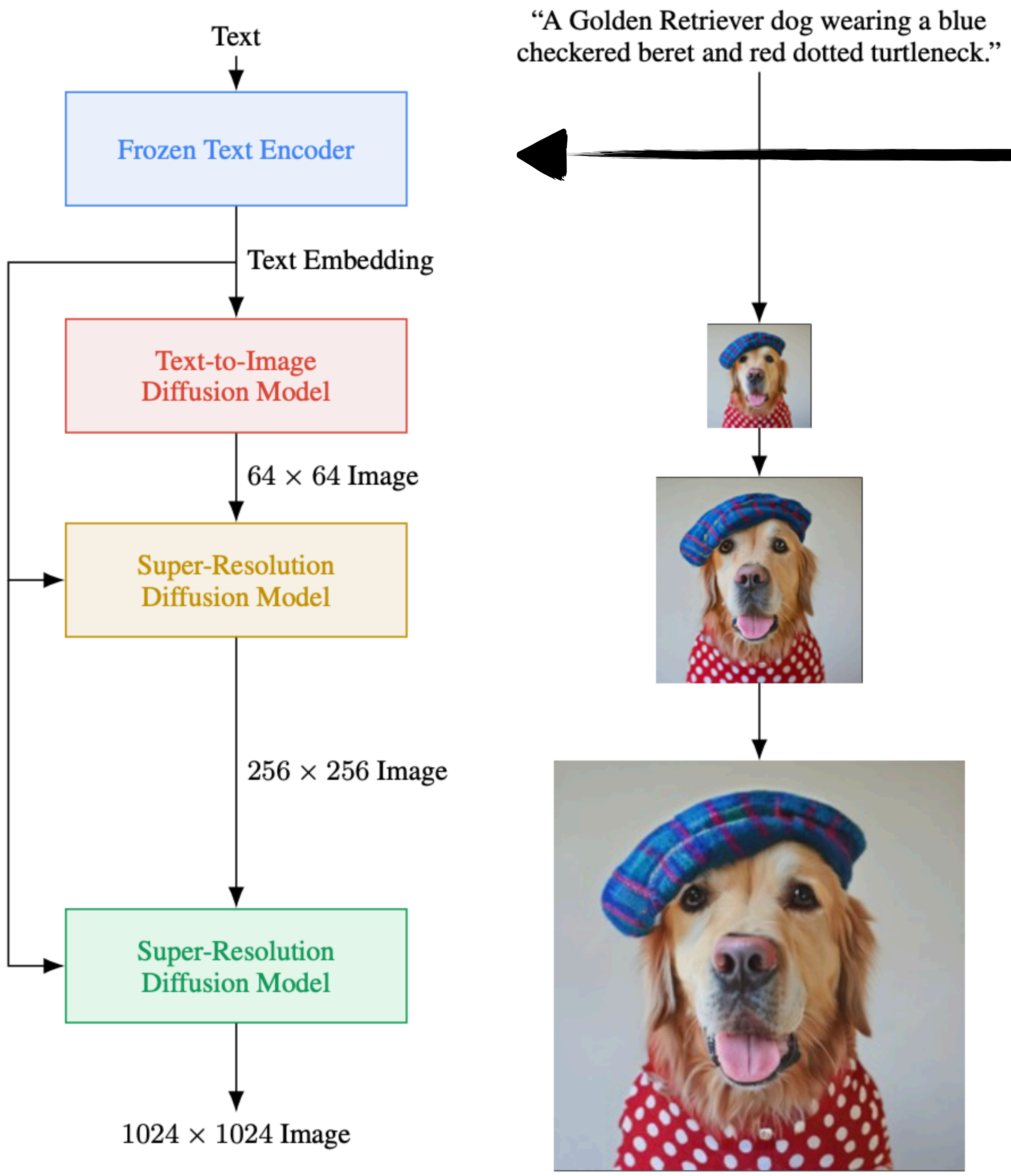
Architecture



- Uses a fixed and pertained encoder

https://arxiv.org/pdf/2205.11487

Architecture

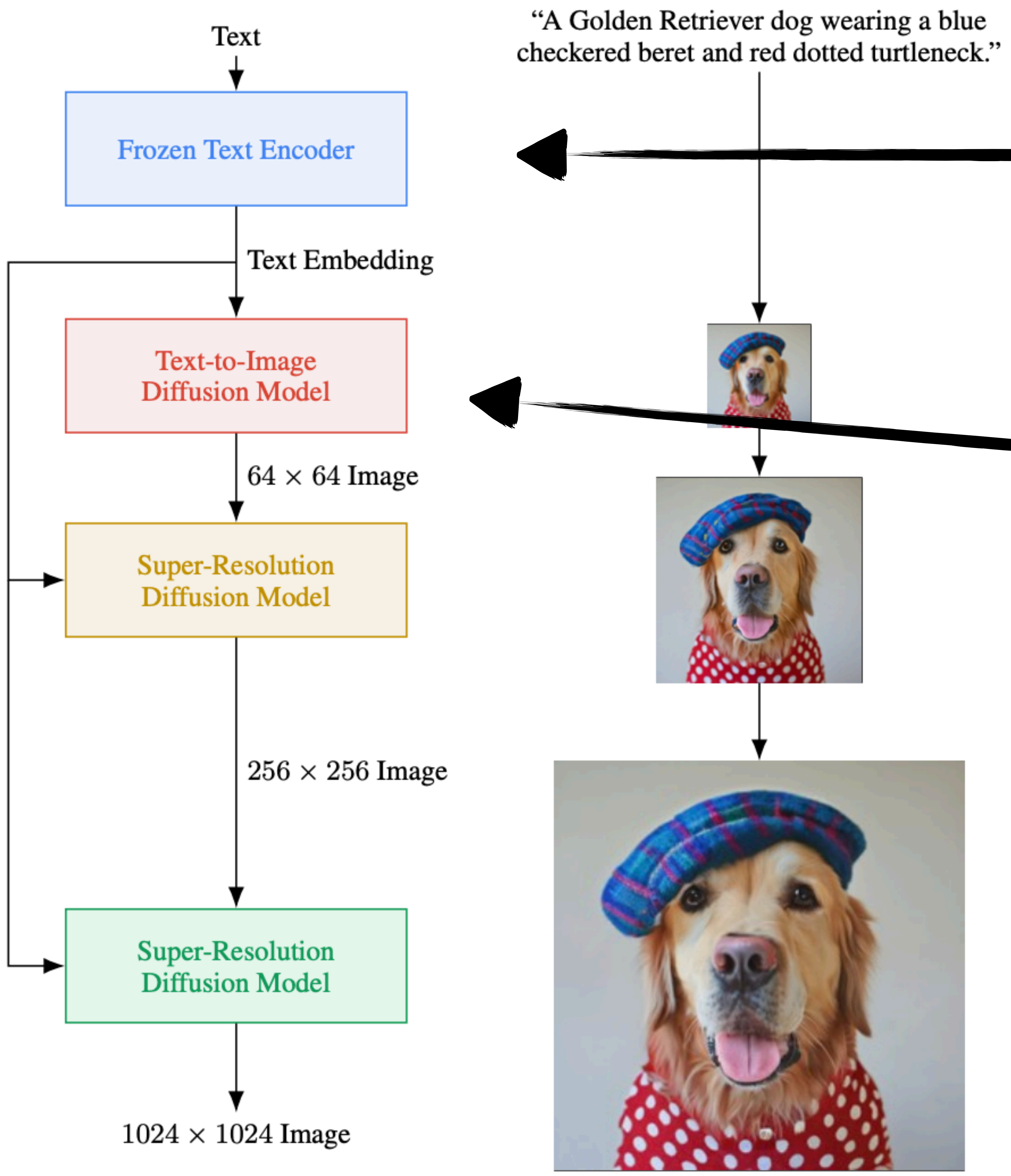


- Uses a fixed and pertained encoder

For example a transformer learned from a large-scale text dataset to predict the next word. **No image!**

https://arxiv.org/pdf/2205.11487

Architecture



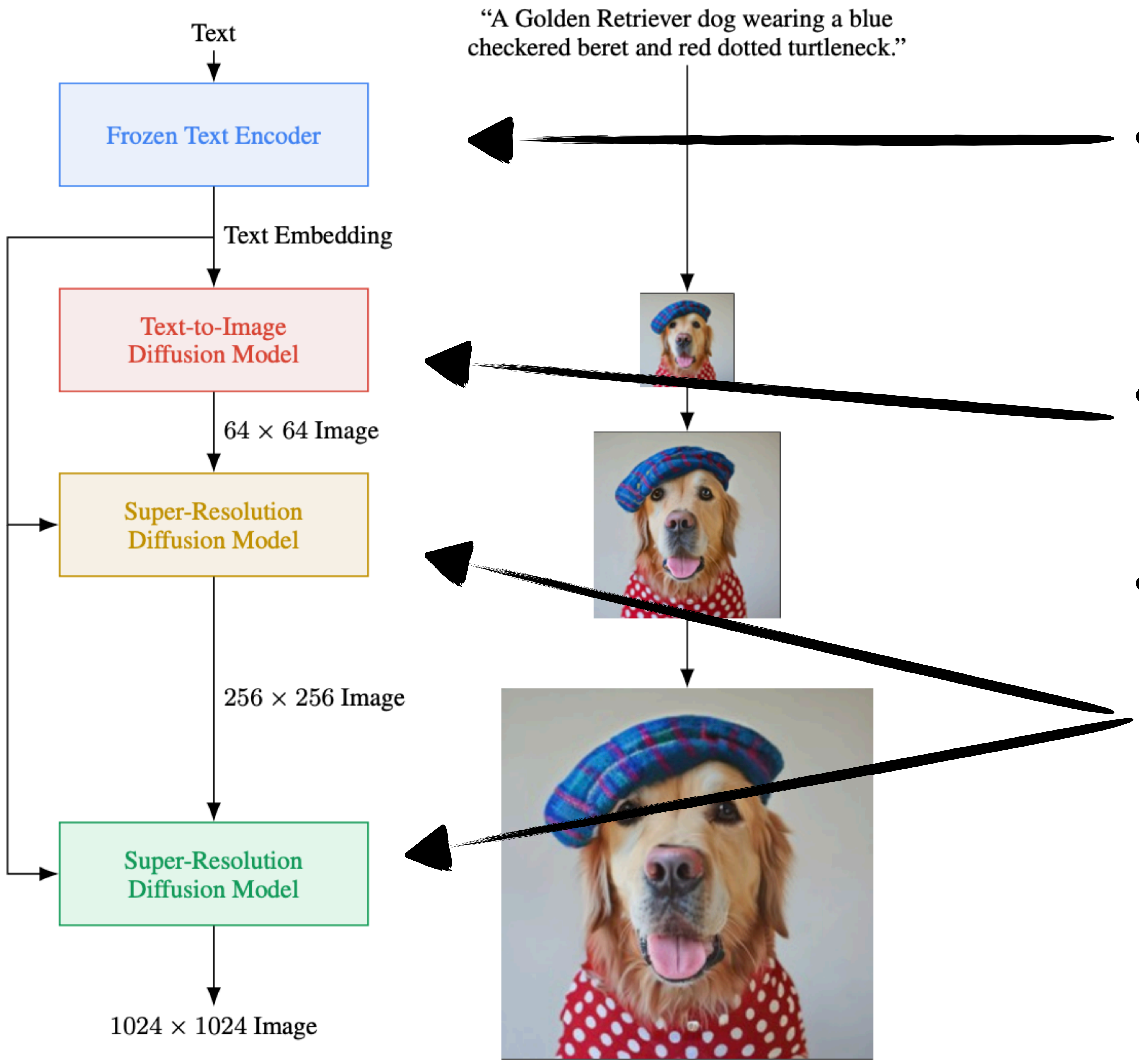
- Uses a fixed and pertained encoder

For example a transformer learned from a large-scale text dataset to predict the next word. **No image!**

- *Followed by a diffusion model to obtain a first image*

https://arxiv.org/pdf/2205.11487

Architecture



- Uses a fixed and pertained encoder

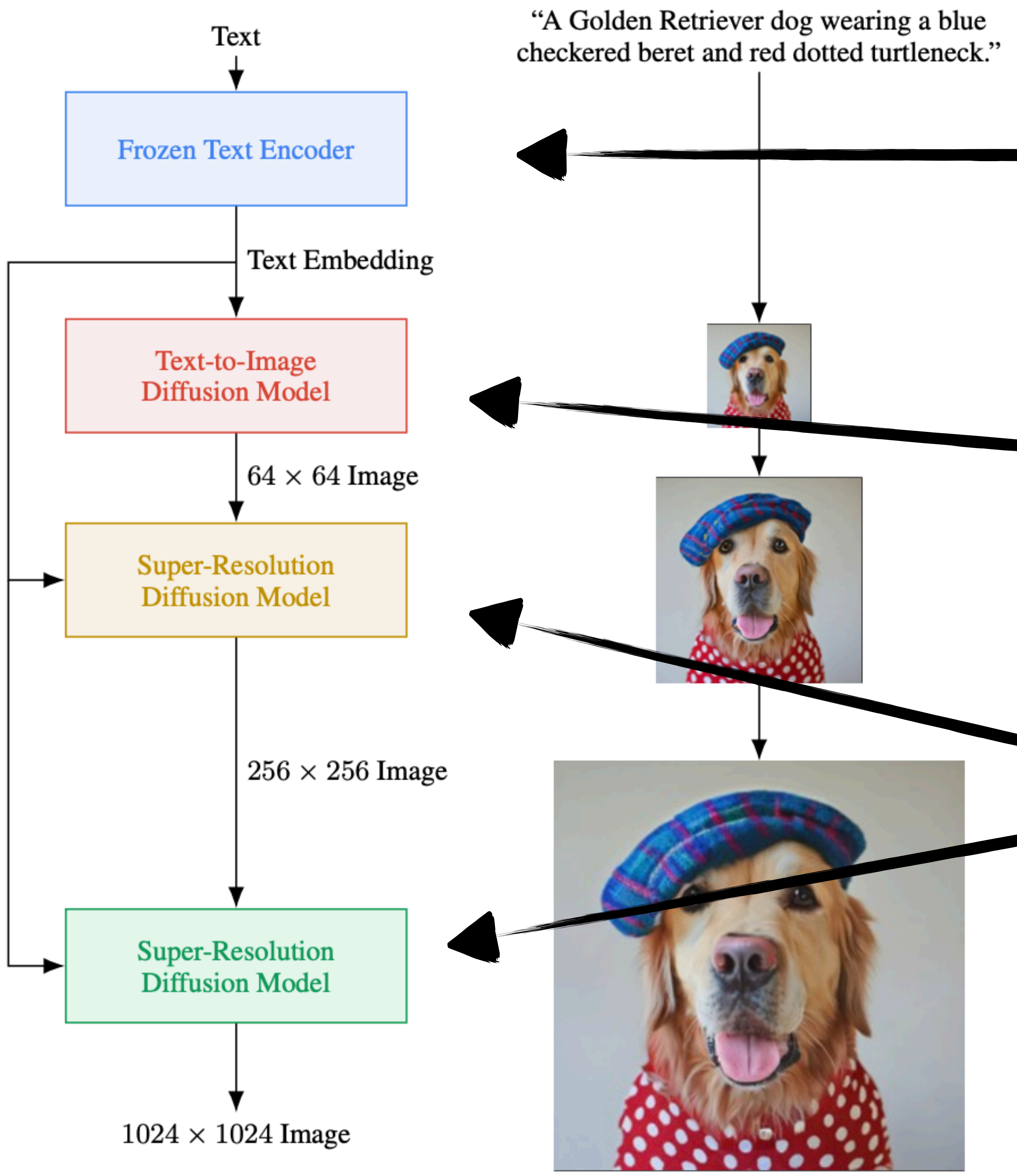
  For example a transformer learned from a large-scale text dataset to predict the next word. **No image!**

- *Followed by a diffusion model to obtain a first image*

- Followed by a few other diffusion models to obtain images of higher and higher resolution

https://arxiv.org/pdf/2205.11487

Architecture



- Uses a fixed and pertained encoder

  For example a transformer learned from a large-scale text dataset to predict the next word. **No image!**

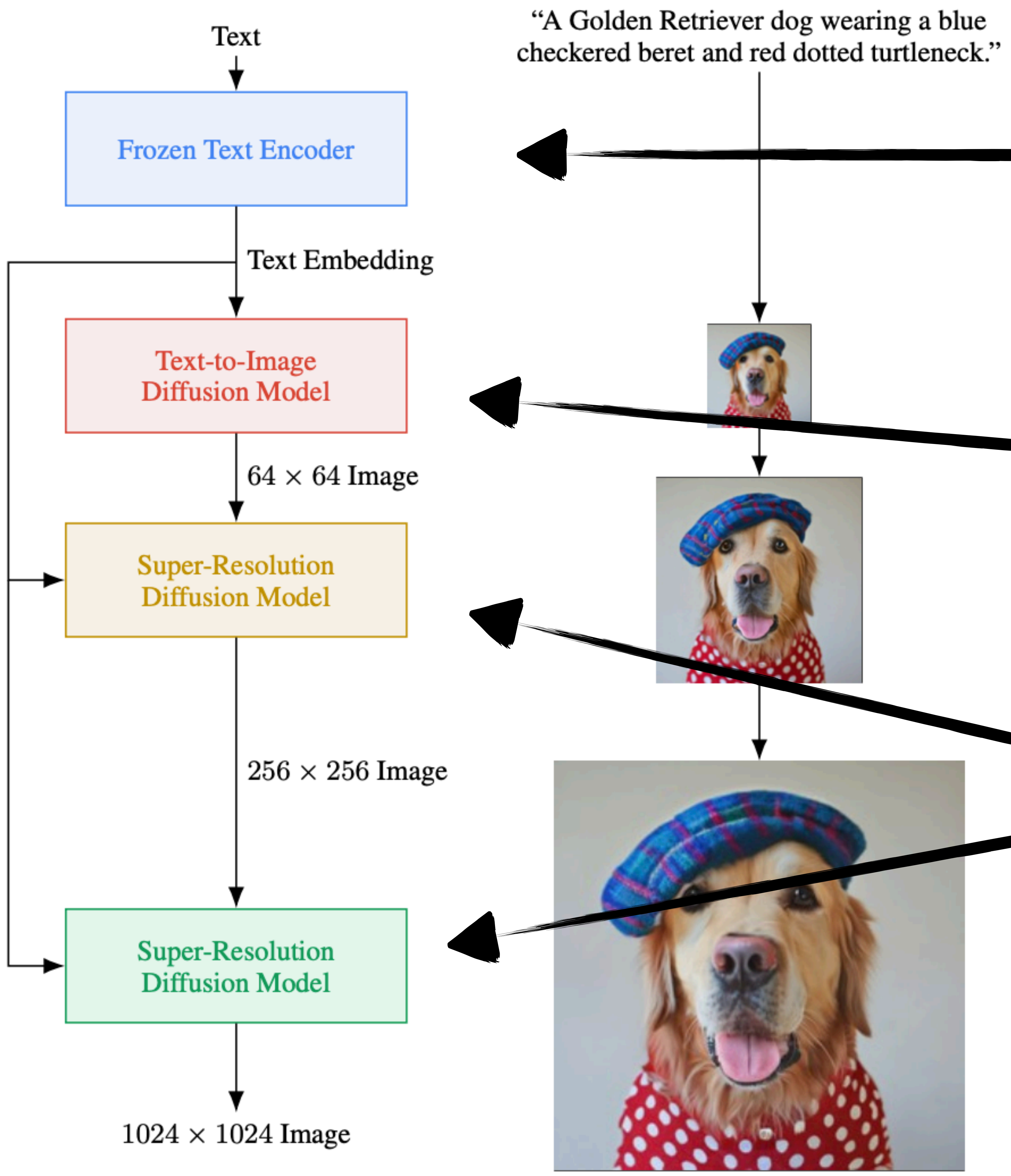
- *Followed by a diffusion model to obtain a first image*

- Followed by a few other diffusion models to obtain images of higher and higher resolution

  - The diffusion models use an attention mechanism on the text representation

https://arxiv.org/pdf/2205.11487

Architecture



- Uses a fixed and pertained encoder

For example a transformer learned from a large-scale text dataset to predict the next word. **No image!**

- *Followed by a diffusion model to obtain a first image*
- Followed by a few other diffusion models to obtain images of higher and higher resolution
  - The diffusion models use an attention mechanism on the text representation
- The diffusion moles are parametrized using a U-Net

https://arxiv.org/pdf/2205.11487

« Classifier-free » guidance

- The diffusion model is trained using two objectives

  1. Generate images from the text

  2. (Also) Generate images

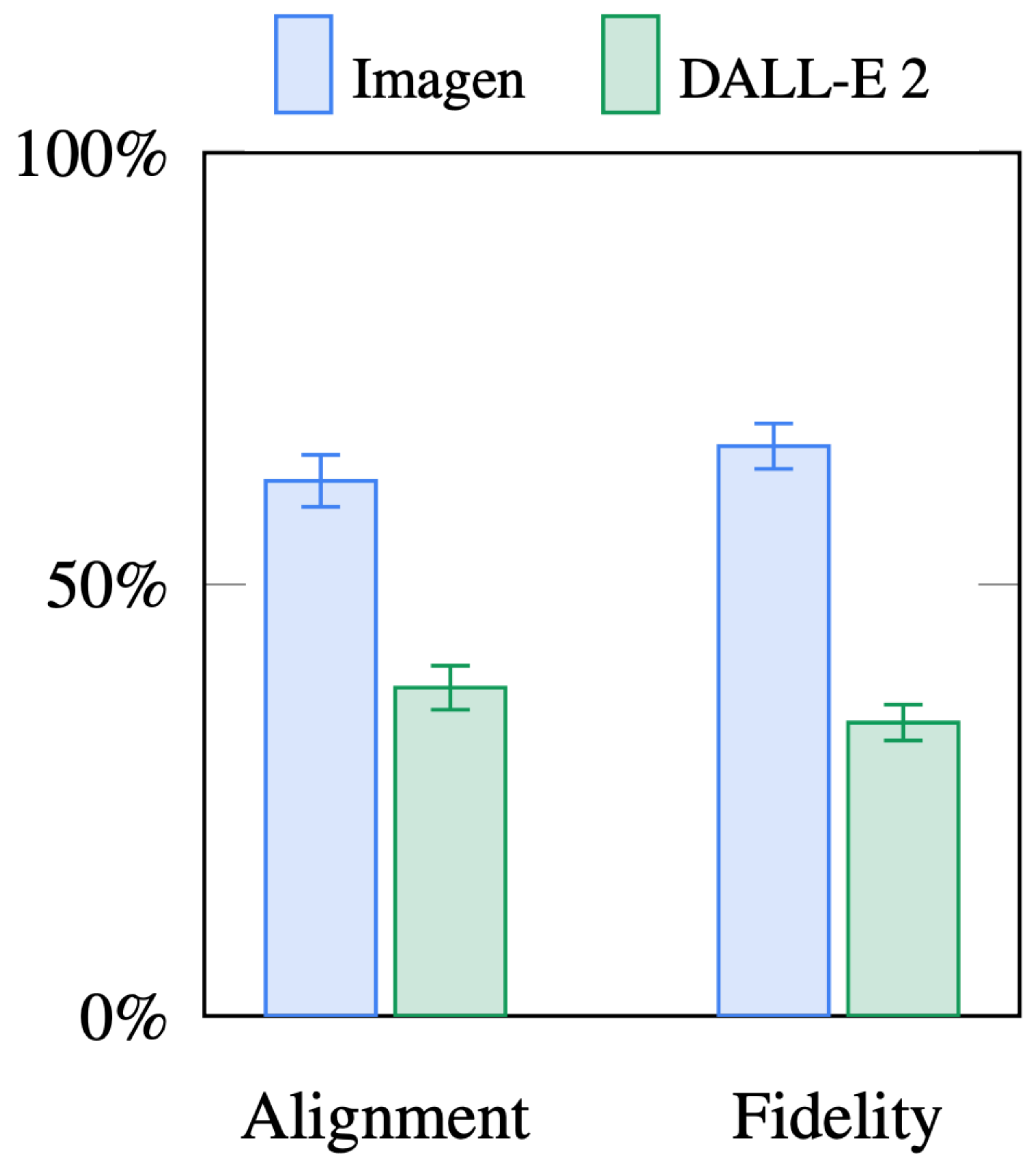
     - This allows to obtain high-quality images (1) that are diversified (2)

- Imagen proposes a method to ensure pixels don't saturate during diffusion (somewhat similar problem to clipping in RNNs)

Avant Imagen

Imagen

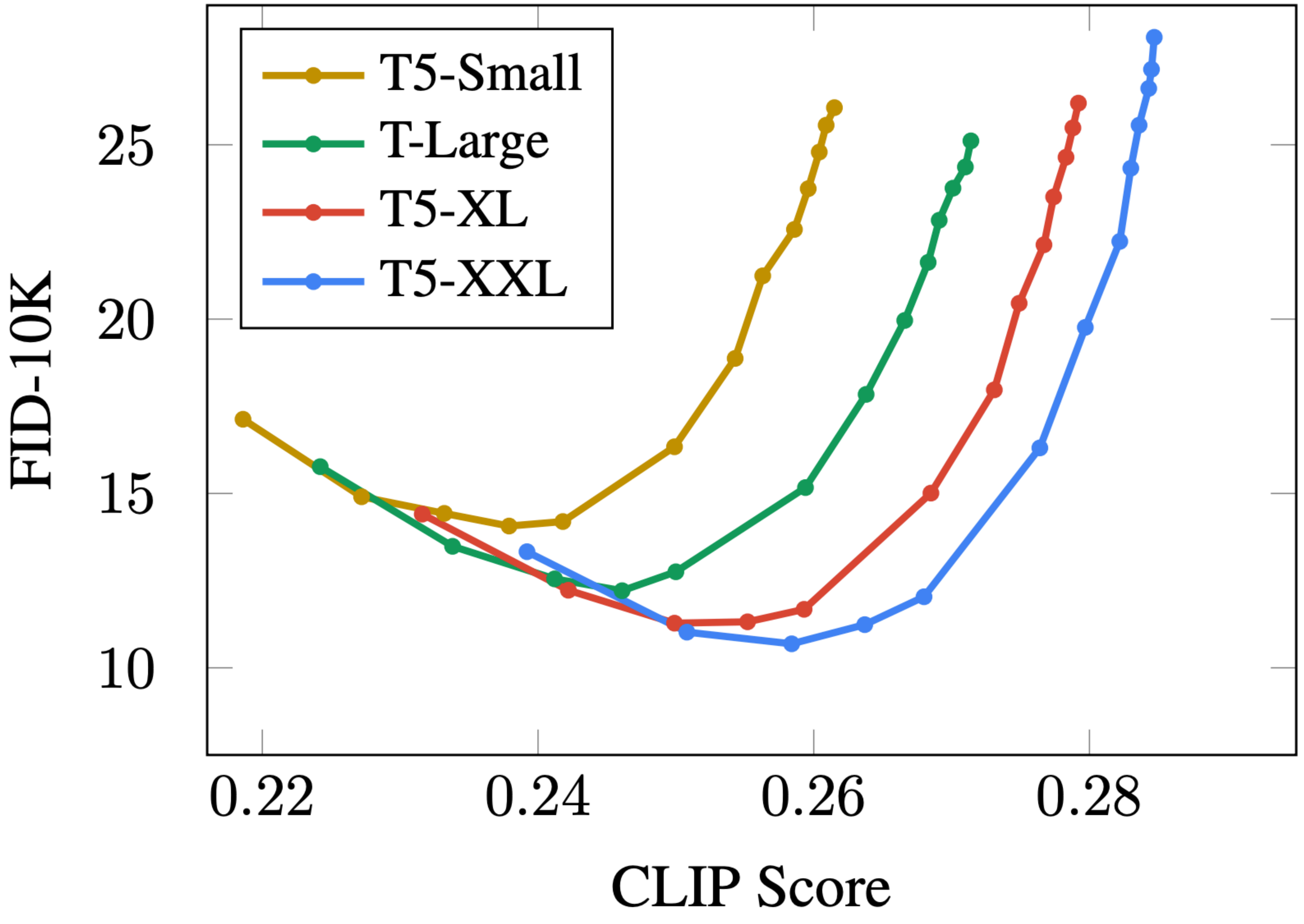
Compared to Dall-E 2



- Better empirical performance (according to a human study)
  - “Alignment” -> “Does the caption accurately describe the above image”
  - “Fidelity” -> “Which image is more photorealistic”
- Users a simpler architecture (no CLIP)

The size of the text encoder is an important hyper-ammeter

Diversity of Images

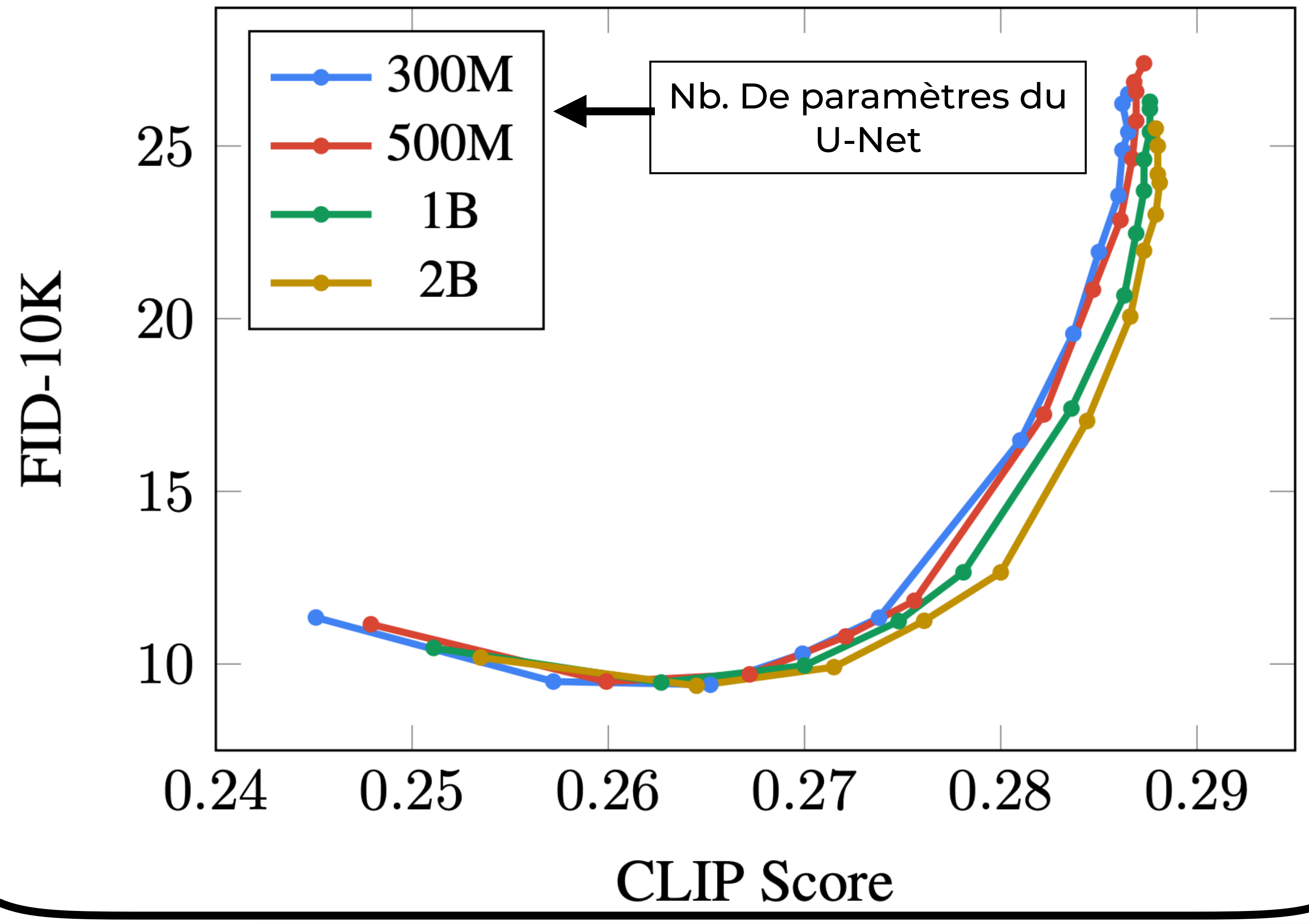


Alignment between images and the text

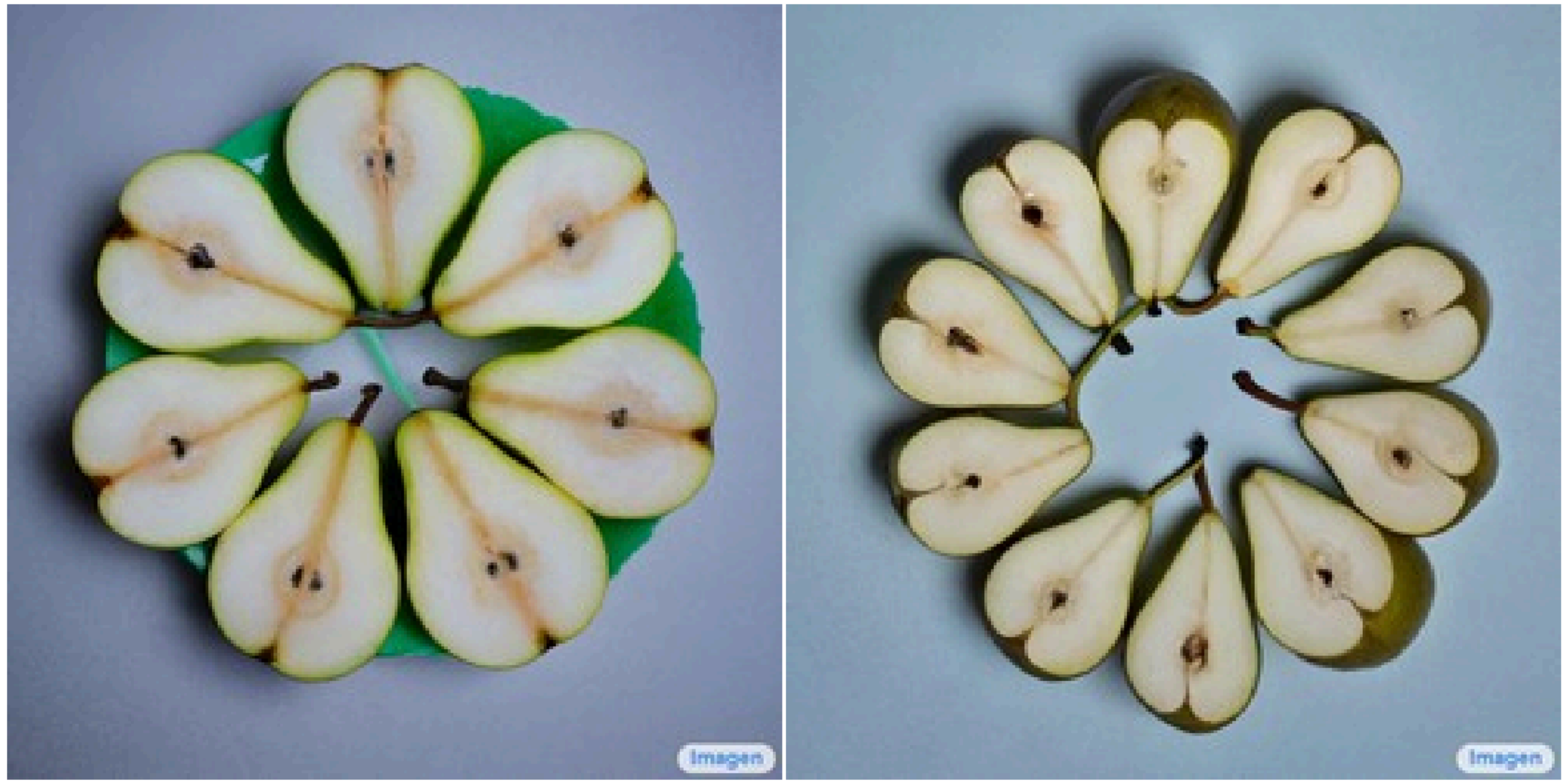
T5 XXL (2.6B)

- Trans. Encodeur-Decodeur
  - Uses only the encoder

Comparatively, the size of the image generation model is less important



Still far from perfect...



A pear cut into seven pieces arranged in a ring.

One cat and two dogs sitting on the grass.

- E.g., operations that require counting and logic remain difficult