Variational Auto Encoders

Sergey Tulyakov

Today

- Generative vs Discriminative
- AutoEncoders - AEs
- Variational AutoEncoders - VAEs
  - AEs vs VAEs
  - Derivation
  - Intuition
    - Reconstruction losses
  - Applications
  - Extensions

Generative vs Discriminative

From C. Bishop:

Approaches that **explicitly or implicitly model the distribution of inputs as well as outputs are known** as generative models, because by sampling from them it is possible to generate synthetic data points in the input space.

Generative vs Discriminative

From C. Bishop:

Approaches that **explicitly or implicitly model the distribution of inputs as well as outputs are known** as generative models, because by sampling from them it is possible to generate synthetic data points in the input space.

Sampling

$x \sim p(x)$

Generative vs Discriminative

From C. Bishop:

Approaches that **explicitly or implicitly model the distribution of inputs as well as outputs are known** as generative models, because by sampling from them it is possible to generate synthetic data points in the input space.

Sampling	Style Transfer
$x \sim p(x)$	$x_k \sim p(x\|C_k)$

Generative vs Discriminative

From C. Bishop:

Approaches that **explicitly or implicitly model the distribution of inputs as well as outputs are known** as generative models, because by sampling from them it is possible to generate synthetic data points in the input space.

Sampling	Style Transfer	Classification
$x \sim p(x)$	$x_k \sim p(x\|C_k)$	$C_k \sim p(C_k\|x)$

Generative vs Discriminative

From C. Bishop:

Approaches that **explicitly or implicitly model the distribution of inputs as well as outputs are known** as generative models, because by sampling from them it is possible to generate synthetic data points in the input space.

Sampling	Style Transfer	Classification	Evaluating Likelihood
$x \sim p(x)$	$x_k \sim p(x\|C_k)$	$C_k \sim p(C_k\|x)$	$log\ p(x)$

Generative vs Discriminative

From C. Bishop:

Approaches that **explicitly or implicitly model the distribution of inputs as well as outputs are known** as generative models, because by sampling from them it is possible to generate synthetic data points in the input space.

Sampling	Style Transfer	Classification	Evaluating Likelihood
$x \sim p(x)$	$x_k \sim p(x\|C_k)$	$C_k \sim p(C_k\|x)$	$log\ p(x)$

Approaches that model the posterior probabilities directly are called discriminative models.

Generative vs Discriminative

From C. Bishop:

Approaches that **explicitly or implicitly model the distribution of inputs as well as outputs are known** as generative models, because by sampling from them it is possible to generate synthetic data points in the input space.

Sampling	Style Transfer	Classification	Evaluating Likelihood
$x \sim p(x)$	$x_k \sim p(x\|C_k)$	$C_k \sim p(C_k\|x)$	$log\ p(x)$

Approaches that model the posterior probabilities directly are called discriminative models.

Classification

$$p(C_k|x) = f(x)$$

Generative vs Discriminative

From C. Bishop:

Approaches that **explicitly or implicitly model the distribution of inputs as well as outputs are known** as generative models, because by sampling from them it is possible to generate synthetic data points in the input space.

Sampling	Style Transfer	Classification	Evaluating Likelihood
$x \sim p(x)$	$x_k \sim p(x\|C_k)$	$C_k \sim p(C_k\|x)$	$log\ p(x)$

Approaches that model the posterior probabilities directly are called discriminative models.

Modeled with NN

Classification

$$p(C_k|x) = f(x)$$

Generative vs Discriminative

From C. Bishop:

Approaches that **explicitly or implicitly model the distribution of inputs as well as outputs are known** as generative models, because by sampling from them it is possible to generate synthetic data points in the input space.

Sampling	Style Transfer	Classification	Evaluating Likelihood
$x \sim p(x)$	$x_k \sim p(x\|C_k)$	$C_k \sim p(C_k\|x)$	$log\ p(x)$

Approaches that model the posterior probabilities directly are called discriminative models.

Modeled with NN

not NN

Classification

$$p(C_k|x) = f(x)$$

Auto encoders

AutoEncoders (AEs) consist of two models:

Encoder:

$$h = \mathrm{Enc}(x)$$

Decoder:

$$\hat{x} = \mathrm{Dec}(h)$$

Auto encoders

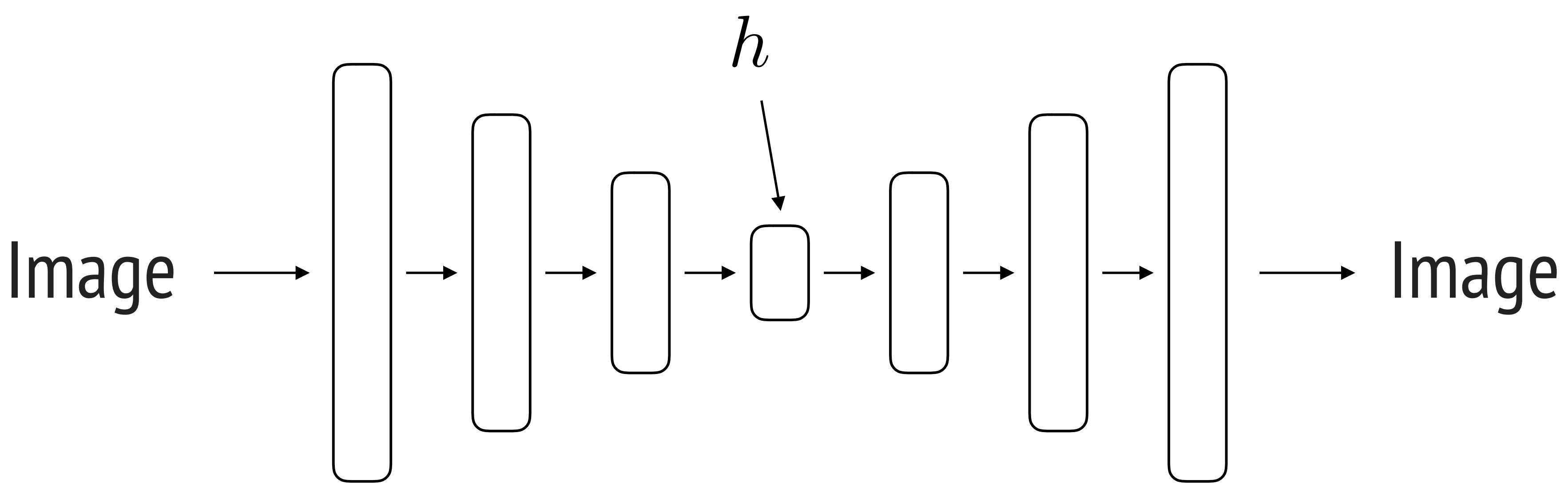
AutoEncoders (AEs) consist of two models:

Encoder:

$$h = \mathrm{Enc}(x)$$

Decoder:

$$\hat{x} = \mathrm{Dec}(h)$$



Auto encoders

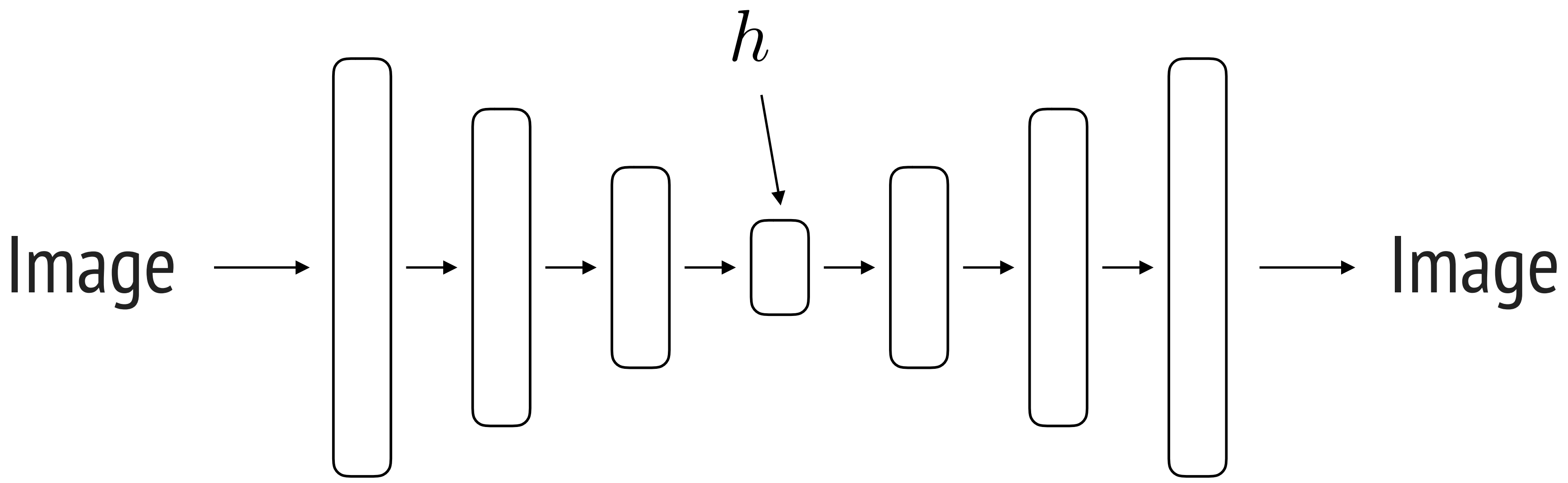
AutoEncoders (AEs) consist of two models:

Encoder:

$$h = \mathrm{Enc}(x)$$

Decoder:

$$\hat{x} = \mathrm{Dec}(h)$$



Training using reconstruction loss:

Auto encoders

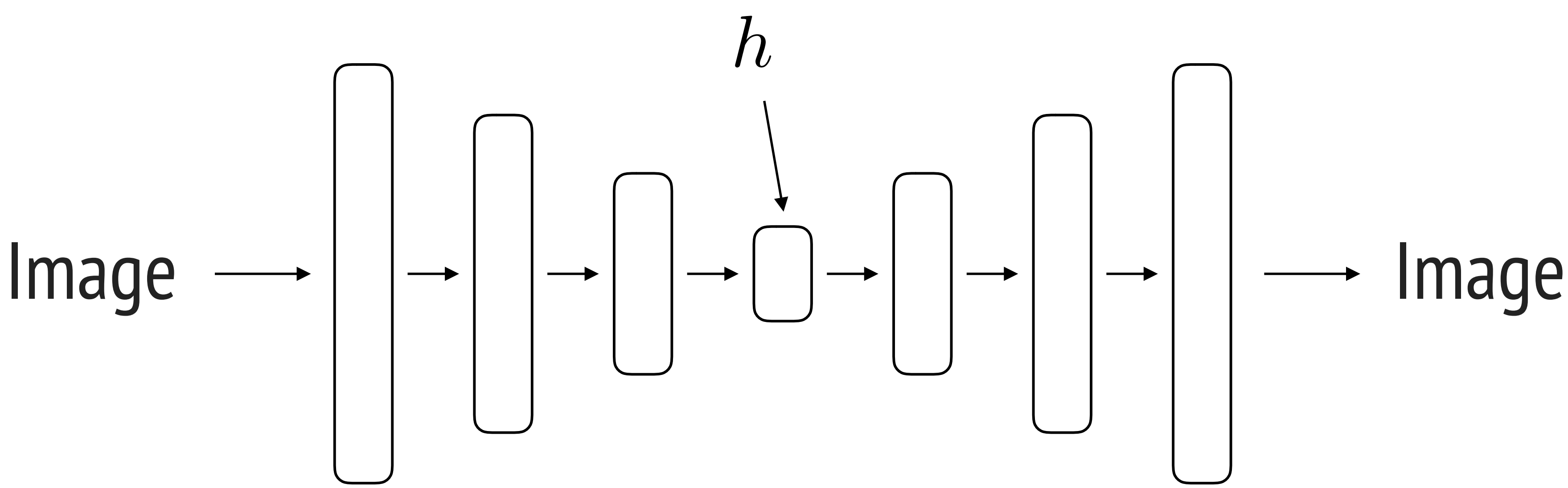
AutoEncoders (AEs) consist of two models:

Encoder:

$$h = \mathrm{Enc}(x)$$

Decoder:

$$\hat{x} = \mathrm{Dec}(h)$$



Training using reconstruction loss:

$$\hat{x} = \mathrm{Dec}(\mathrm{Enc}(x))$$

Auto encoders

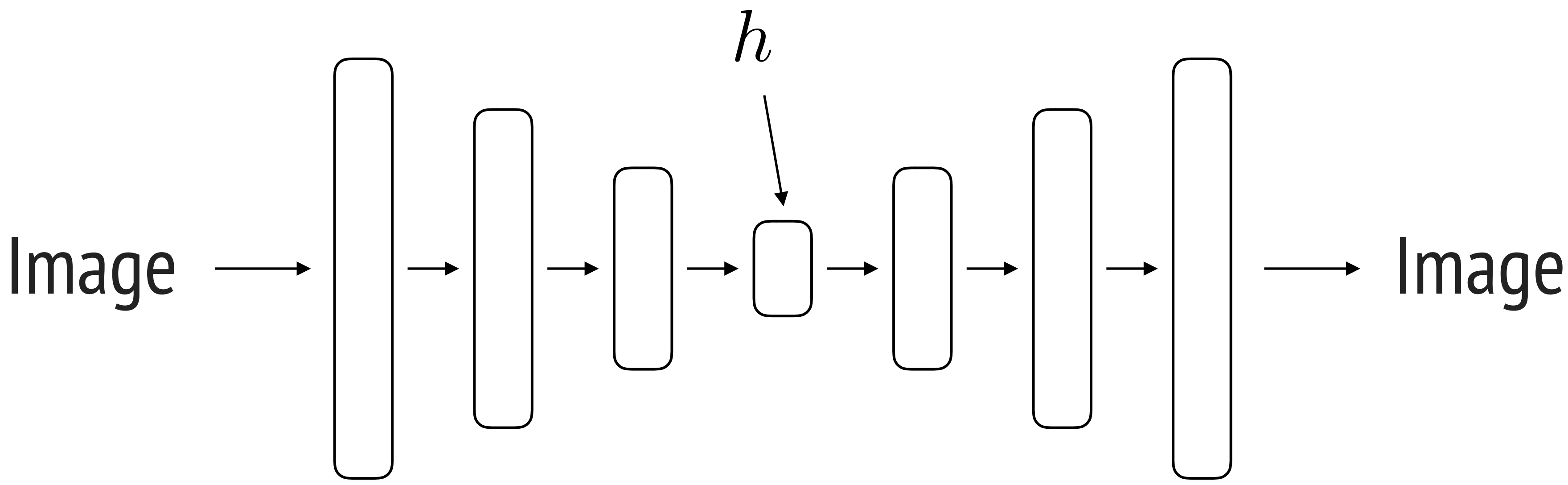
AutoEncoders (AEs) consist of two models:

Encoder:

$$h = \mathrm{Enc}(x)$$

Decoder:

$$\hat{x} = \mathrm{Dec}(h)$$



Training using reconstruction loss:

$$\hat{x} = \mathrm{Dec}(\mathrm{Enc}(x)) \qquad \mathcal{L} = f(\hat{x}, x)$$

Auto encoders

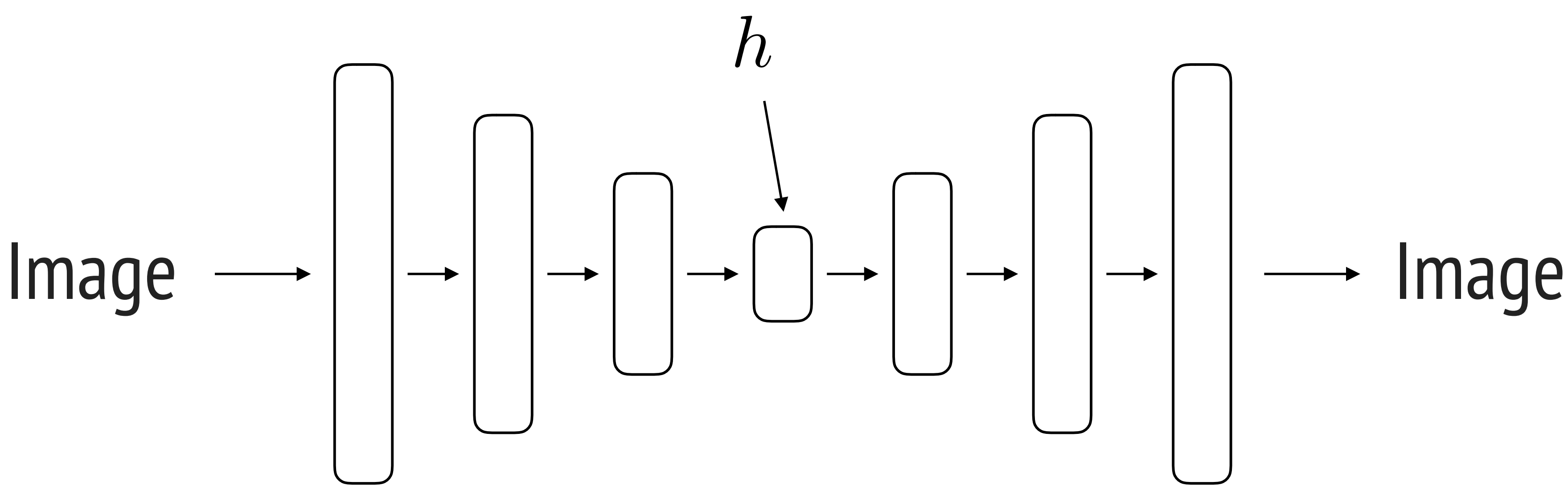
AutoEncoders (AEs) consist of two models:

Encoder:

$$h = \mathrm{Enc}(x)$$

Decoder:

$$\hat{x} = \mathrm{Dec}(h)$$



Training using reconstruction loss:

$$\hat{x} = \mathrm{Dec}(\mathrm{Enc}(x)) \qquad \mathcal{L} = f(\hat{x}, x)$$

l1, l2 - norms

Perception similarity

Structural similarity

Examples: De-noising Auto Encoders

Learning more robust representations:

- Randomly zero pixels in the input and reconstruct the uncorrupted image

Examples: De-noising Auto Encoders

Learning more robust representations:

- Randomly zero pixels in the input and reconstruct the uncorrupted image

$$h = \mathrm{Enc}(x \cdot m)$$
$$\hat{x} = \mathrm{Dec}(h),$$
$$m \in \{0, 1\}$$

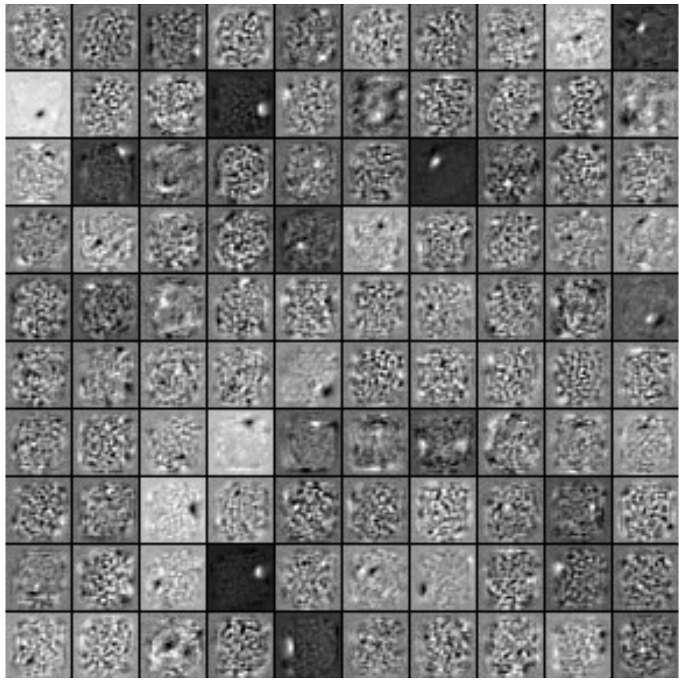
Examples: De-noising Auto Encoders

Learning more robust representations:

- Randomly zero pixels in the input and reconstruct the uncorrupted image

$$h = \mathrm{Enc}(x \cdot m)$$
$$\hat{x} = \mathrm{Dec}(h),$$
$$m \in \{0, 1\}$$

Without noise



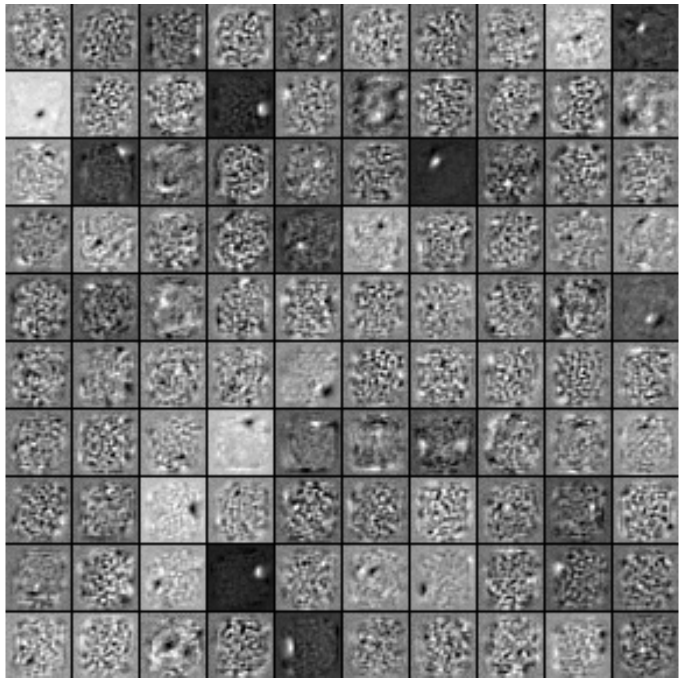
Examples: De-noising Auto Encoders

Learning more robust representations:

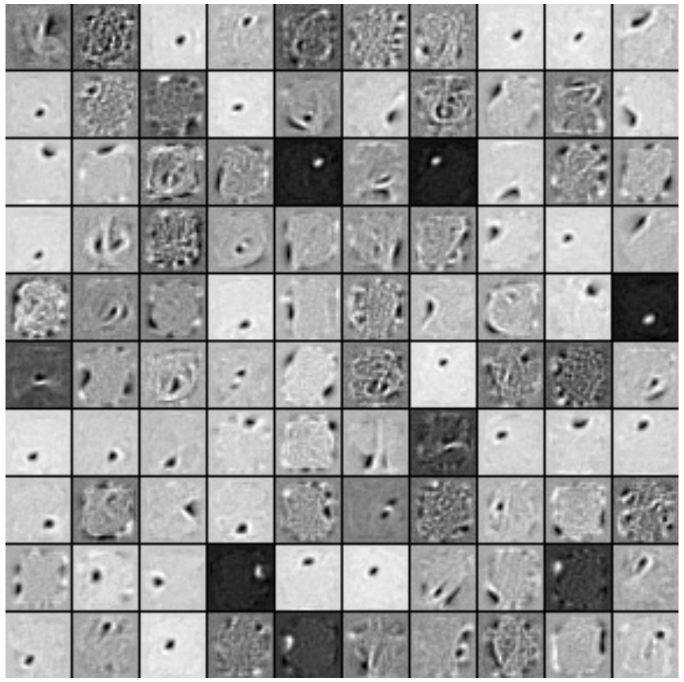
- Randomly zero pixels in the input and reconstruct the uncorrupted image

$$h = \mathrm{Enc}(x \cdot m)$$
$$\hat{x} = \mathrm{Dec}(h),$$
$$m \in \{0, 1\}$$

Without noise



With noise



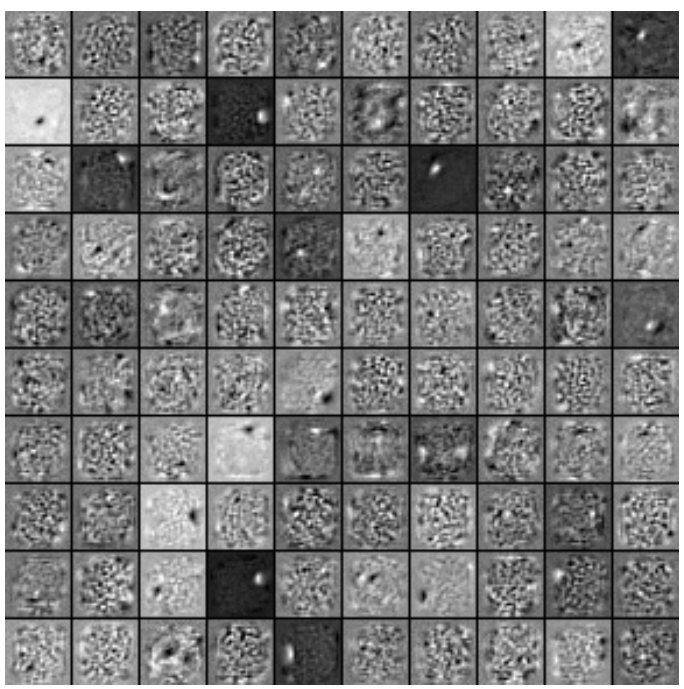
Examples: De-noising Auto Encoders

Learning more robust representations:

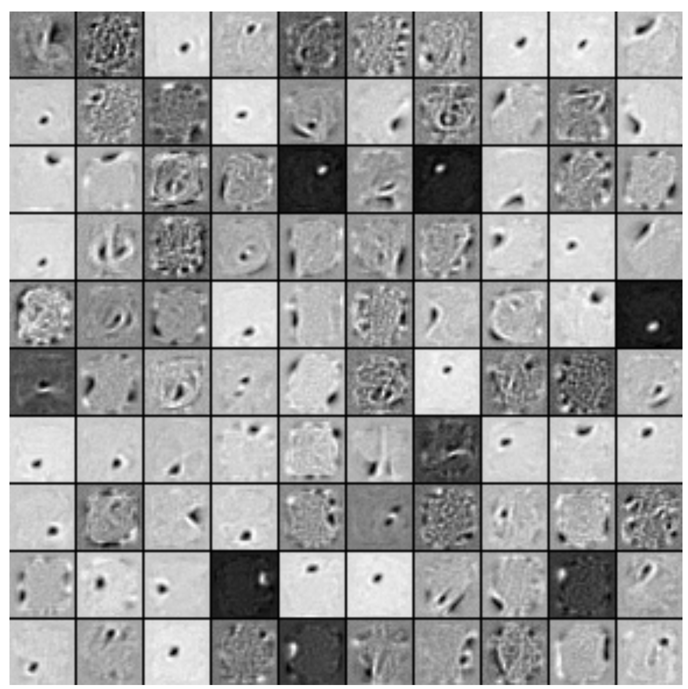
- Randomly zero pixels in the input and reconstruct the uncorrupted image

$$h = \mathrm{Enc}(x \cdot m)$$
$$\hat{x} = \mathrm{Dec}(h),$$
$$m \in \{0, 1\}$$

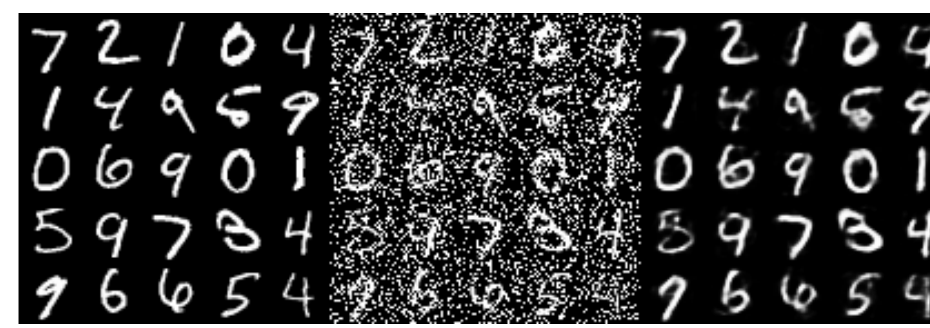
Without noise



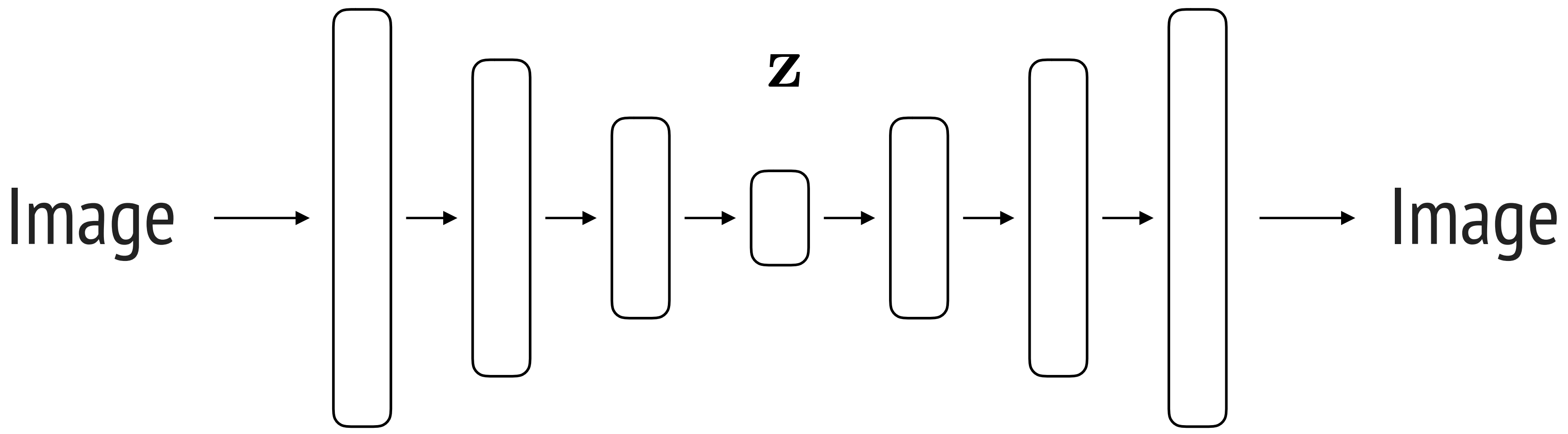
With noise



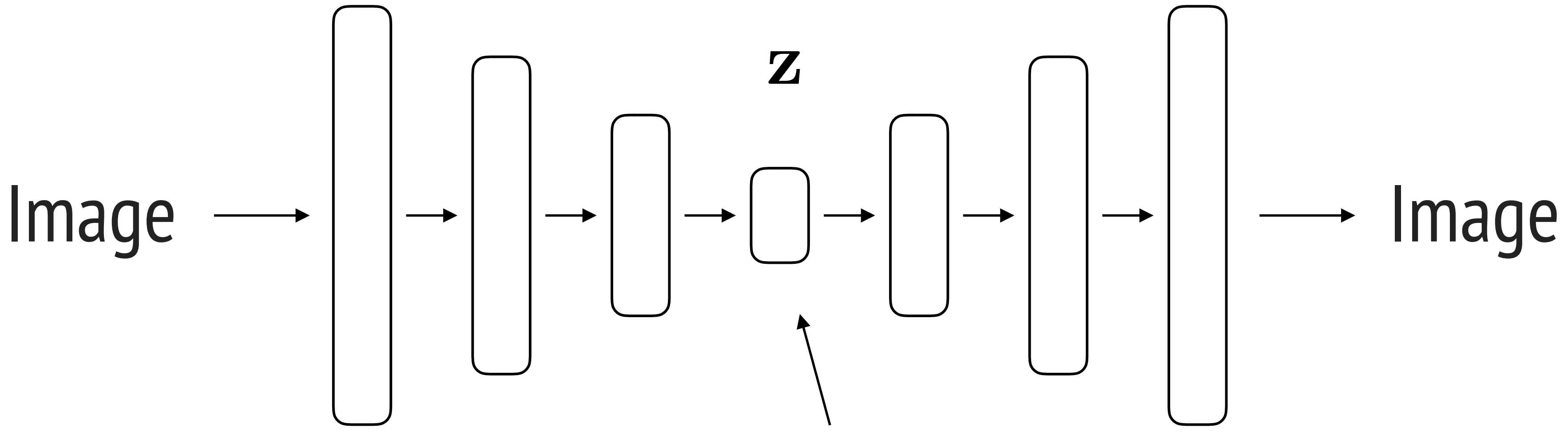
Original Corrupted Output



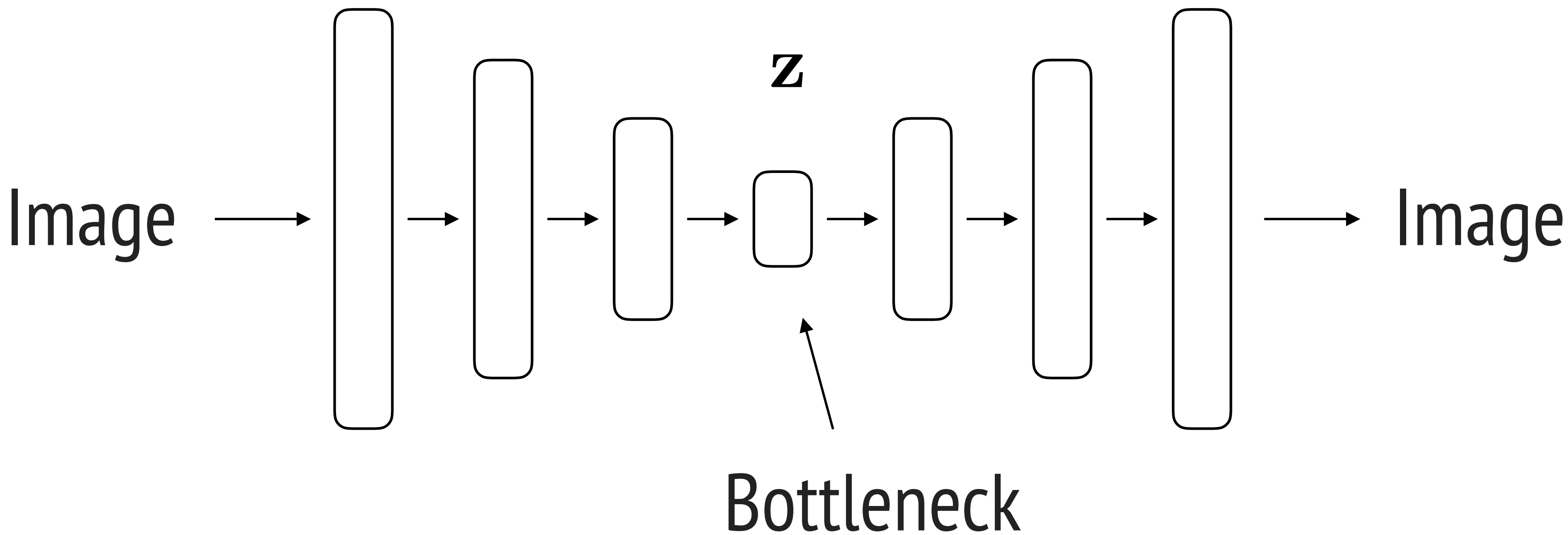
Examples: De-noising Auto Encoders



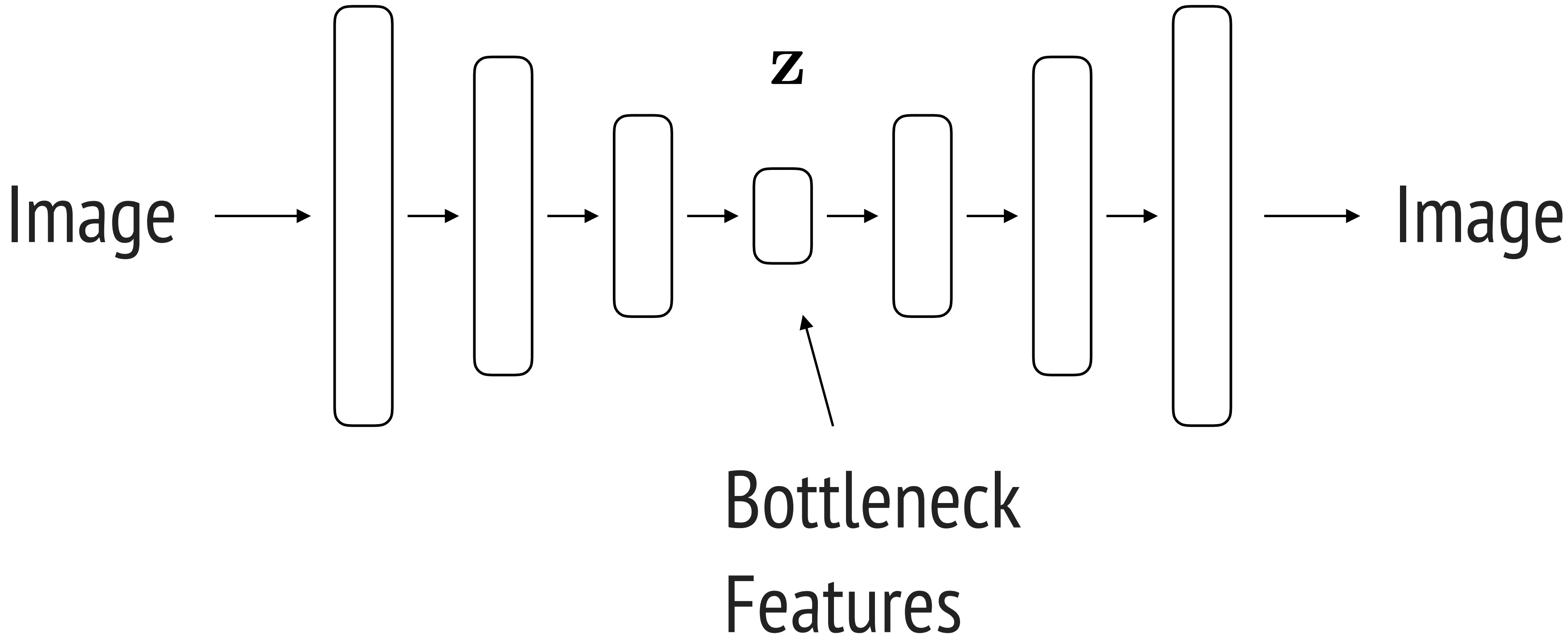
Examples: De-noising Auto Encoders



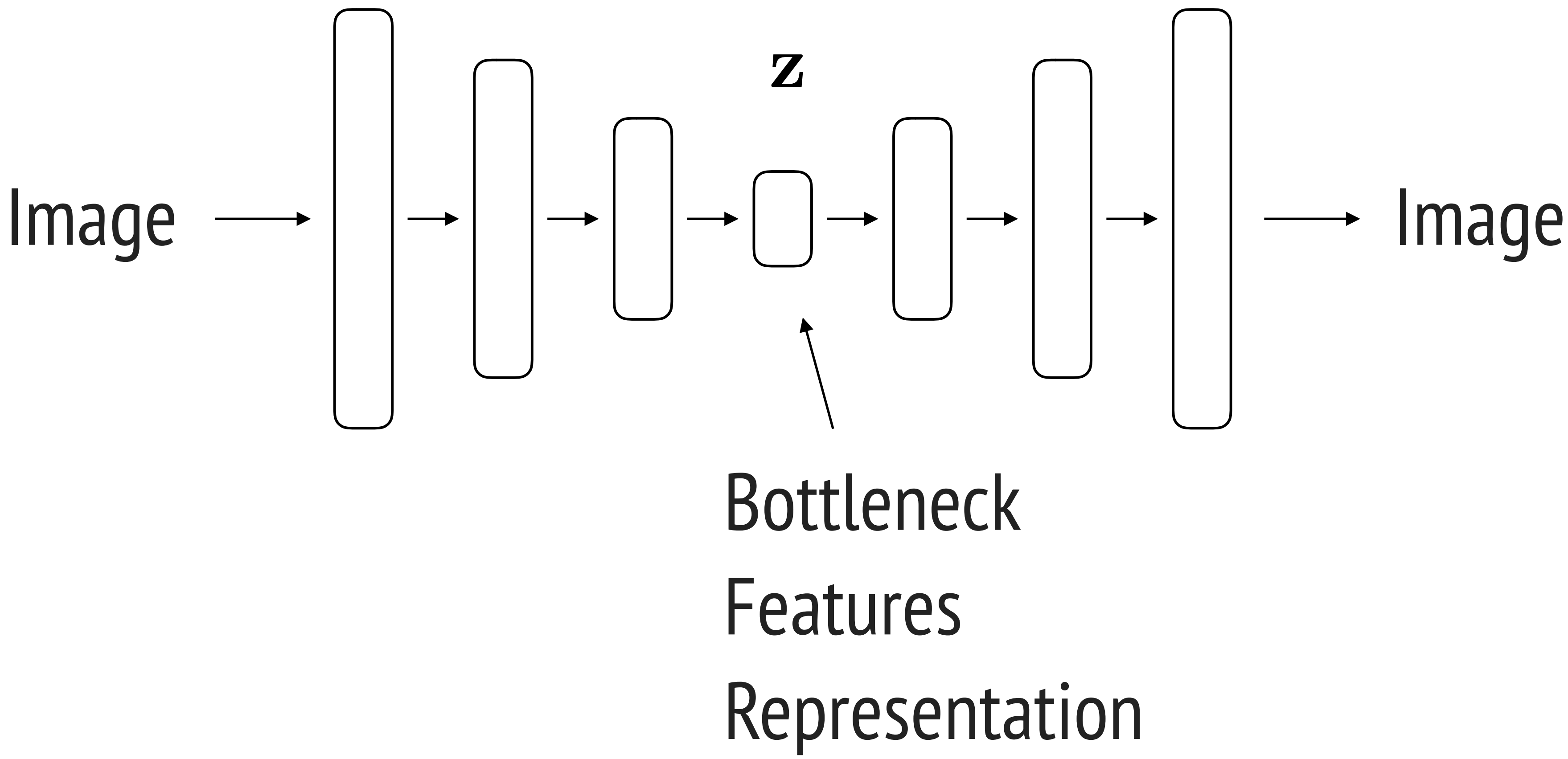
Examples: De-noising Auto Encoders



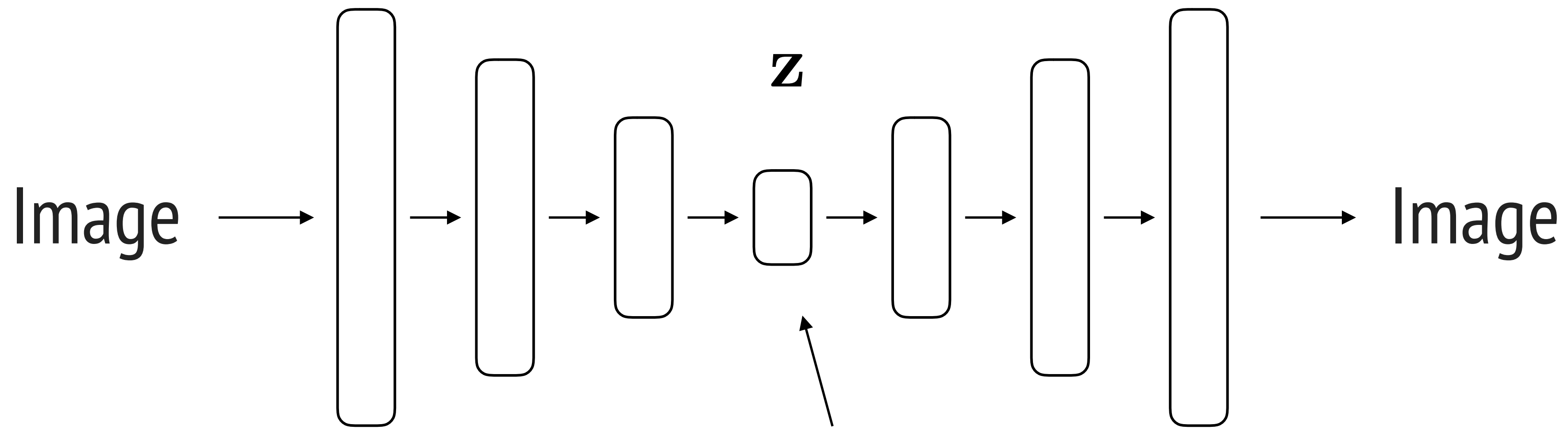
Examples: De-noising Auto Encoders



Examples: De-noising Auto Encoders

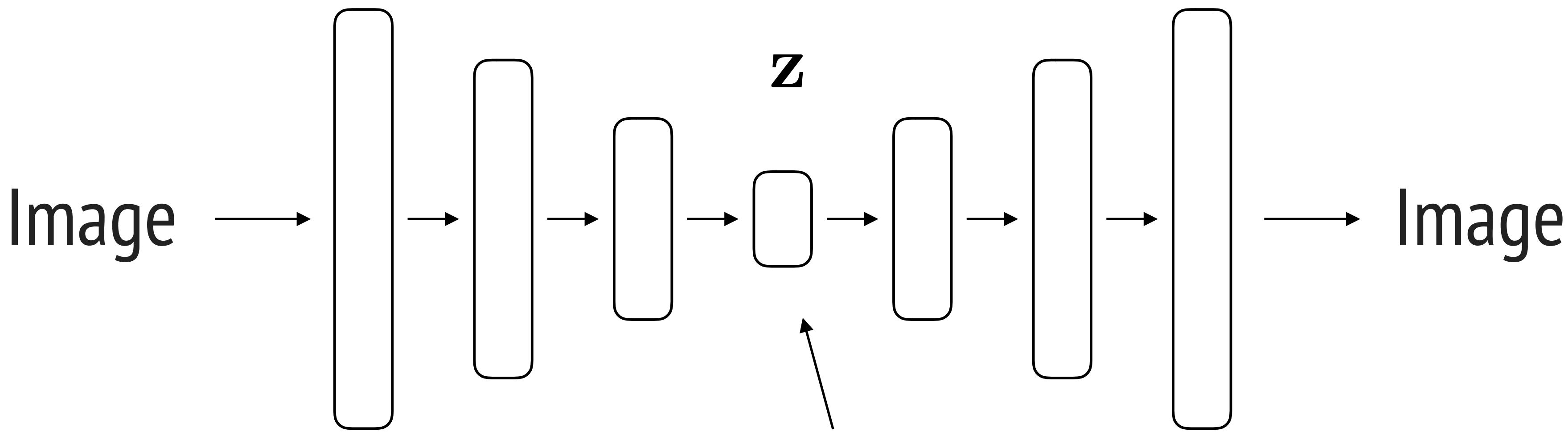


Examples: De-noising Auto Encoders



Bottleneck
Features
Representation
Code

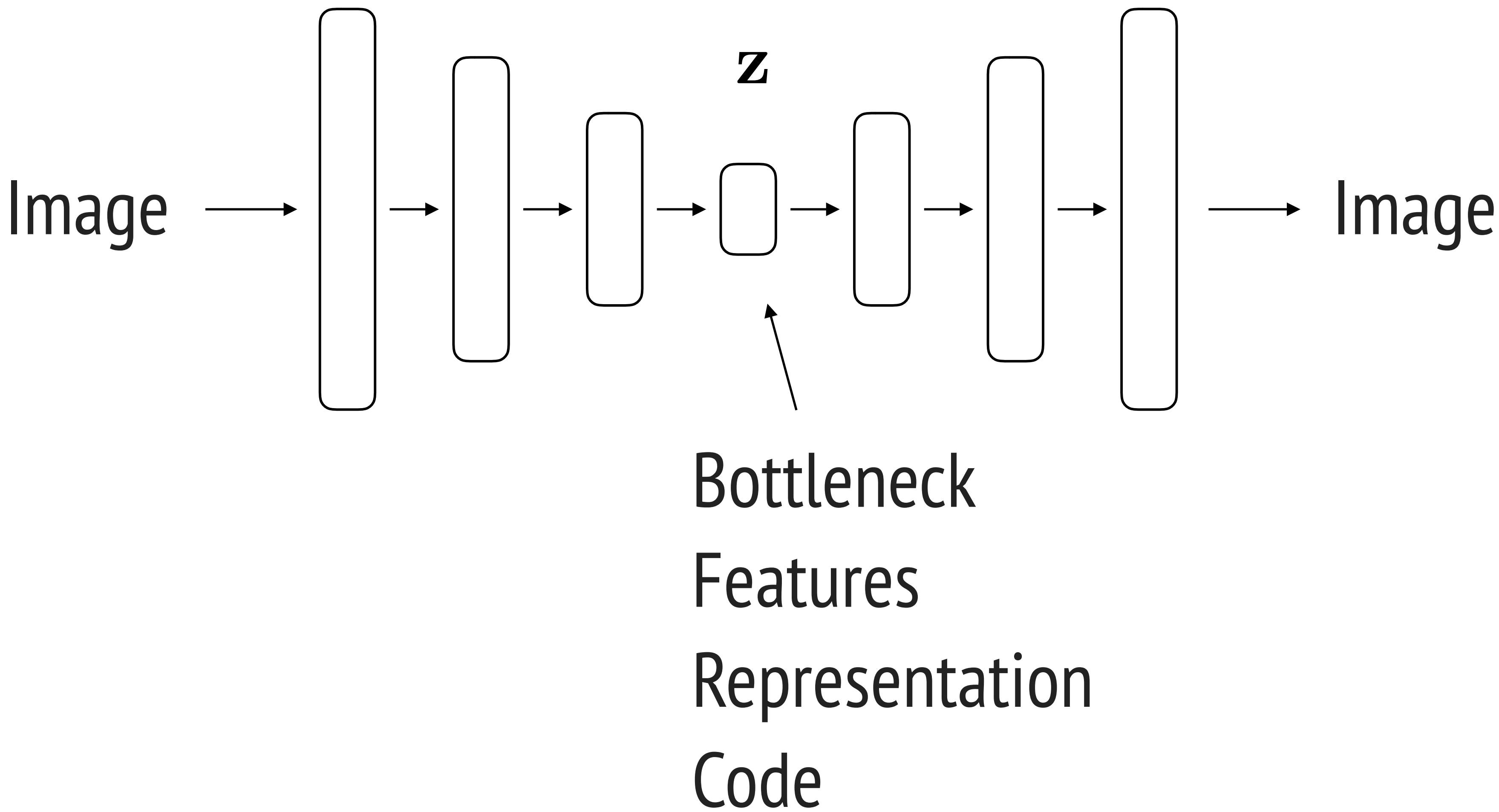
Examples: De-noising Auto Encoders



Bottleneck
Features
Representation
Code

Can we

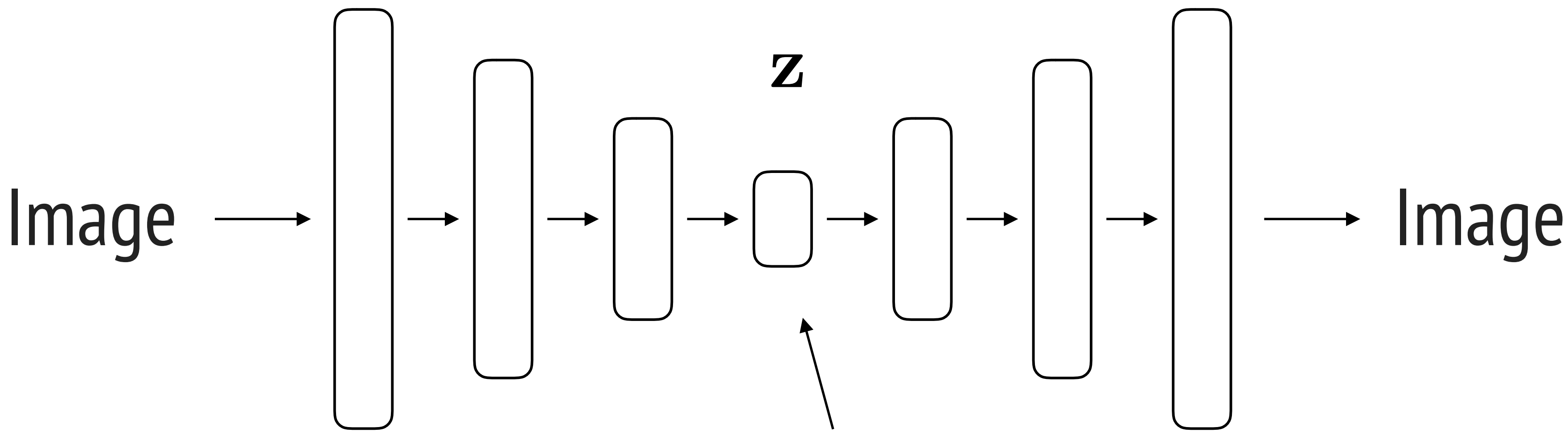
Examples: De-noising Auto Encoders



Can we

- generate an image?

Examples: De-noising Auto Encoders



Bottleneck

Features

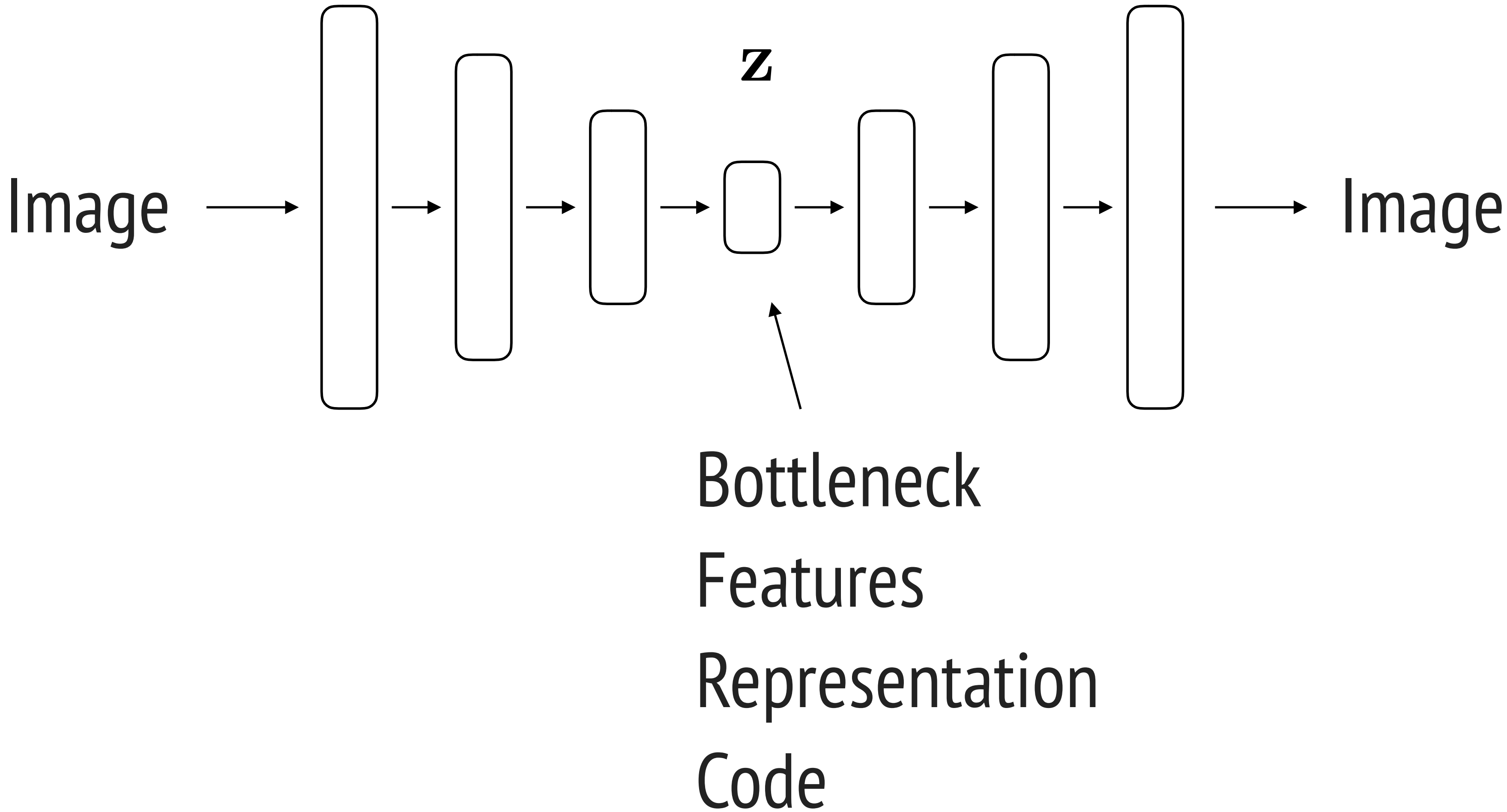
Representation

Code

Can we

- generate an image?
- change an image?

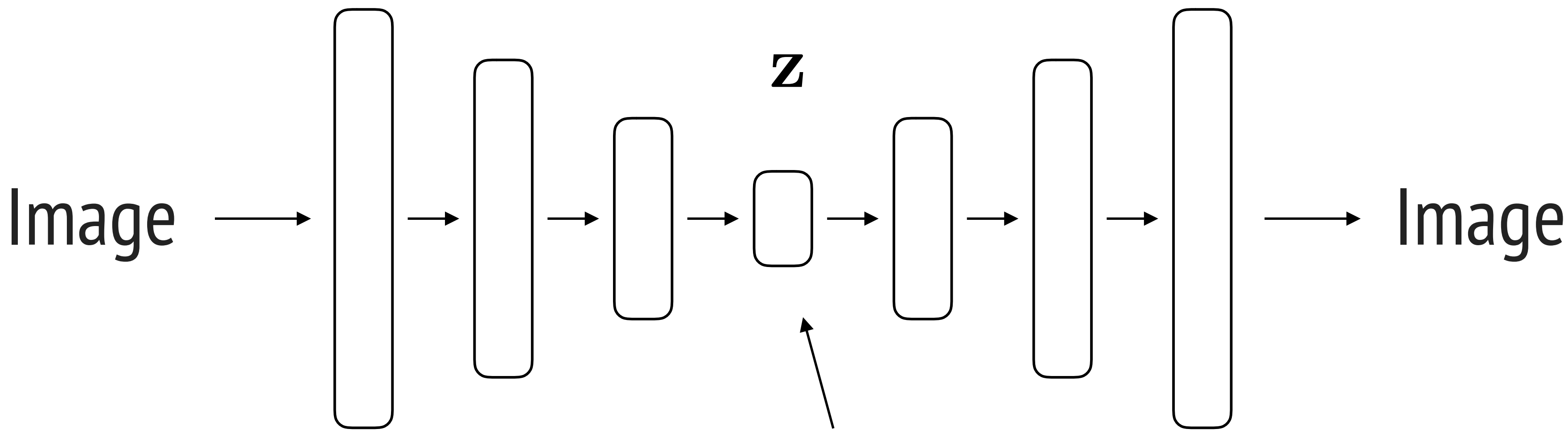
Examples: De-noising Auto Encoders



Can we

- generate an image?
- change an image?
- estimate likelihood?

Examples: De-noising Auto Encoders



Bottleneck

Features

Representation

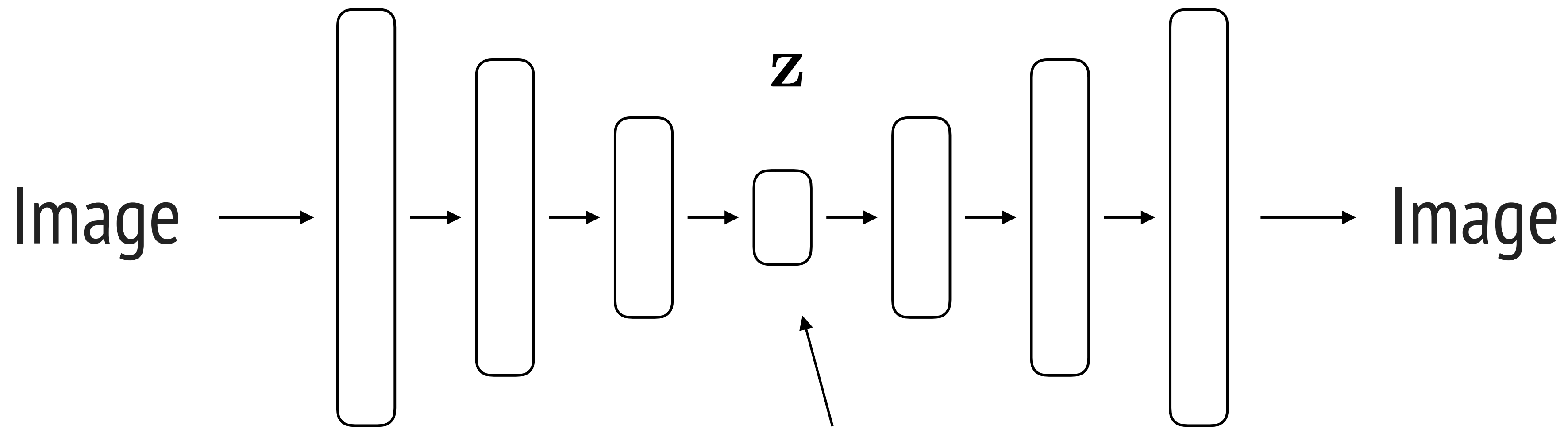
Code

Can we

- generate an image?
- change an image?
- estimate likelihood?

$$\mathbf{z} \sim p(\mathbf{z}), \;\; \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$

Examples: De-noising Auto Encoders



Bottleneck

Features

Representation

Code

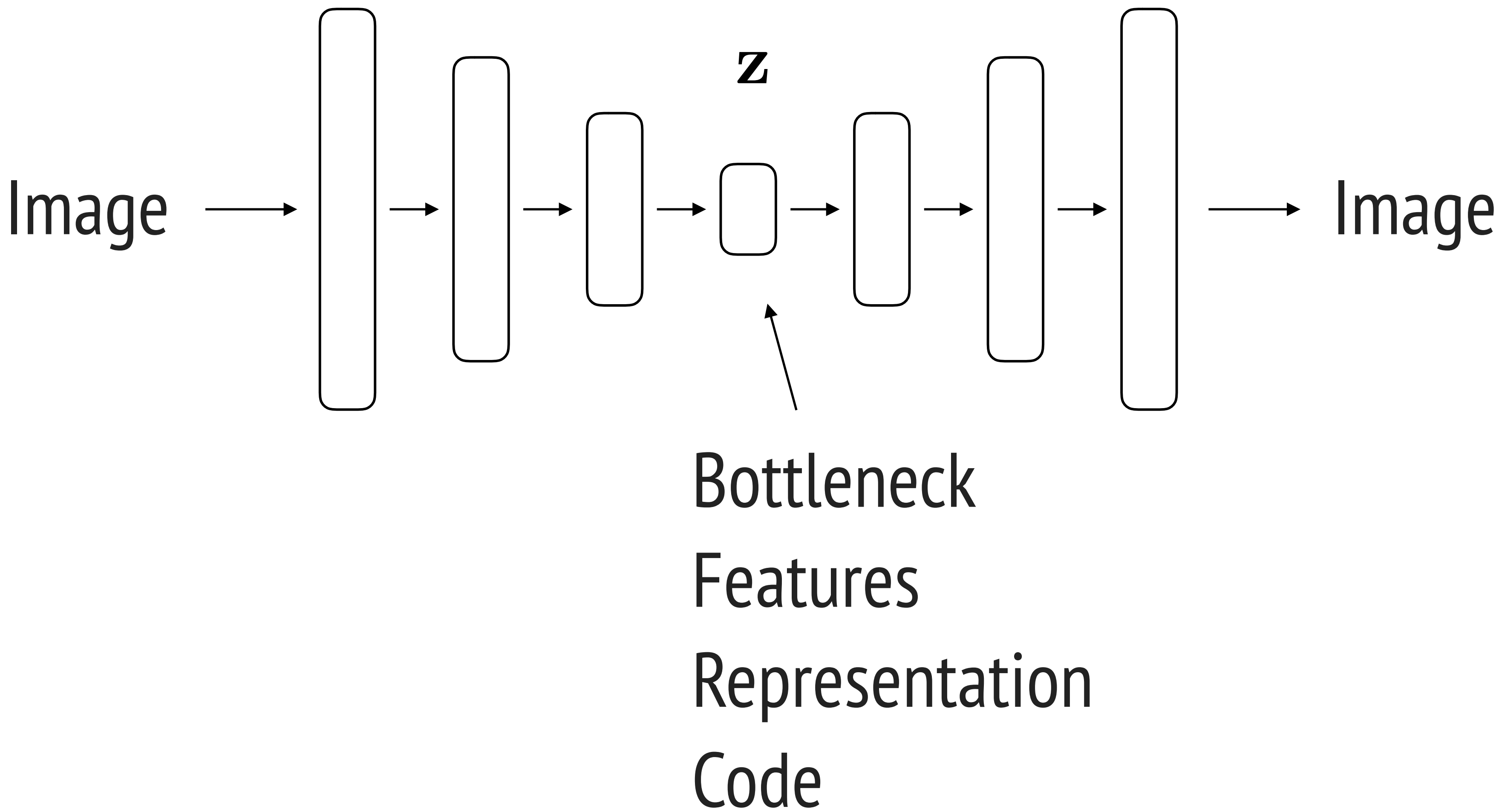
Can we

- generate an image?
- change an image?
- estimate likelihood?

$$\mathbf{z} \sim p(\mathbf{z}), \;\; \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$

$$\mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^*$$

Examples: De-noising Auto Encoders



Can we

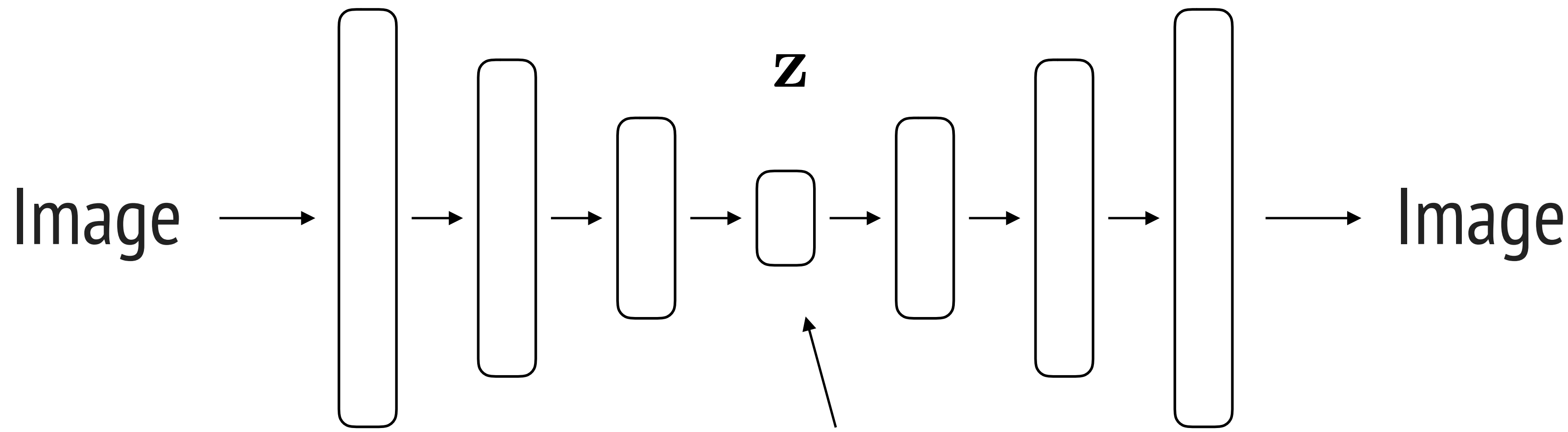
- generate an image?
- change an image?
- estimate likelihood?

$$\mathbf{z} \sim p(\mathbf{z}), \;\; \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$

$$\mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^*$$

$$\log p(\mathbf{x})$$

Examples: De-noising Auto Encoders



Bottleneck
Features
Representation
Code

Can we

- generate an image?
- change an image?
- estimate likelihood?

No or Tricky

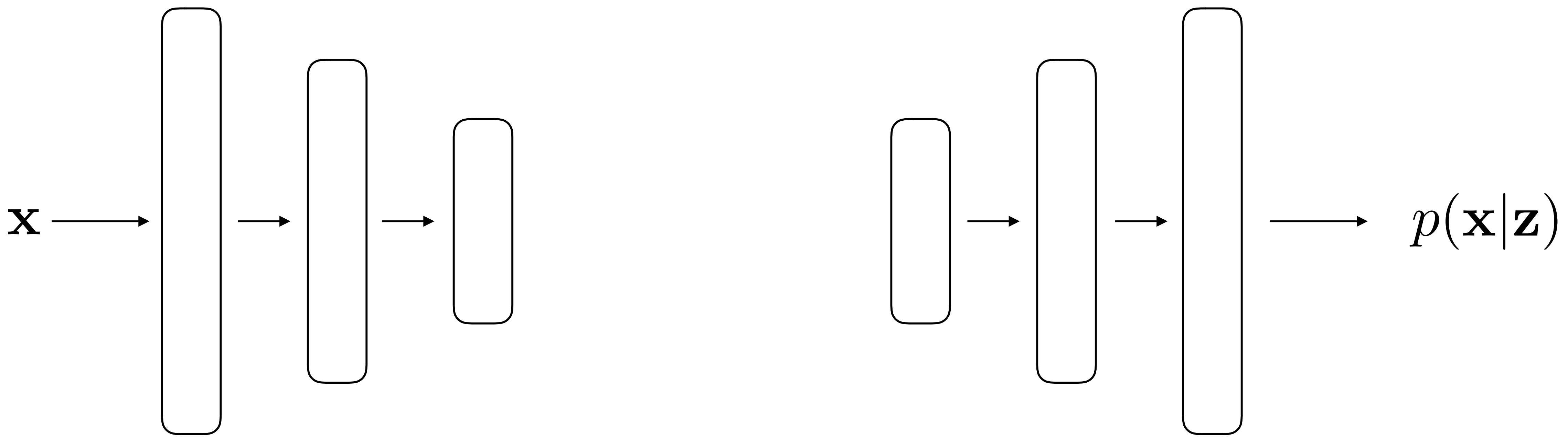
$$\mathbf{z} \sim p(\mathbf{z}), \;\; \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$

$$\mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^*$$

$$\log p(\mathbf{x})$$

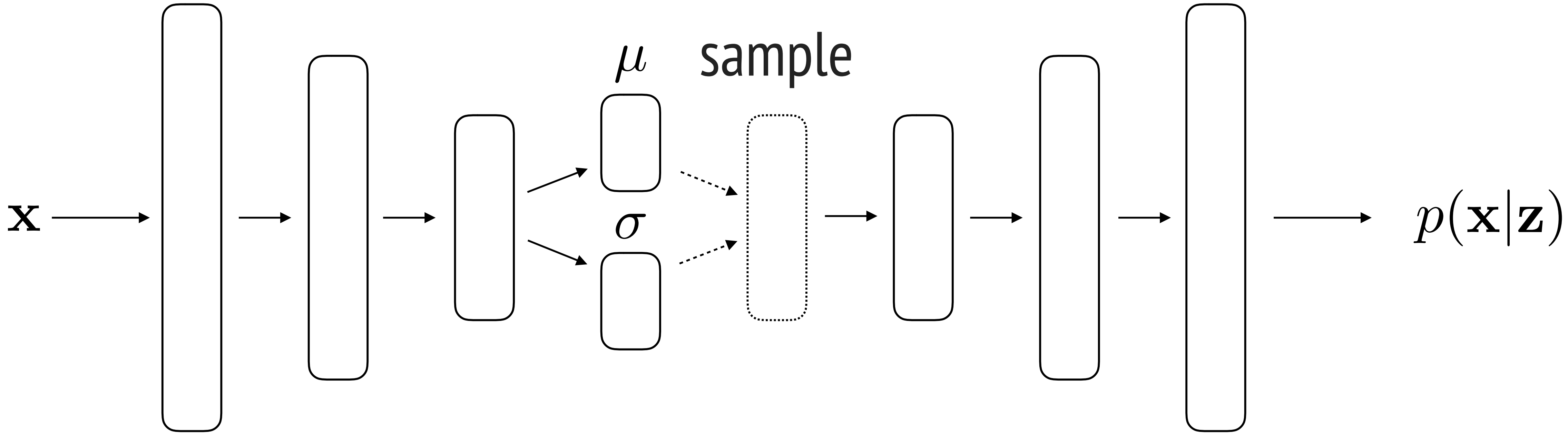
Variational Auto Encoders

$\mathbf{z} \sim p(\mathbf{z}),\ \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$ $\qquad$ $\mathbf{x} \to \mathbf{z} \to \mathbf{z}^* \to \mathbf{x}^*$ $\qquad$ $\log\, p(\mathbf{x})$



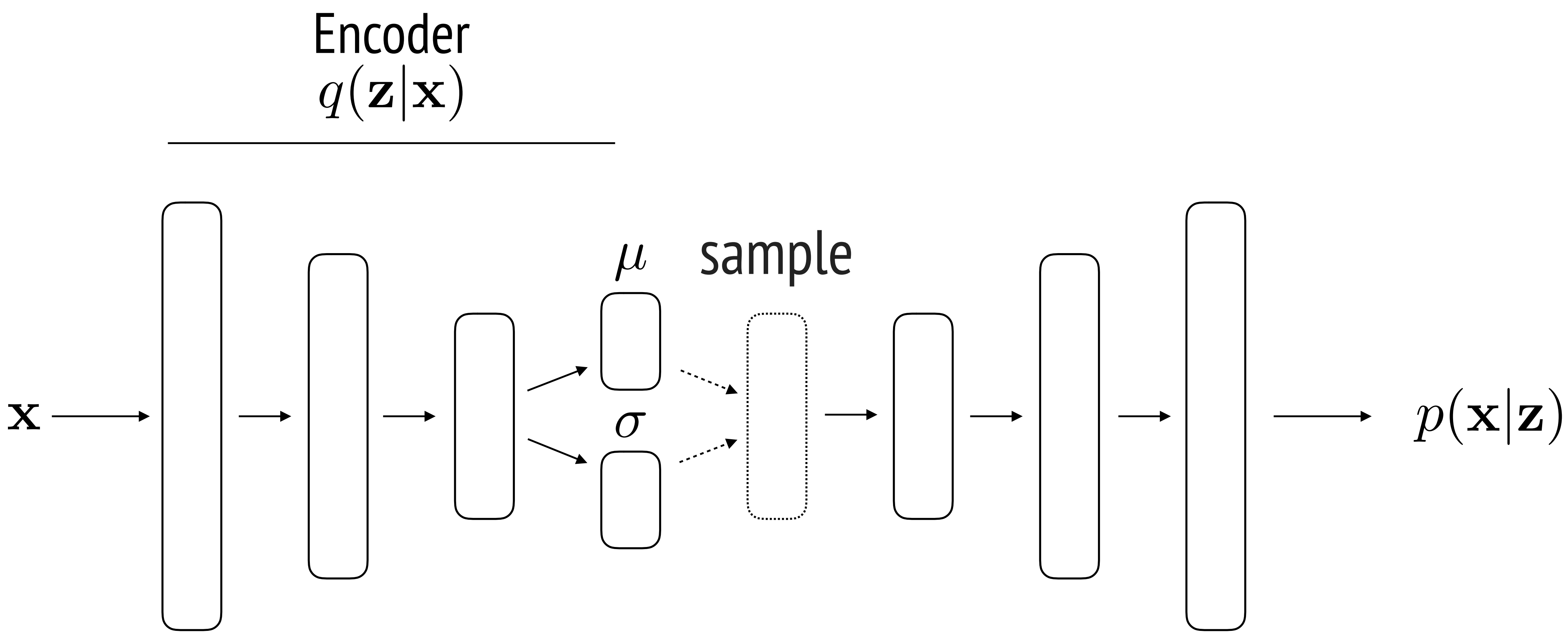
Variational Auto Encoders

$\mathbf{z} \sim p(\mathbf{z}), \ \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$ $\mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^*$ $\log\, p(\mathbf{x})$



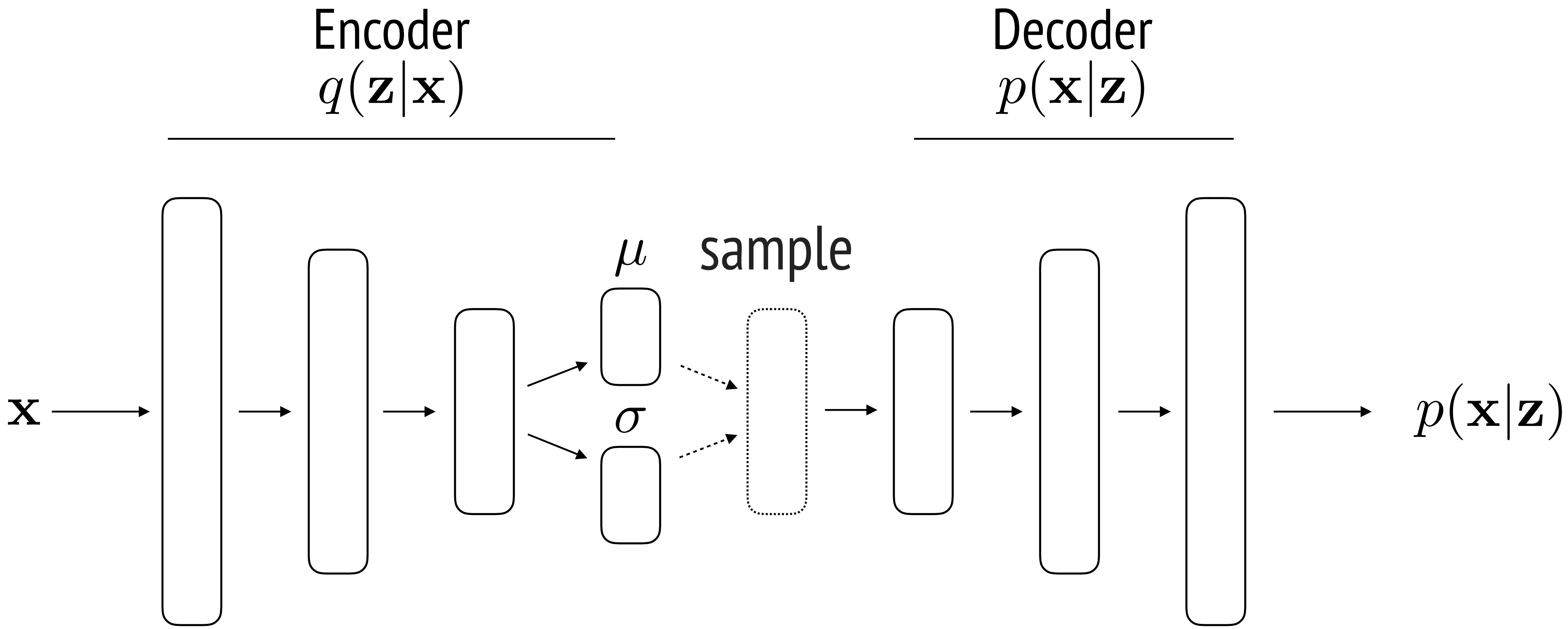
Variational Auto Encoders

$\mathbf{z} \sim p(\mathbf{z}), \ \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$ $\mathbf{x} \to \mathbf{z} \to \mathbf{z}^* \to \mathbf{x}^*$ $\log\, p(\mathbf{x})$



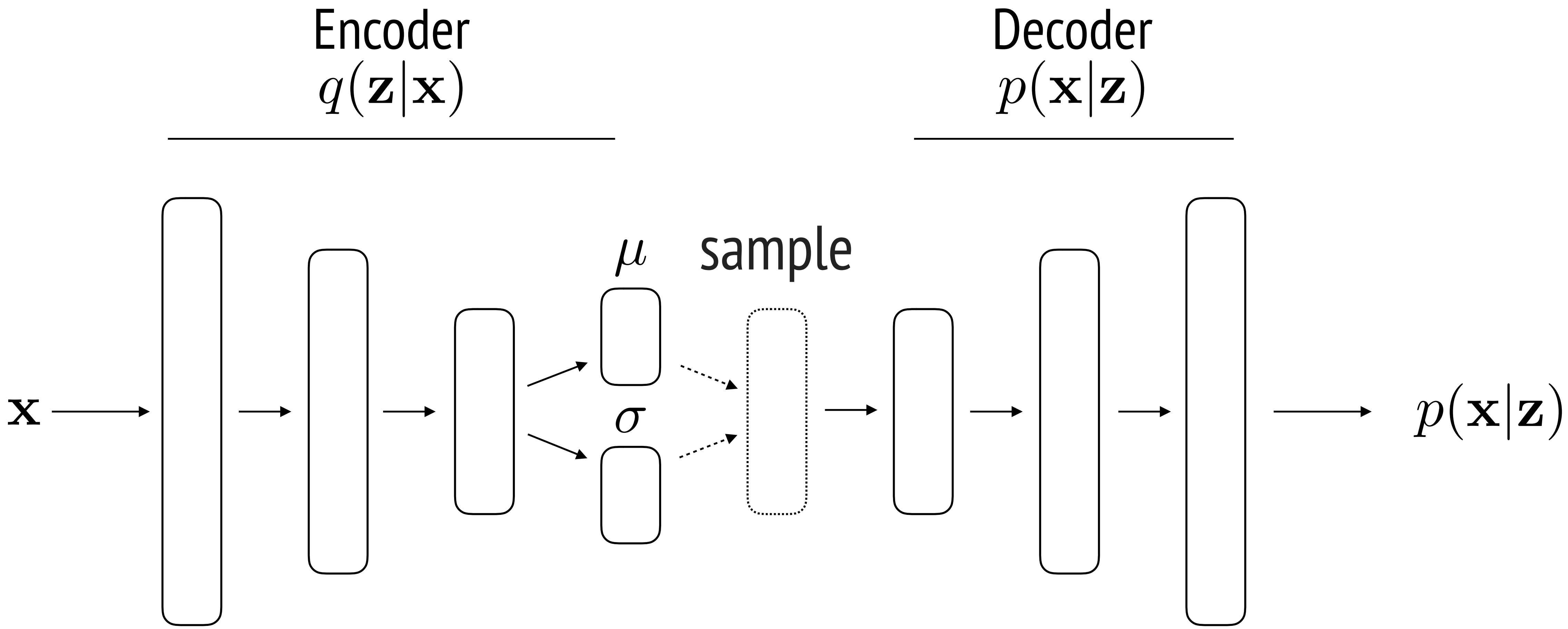
Variational Auto Encoders

$$\mathbf{z} \sim p(\mathbf{z}), \;\; \mathbf{x} \sim p(\mathbf{x}|\mathbf{z}) \qquad \mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^* \qquad \log\, p(\mathbf{x})$$



Variational Auto Encoders

$\mathbf{z} \sim p(\mathbf{z}),\ \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$ $\quad \mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^*$ $\quad \log\, p(\mathbf{x})$



$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z} \sim \mathbf{q}(\mathbf{z}|\mathbf{x})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D_{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Variational Auto Encoders

$$\mathbf{z} \sim p(\mathbf{z}),\ \mathbf{x} \sim p(\mathbf{x}|\mathbf{z}) \qquad \mathbf{x} \to \mathbf{z} \to \mathbf{z}^* \to \mathbf{x}^* \qquad \log\, p(\mathbf{x})$$

$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{p}(\mathbf{z})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Variational Auto Encoders

$$\mathbf{z} \sim p(\mathbf{z}),\ \mathbf{x} \sim p(\mathbf{x}|\mathbf{z}) \qquad \mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^* \qquad \log\ p(\mathbf{x})$$

Neural Networks

$$\log\ p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{p}(\mathbf{z})}[\log\ p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Variational Auto Encoders

$$\mathbf{z} \sim p(\mathbf{z}), \;\; \mathbf{x} \sim p(\mathbf{x}|\mathbf{z}) \qquad\qquad \mathbf{x} \to \mathbf{z} \to \mathbf{z}^* \to \mathbf{x}^* \qquad\qquad \log\, p(\mathbf{x})$$

Neural Networks

$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{p}(\mathbf{z})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Evaluate Likelihood

Variational Auto Encoders

$$\mathbf{z} \sim p(\mathbf{z}), \;\; \mathbf{x} \sim p(\mathbf{x}|\mathbf{z}) \qquad \mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^* \qquad \log\, p(\mathbf{x})$$

Neural Networks

$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{p(z)}}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Evaluate Likelihood

Expectation over sampling

Variational Auto Encoders

$\mathbf{z} \sim p(\mathbf{z}),\ \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$ $\quad$ $\mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^*$ $\quad$ $\log\, p(\mathbf{x})$

Neural Networks

$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z} \sim \mathbf{p}(\mathbf{z})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Evaluate Likelihood

Generate Image

Expectation over sampling

Variational Auto Encoders

$$\mathbf{z} \sim p(\mathbf{z}), \;\; \mathbf{x} \sim p(\mathbf{x}|\mathbf{z}) \qquad \mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^* \qquad \log\, p(\mathbf{x})$$

Neural Networks

$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{p}(\mathbf{z})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Evaluate Likelihood

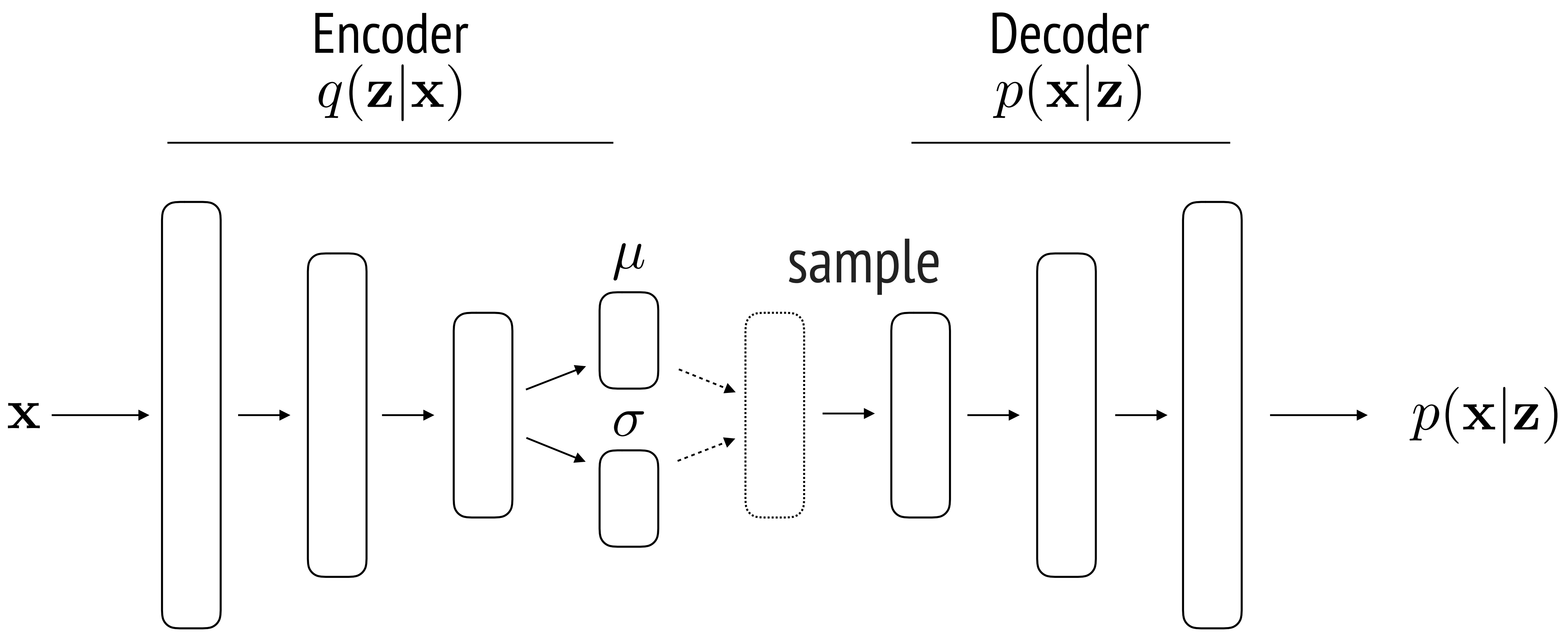
Generate Image

Expectation over sampling

“Regularization”

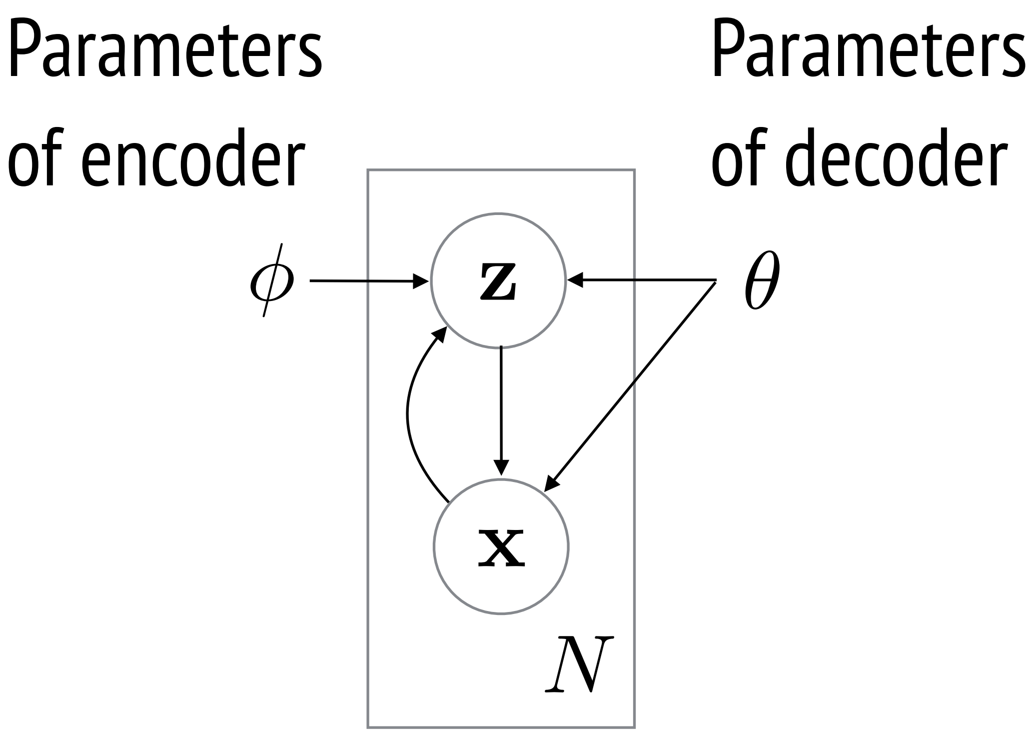
Variational Auto Encoders

$$\mathbf{z} \sim p(\mathbf{z}),\; \mathbf{x} \sim p(\mathbf{x}|\mathbf{z}) \qquad \mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^* \qquad \log\, p(\mathbf{x})$$

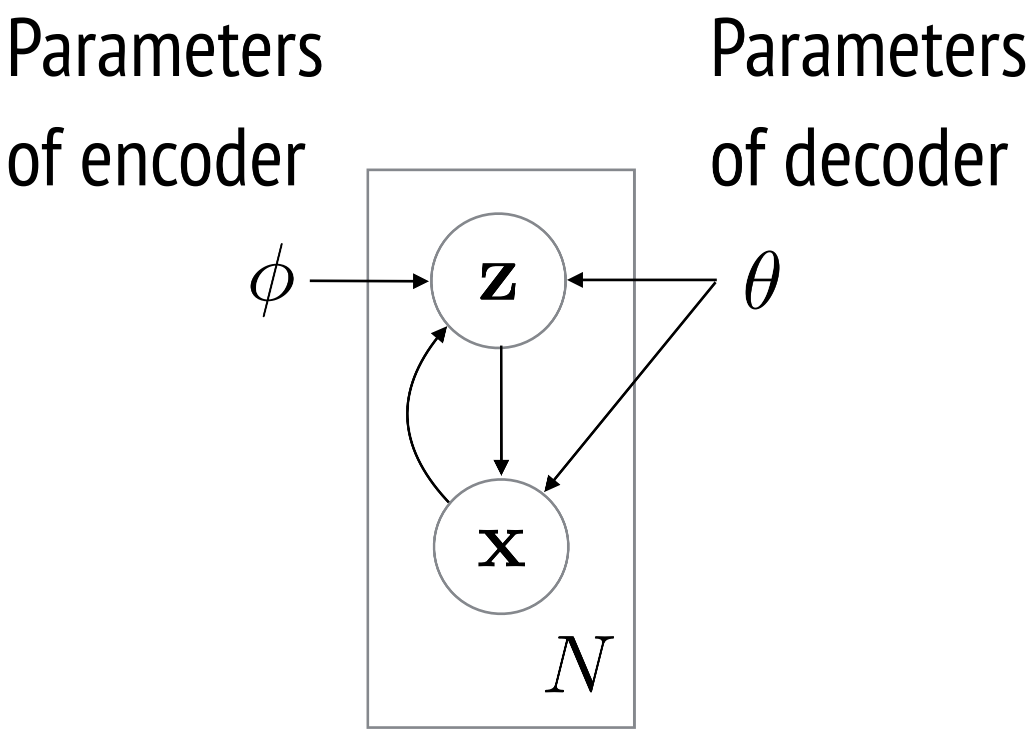


$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D_{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Graphical Model Perspective



Graphical Model Perspective

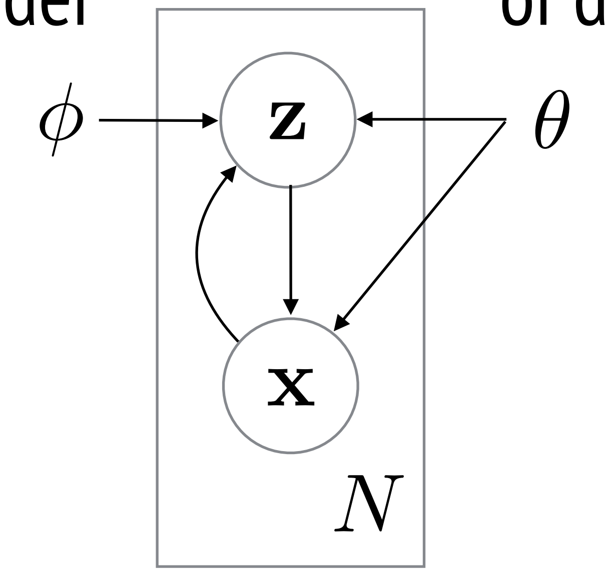


$$\log p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Graphical Model Perspective

Parameters of encoder

Parameters of decoder



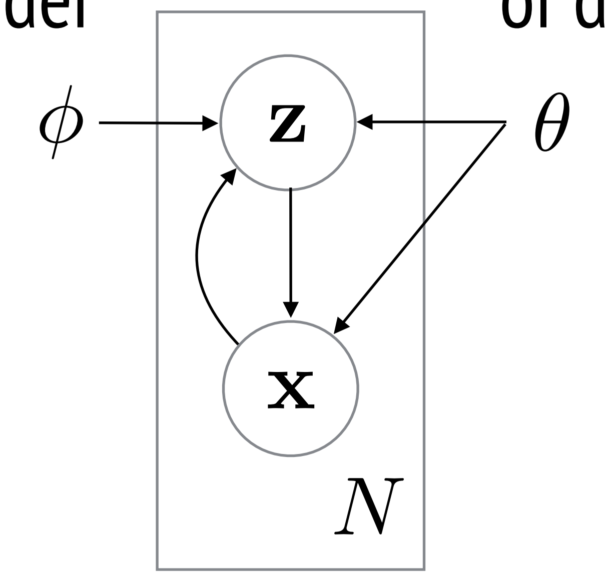
Intractable

$$\log p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Graphical Model Perspective

Parameters of encoder

Parameters of decoder



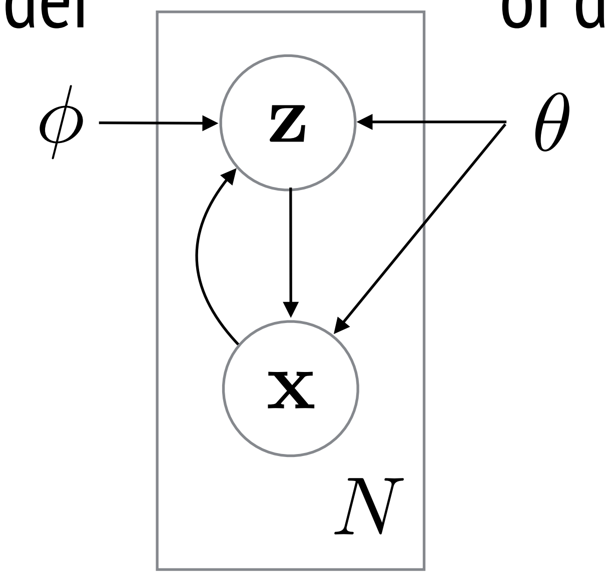
Intractable

$$\log p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Graphical Model Perspective

Parameters of encoder

Parameters of decoder



Intractable

$$\log p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

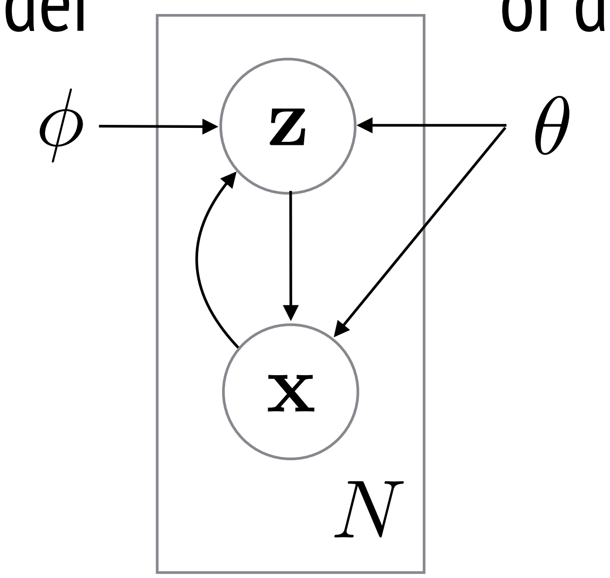
How to choose prior?

How to evaluate the likelihood?

Graphical Model Perspective

Parameters of encoder

Parameters of decoder



Intractable

$$\log p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

How to choose prior?

How to evaluate the likelihood?

Approximate it with with Variational Inference

Derivation

- Via KL-Divergence

$$D_{\mathrm{KL}}(P||Q) = -\sum_{x\in\mathcal{X}} P(x)\log\left(\frac{Q(x)}{P(x)}\right) = -\mathbb{E}_{x\sim P(x)}\left[\log\left(\frac{Q(x)}{P(x)}\right)\right]$$

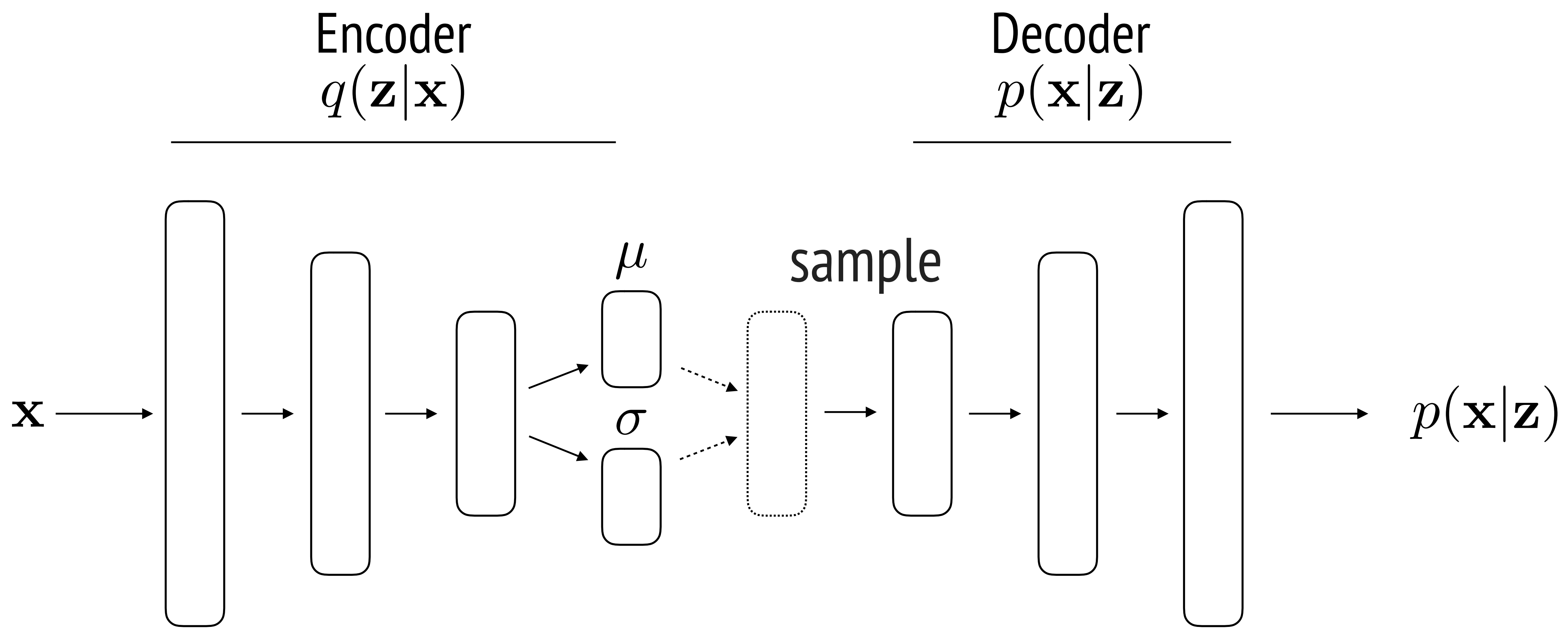
- Via Likelihood

$$\log\, p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Variational Auto Encoders

$$\mathbf{z} \sim p(\mathbf{z}), \;\; \mathbf{x} \sim p(\mathbf{x}|\mathbf{z}) \qquad \mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{z}^* \rightarrow \mathbf{x}^* \qquad \log\, p(\mathbf{x})$$



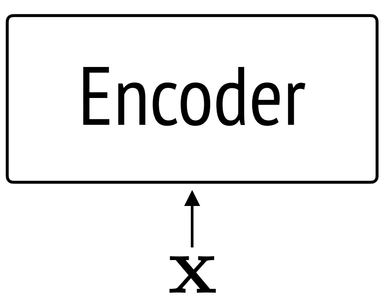
$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Training VAEs

$$\uparrow$$
$$\mathbf{x}$$

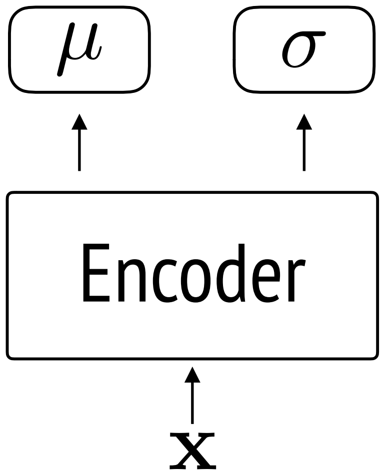
$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Training VAEs



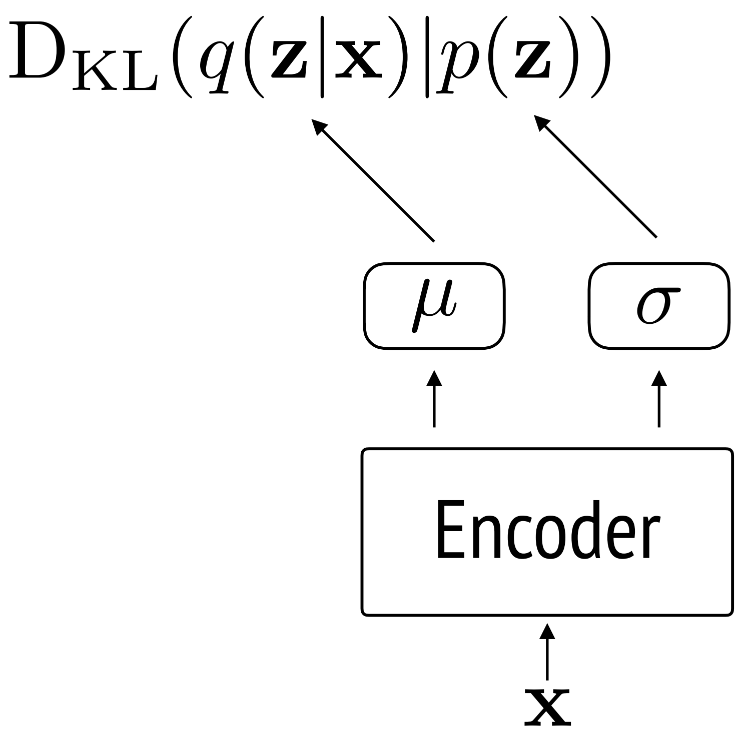
$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Training VAEs



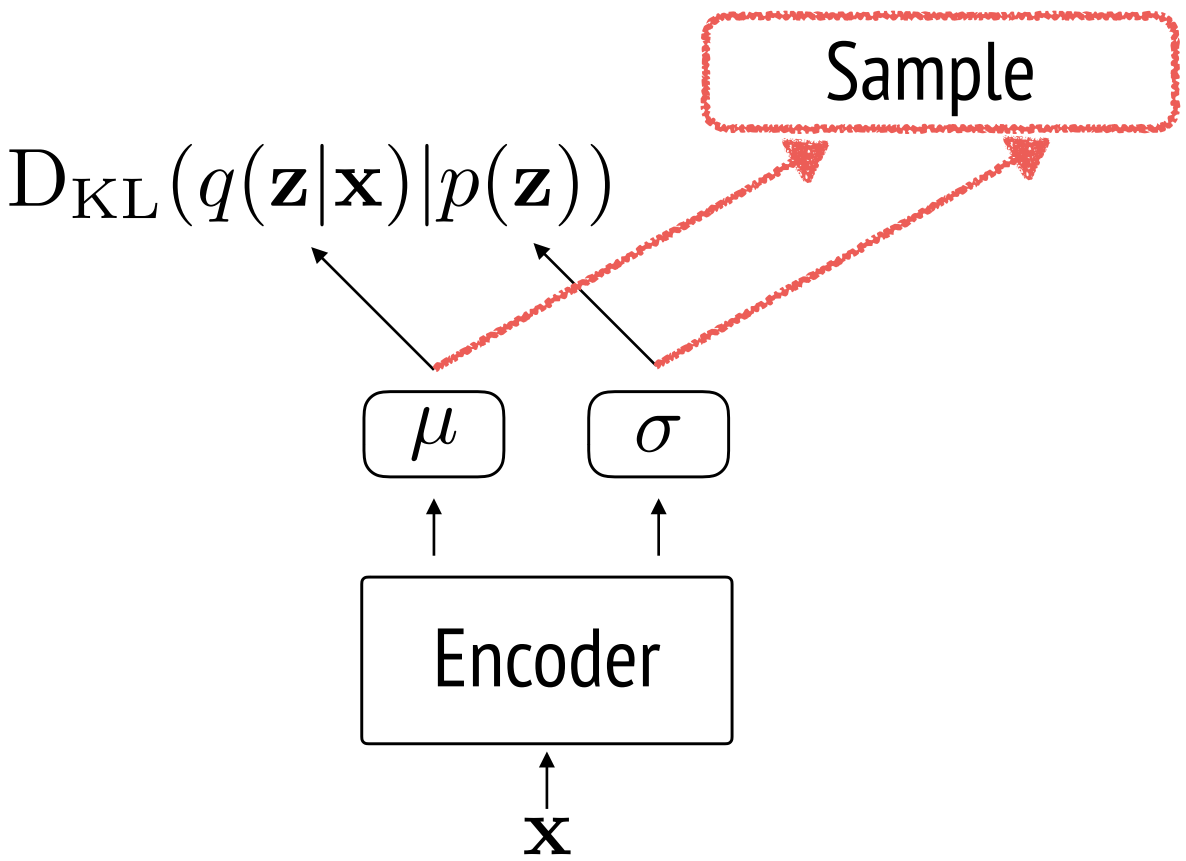
$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Training VAEs



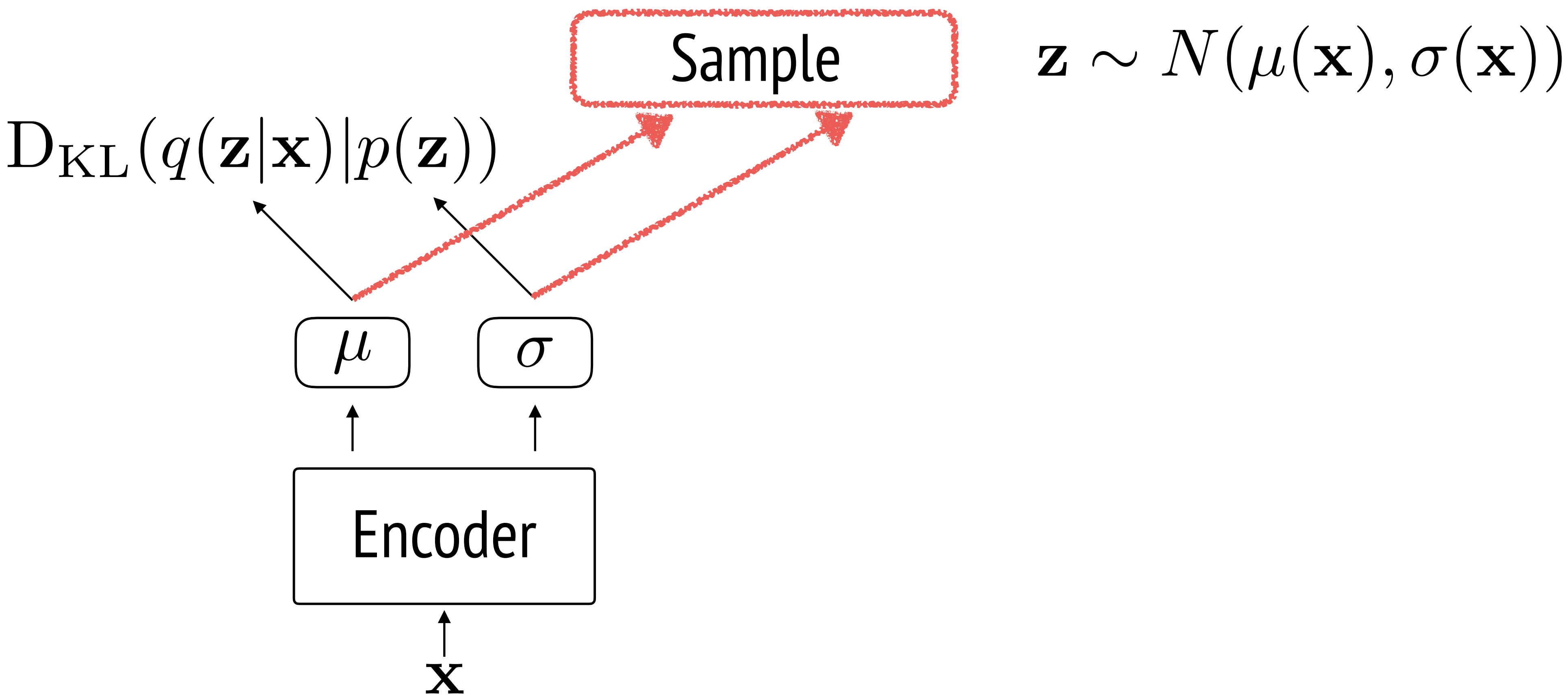
$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{D_{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Training VAEs



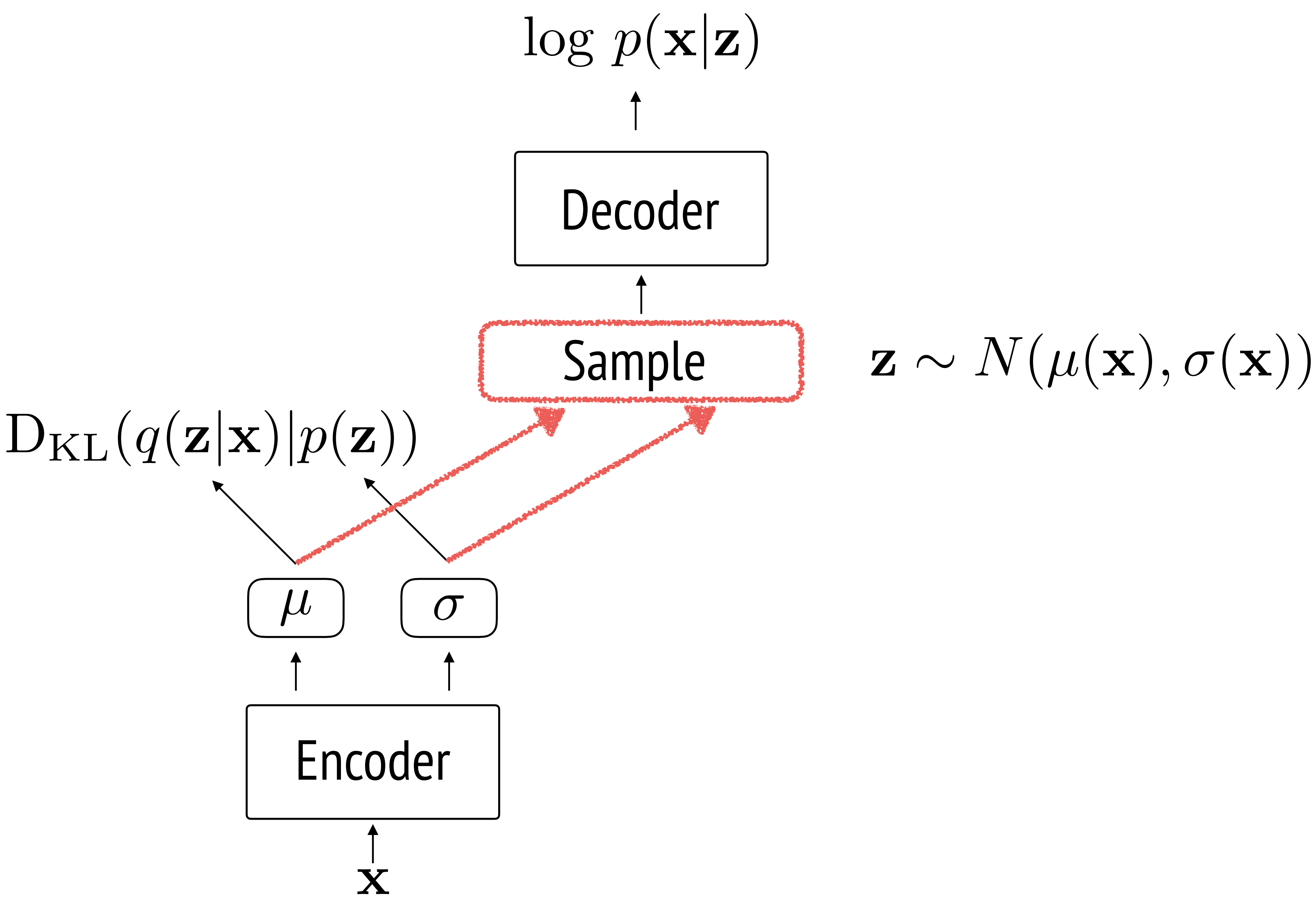
$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Training VAEs



$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D_{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Training VAEs



$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{D_{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Reparametrization Trick

Given some Gaussian how to make it Normal?

$$N(\mu, \sigma) \rightarrow N(0, 1)$$

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z} \sim \mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Reparametrization Trick

Given some Gaussian how to make it Normal?

$$N(\mu, \sigma) \rightarrow N(0,1)$$

Given a Normal how to make it any Gaussian?

$$N(0,1) \rightarrow N(\mu(\mathbf{x}), \sigma(\mathbf{x}))$$

$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Reparametrization Trick

Given some Gaussian how to make it Normal?

$$N(\mu, \sigma) \rightarrow N(0,1)$$

Given a Normal how to make it any Gaussian?

$$N(0,1) \rightarrow N(\mu(\mathbf{x}), \sigma(\mathbf{x}))$$

Reparametrization

$$\mathbf{z} \sim N(\mu(\mathbf{x}), \sigma(\mathbf{x}))$$

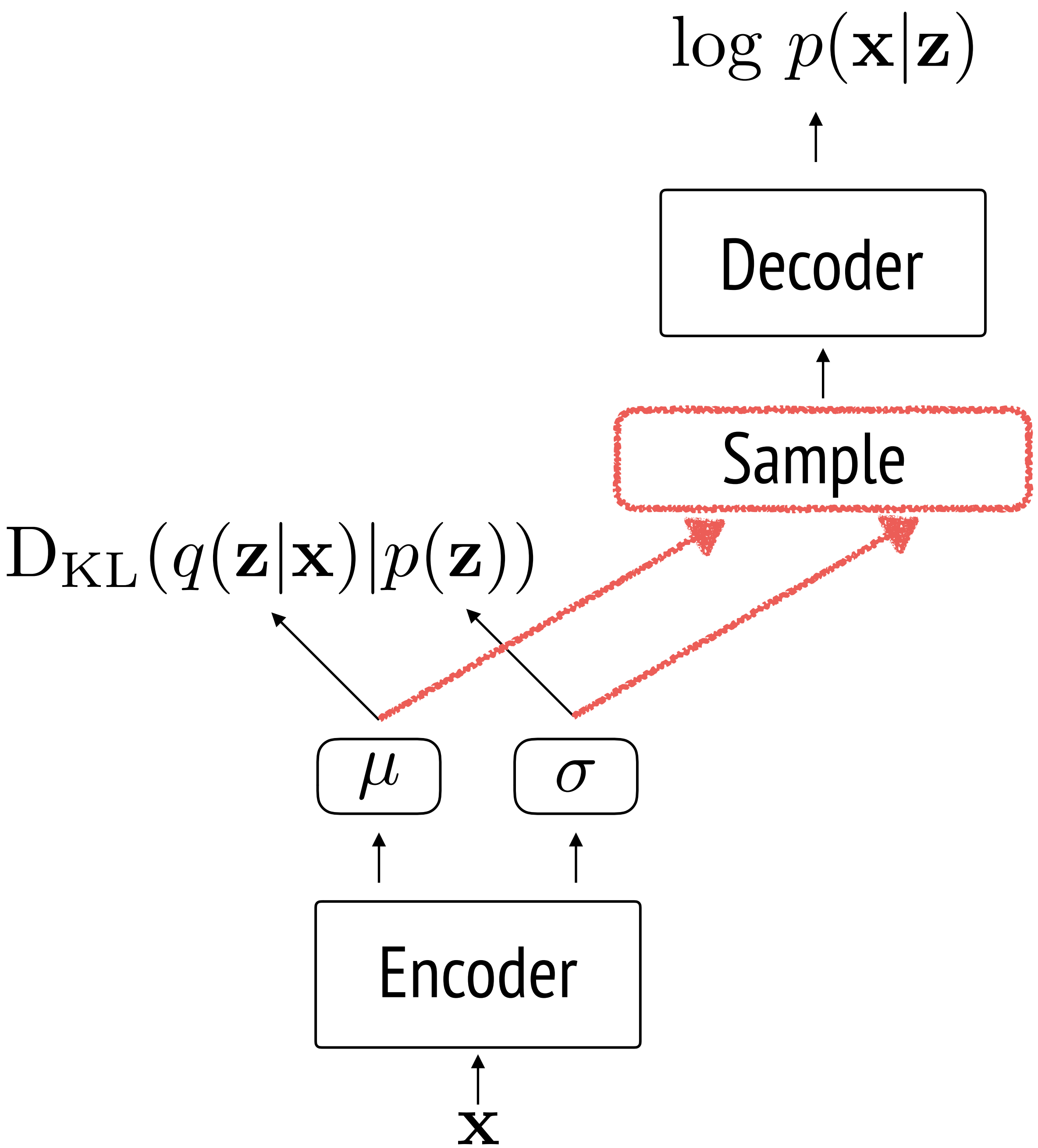
$$\downarrow$$

$$\epsilon \sim N(0,1)$$

$$\mathbf{z} = \mu(\mathbf{z}) + \epsilon \cdot \sigma(\mathbf{z})$$

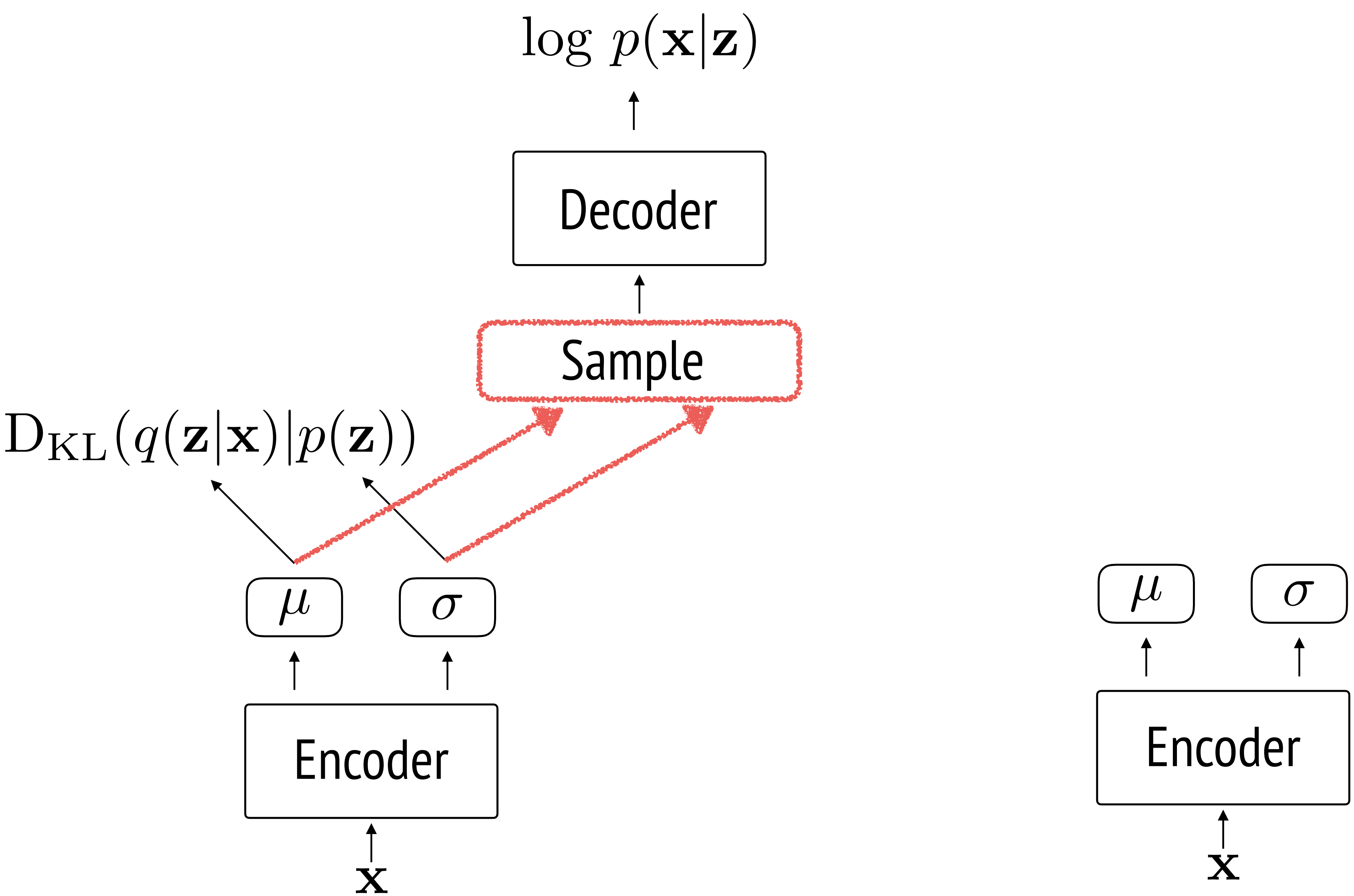
$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z} \sim \mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{D_{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Sampling



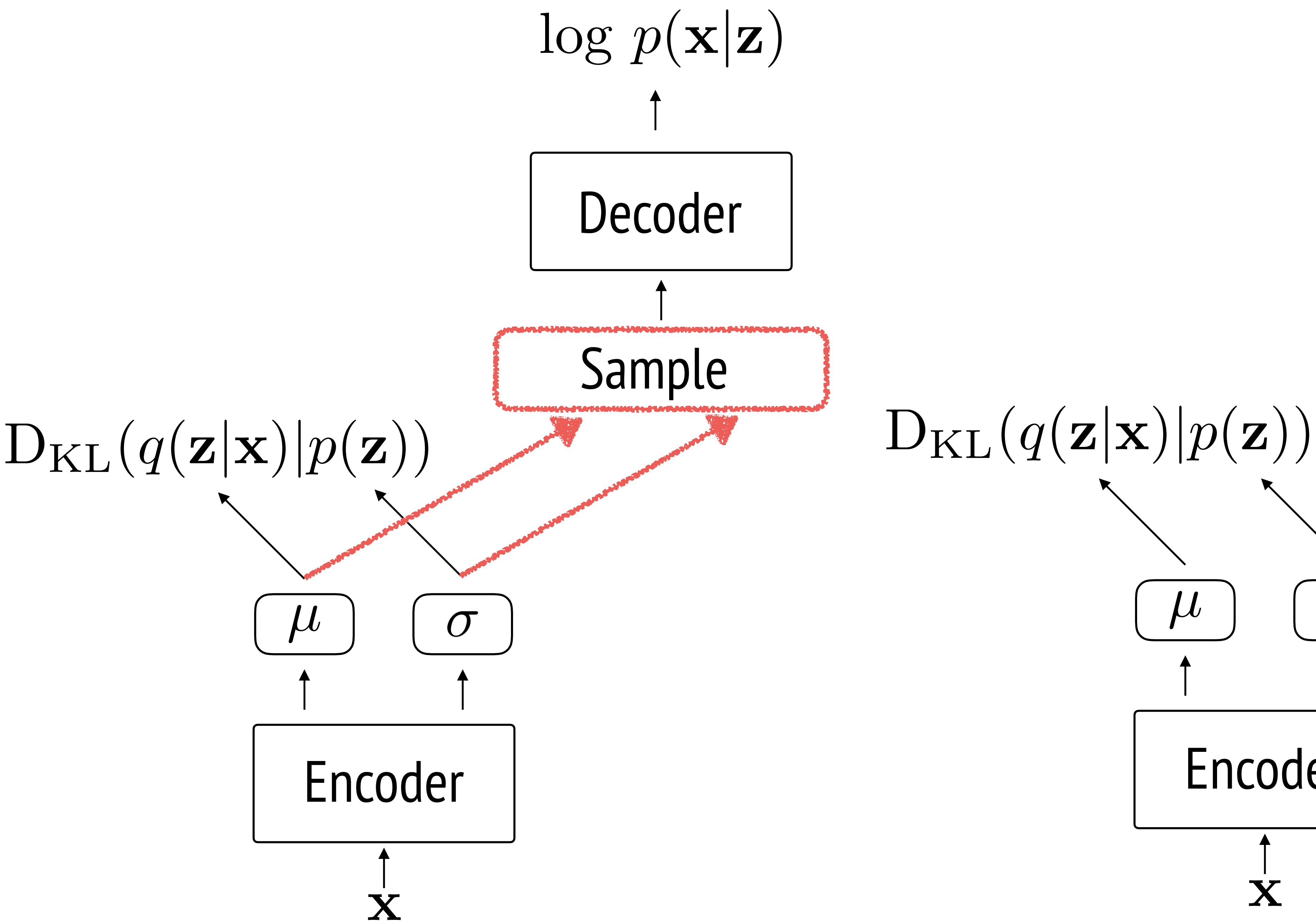
$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{D_{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Sampling



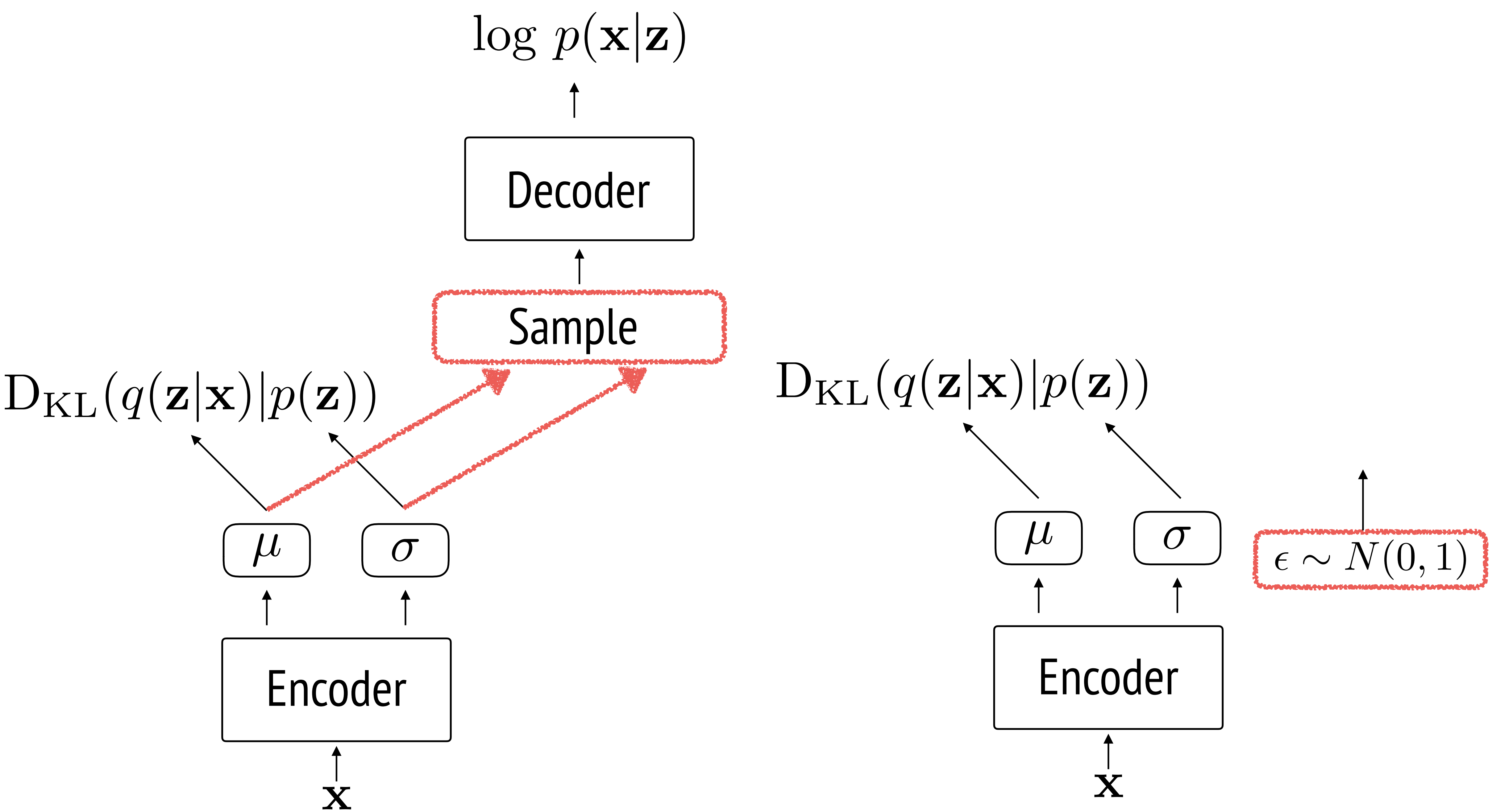
$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Sampling



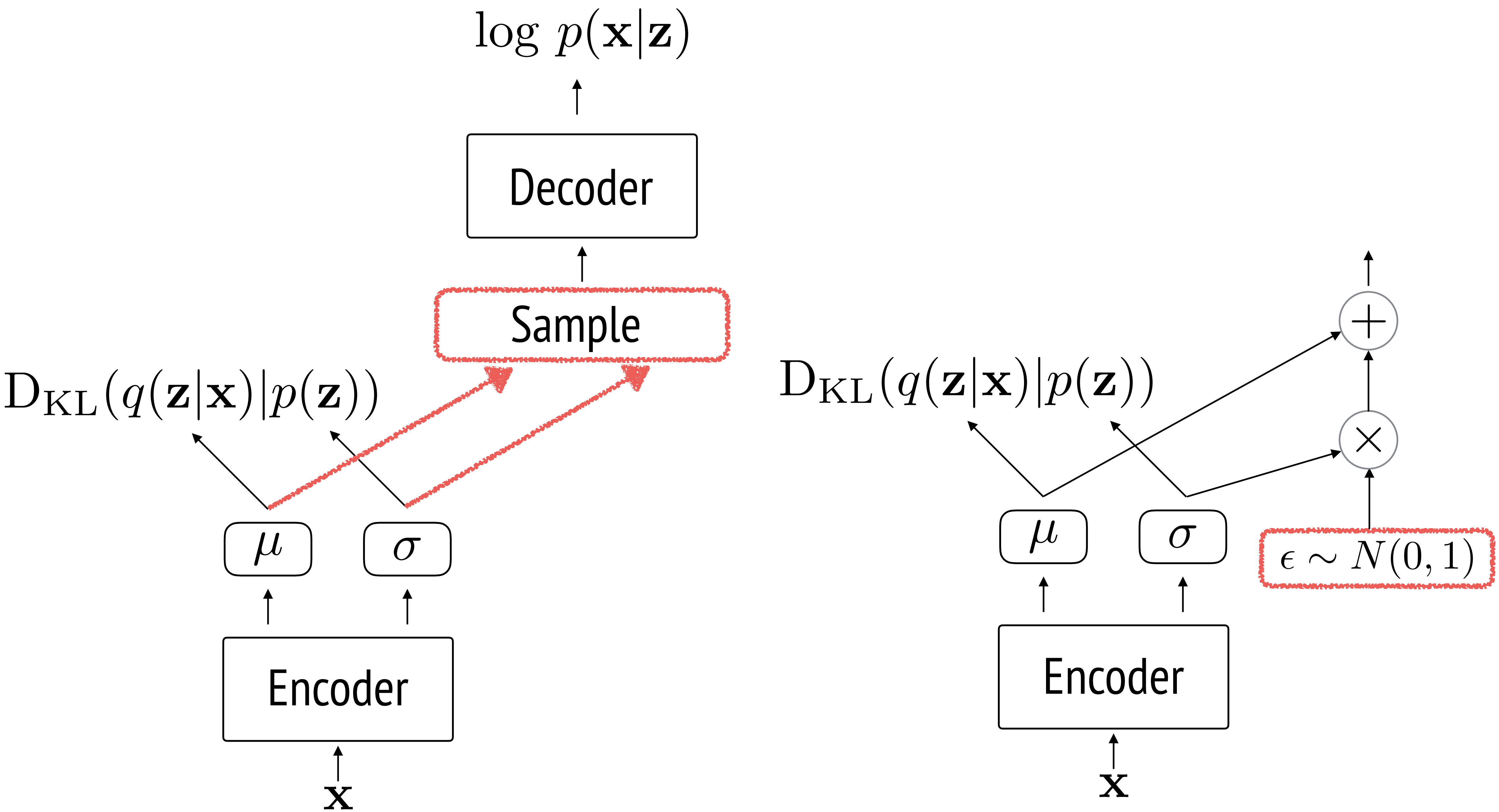
$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D_{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Sampling



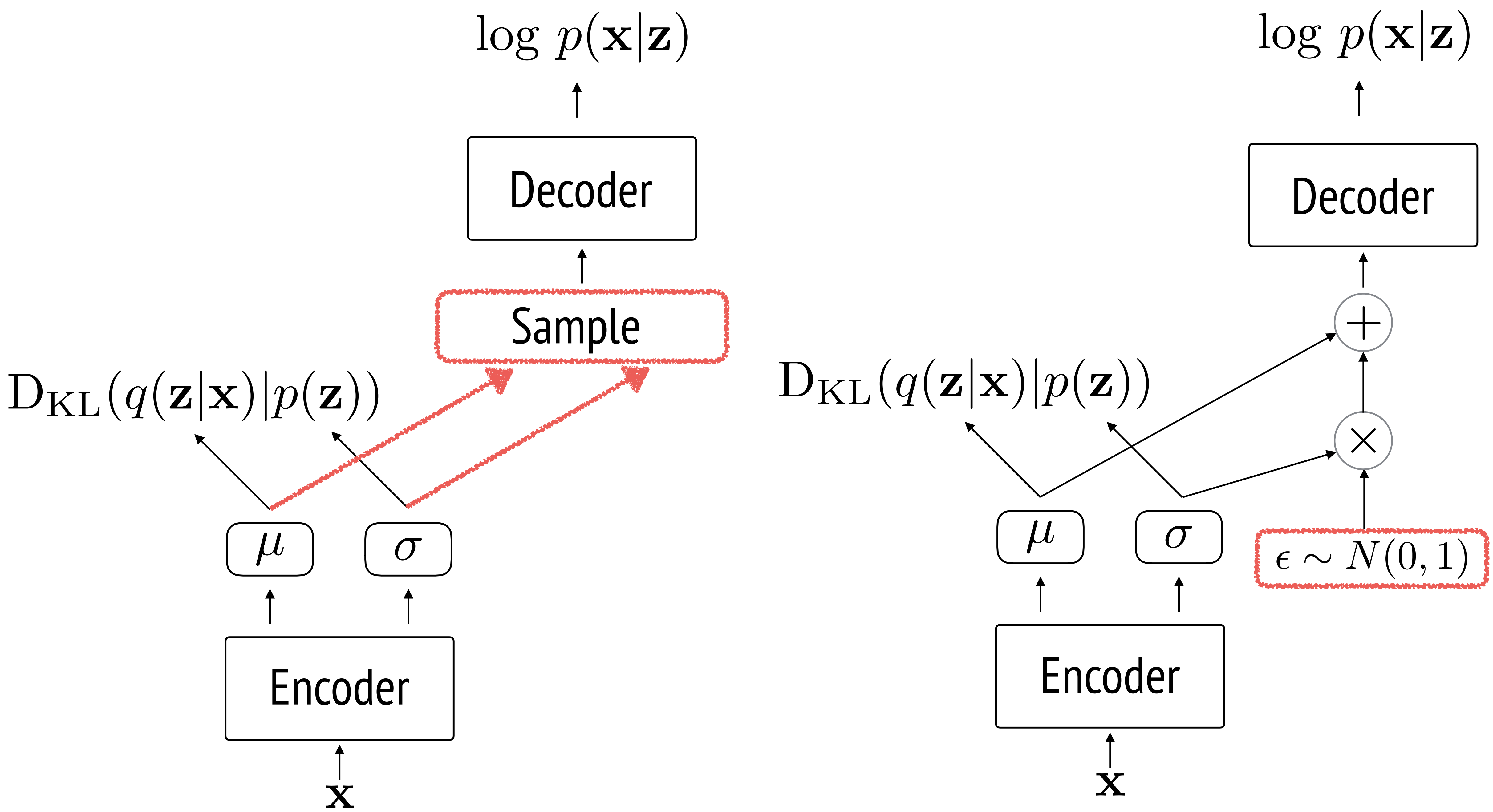
$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - D_{KL}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Sampling



$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Sampling



$$\log\, p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log\, p(\mathbf{x}|\mathbf{z})] - \mathrm{D_{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Reconstruction

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\underline{\log p(\mathbf{x}|\mathbf{z})}] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Reconstruction

- Gaussian likelihood, l2-loss

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\underline{\log p(\mathbf{x}|\mathbf{z})}] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Reconstruction

- Gaussian likelihood, l2-loss

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Laplace likelihood, l1-loss

$$f(x|\mu,b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\underline{\log p(\mathbf{x}|\mathbf{z})}] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Reconstruction

- Gaussian likelihood, l2-loss

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Laplace likelihood, l1-loss

$$f(x|\mu,b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$

Predicted by NN

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Reconstruction

- Gaussian likelihood, l2-loss

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Laplace likelihood, l1-loss

Data

Predicted by NN

$$f(x|\mu,b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\underline{\log p(\mathbf{x}|\mathbf{z})}] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: Reconstruction

- Gaussian likelihood, l2-loss

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Predicted by NN

- Laplace likelihood, l1-loss

Data

$$f(x|\mu,b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$

- Perception-loss, l1 or l2 loss in the feature space of a pre-trained network, such as VGG, MobileNet etc

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}\left[\underline{\log p(\mathbf{x}|\mathbf{z})}\right] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Computing Objective: KL Divergence

$$D_{\mathrm{KL}}(P||Q) = -\sum_{x\in\mathcal{X}} P(x)\log\left(\frac{Q(x)}{P(x)}\right) = -\mathbb{E}_{x\sim P(x)}\left[\log\left(\frac{Q(x)}{P(x)}\right)\right]$$

- Laplace, Gaussian - closed form
- Sampling - when closed form is not available

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{D_{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Using a VAE

$\epsilon \sim N(0,1) \rightarrow \boxed{\text{Decoder}}$

Using a VAE

$$\epsilon \sim N(0,1) \rightarrow \boxed{\text{Decoder}}$$

Using a VAE

$$\epsilon \sim N(0,1) \rightarrow \boxed{\text{Decoder}}$$

$$\epsilon_1 \sim N(0,1)$$

$$\vdots$$

$$\epsilon_k \sim N(0,1)$$

Decoder

Using a VAE

$\epsilon \sim N(0,1) \rightarrow$ Decoder

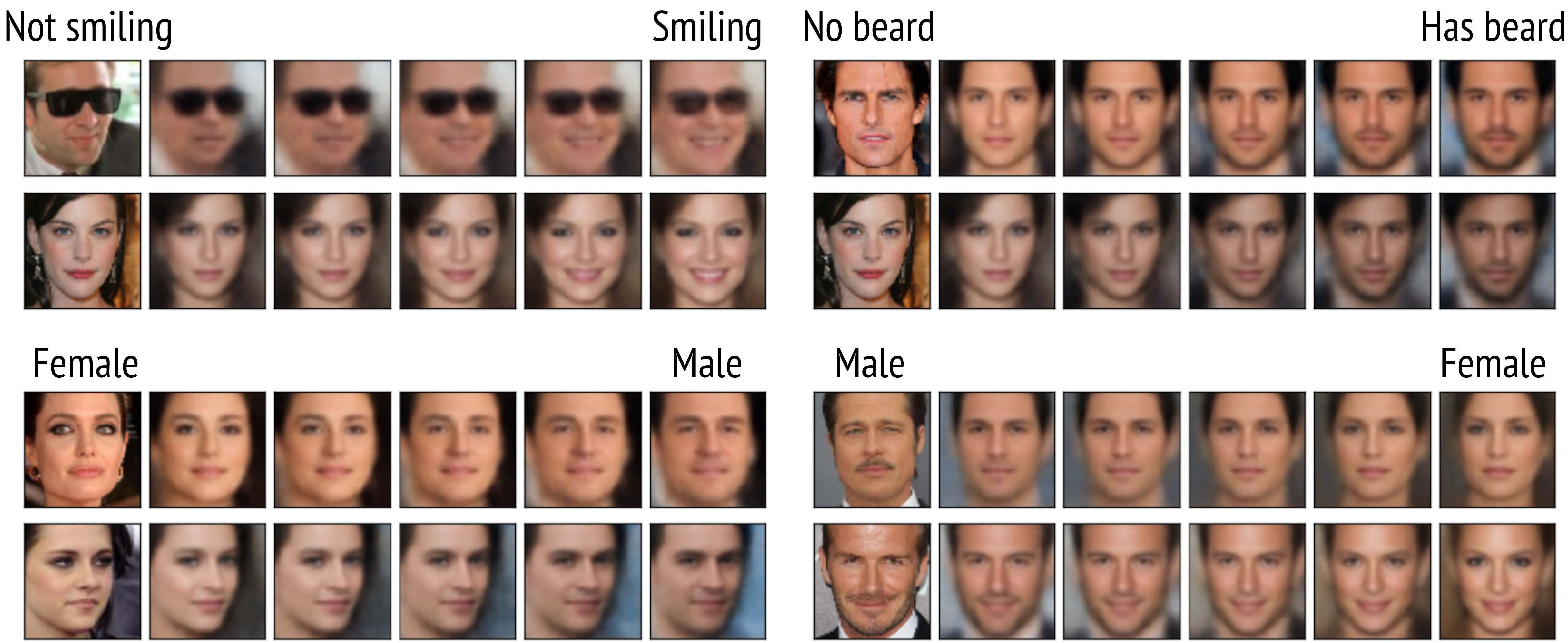
$\epsilon_1 \sim N(0,1)$

$\vdots$

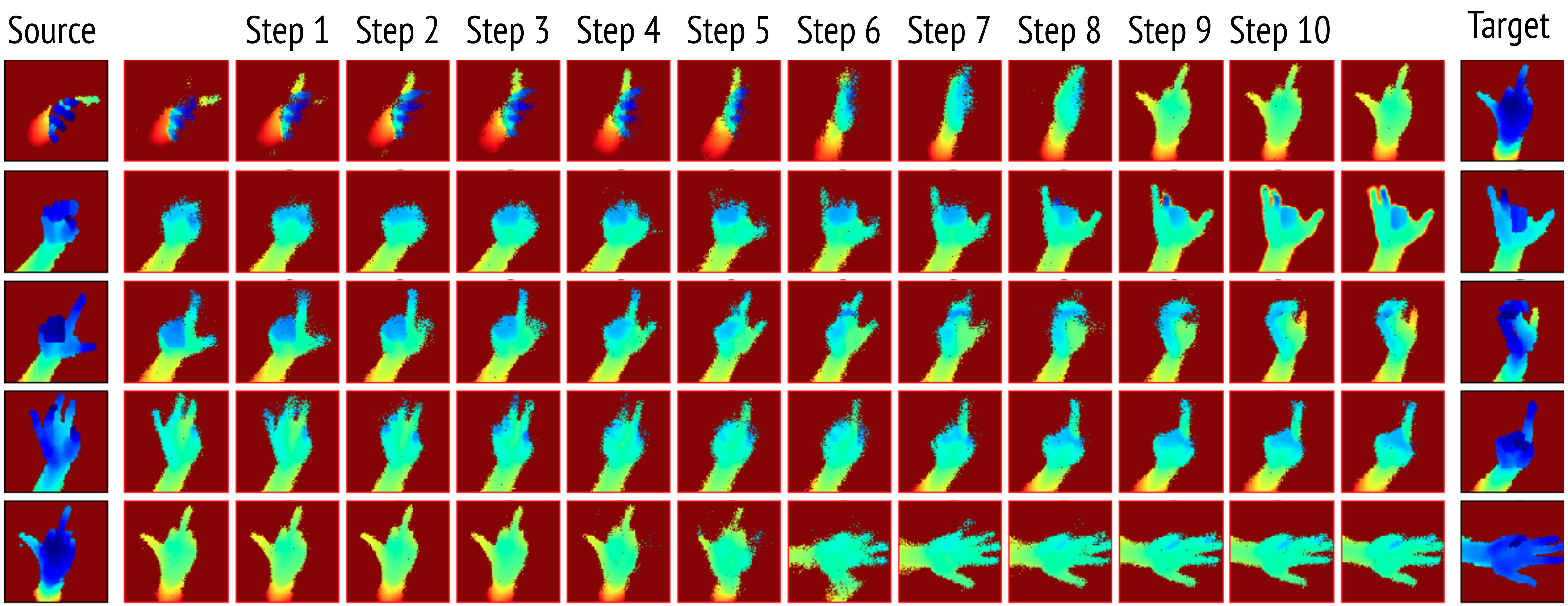
$\epsilon_k \sim N(0,1)$

Decoder

Analyzing hidden dimensions



Generating depth images



Extensions

- Hybrid VAE - multiple observations, learning with partial observations
- Multi Level VAE - style-content disentanglement
- VAE-GANs, CVAE-GAN - not solid mathematically, but higher quality
- Variational RNN - modeling temporal sequences

Single vs Multiple Observations

- Single Observation

$$\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$

Single vs Multiple Observations

- Single Observation

$$\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$


- Multiple Observations

Single vs Multiple Observations

- Single Observation

$$\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$

- Multiple Observations

$$\mathbf{d}, \mathbf{h} \sim p(\mathbf{d}, \mathbf{h}|\mathbf{z})$$

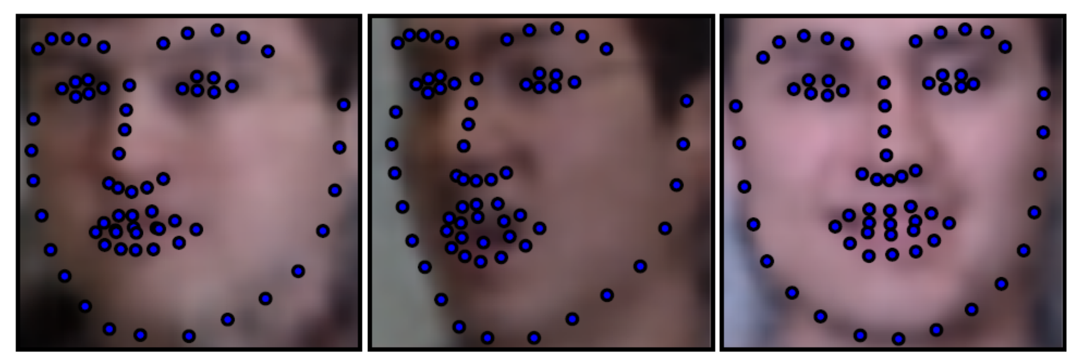
Single vs Multiple Observations

- Single Observation

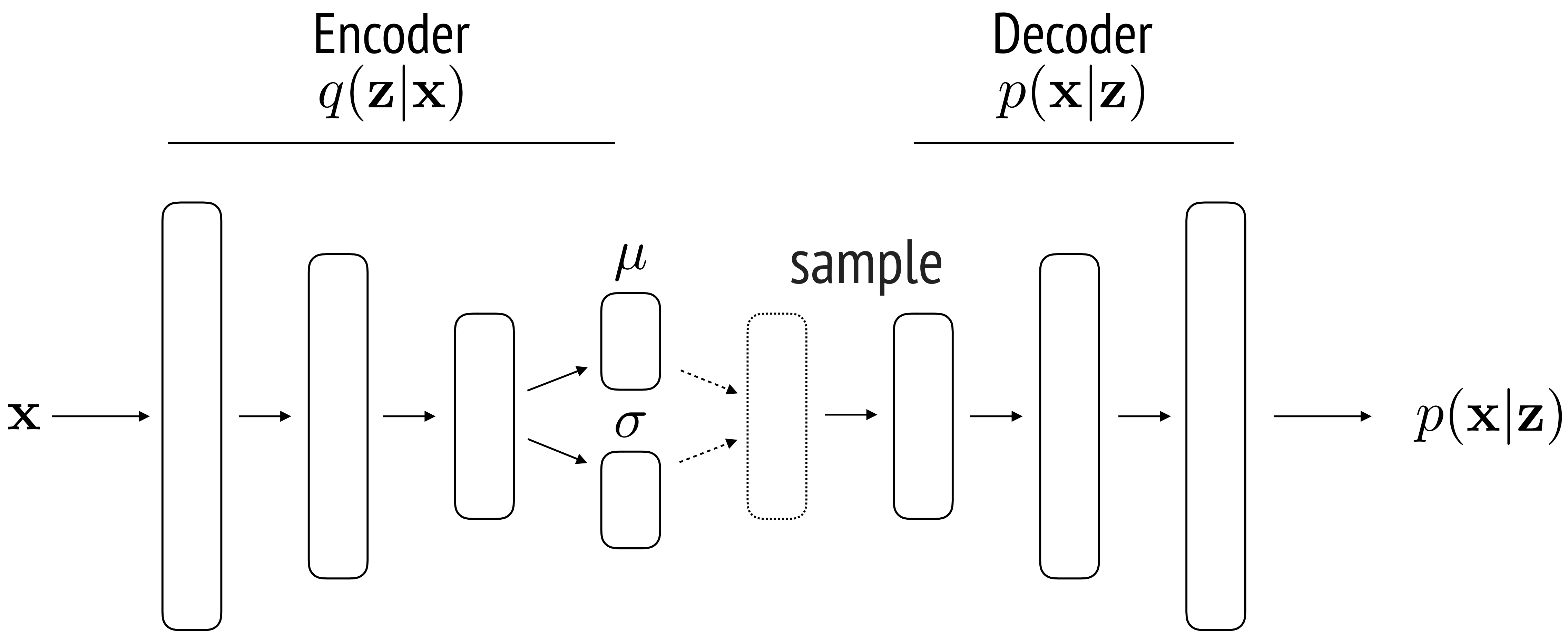
$$\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$

- Multiple Observations

$$\mathbf{d}, \mathbf{h} \sim p(\mathbf{d}, \mathbf{h}|\mathbf{z})$$

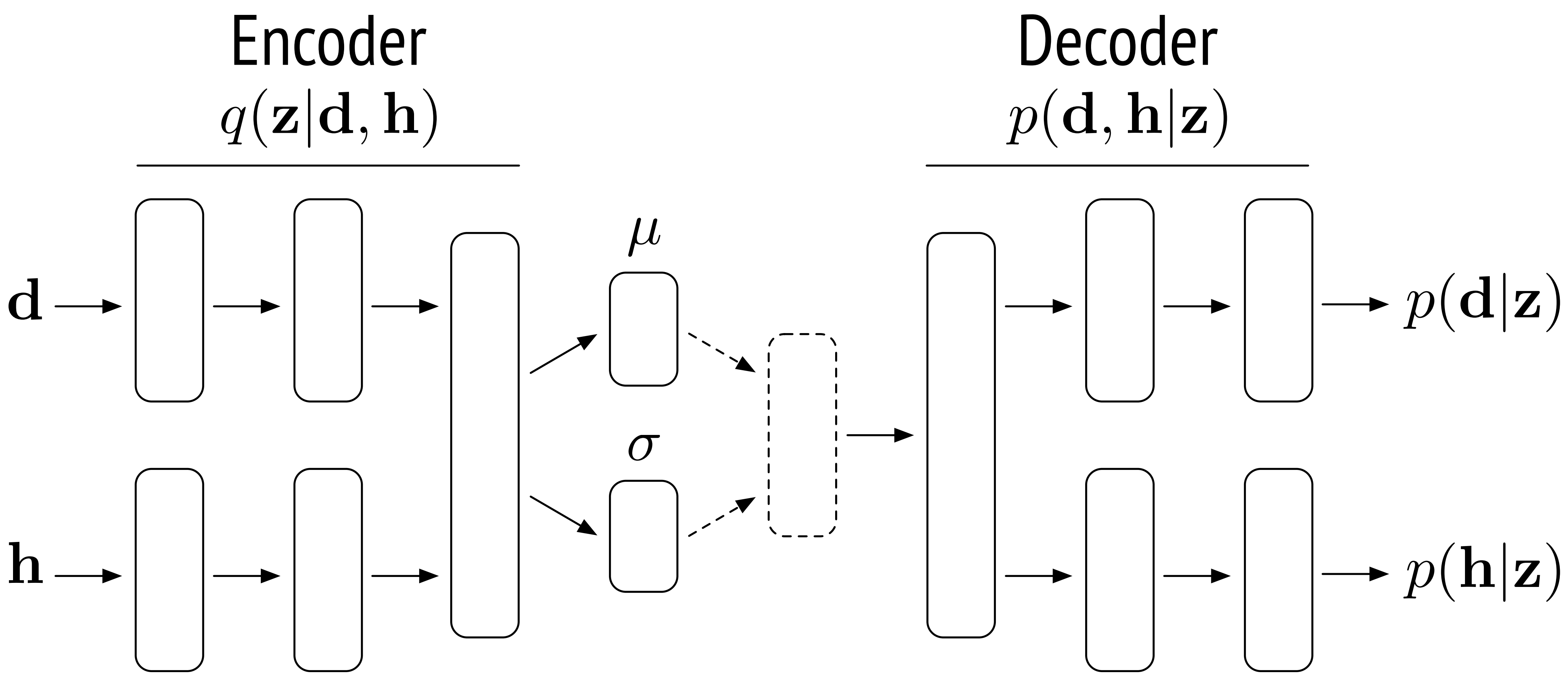


Single Observation



$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))$$

Multiple Observations



$$\log\, p(\mathbf{d},\mathbf{h}) \geq \mathbb{E}_{\mathbf{z}\sim\mathbf{q}(\mathbf{z}|\mathbf{d},\mathbf{h})}[\log\, p(\mathbf{d},\mathbf{h}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{d},\mathbf{h})|p(\mathbf{z}))$$

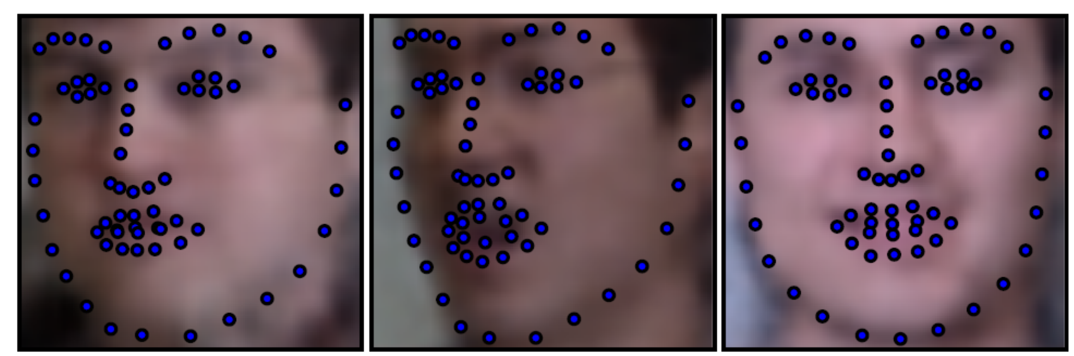
Single vs Multiple Observations

- Single Observation

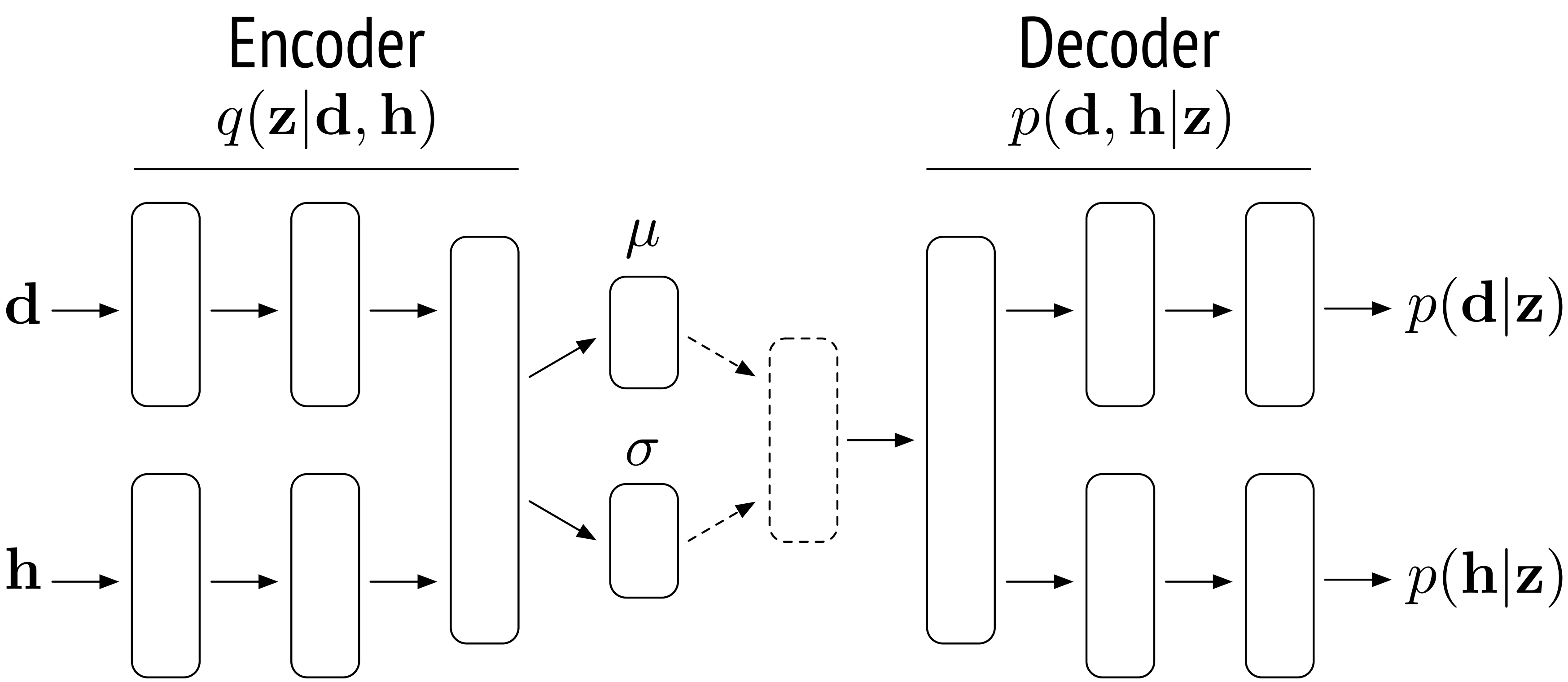
$$\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$

- Multiple Observations

$$\mathbf{d}, \mathbf{h} \sim p(\mathbf{d}, \mathbf{h}|\mathbf{z})$$

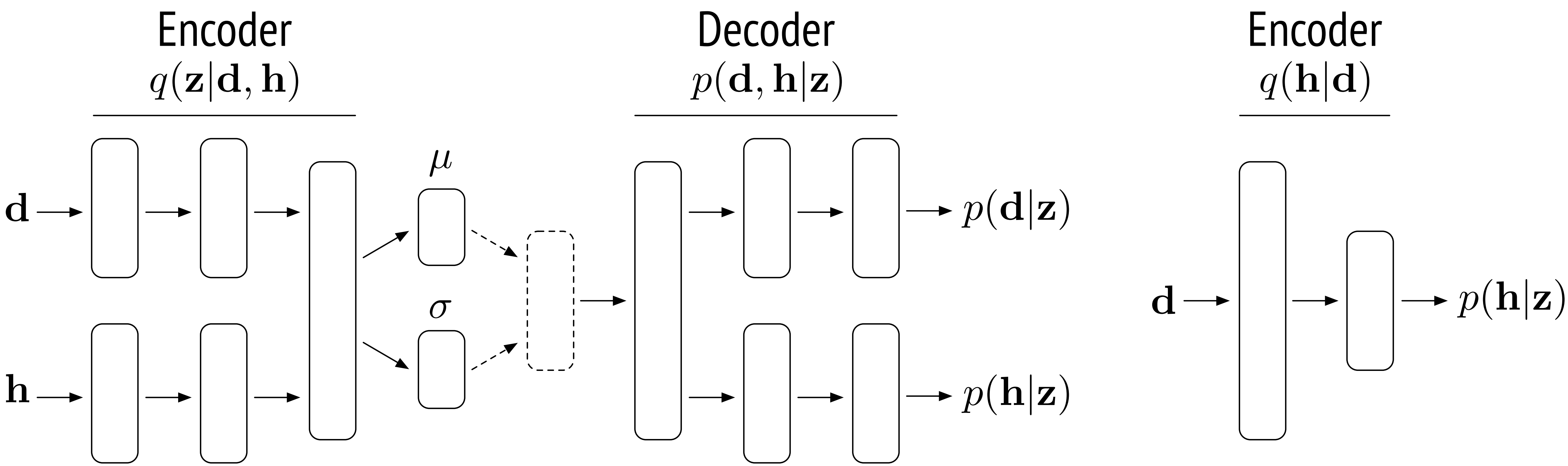


Hybrid VAEs



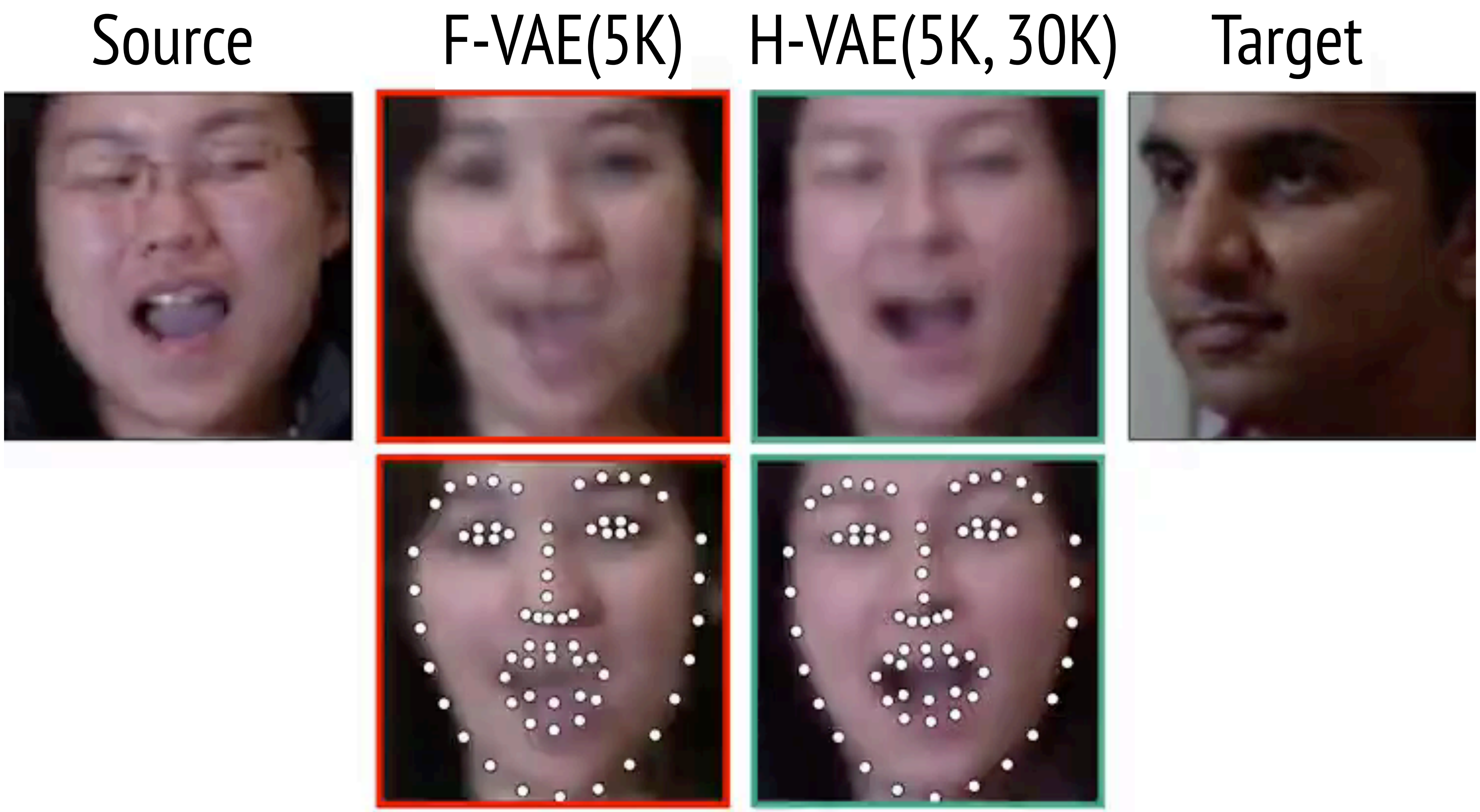
$$\log\, p(\mathbf{d}) \;\geq \mathbb{E}_{\mathbf{z}\sim\mathbf{p}(\mathbf{z})}[\log\, p(\mathbf{d},\mathbf{h}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{d},\mathbf{h})|p(\mathbf{z}))$$

Derivation



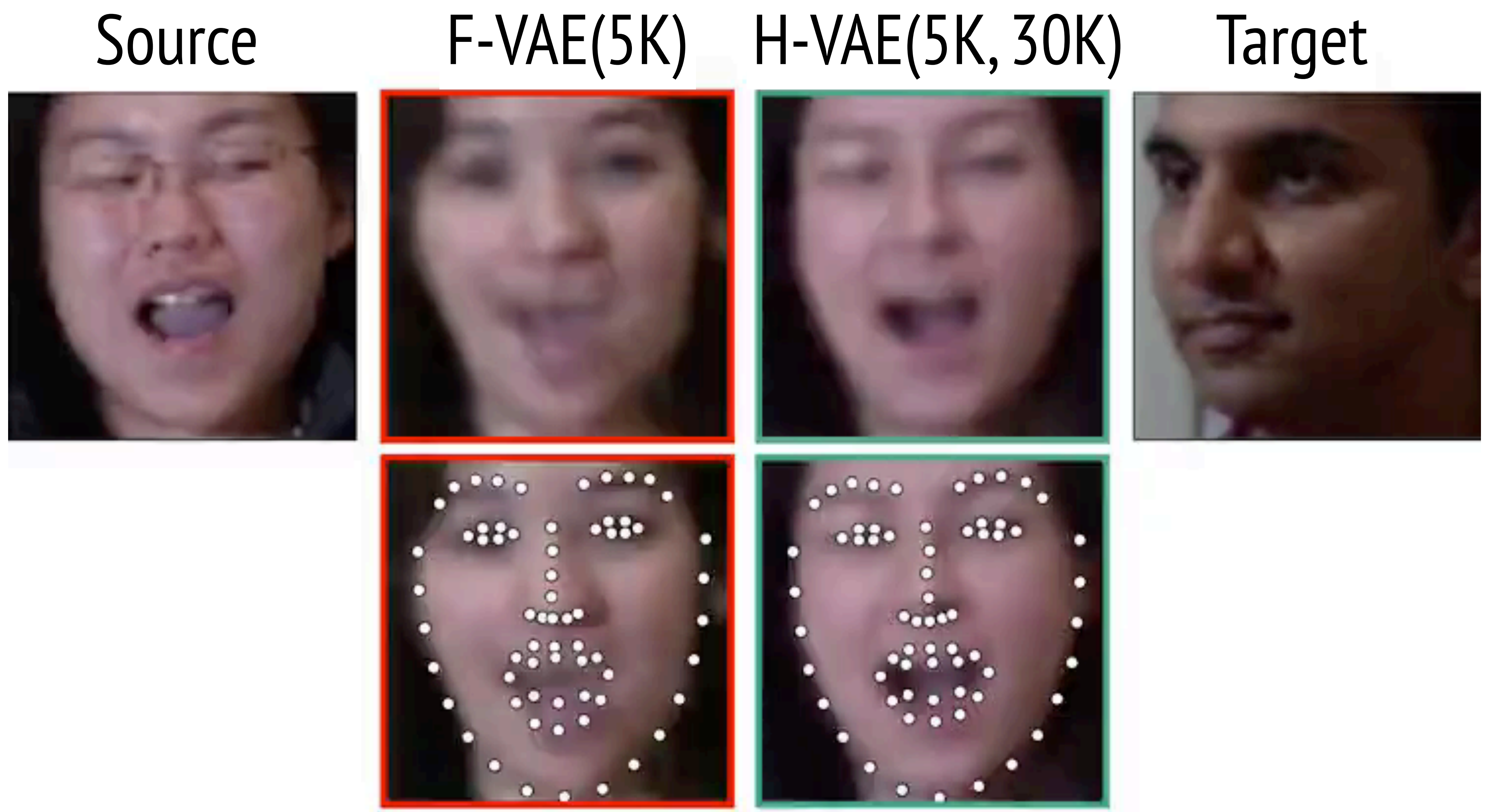
$$
\begin{aligned}
\log\, p(\mathbf{d}) \;\; &\geq \mathbb{E}_{\mathbf{z}\sim\mathbf{p}(\mathbf{z})}[\log\, p(\mathbf{d},\mathbf{h}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{d},\mathbf{h})|p(\mathbf{z})) \\
&+ \mathbb{E}_{\mathbf{h}\sim q(\mathbf{h}|\mathbf{d})}\left[\mathbb{E}_{\mathbf{z}\sim\mathbf{p}(\mathbf{z})}[\log\, p(\mathbf{d},\mathbf{h}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{d},\mathbf{h})|p(\mathbf{z}))\right] \\
&+ H(q(\mathbf{h}|\mathbf{d}))
\end{aligned}
$$

Benefits of H-VAE



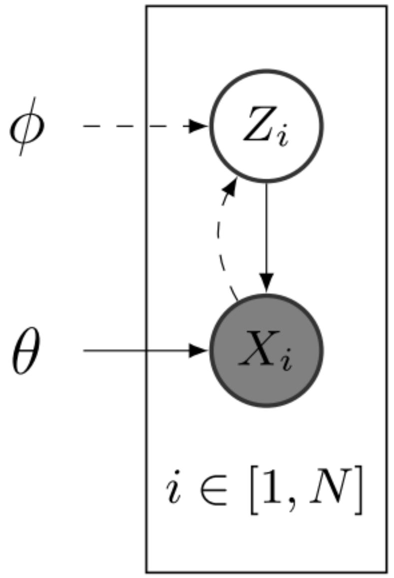
Both models trained on MultiPIE

Benefits of H-VAE



Both models trained on MultiPIE

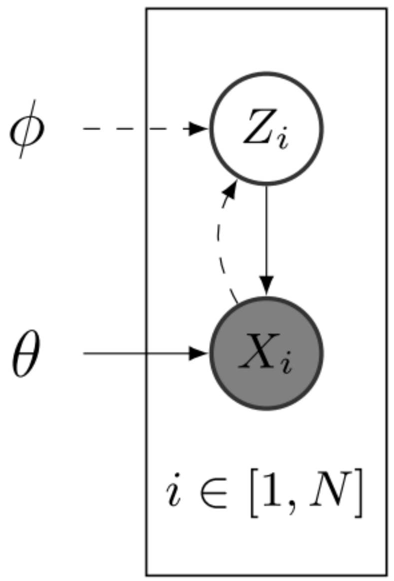
Multi-Level VAE



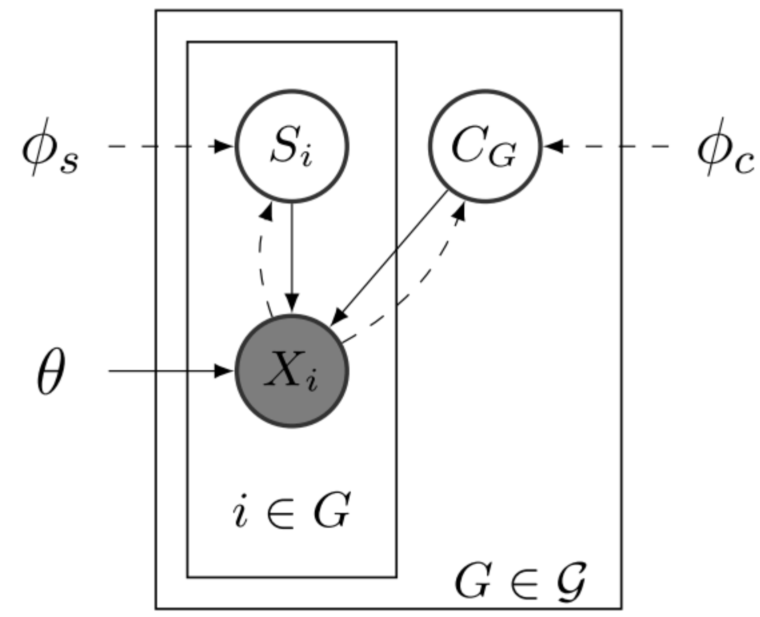
VAE

[2017] Bouchacourt, D., Multi-Level Variational Autoencoder: Learning Disentangled Representations from Grouped Observations

Multi-Level VAE



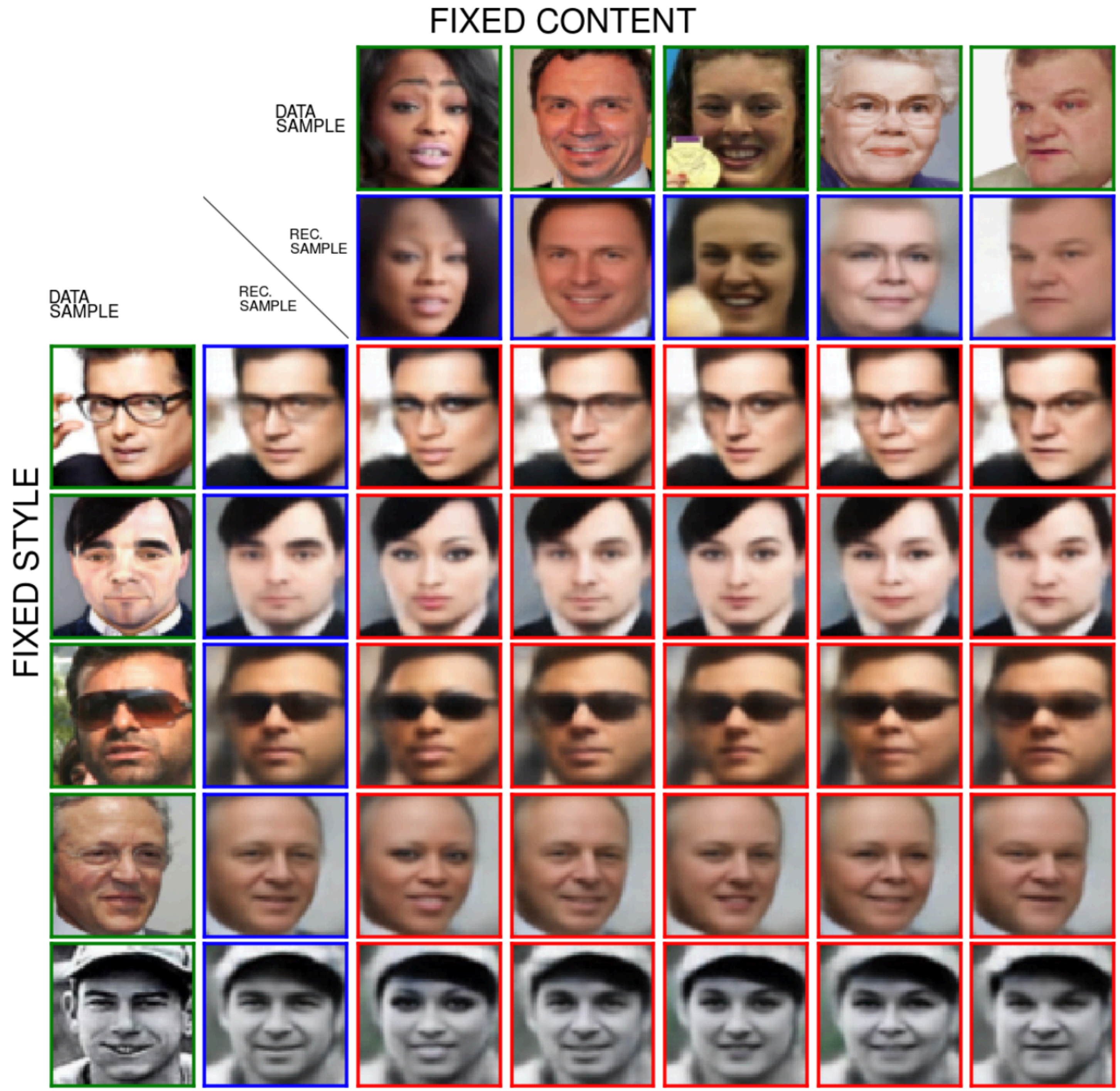
VAE



ML-VAE

[2017] Bouchacourt, D., Multi-Level Variational Autoencoder: Learning Disentangled Representations from Grouped Observations

Multi-Level VAE



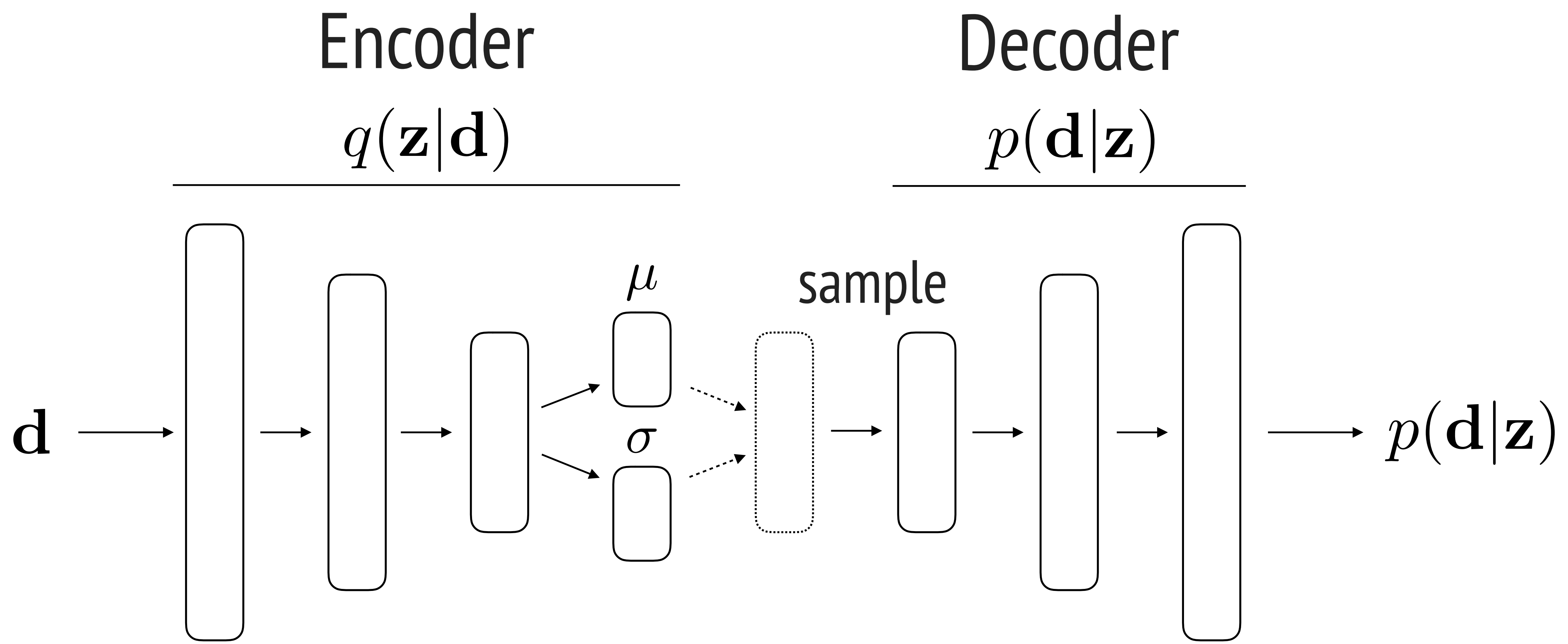
Multi-Level VAE

$$\mathrm{ELBO}(G;\theta,\phi_s,\phi_c) = \sum_{i\in G} \mathbb{E}_{q(C_G|\mathbf{X}_G;\phi_c)}[\mathbb{E}_{q(S_i|X_i;\phi_s)}[\log p(X_i|C_G,S_i;\theta)]] - \sum_{i\in G} \mathrm{KL}(q(S_i|X_i;\phi_s)||p(S_i)) - \mathrm{KL}(q(C_G|\mathbf{X}_G;\phi_c)||p(C_G)).$$

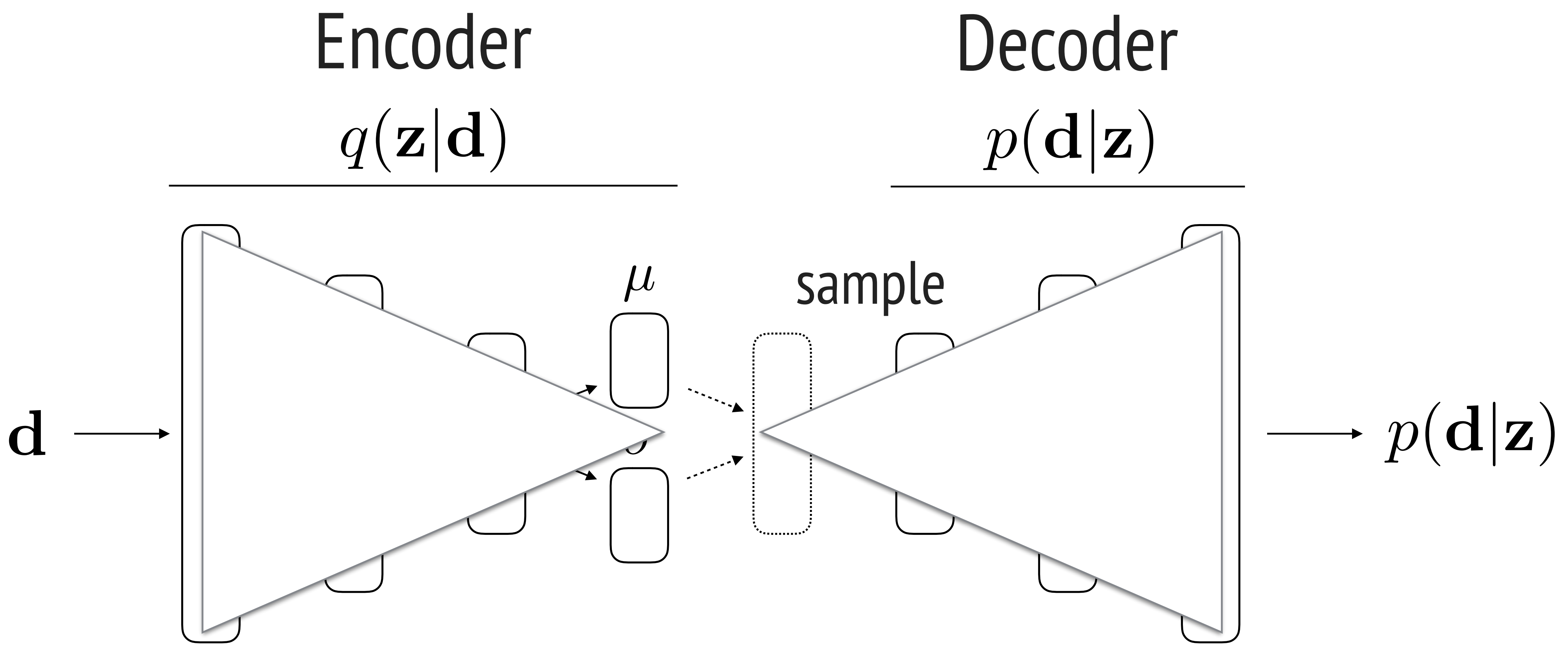
Algorithm 1: ML-VAE training algorithm.

for *Each epoch* **do**
  Sample minibatch of groups $\mathcal{G}_b$,
  for $G \in \mathcal{G}_b$ **do**
    for $i \in G$ **do**
      Encode x_i into $q(C_G|X_i = x_i;\phi_c)$, $q(S_i|X_i = x_i;\phi_s)$,
    end
    Construct $q(C_G|\mathbf{X}_G = \mathbf{x}_G;\phi_c)$ using $q(C_G|X_i = x_i;\phi_c), \forall i \in G$,
    for $i \in G$ **do**
      Sample $c_{G,i} \sim q(C_G|\mathbf{X}_G = \mathbf{x}_G;\phi_c)$, $s_i \sim q(S_i|X_i = x_i;\phi_s)$,
      Decode $c_{G,i}, s_i$ to obtain $p(X_i|C_G = c_{G,i}, S_i = s_i;\theta)$,
    end
  end
  Update θ, ϕ_c, ϕ_s by taking a gradient step of Equation (5): $\nabla_{\theta,\phi_c,\phi_s}\mathcal{L}(\mathcal{G}_b,\theta,\phi_c,\phi_s)$
end

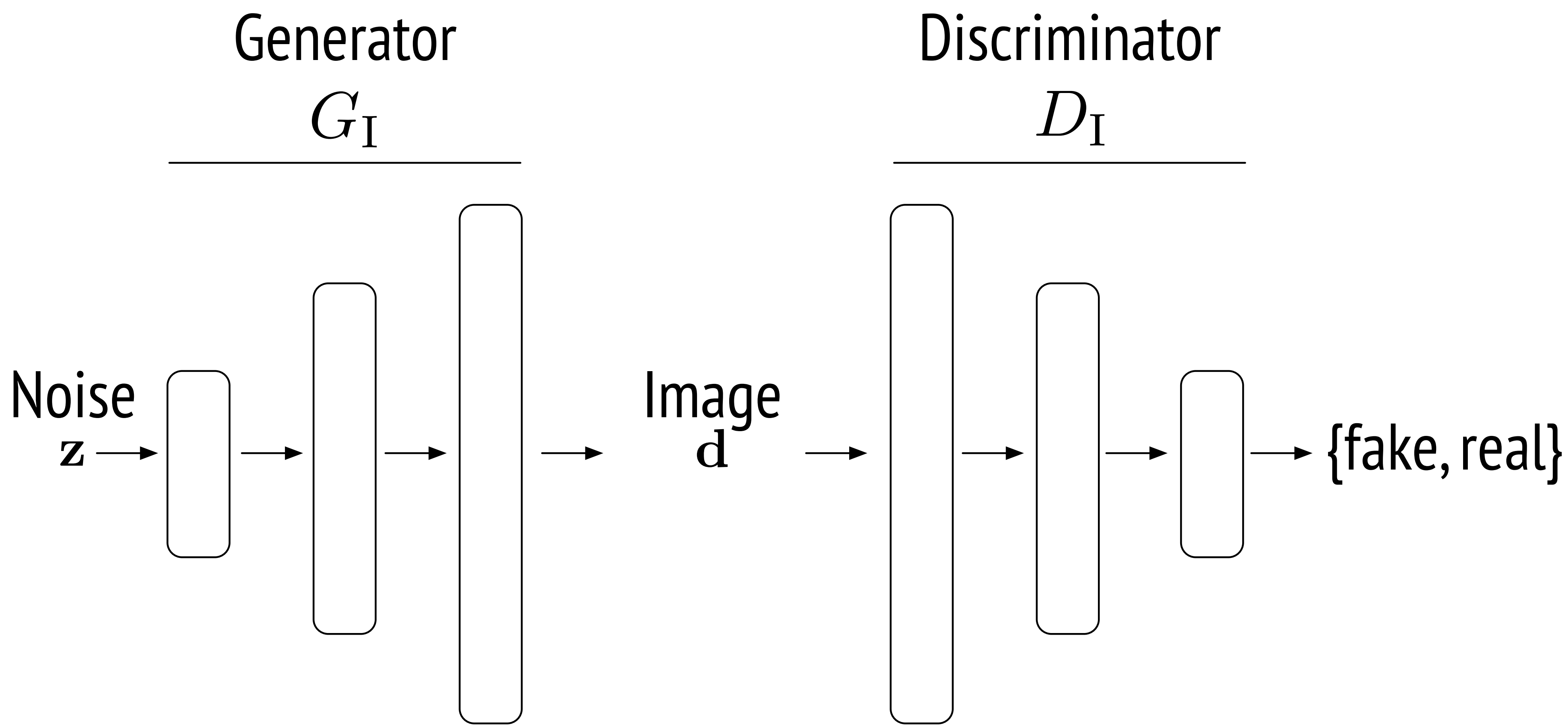
Variational Auto Encoders - VAEs



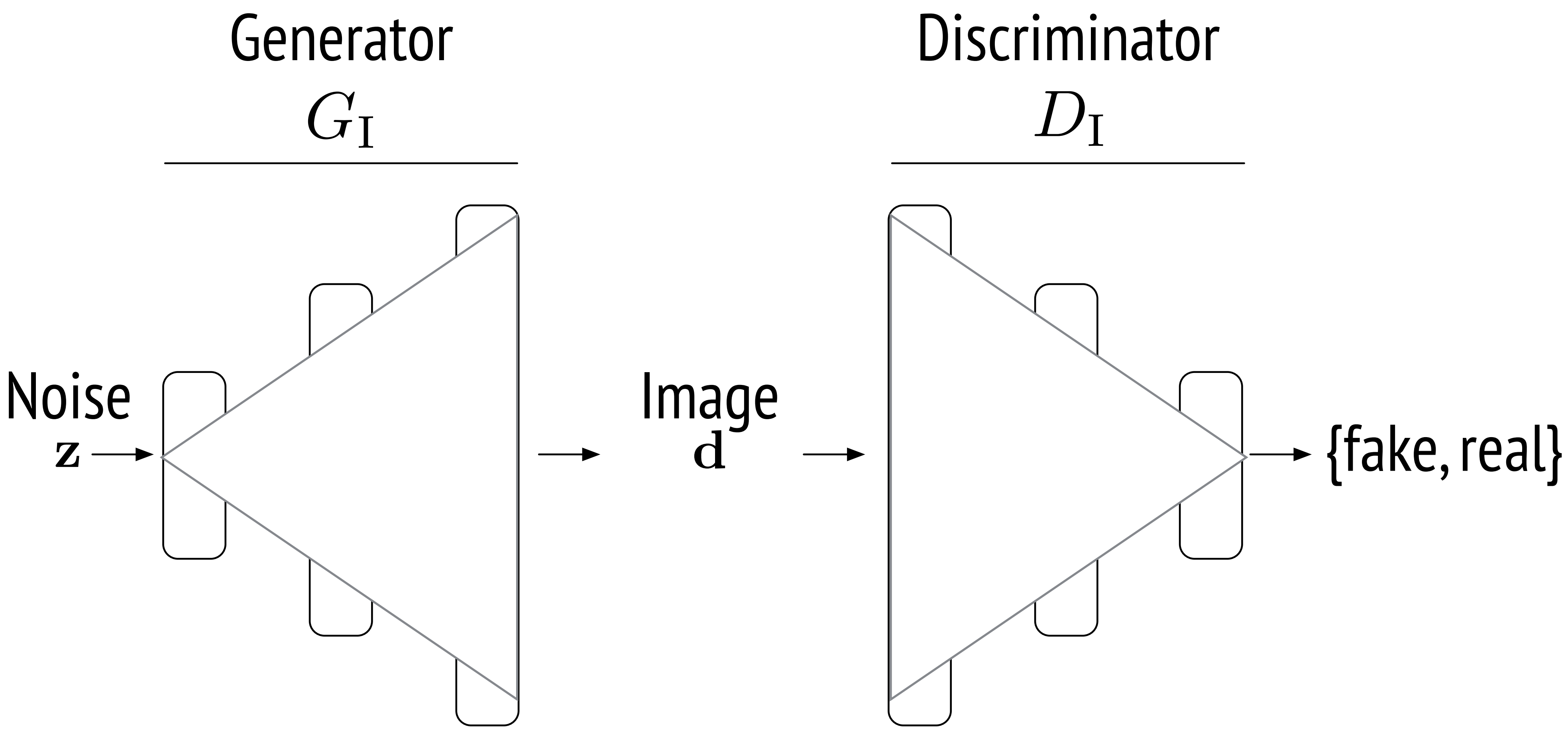
Variational Auto Encoders - VAEs



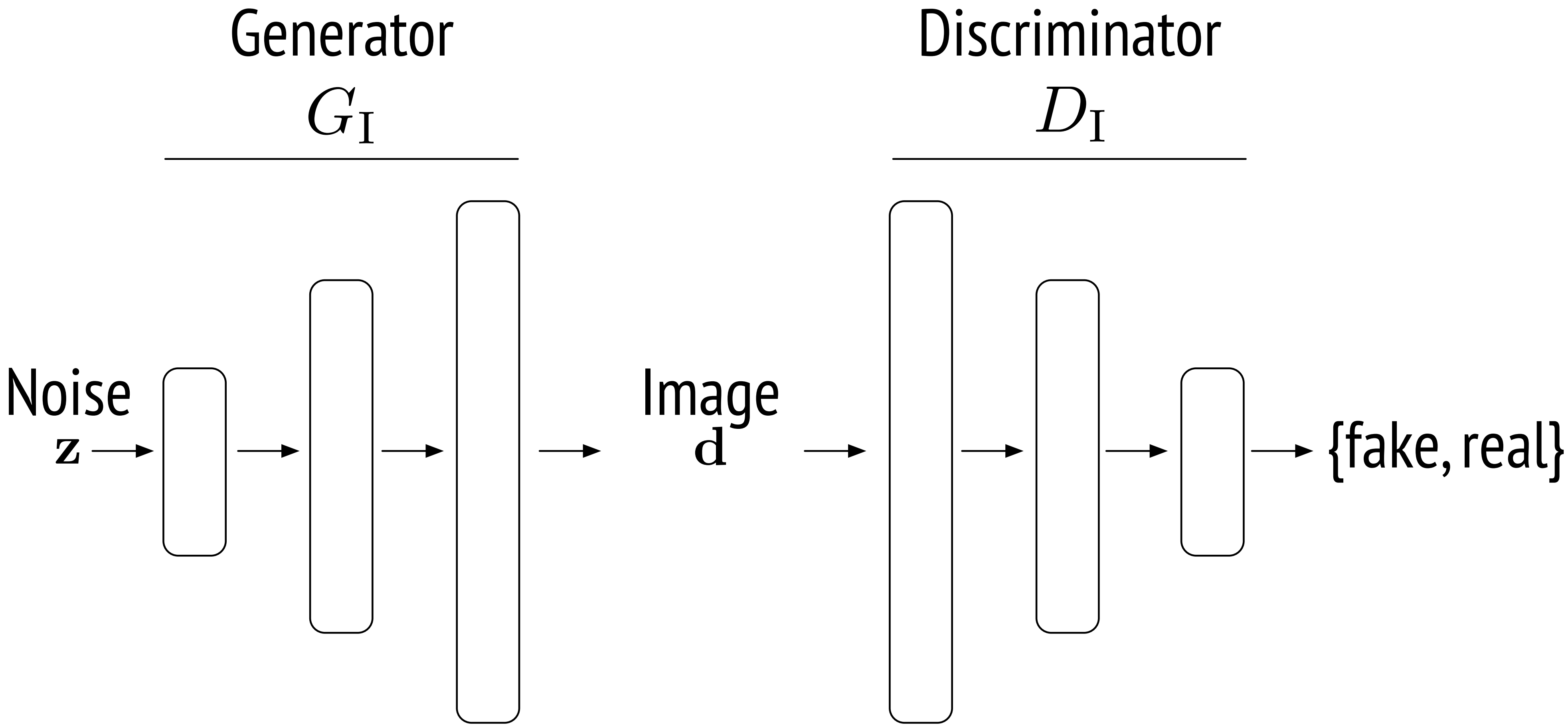
Generative Adversarial Networks - GANs



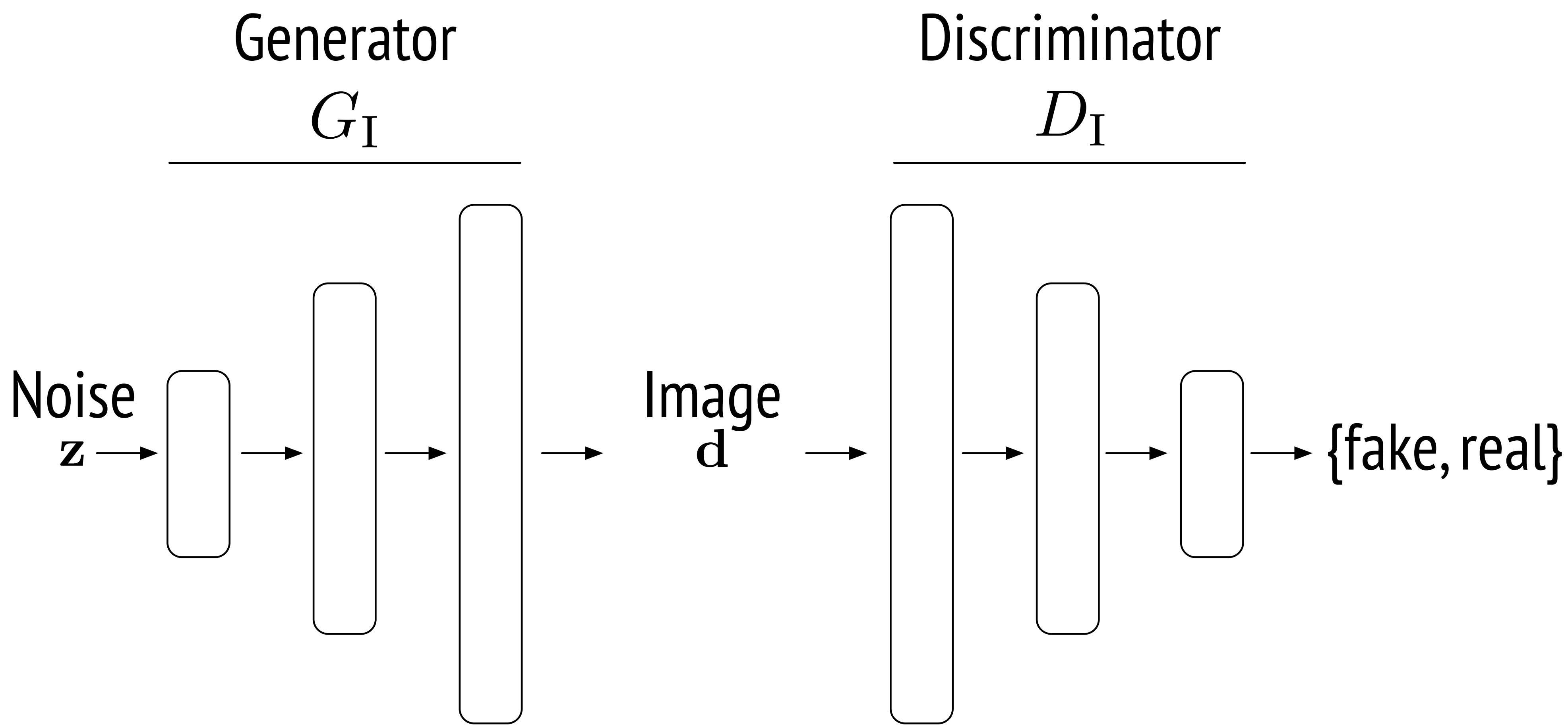
Generative Adversarial Networks - GANs



Generative Adversarial Networks - GANs

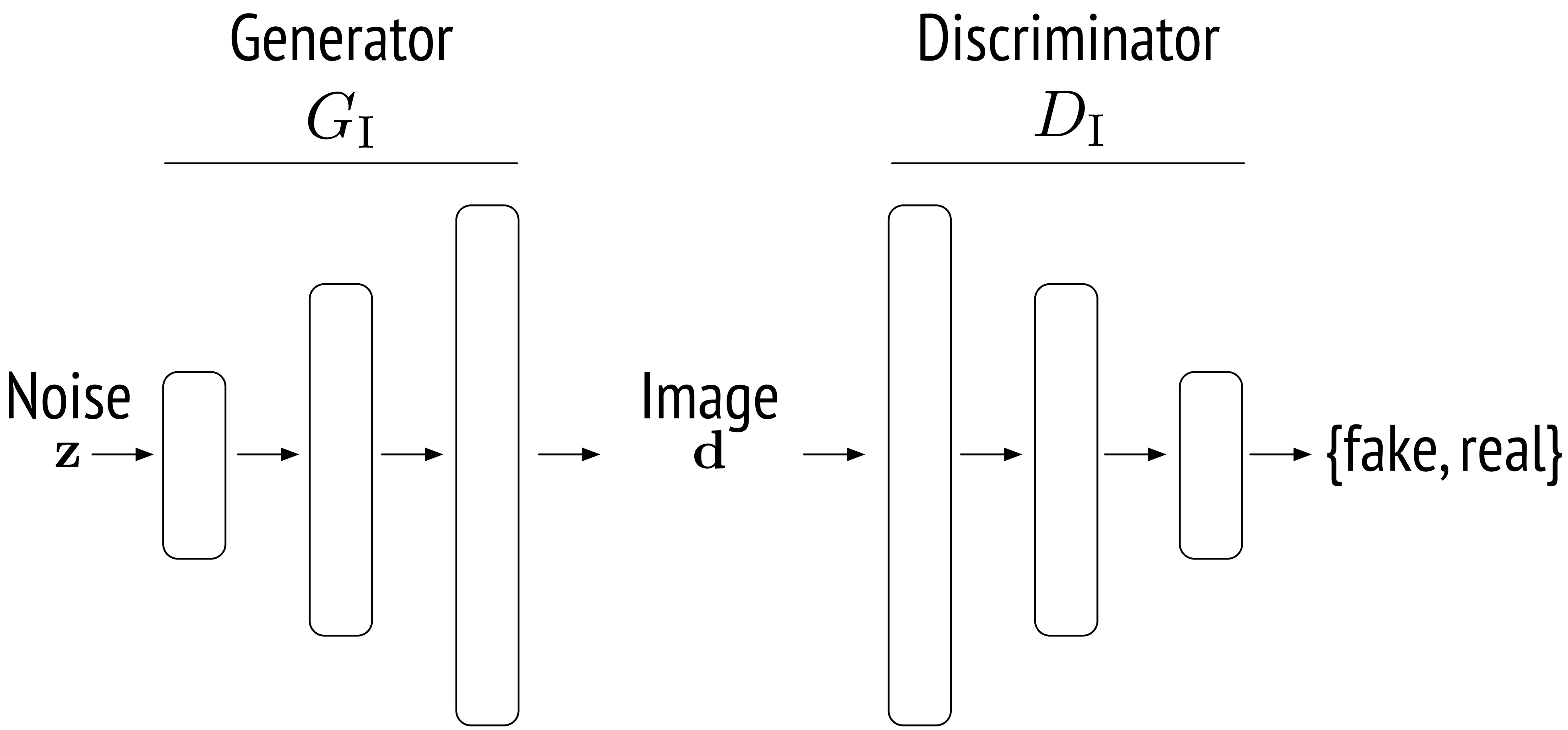


Generative Adversarial Networks - GANs



$$\text{For } D_{\mathrm{I}}: \quad \min_{D_{\mathrm{I}}} \mathbb{E}_{\mathbf{d}\sim p(\mathbf{d})}[-\log D_{\mathrm{I}}(\mathbf{d})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}_{\mathrm{I}})}[-\log(1 - D_{\mathrm{I}}(G_{\mathrm{I}}(\mathbf{z})))]$$

Generative Adversarial Networks - GANs

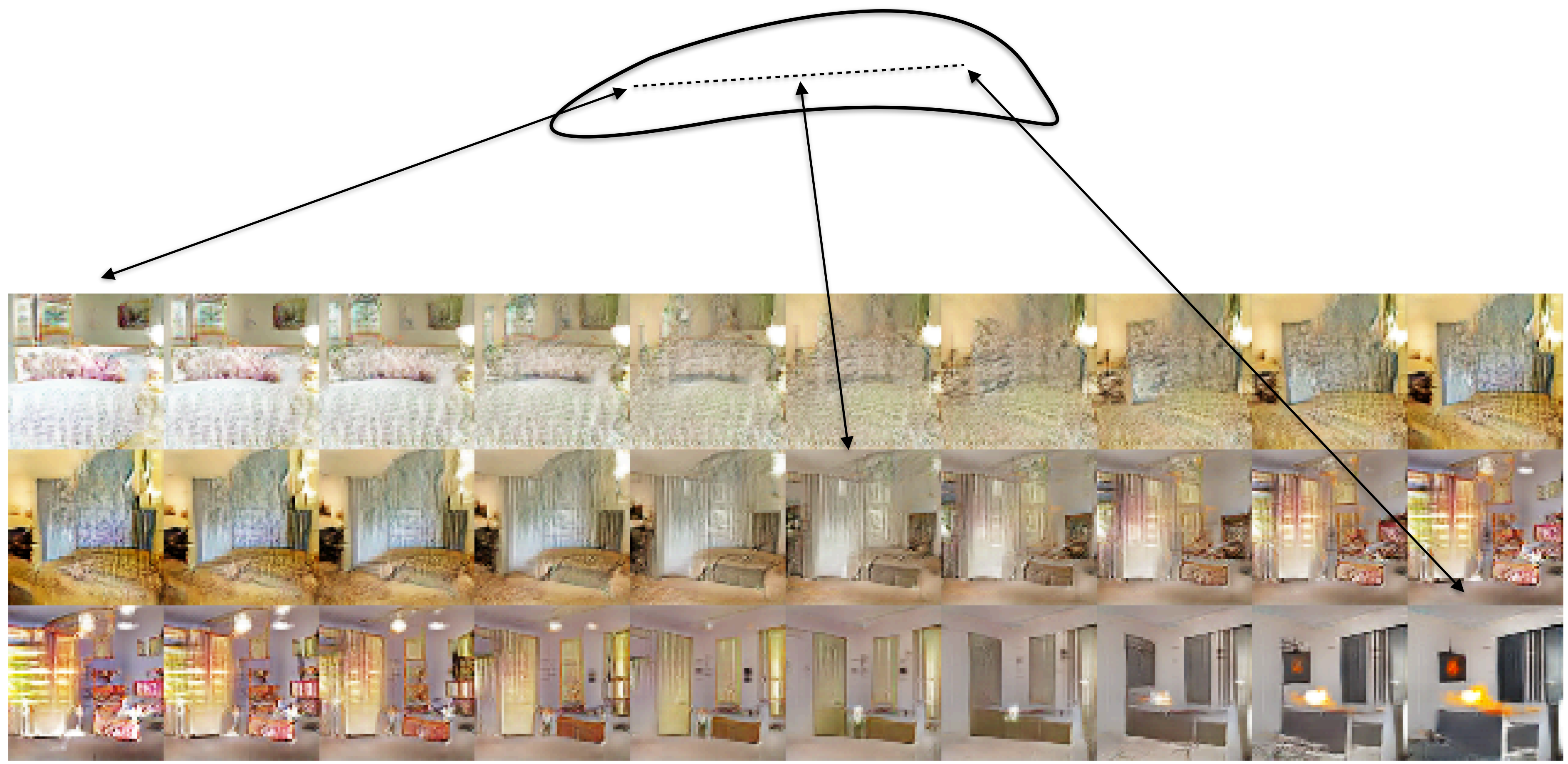


$$\text{For } D_{\mathrm{I}}: \quad \min_{D_{\mathrm{I}}} \mathbb{E}_{\mathbf{d}\sim p(\mathbf{d})}[-\log D_{\mathrm{I}}(\mathbf{d})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}_{\mathrm{I}})}[-\log(1 - D_{\mathrm{I}}(G_{\mathrm{I}}(\mathbf{z})))]$$

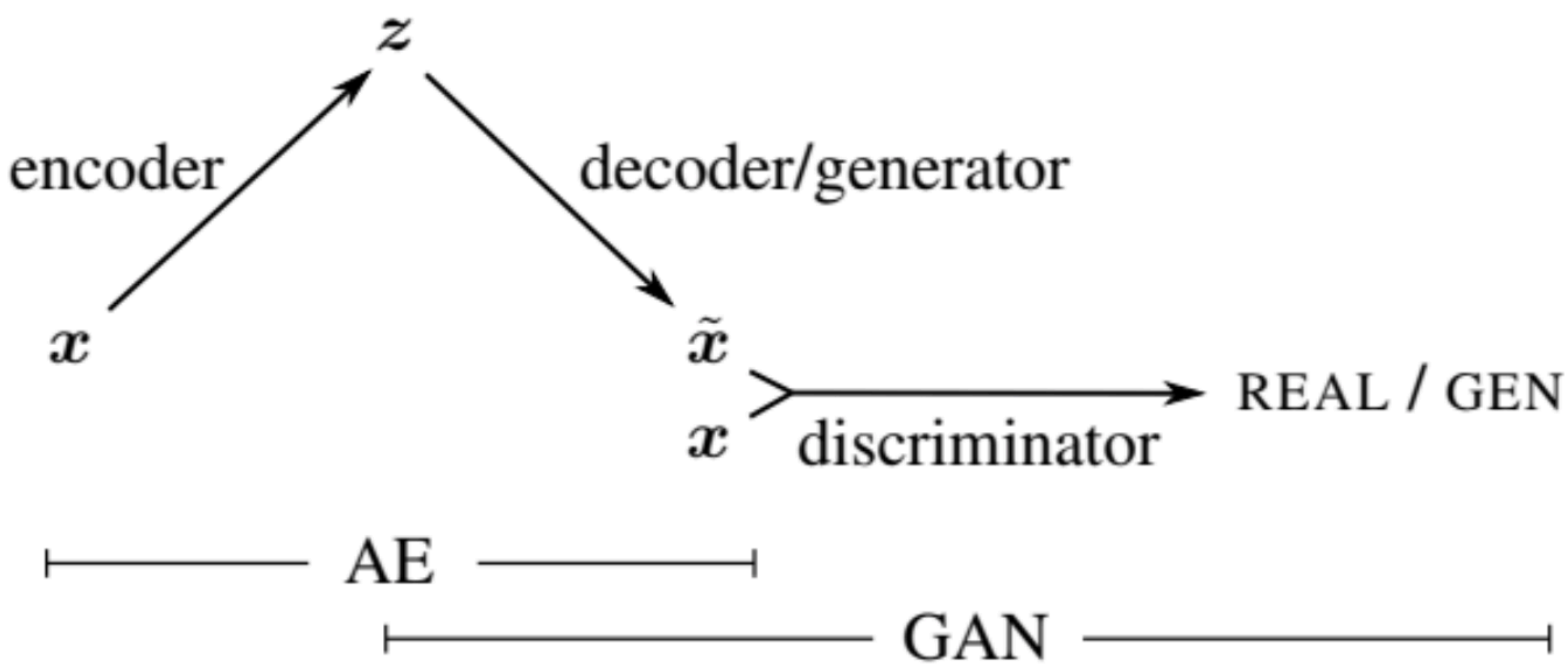
$$\text{For } G_{\mathrm{I}}: \quad \max_{G_{\mathrm{I}}} \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}_{\mathrm{I}})}[-\log(1 - D_{\mathrm{I}}(G_{\mathrm{I}}(\mathbf{z})))]$$

Generative Adversarial Networks

Hidden space: every point is an image

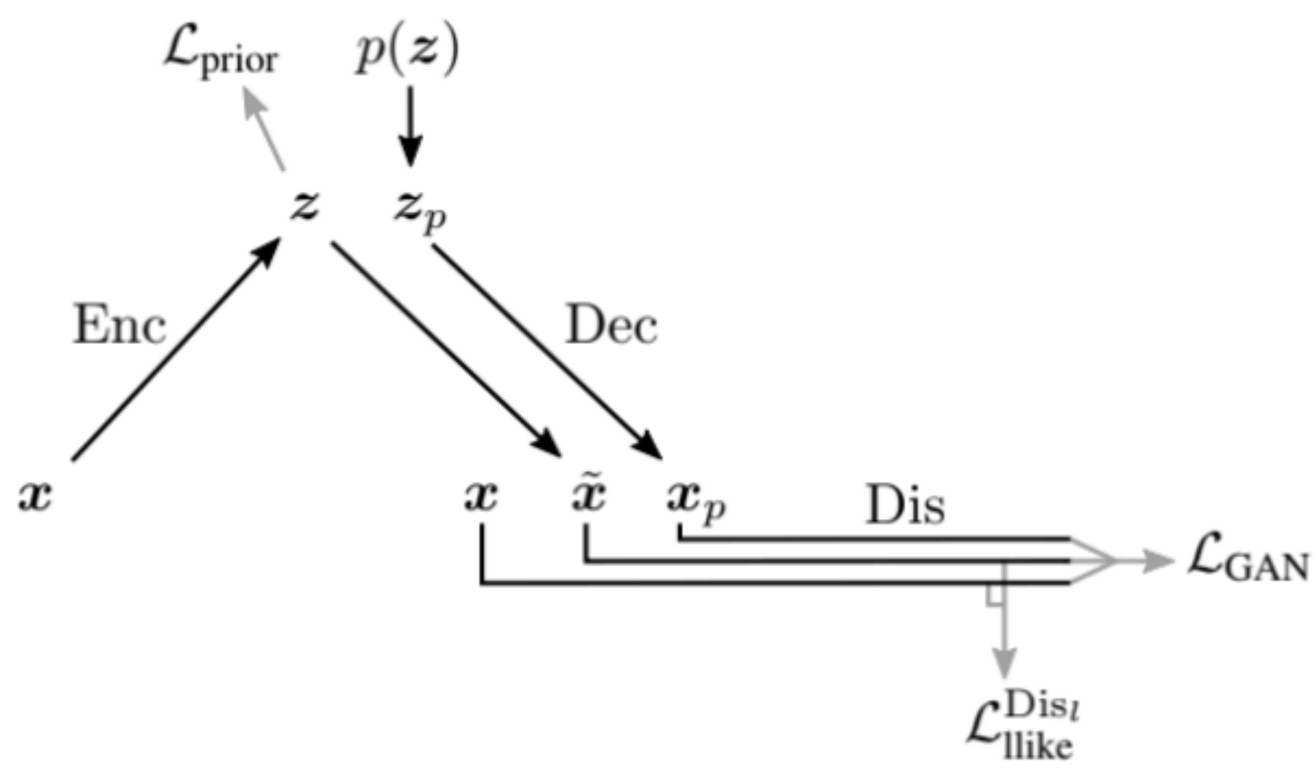


Making VAE sharper - VAE-GANs



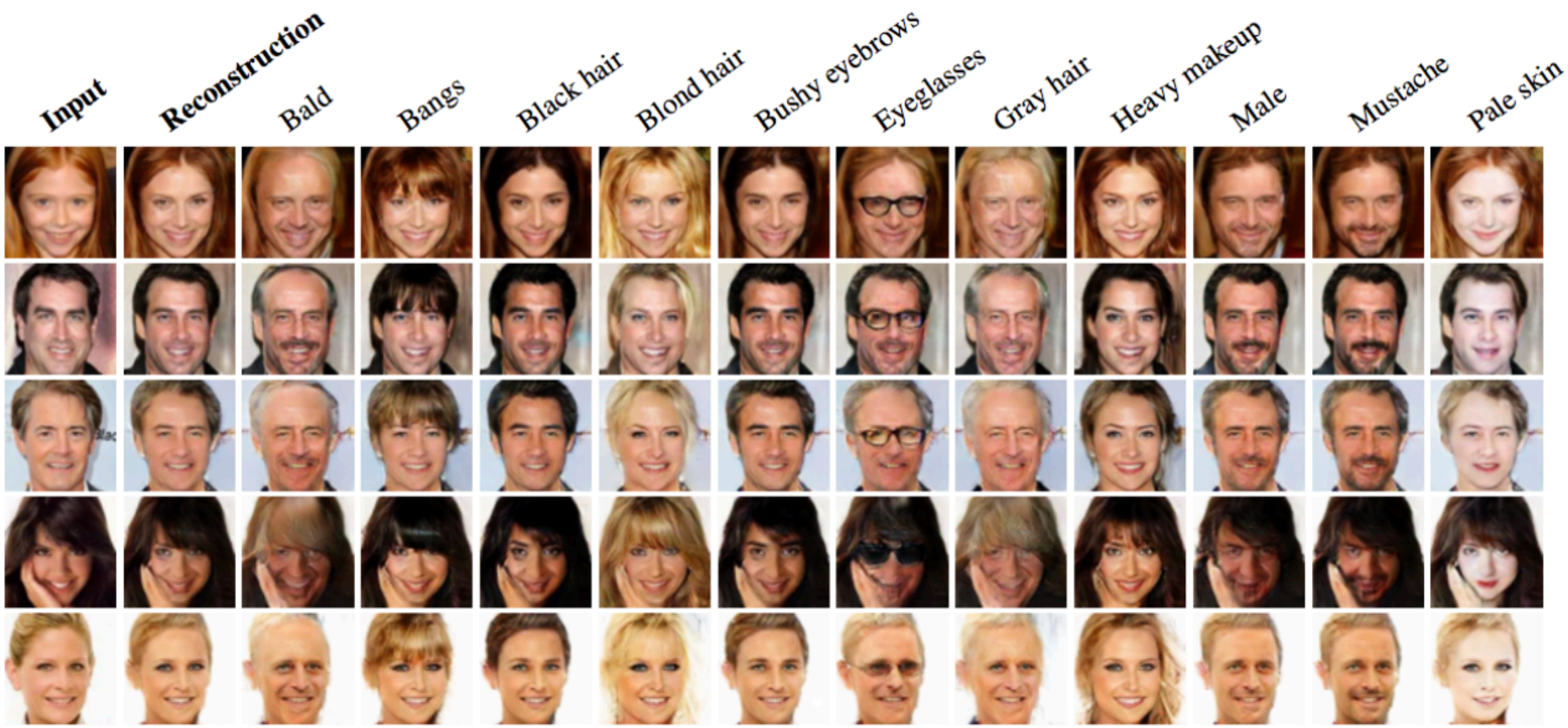
[2015] Larsen, A. B. L. et al. Autoencoding beyond pixels using a learned similarity metric

Making VAE sharper - VAE-GANs



[2015] Larsen, A. B. L. et al. Autoencoding beyond pixels using a learned similarity metric

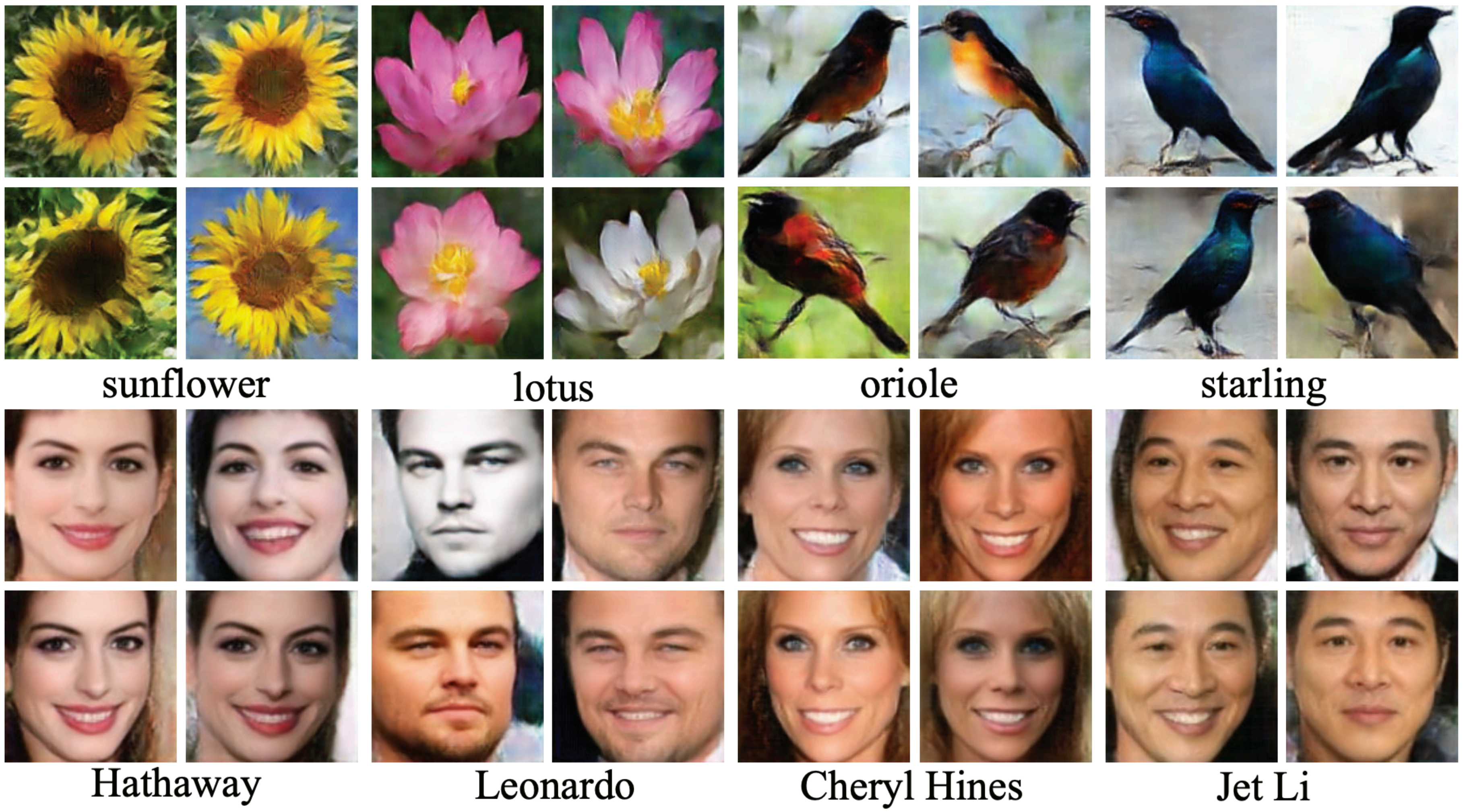
Making VAE sharper - VAE-GANs



[2015] Larsen, A. B. L. et al. Autoencoding beyond pixels using a learned similarity metric

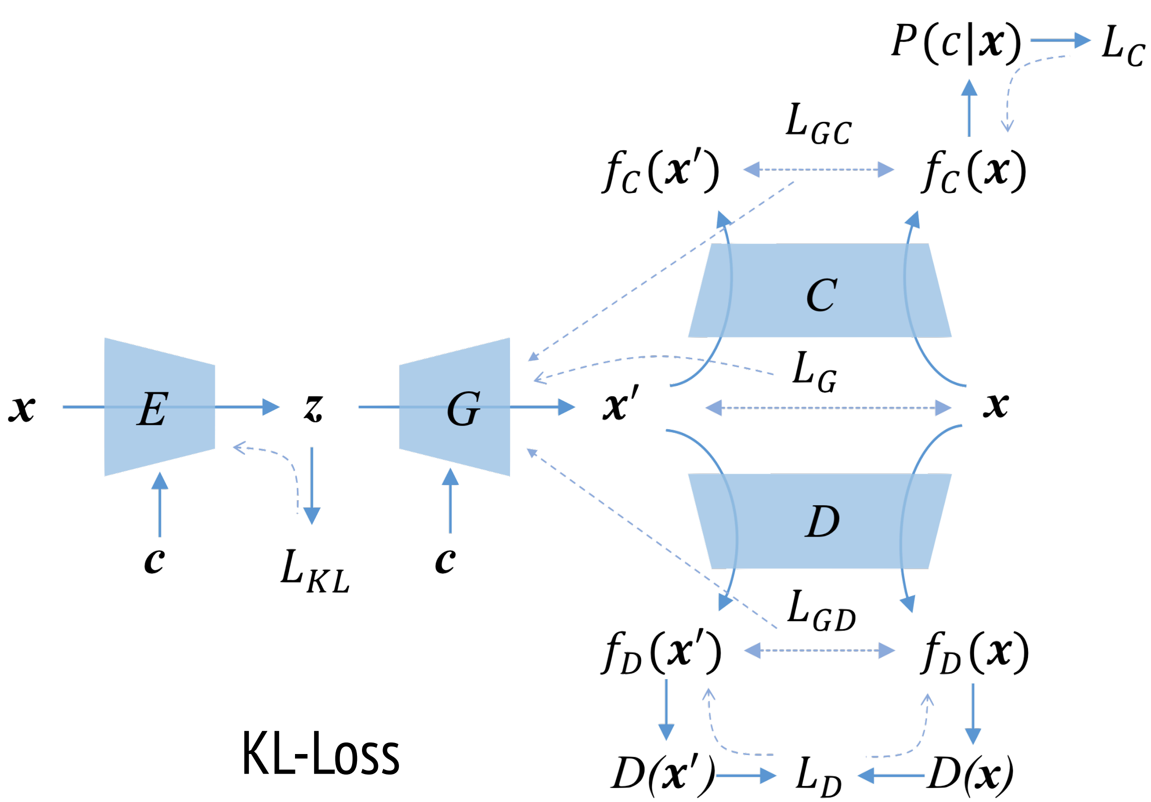
Making VAE sharper - CVAE-GAN

C - conditioned. We want to generate an object of a particular category



Making VAE sharper - CVAE-GAN

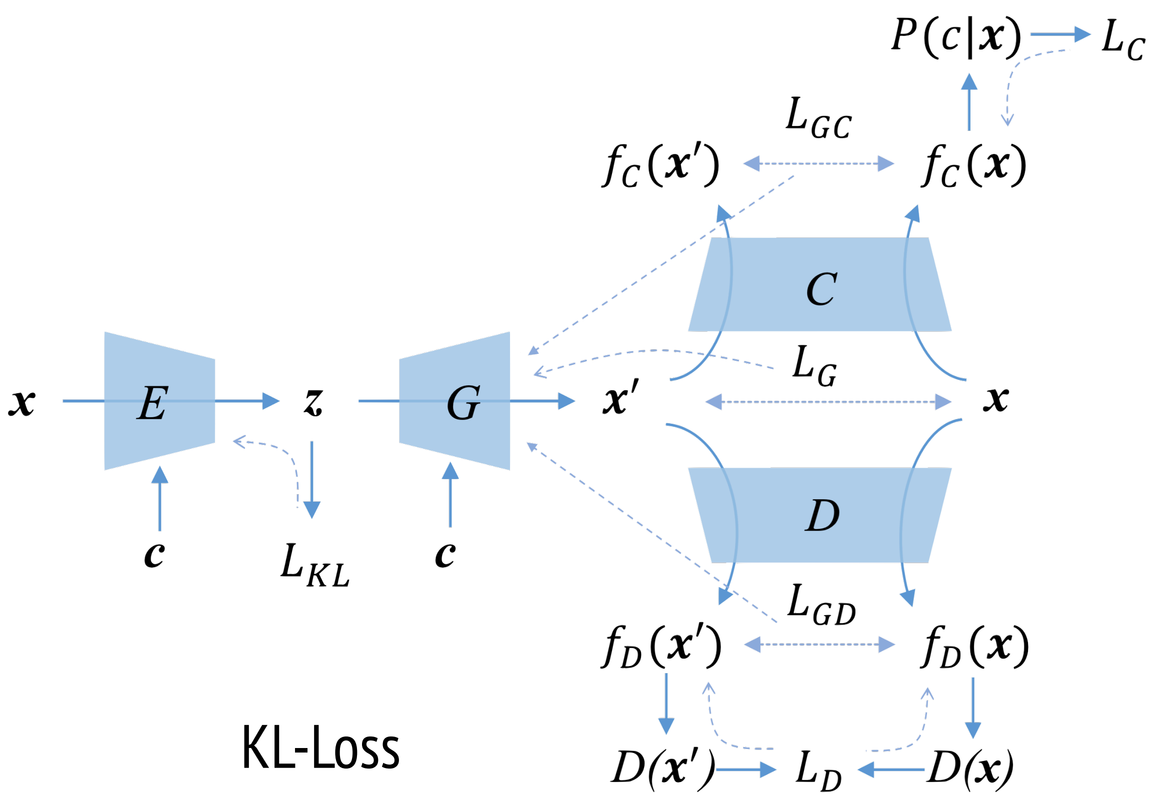
C - conditioned. We want to generate an object of a particular category



KL-Loss

Making VAE sharper - CVAE-GAN

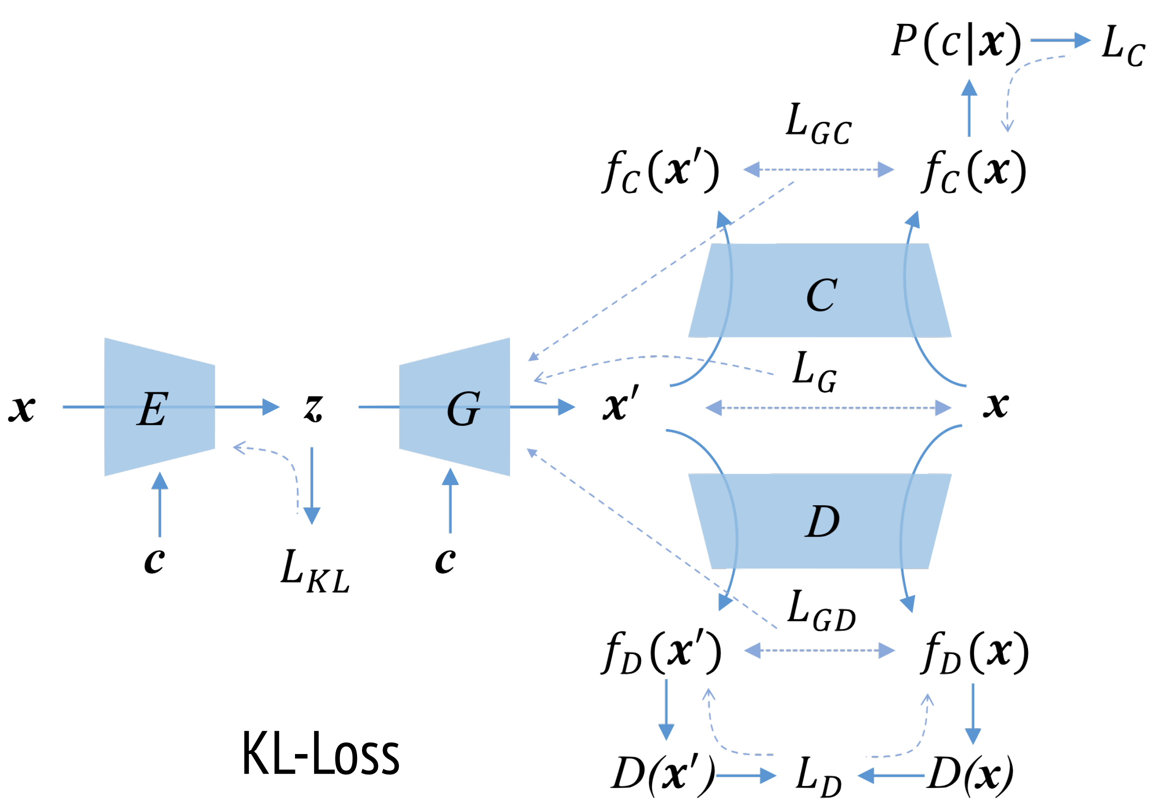
C - conditioned. We want to generate an object of a particular category



InfoGAN: $\mathcal{L}_C = -\mathbb{E}_{\boldsymbol{x}\sim P_r}[\log P(c|\boldsymbol{x})].$

Making VAE sharper - CVAE-GAN

C - conditioned. We want to generate an object of a particular category



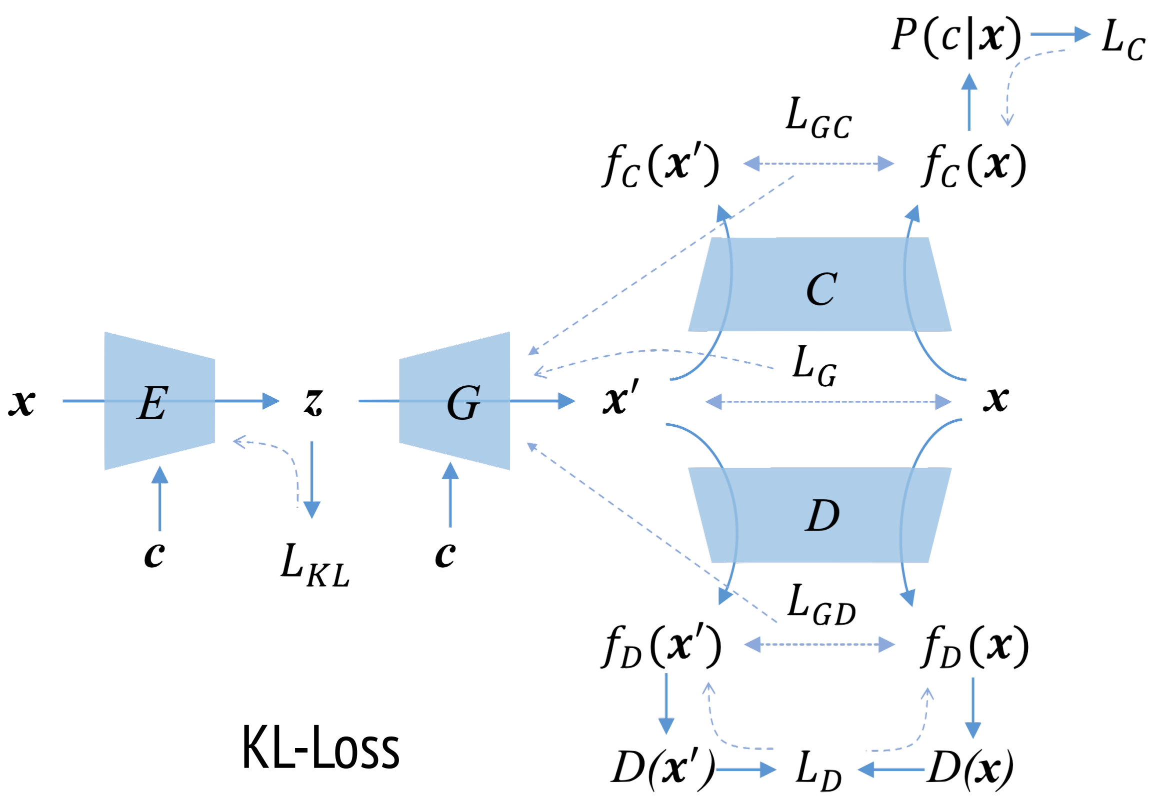
KL-Loss

InfoGAN: $\mathcal{L}_C = -\mathbb{E}_{\boldsymbol{x}\sim P_r}[\log P(c|\boldsymbol{x})].$

Feature Matching: $\mathcal{L}_{GC} = \frac{1}{2}\sum_c ||\mathbb{E}_{\boldsymbol{x}\sim P_r} f_C(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim P_{\boldsymbol{z}}} f_C(G(\boldsymbol{z}, c))||_2^2.$

Making VAE sharper - CVAE-GAN

C - conditioned. We want to generate an object of a particular category



InfoGAN: $\mathcal{L}_C = -\mathbb{E}_{\boldsymbol{x} \sim P_r}[\log P(c|\boldsymbol{x})].$

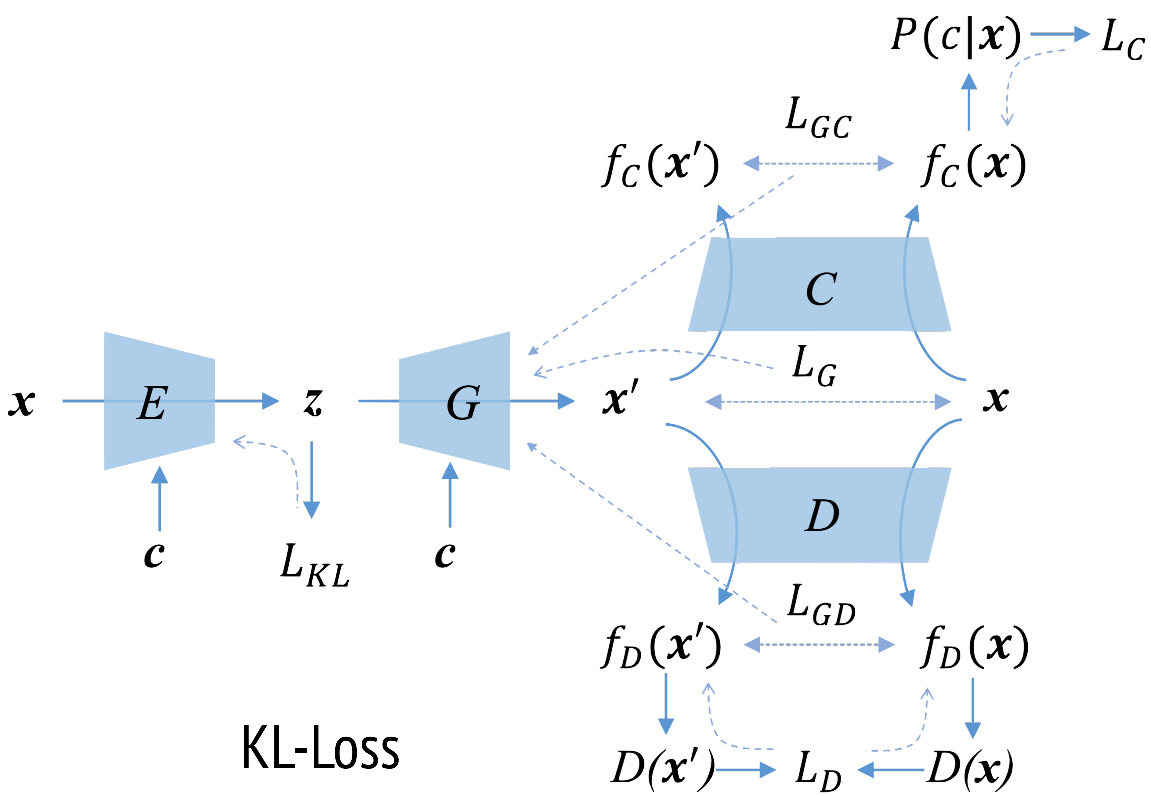
Feature Matching: $\mathcal{L}_{GC} = \frac{1}{2} \sum_c ||\mathbb{E}_{\boldsymbol{x} \sim P_r} f_C(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z} \sim P_{\boldsymbol{z}}} f_C(G(\boldsymbol{z}, c))||_2^2.$

Pairwise Feature Matching:

$$\mathcal{L}_G = \frac{1}{2}(||\boldsymbol{x} - \boldsymbol{x}'||_2^2 + ||f_D(\boldsymbol{x}) - f_D(\boldsymbol{x}')||_2^2 + ||f_C(\boldsymbol{x}) - f_C(\boldsymbol{x}')||_2^2)$$

Making VAE sharper - CVAE-GAN

C - conditioned. We want to generate an object of a particular category



InfoGAN: $\mathcal{L}_C = -\mathbb{E}_{\boldsymbol{x}\sim P_r}[\log P(c|\boldsymbol{x})].$

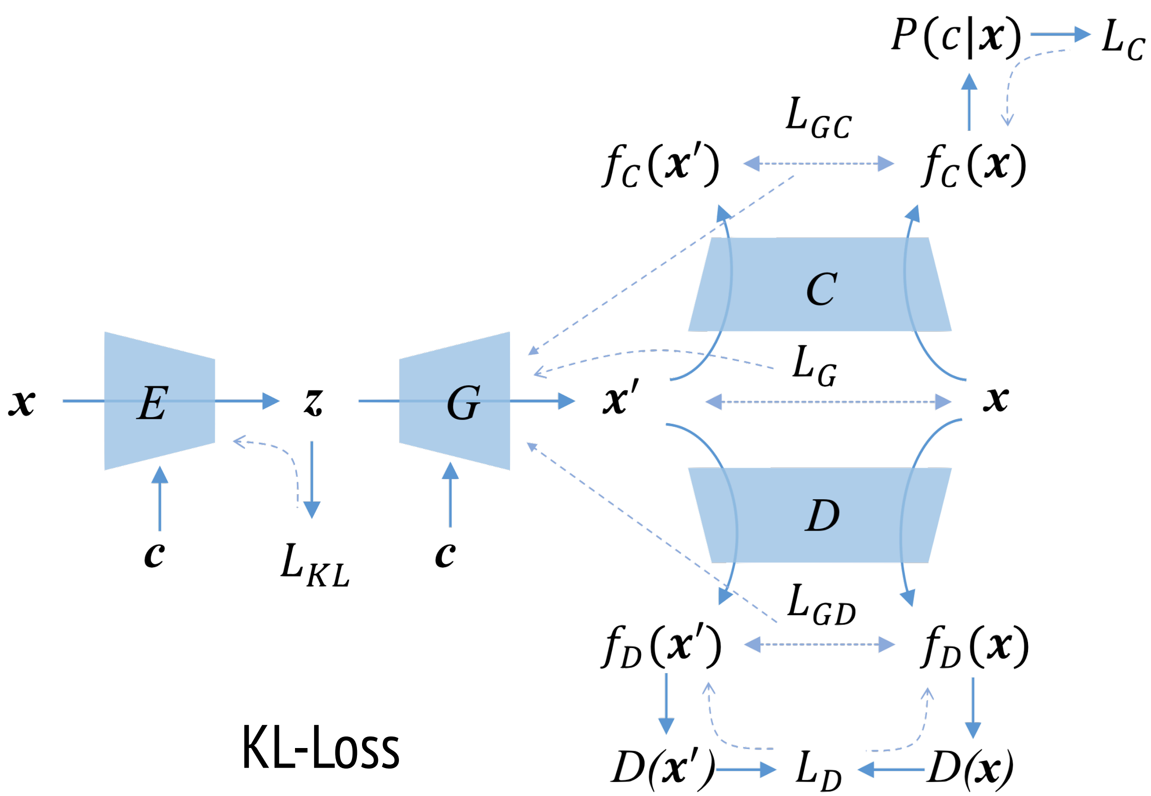
Feature Matching: $\mathcal{L}_{GC} = \frac{1}{2}\sum_c ||\mathbb{E}_{\boldsymbol{x}\sim P_r} f_C(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim P_{\boldsymbol{z}}} f_C(G(\boldsymbol{z}, c))||_2^2.$

Pairwise Feature Matching: $$\mathcal{L}_G = \frac{1}{2}(||\boldsymbol{x} - \boldsymbol{x}'||_2^2 + ||f_D(\boldsymbol{x}) - f_D(\boldsymbol{x}')||_2^2 + ||f_C(\boldsymbol{x}) - f_C(\boldsymbol{x}')||_2^2)$$

Feature Matching: $\mathcal{L}_{GD} = \frac{1}{2}||\mathbb{E}_{\boldsymbol{x}\sim P_r} f_D(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim P_{\boldsymbol{z}}} f_D(G(\boldsymbol{z}))||_2^2.$

Making VAE sharper - CVAE-GAN

C - conditioned. We want to generate an object of a particular category



InfoGAN: $\mathcal{L}_C = -\mathbb{E}_{\boldsymbol{x}\sim P_r}[\log P(c|\boldsymbol{x})].$

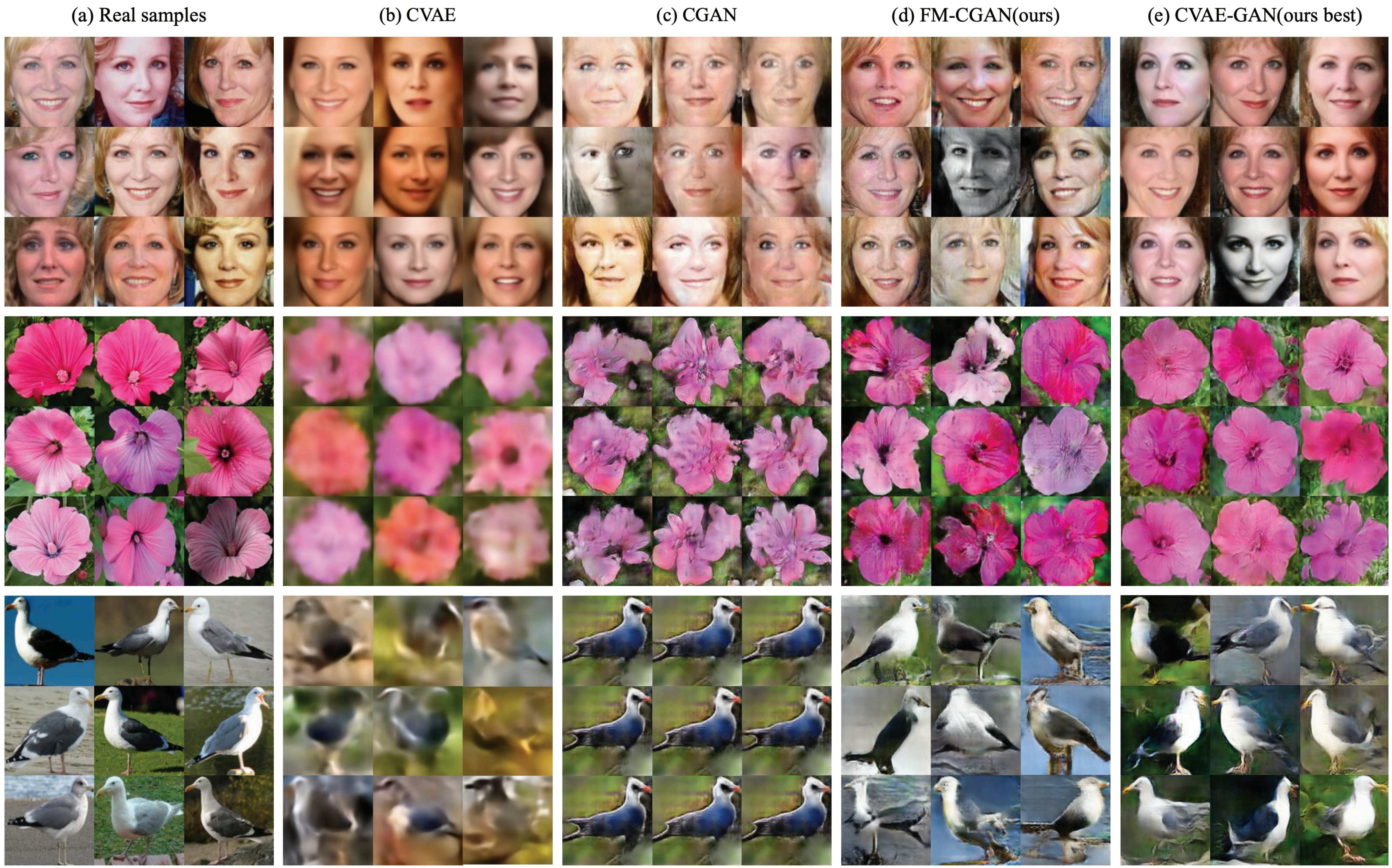
Feature Matching: $\mathcal{L}_{GC} = \frac{1}{2}\sum_c ||\mathbb{E}_{\boldsymbol{x}\sim P_r} f_C(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim P_{\boldsymbol{z}}} f_C(G(\boldsymbol{z}, c))||_2^2.$

Pairwise Feature Matching: $$\mathcal{L}_G = \frac{1}{2}(||\boldsymbol{x} - \boldsymbol{x}'||_2^2 + ||f_D(\boldsymbol{x}) - f_D(\boldsymbol{x}')||_2^2 + ||f_C(\boldsymbol{x}) - f_C(\boldsymbol{x}')||_2^2)$$

Feature Matching: $\mathcal{L}_{GD} = \frac{1}{2}||\mathbb{E}_{\boldsymbol{x}\sim P_r} f_D(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim P_{\boldsymbol{z}}} f_D(G(\boldsymbol{z}))||_2^2.$

Adversarial Loss

CVAE-GAN: Results



CVAE-GAN: Interpolation