Deep Learning (1470)

Randall Balestriero

Class 15: Variational Autoencoders

Recap!

Seq2Seq Translation with Autoregressive Decoding
Strict left-to-right generation: cannot peek at future tokens

ENCODER
DECODER
Autoregressive: $output_t \rightarrow input_{t+1}$
Context
c
Le
chat
s'assoit
<EOS>
h_1^{dec}
h_2^{dec}
h_3^{dec}
h_4^{dec}
<SOS>
Le
chat
s'assoit
Source: Vision
Target: French
Process ALL source first
THEN generate target sequentially

Seq2Seq Translation with Autoregressive Decoding
Strict left-to-right generation: cannot peek at future tokens

ENCODER

DECODER

Autoregressive: $output_t \rightarrow input_{t+1}$

chat
s'assoit
<EOS>
h_2^{dec}
h_3^{dec}
h_4^{dec}
<SOS>
Le
chat
s'assoit

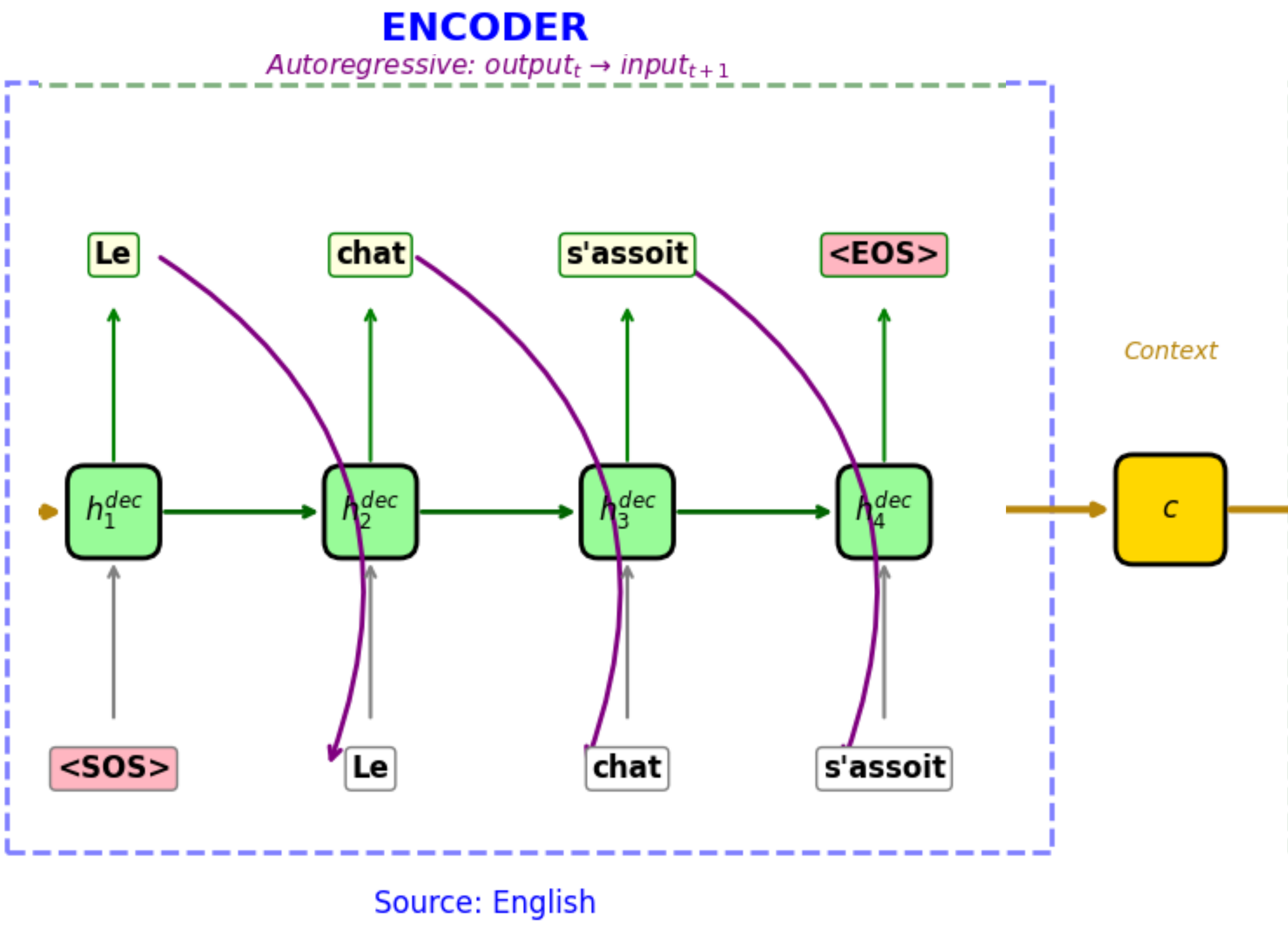
Source: Vision

Target: French

Process ALL source first

THEN generate target sequentially

Seq2Seq Translation with Autoregressive Decoding
Strict left-to-right generation: cannot peek at future tokens



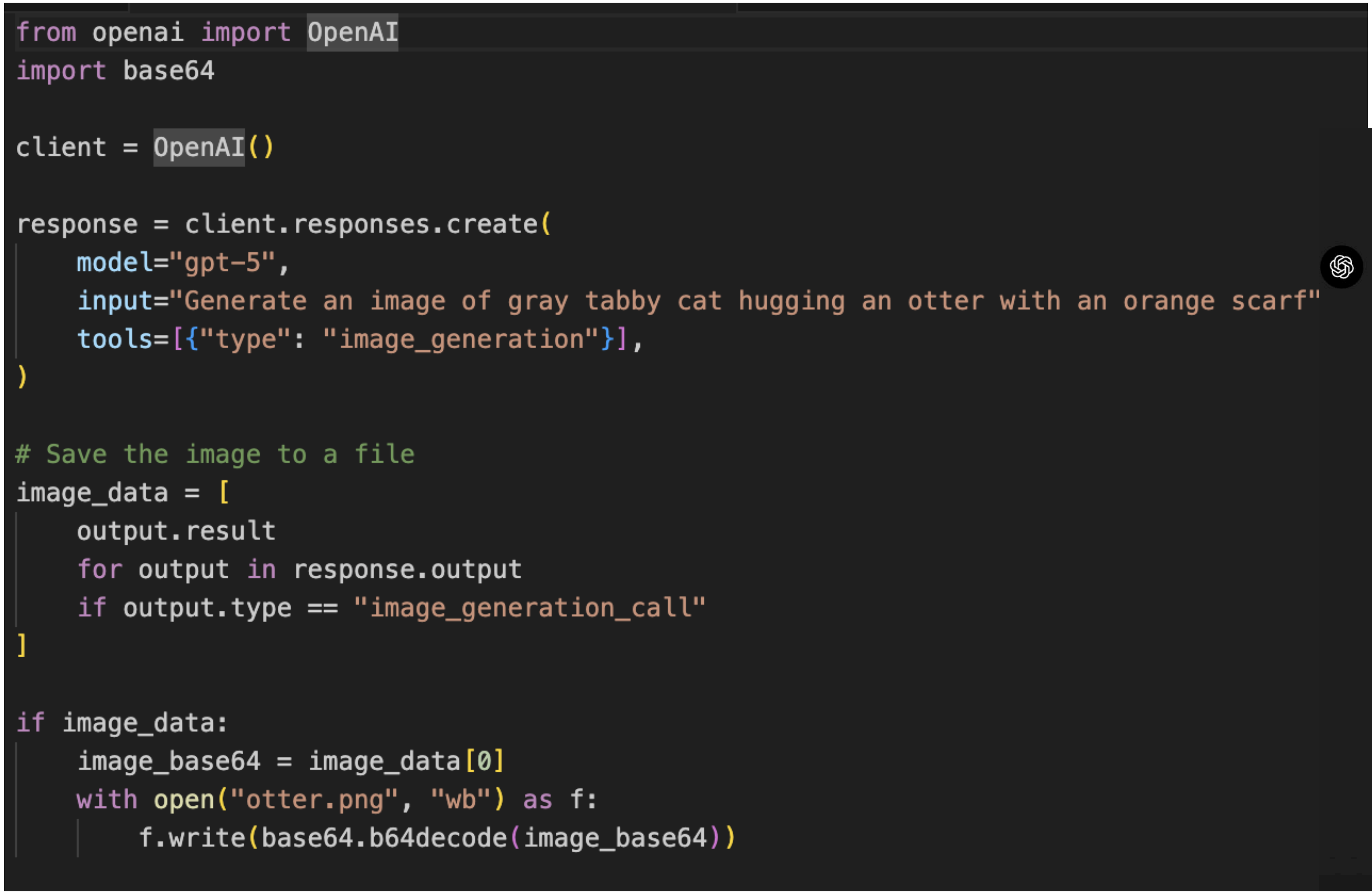
Process ALL source first

THEN generate target sequentially

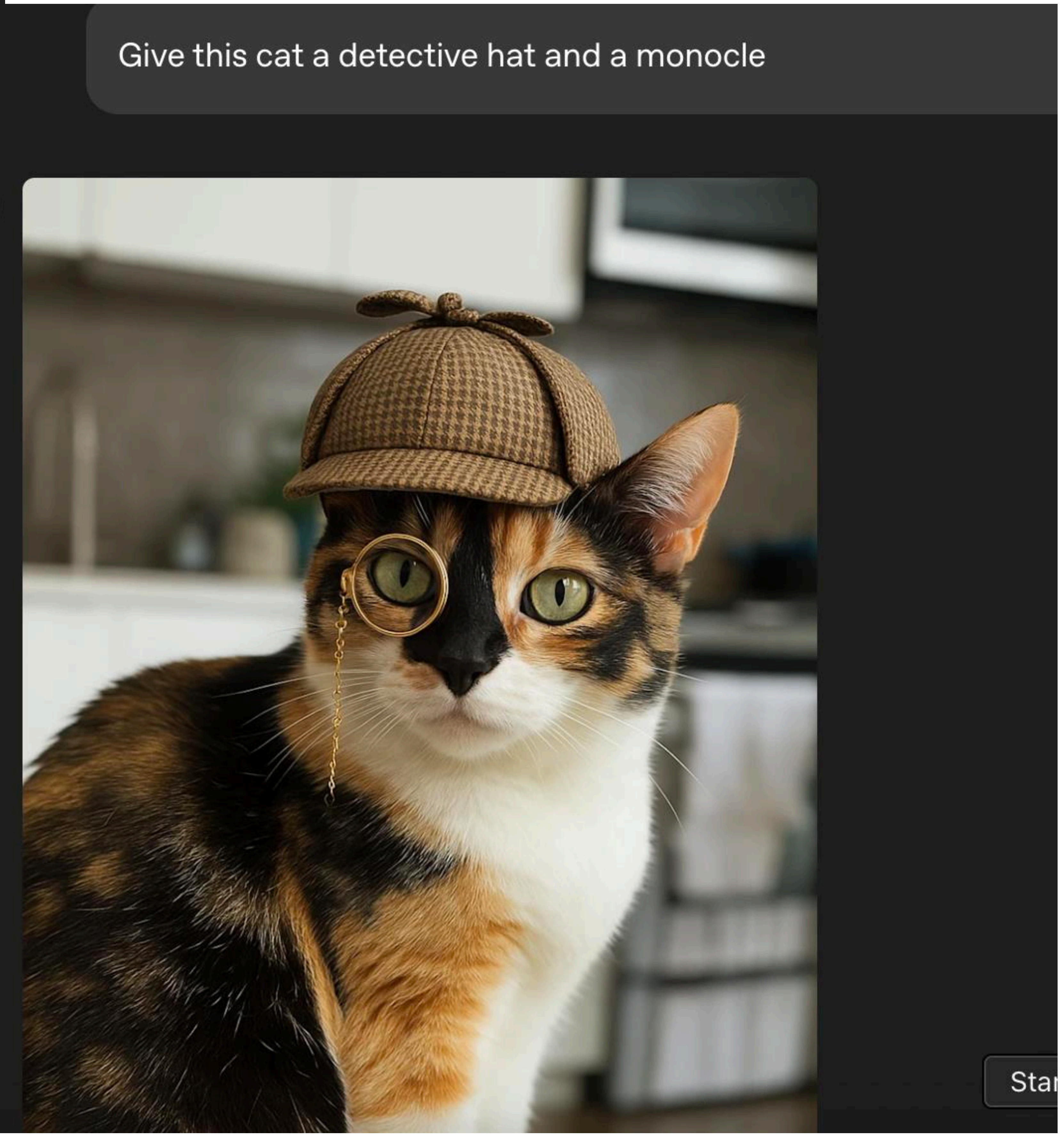
Image Capabilities in LLMs

How might we incorporate other capabilities into our LLMs?

1. Generate Images (call other model)
2. Take images as input and generate text

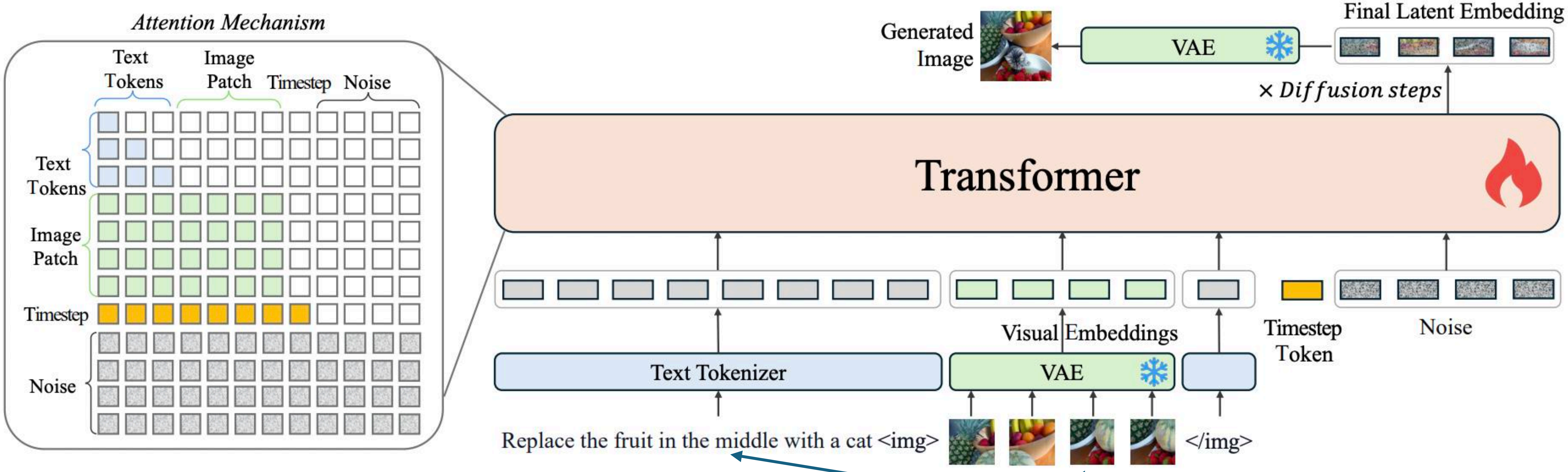


```python
from openai import OpenAI
import base64

client = OpenAI()

response = client.responses.create(
    model="gpt-5",
    input="Generate an image of gray tabby cat hugging an otter with an orange scarf",
    tools=[{"type": "image_generation"}],
)

# Save the image to a file
image_data = [
    output.result
    for output in response.output
    if output.type == "image_generation_call"
]

if image_data:
    image_base64 = image_data[0]
    with open("otter.png", "wb") as f:
        f.write(base64.b64decode(image_base64))
```



Give this cat a detective hat and a monocle

What *might* this look like?



Output image

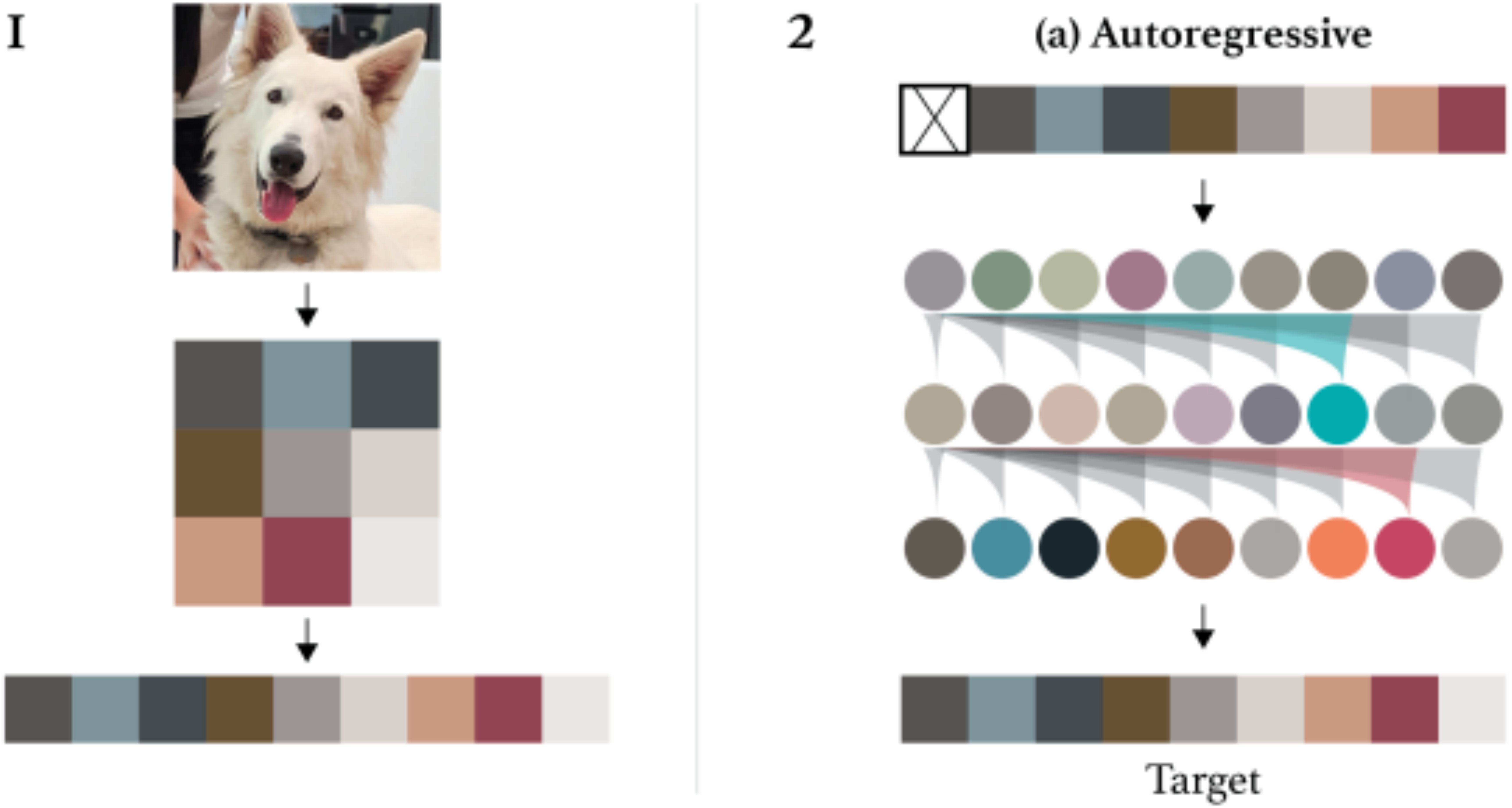
Output embedding vector

Take in multiple modes of input (text, images, etc.)
(Multi-modal learning)

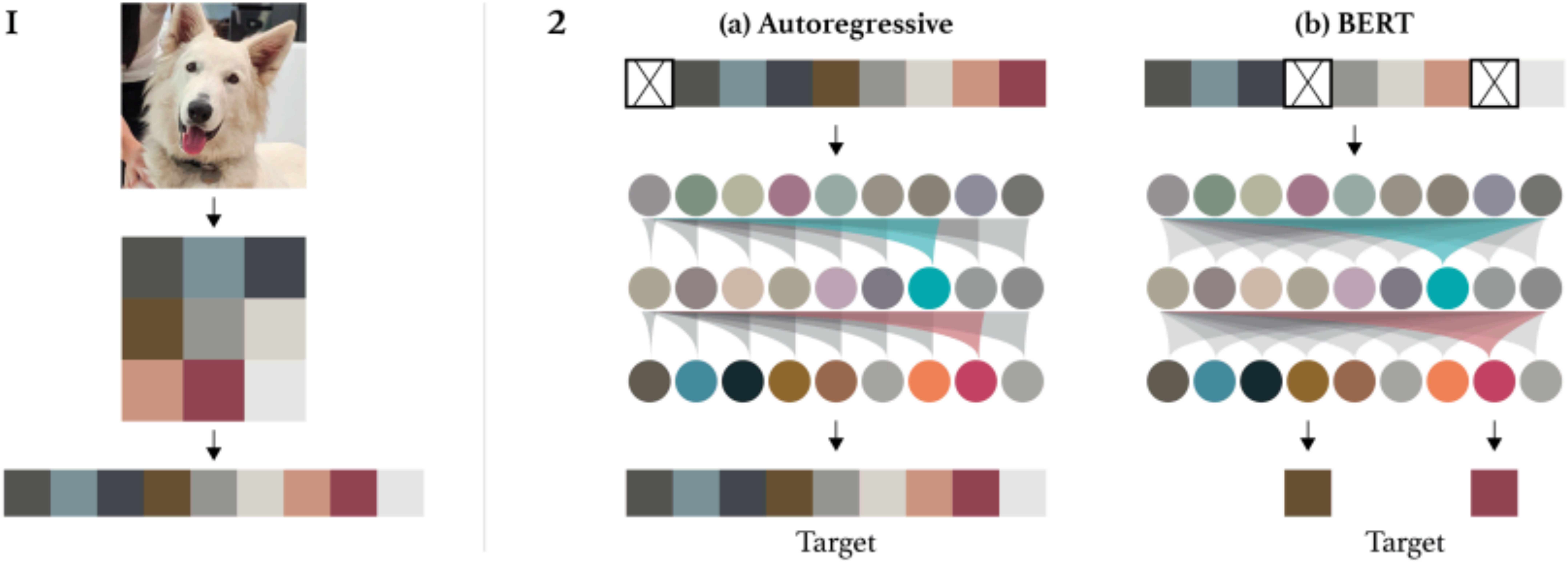
Source: OmniGen: Unified Image Generation: https://arxiv.org/pdf/2409.11340

Any idea? (from last class)

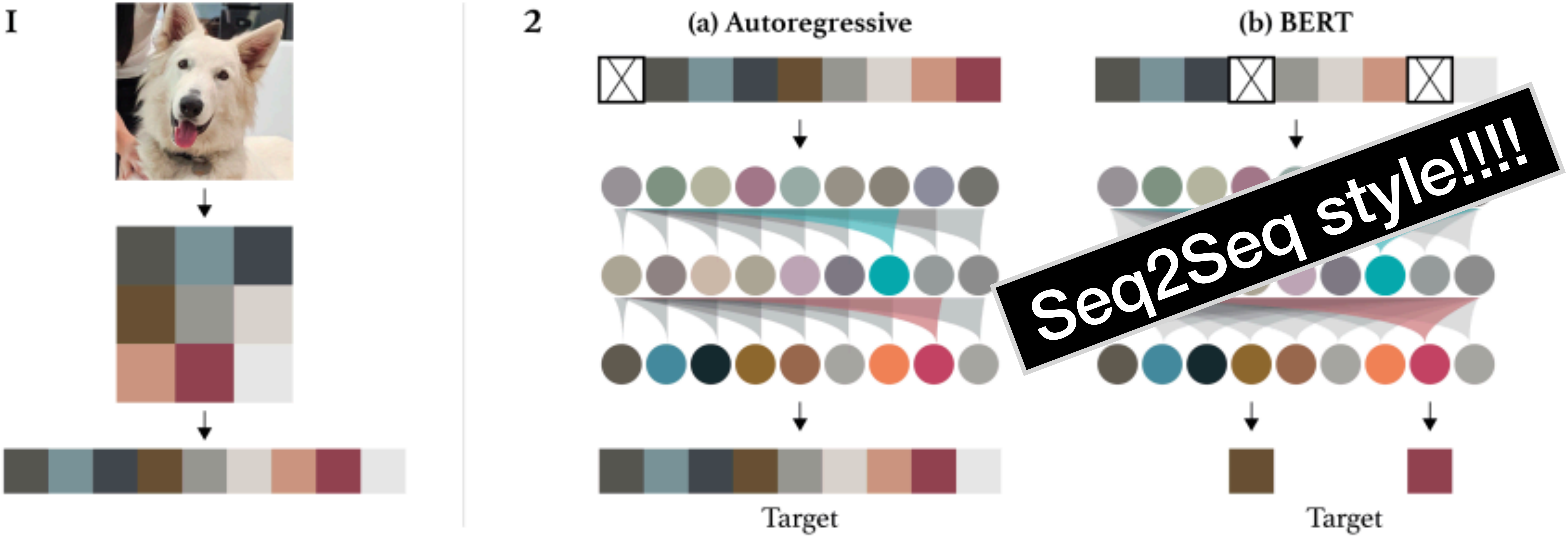
iGPT



iGPT

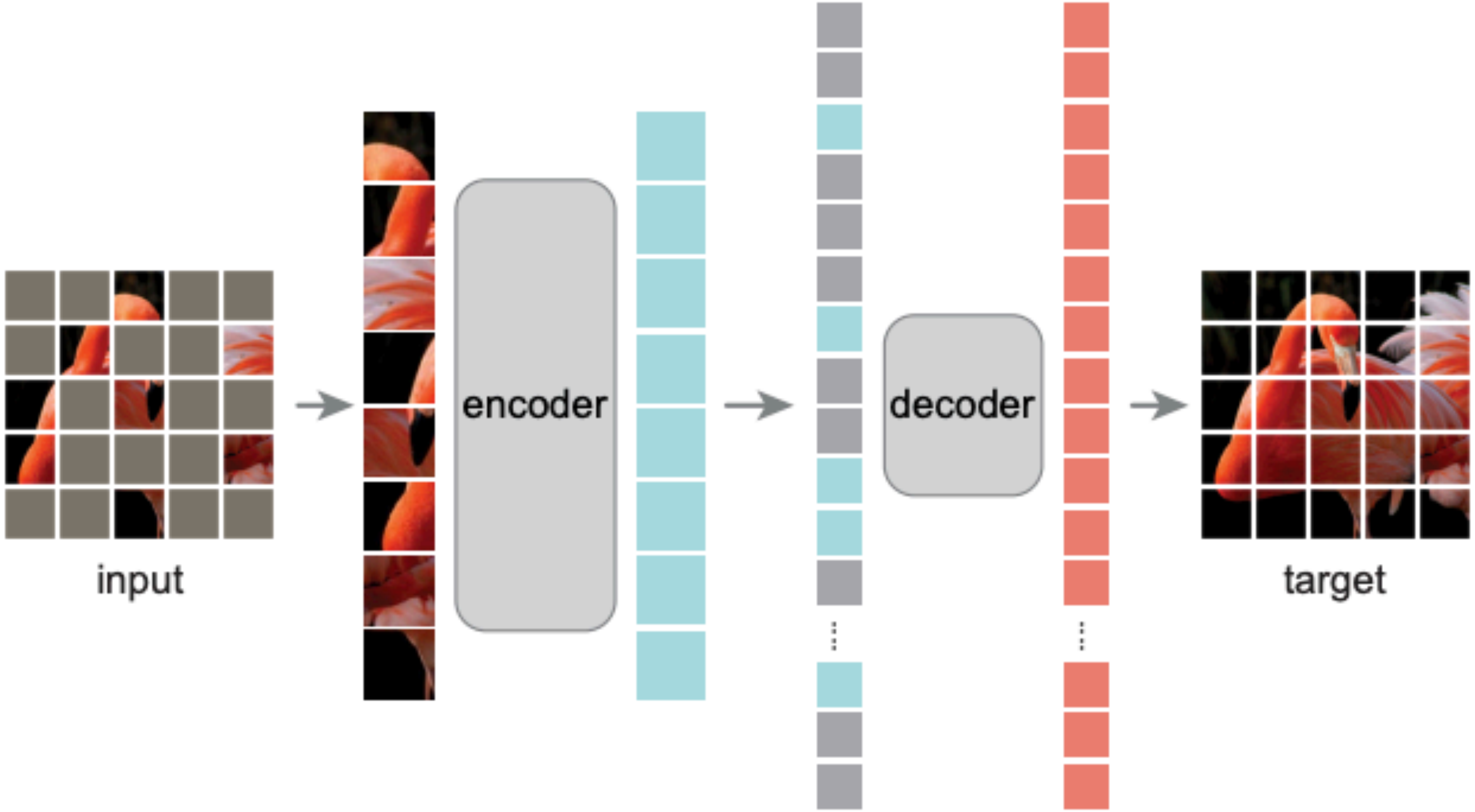


iGPT

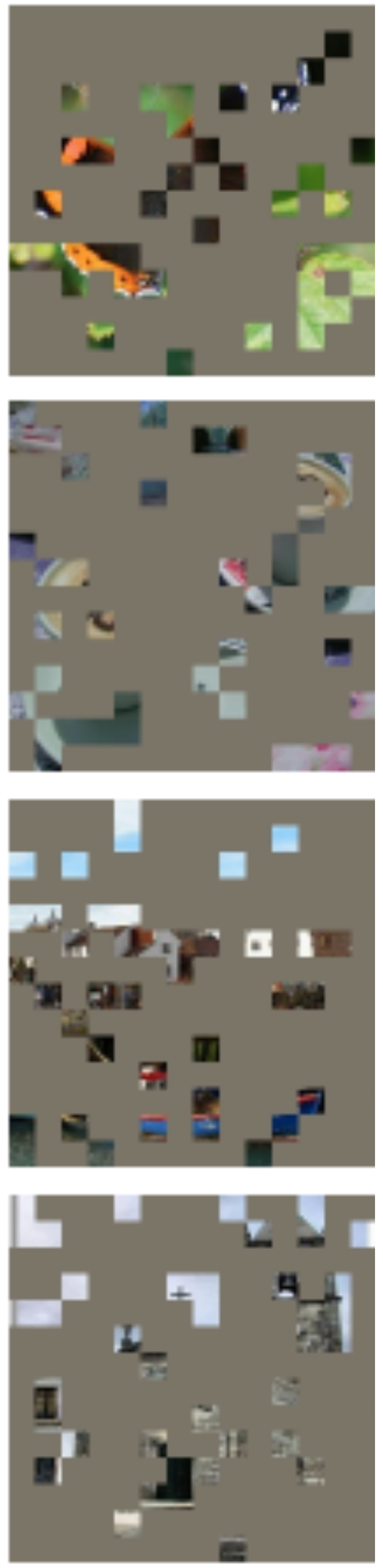


Seq2Seq style!!!!

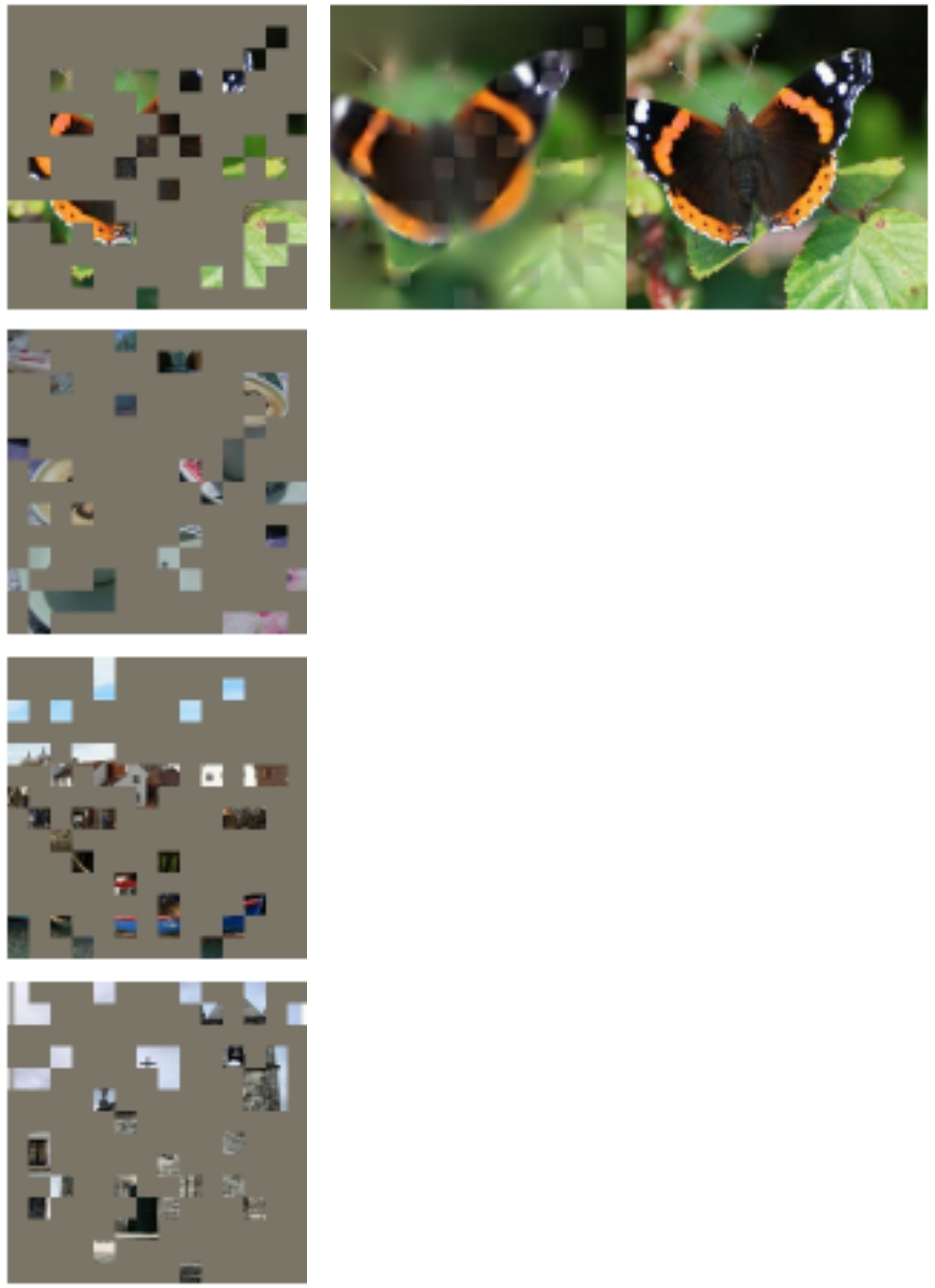
Masked Autoencoder



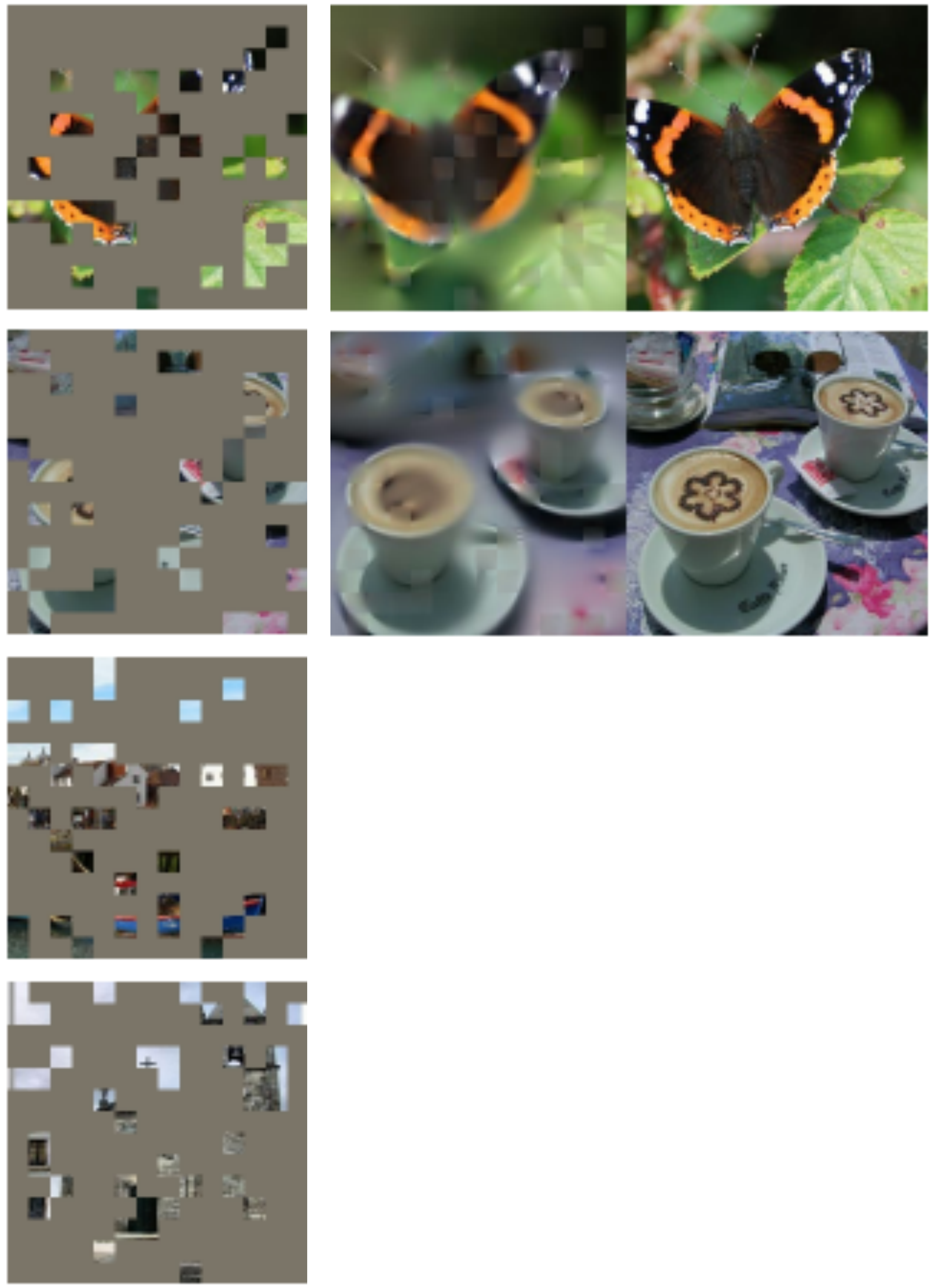
Masked Autoencoder



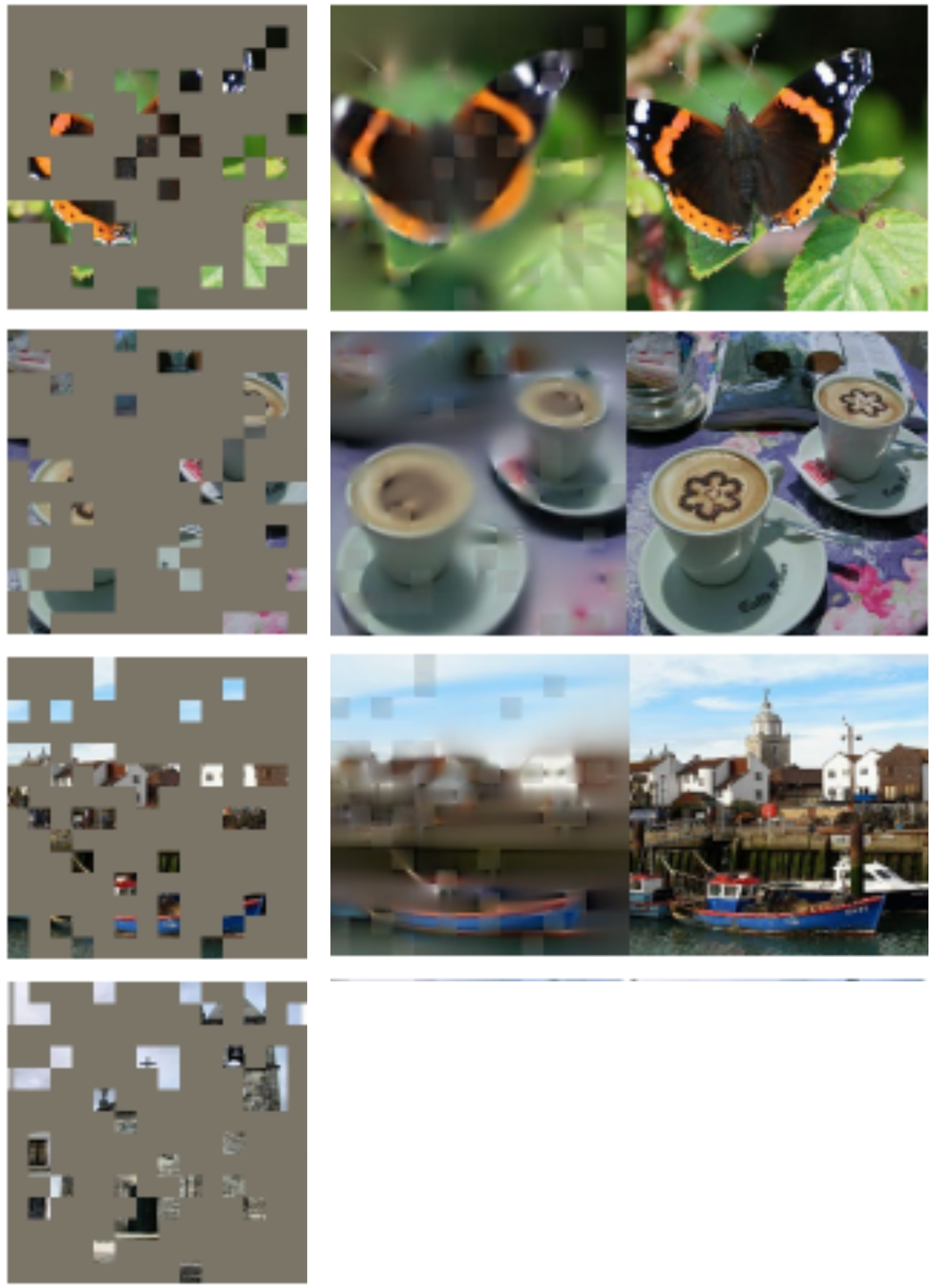
Masked Autoencoder



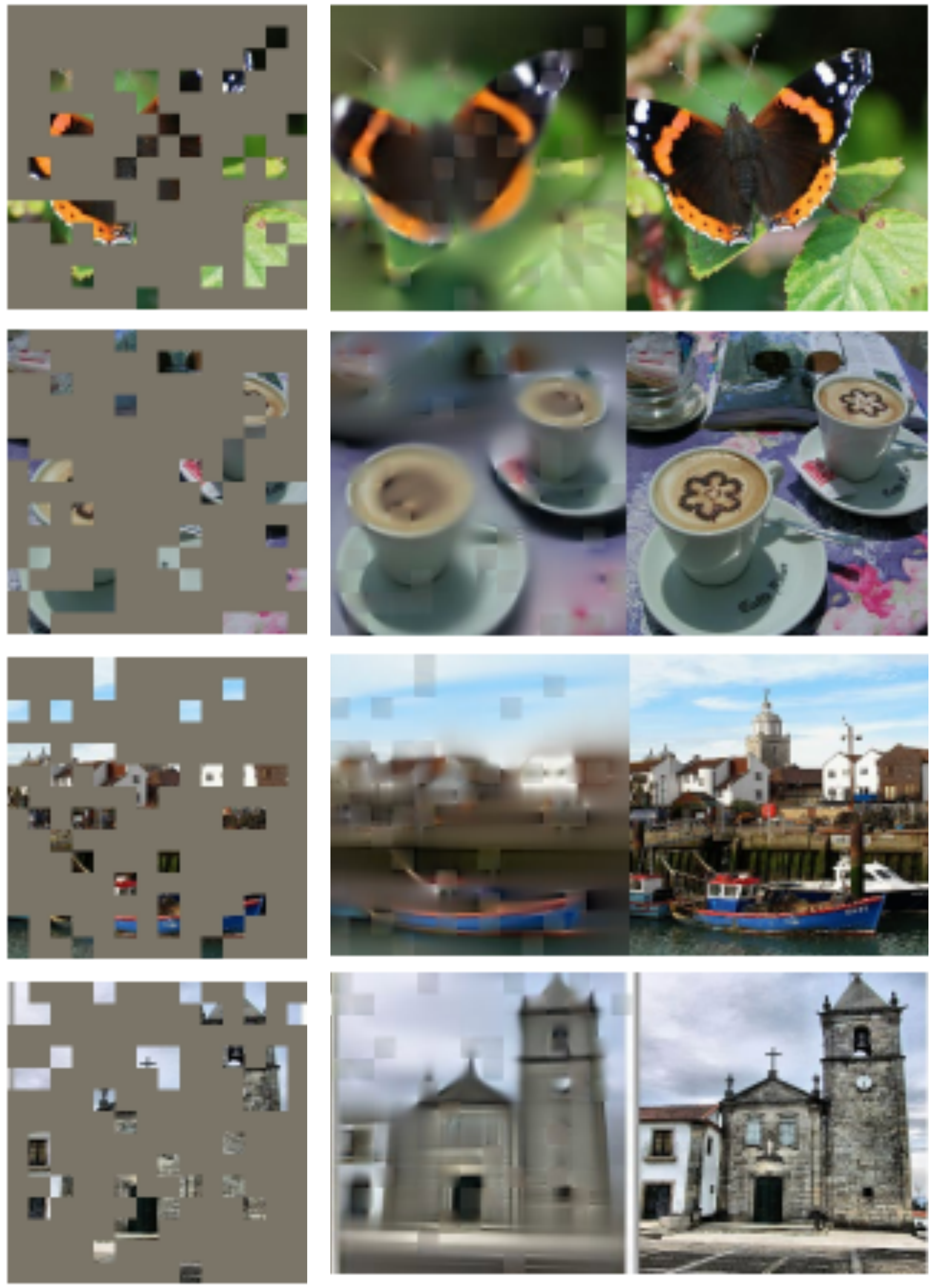
Masked Autoencoder



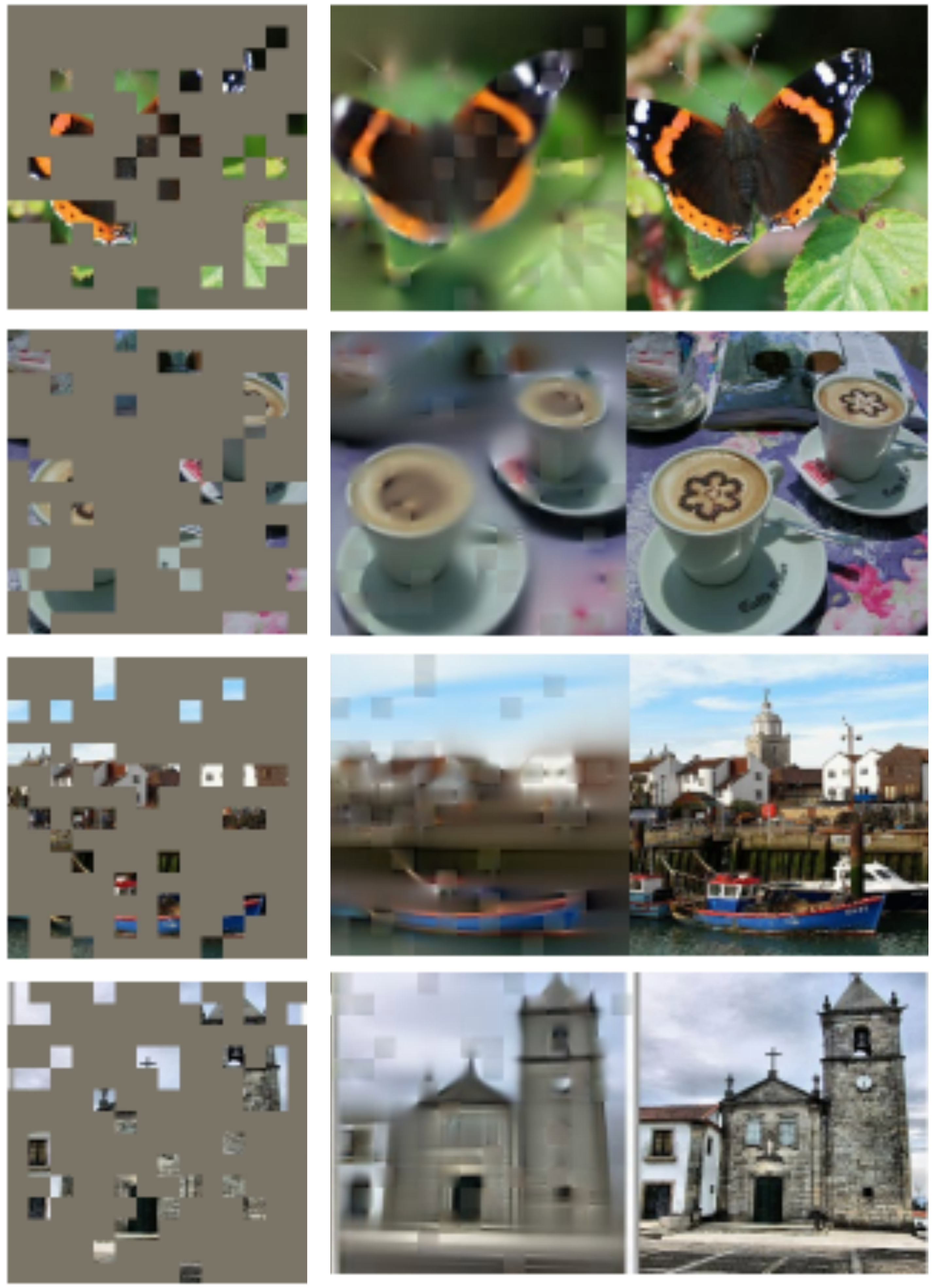
Masked Autoencoder



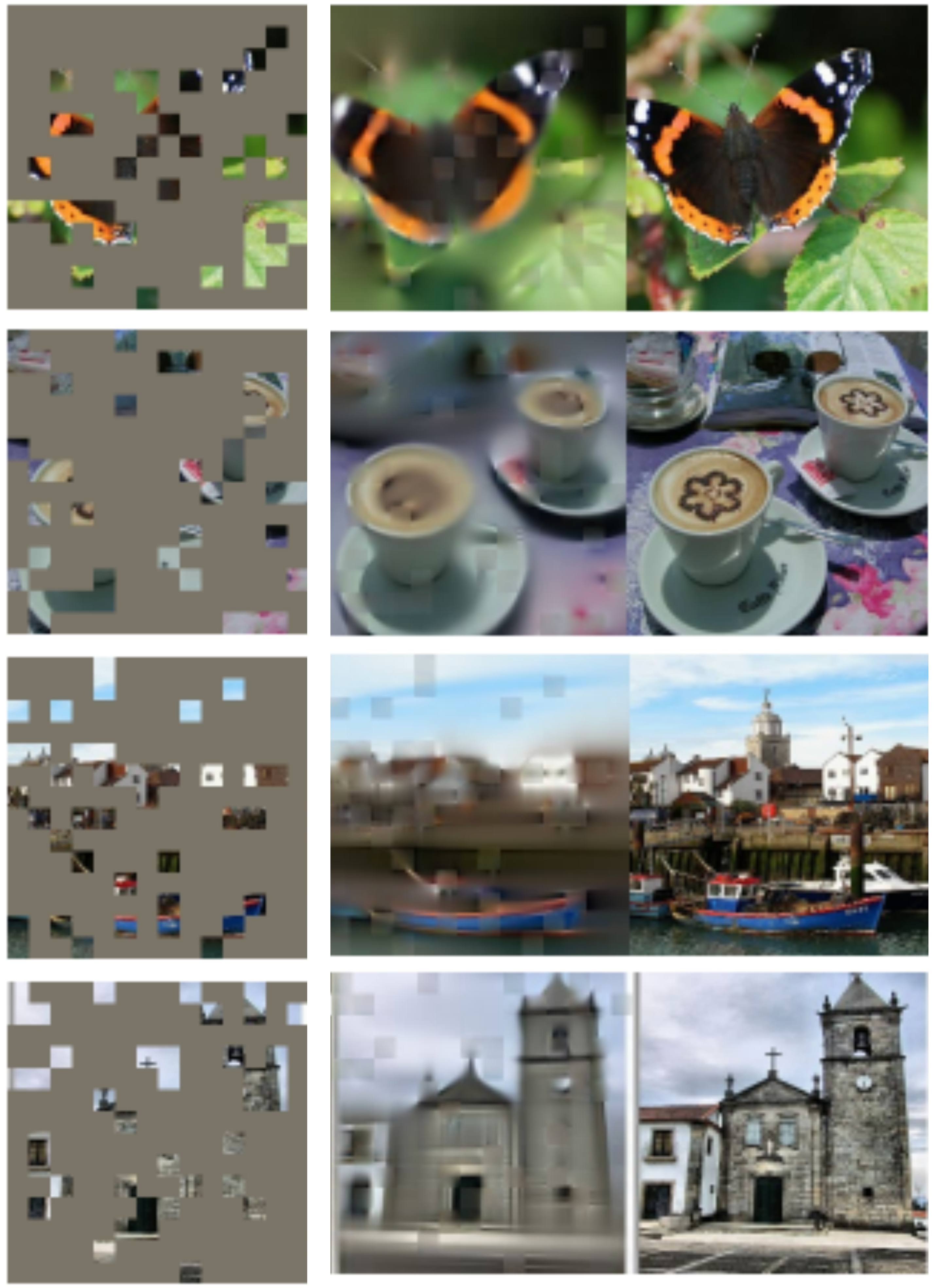
Masked Autoencoder



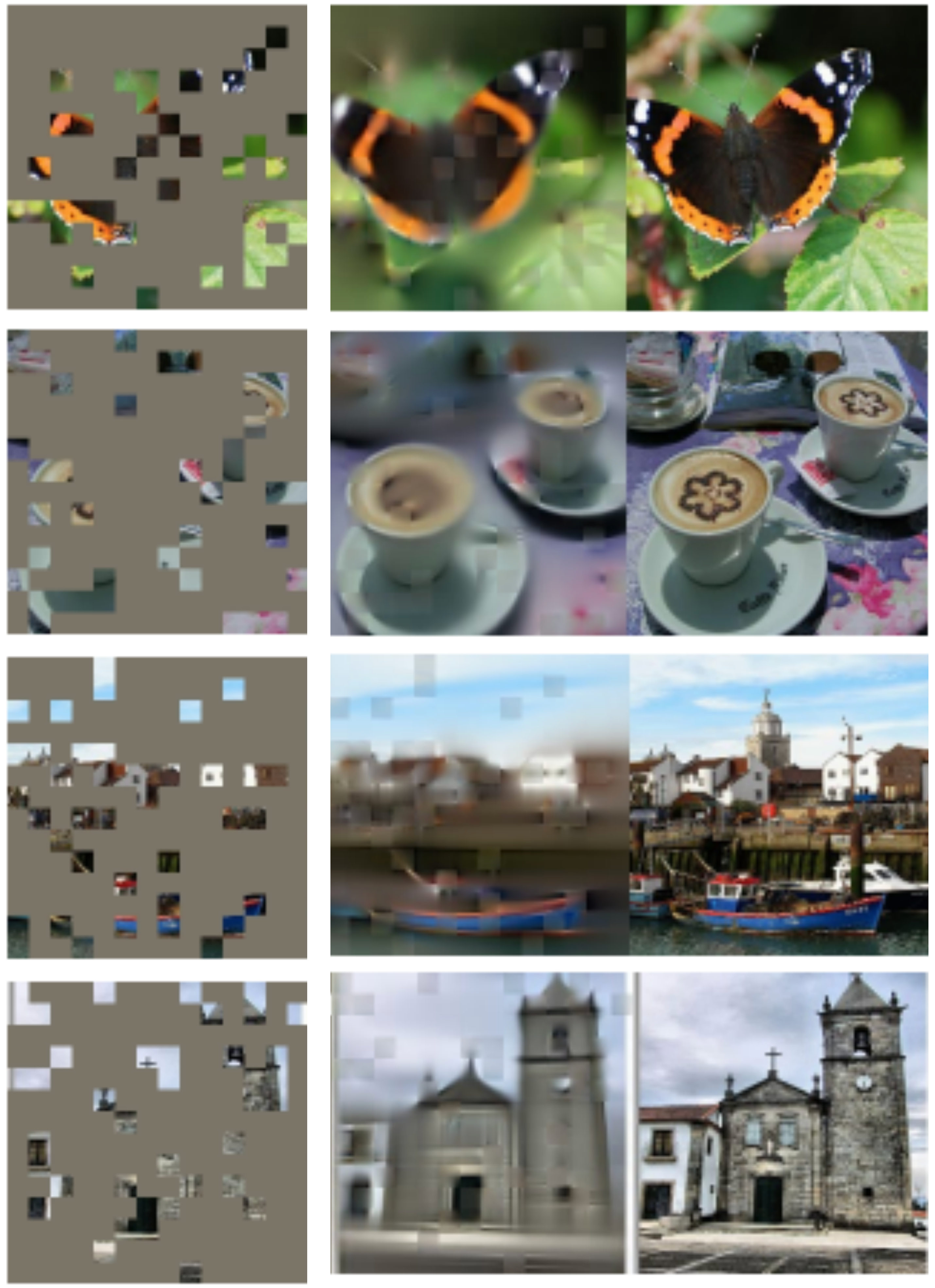
Masked Autoencoder

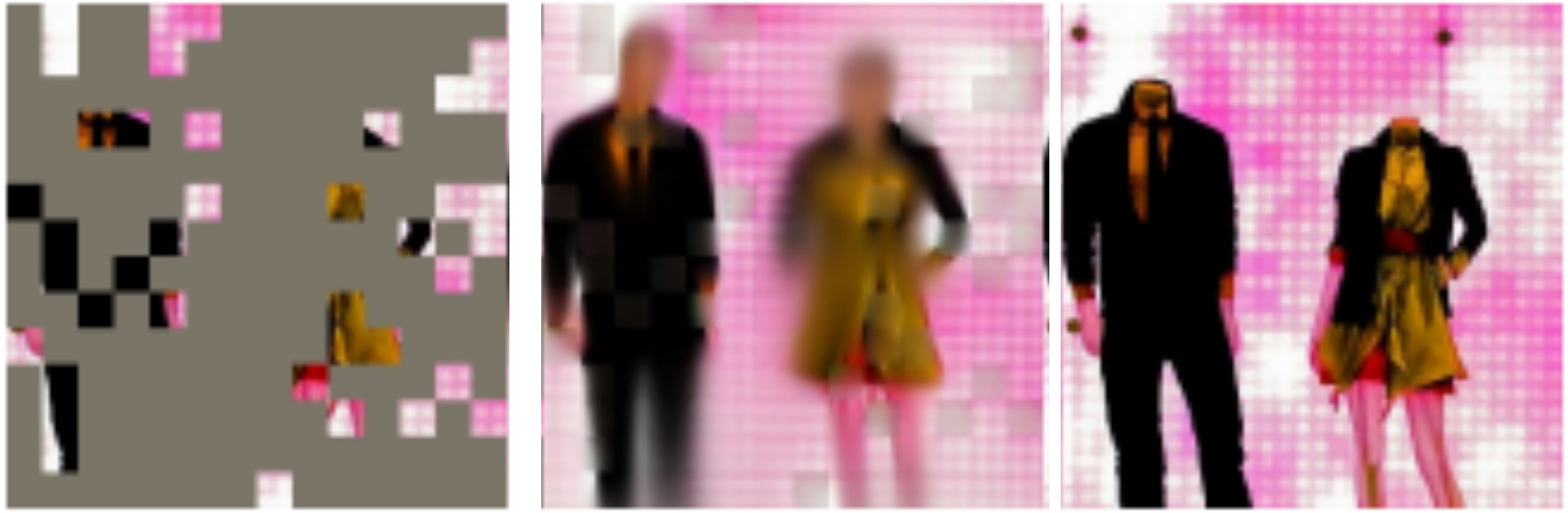


Masked Autoencoder



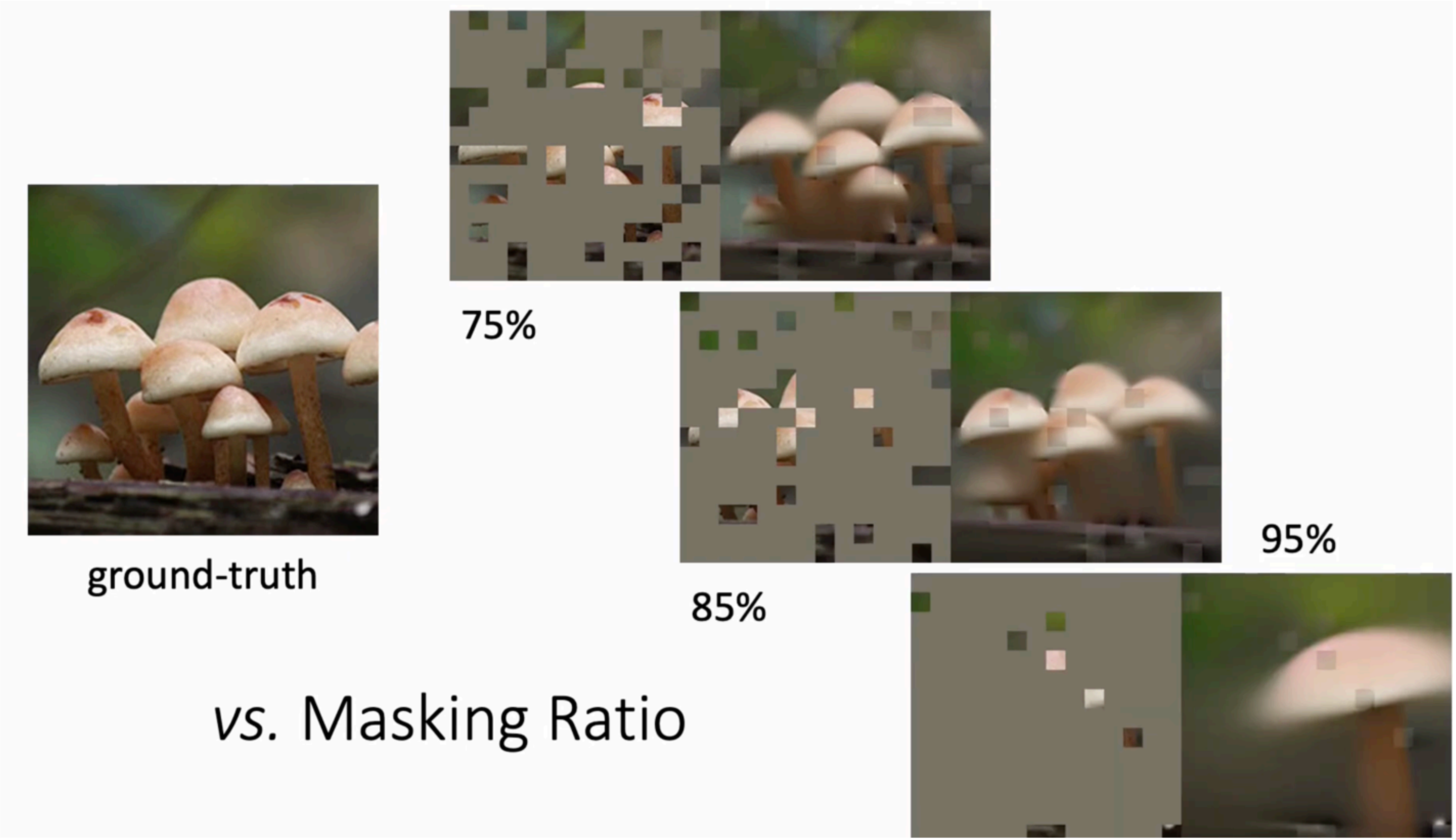
Masked Autoencoder





Is this a generative image model?

Is this a generative image model?



Not “fully” generative

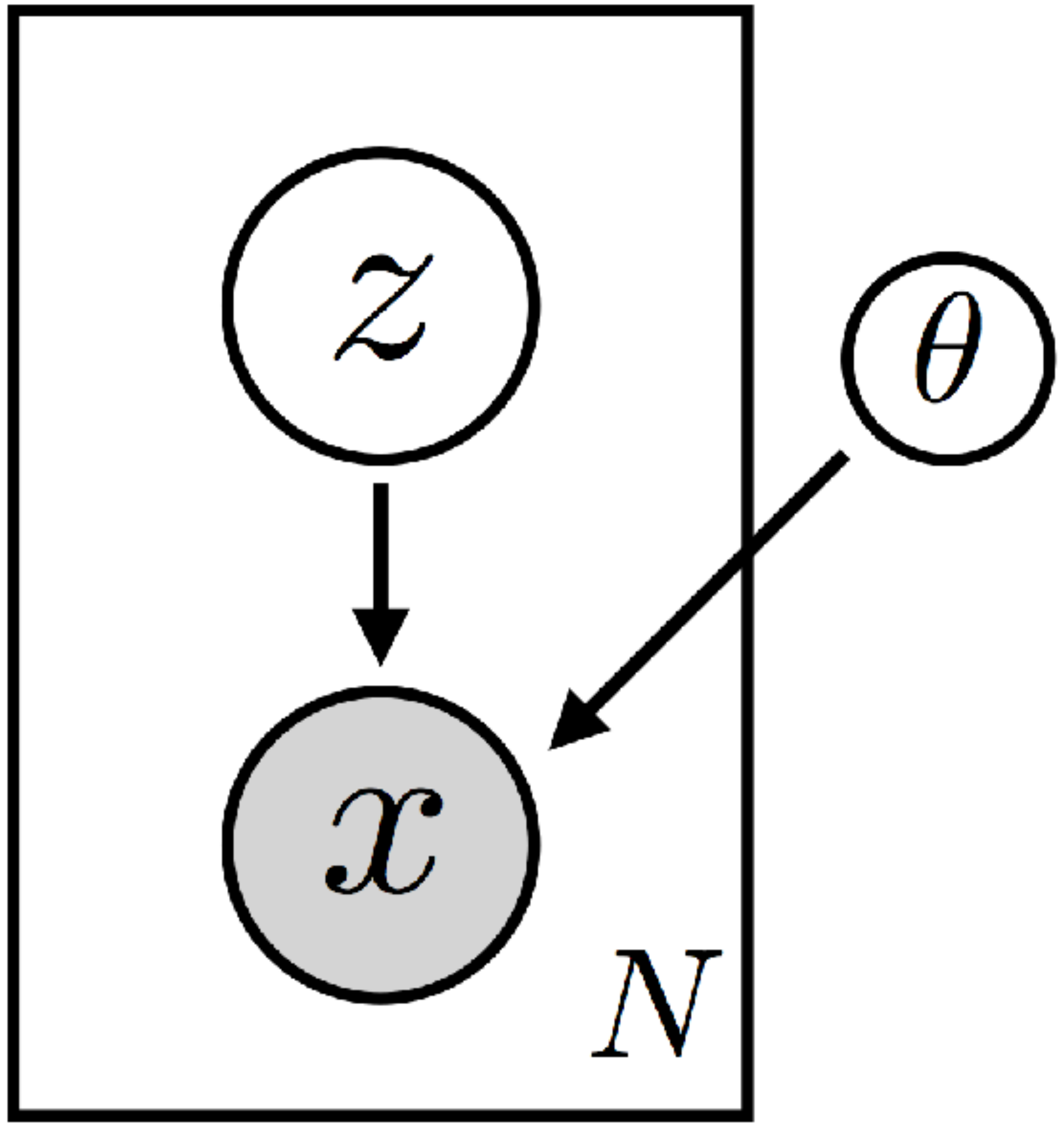
Our goal

- I have nothing (or maybe a caption)
- I produce a new image (maybe following my caption)
- I don’t want to only do image completion…

Any idea?

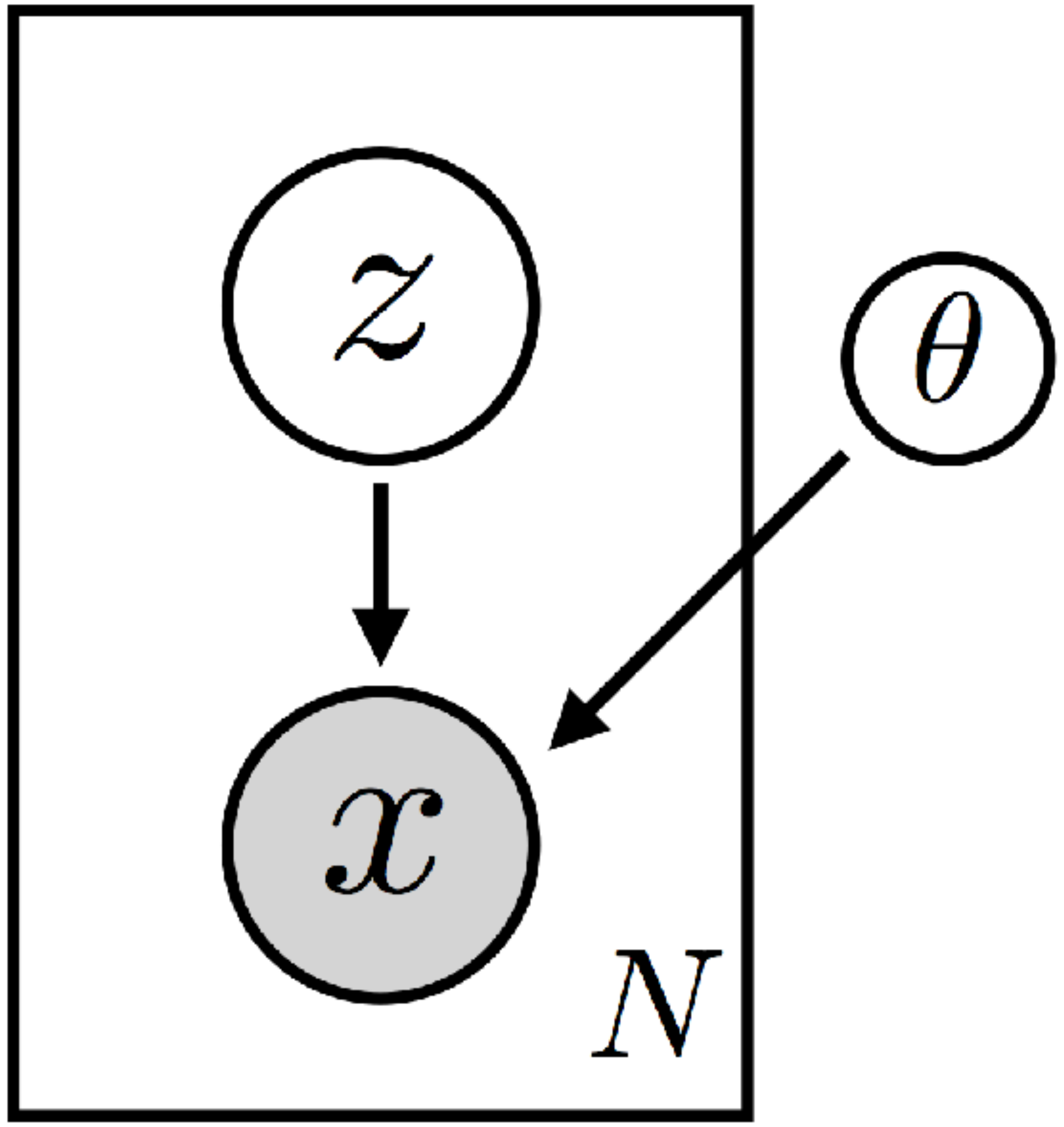
Variational Autoencoders

- Given “nothing”
- Sample a random gaussian vector z
- Generate an image via decoder(z)



Variational Autoencoders: Training

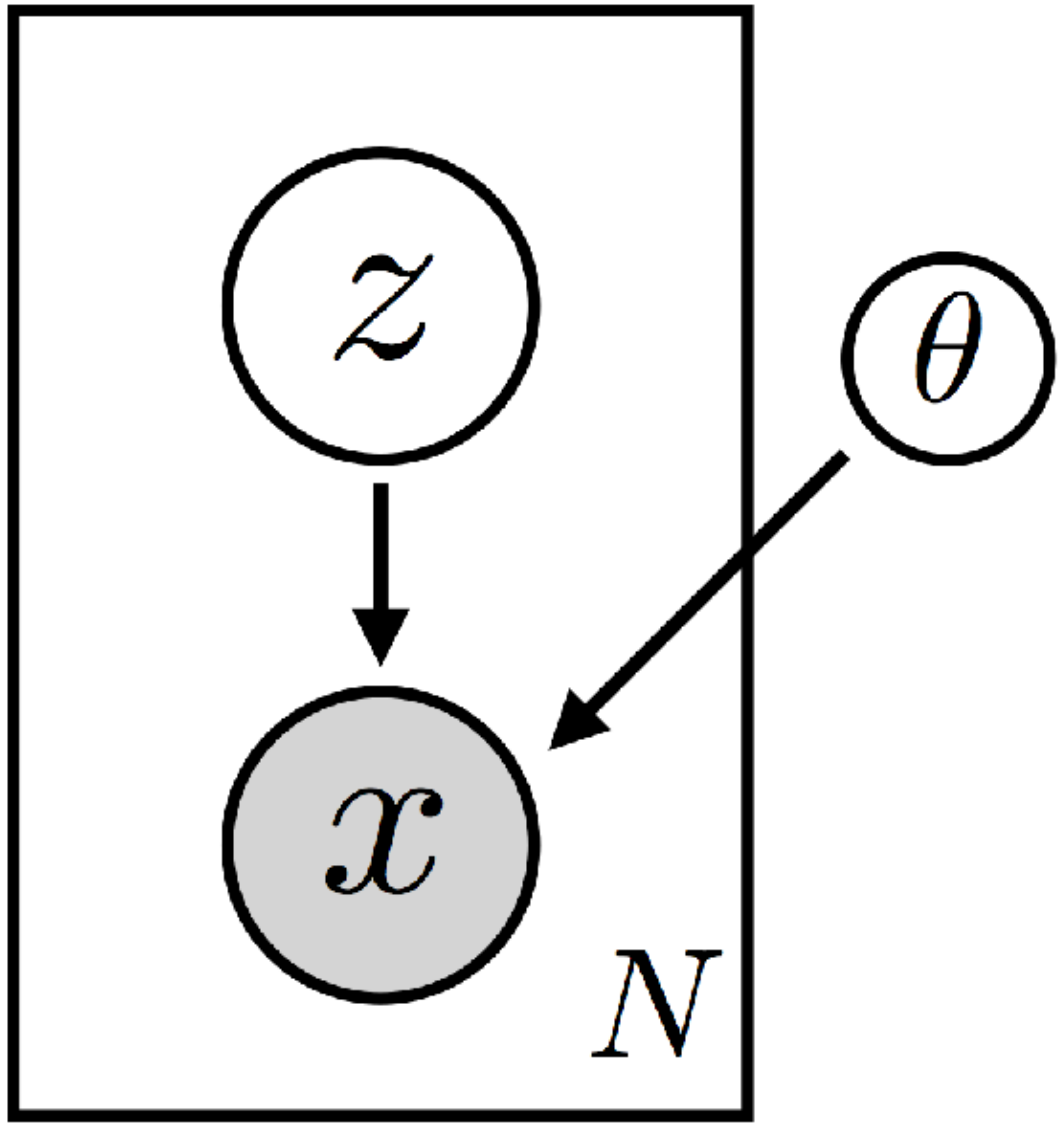
- For every training set image (x)
- There must be a Gaussian z so that
- ||Decoder(z)-x|| < eps



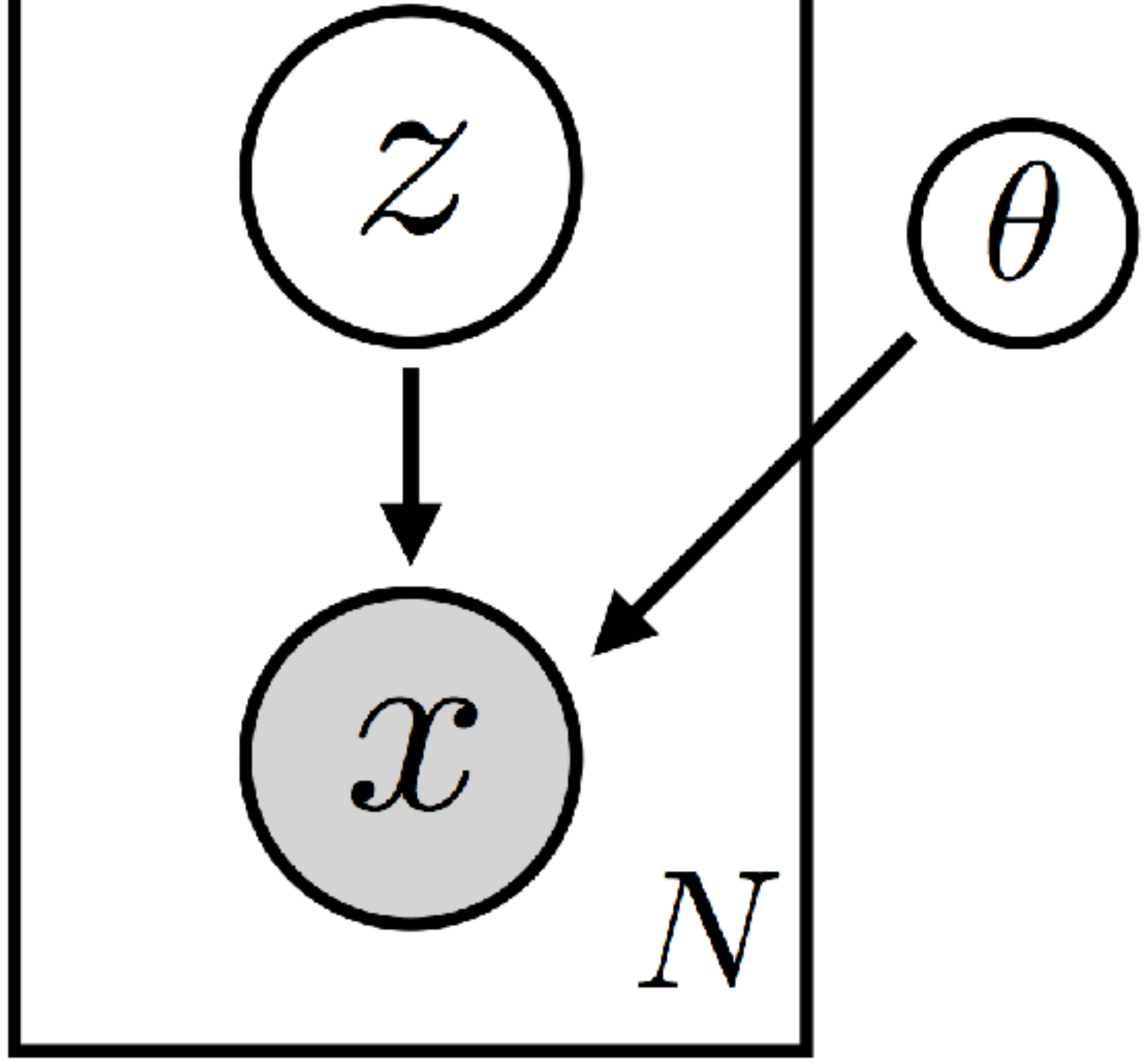
Variational Autoencoders: Training

- For every training set image (x)
- There must be a Gaussian z so that
- ||Decoder(z)-x|| < eps

How to find that Gaussian z?

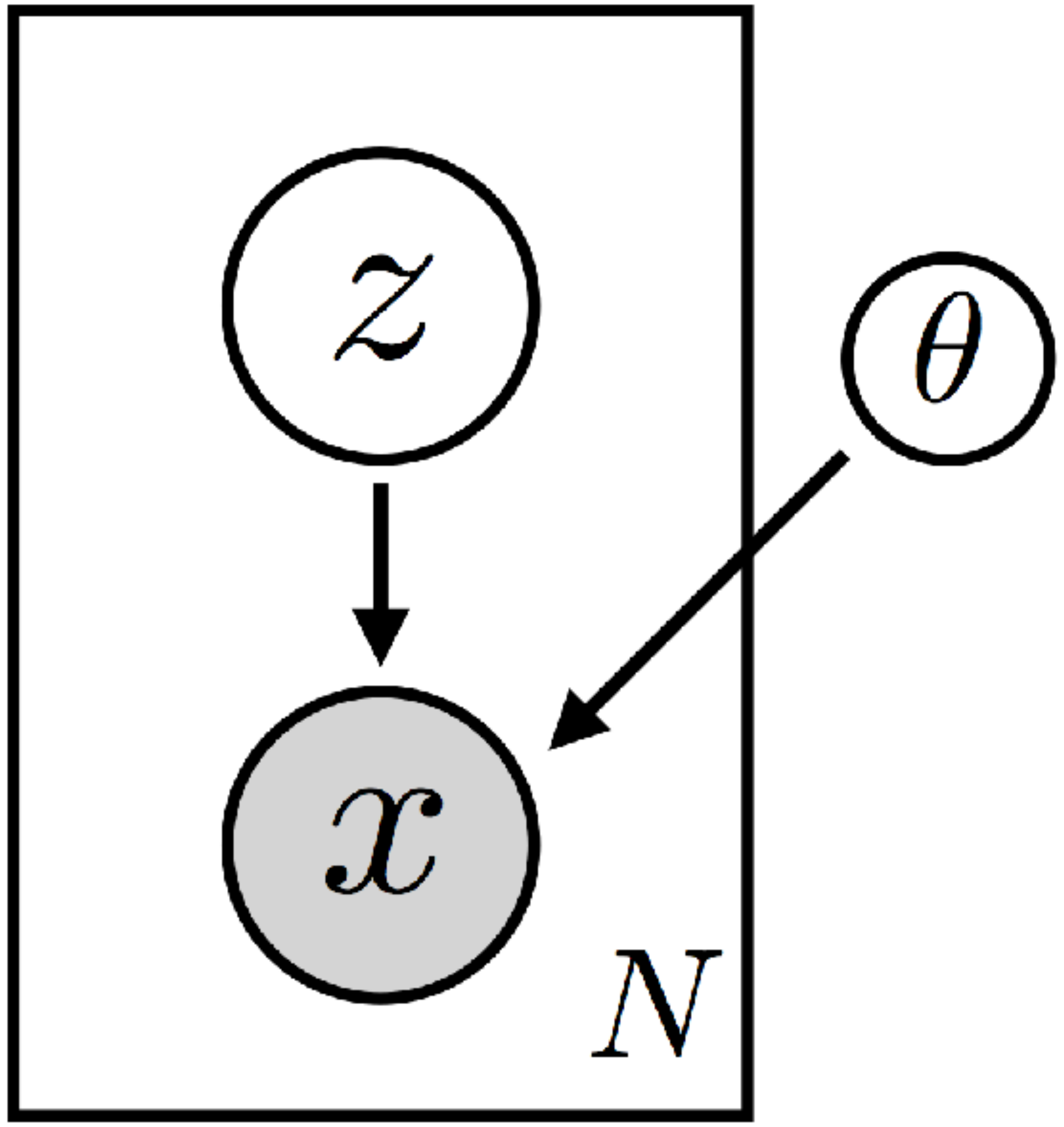


Variational Autoencoders: Training



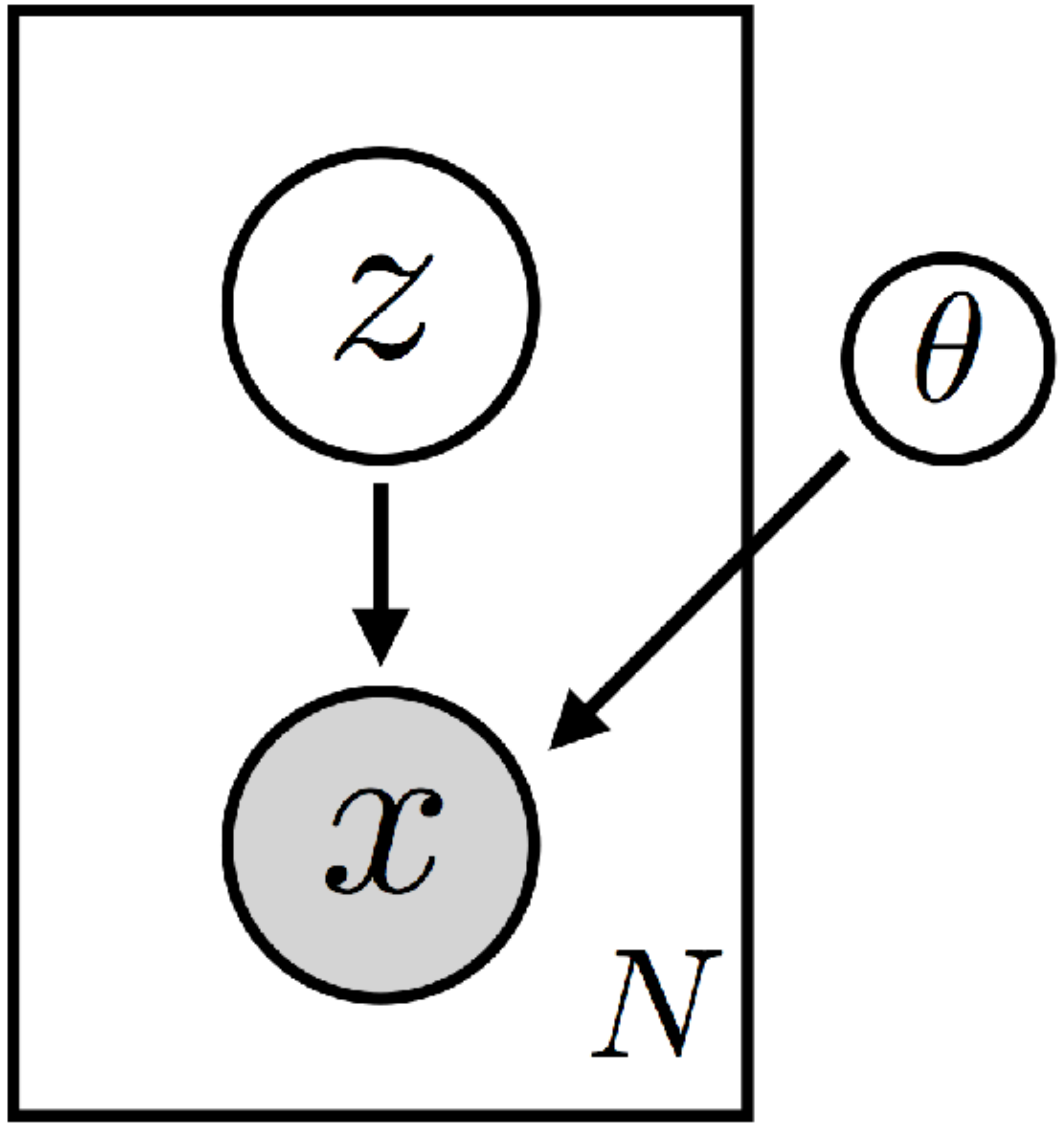
Variational Autoencoders: Training

- Amortized (encoder!) Variational (name!) Inference



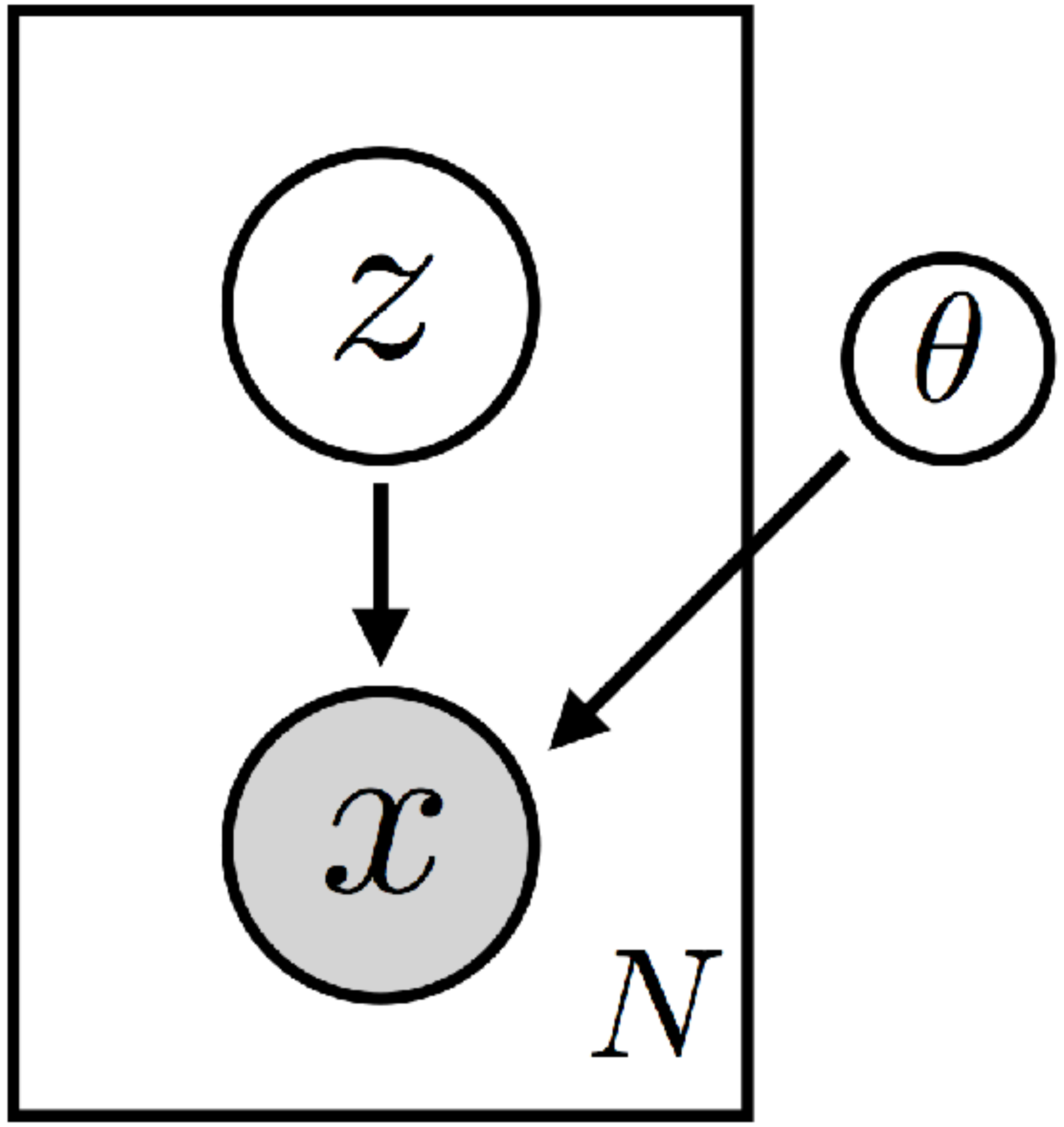
Variational Autoencoders: Training

- Amortized (encoder!) Variational (name!) Inference
- Train an encoder so that



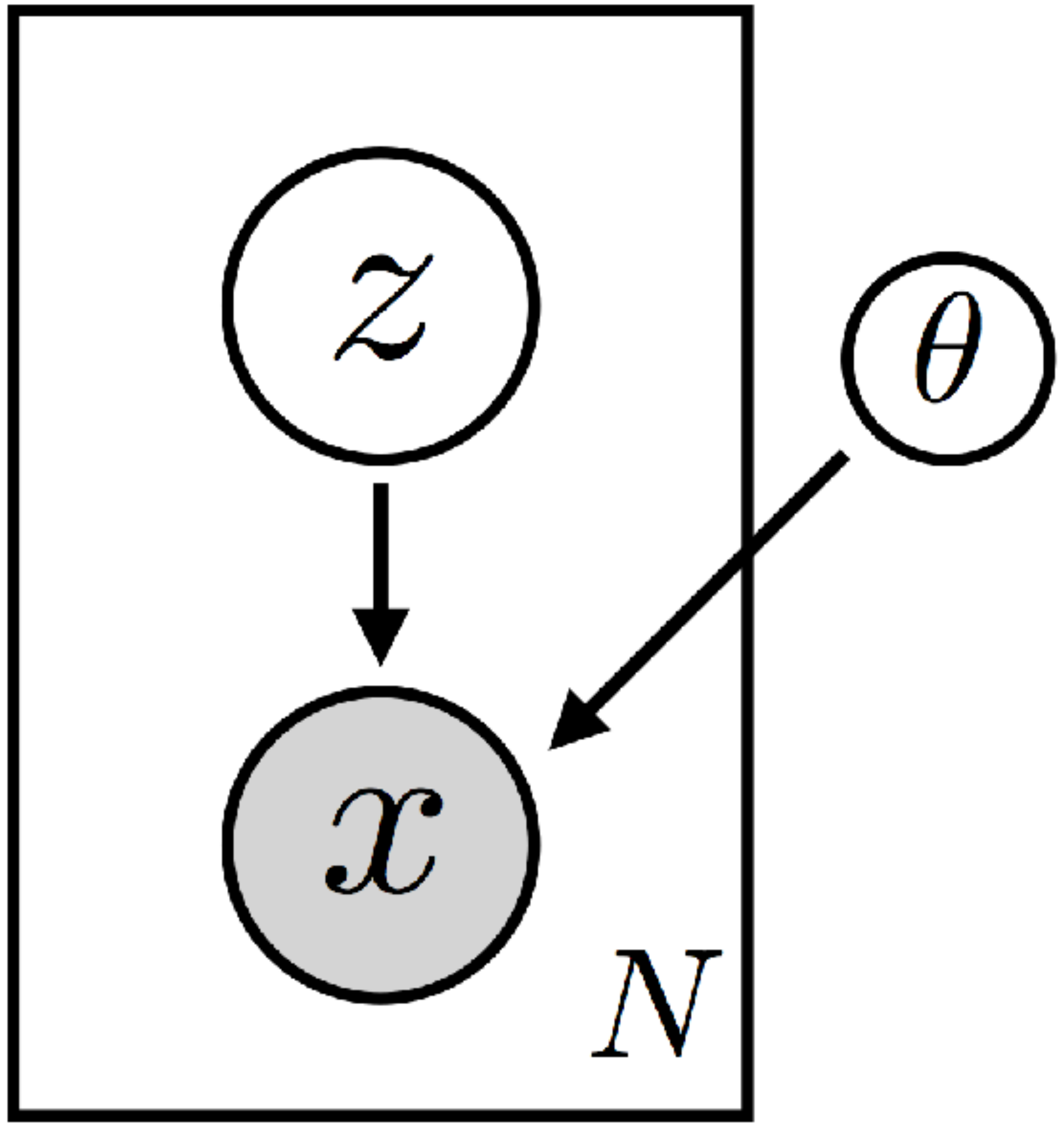
Variational Autoencoders: Training

- Amortized (encoder!) Variational (name!) Inference
- Train an encoder so that
- $z \sim \mathcal{N}(\mu_\theta(x), \Sigma_\theta(x))$



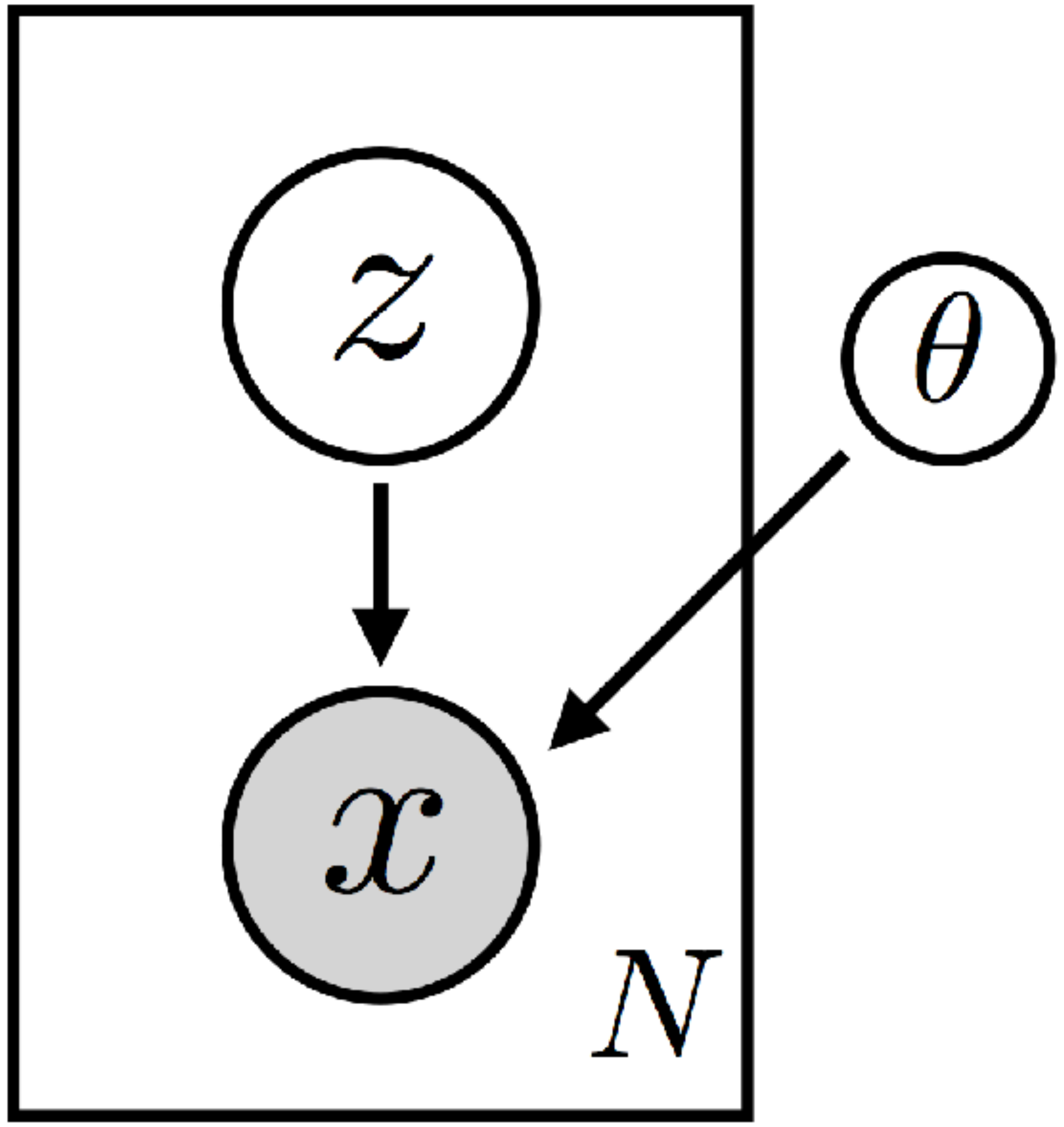
Variational Autoencoders: Training

- Amortized (encoder!) Variational (name!) Inference
- Train an encoder so that
- $z \sim \mathcal{N}(\mu_\theta(x), \Sigma_\theta(x))$
- $||\text{decoder}(z) - x|| < \epsilon$



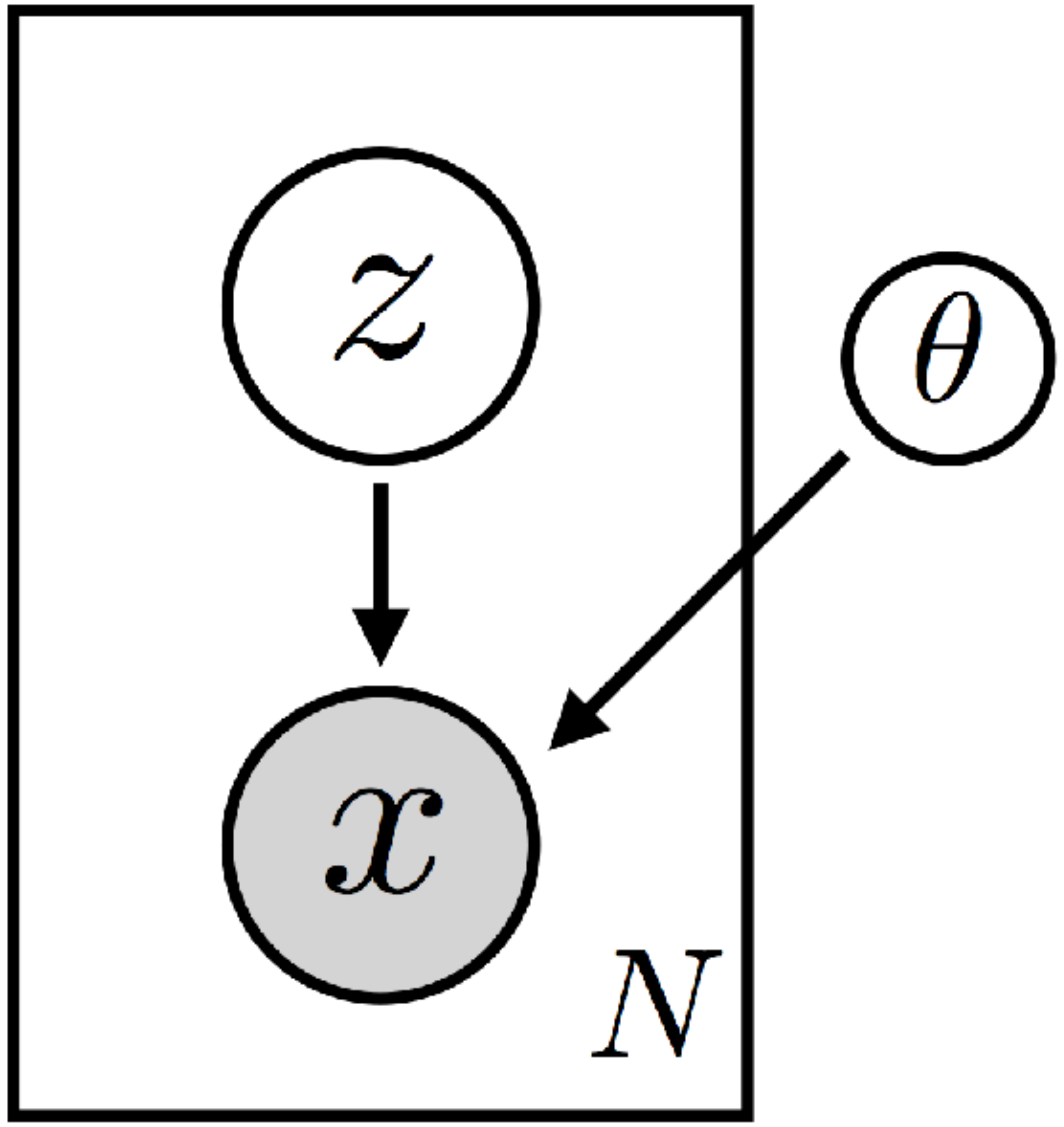
Variational Autoencoders: Training

- Amortized (encoder!) Variational (name!) Inference
- Train an encoder so that
- $z \sim \mathcal{N}(\mu_\theta(x), \Sigma_\theta(x))$
- $||\text{decoder}(z) - x|| < \epsilon$
- While ensuring z is close to our N(0,I) target



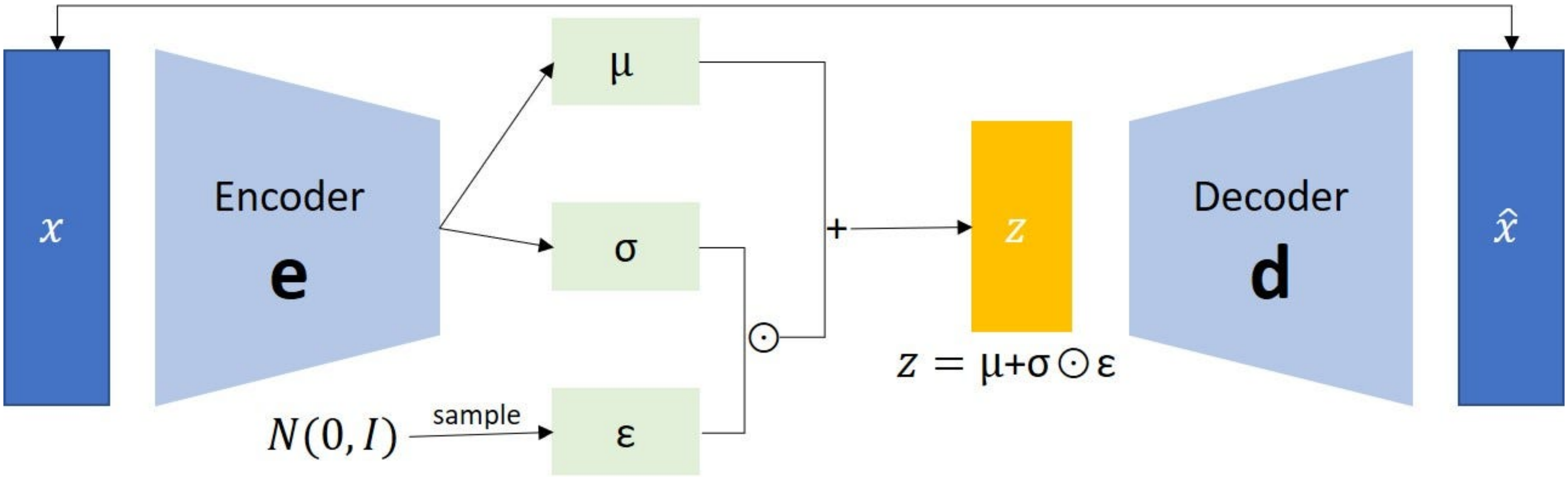
Variational Autoencoders: Training

- Amortized (encoder!) Variational (name!) Inference
- Train an encoder so that
- $z \sim \mathcal{N}(\mu_\theta(x), \Sigma_\theta(x))$
- $||\text{decoder}(z) - x|| < \epsilon$
- While ensuring z is close to our N(0,I) target
- Use KL-divergence to measure that closeness



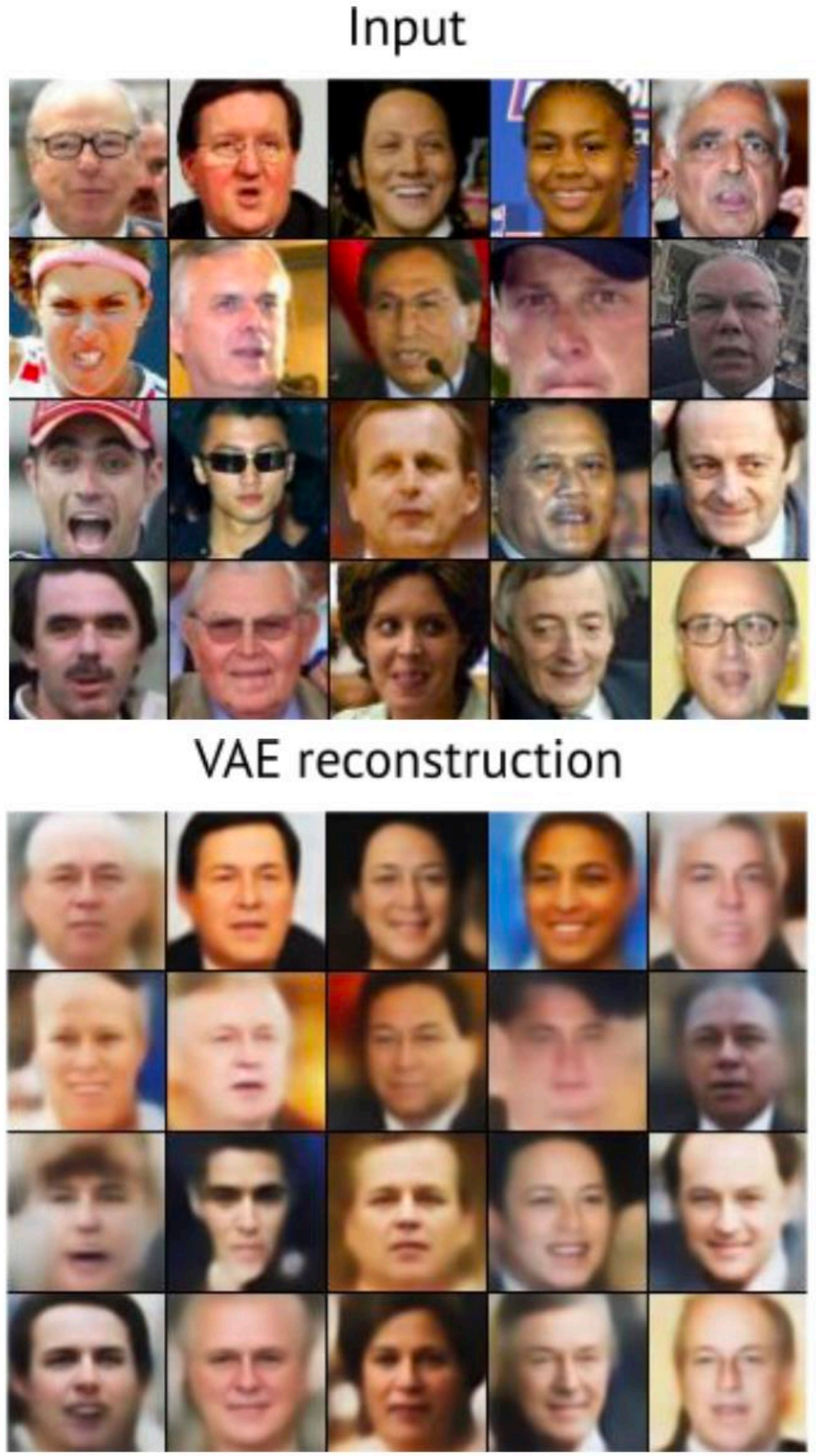
Variational Autoencoders: Overview

Minimize 1:$(x - \hat{x})^2$



Minimize 2: $\frac{1}{2}\sum_{i=1}^{N}(\exp(\sigma_i) - (1+\sigma_i) + \mu_i{}^2)$

VAE output



https://towardsdatascience.com/what-the-heck-are-vae-gans-17b86023588a

What’s the issue here?

Why?

See you on Friday!