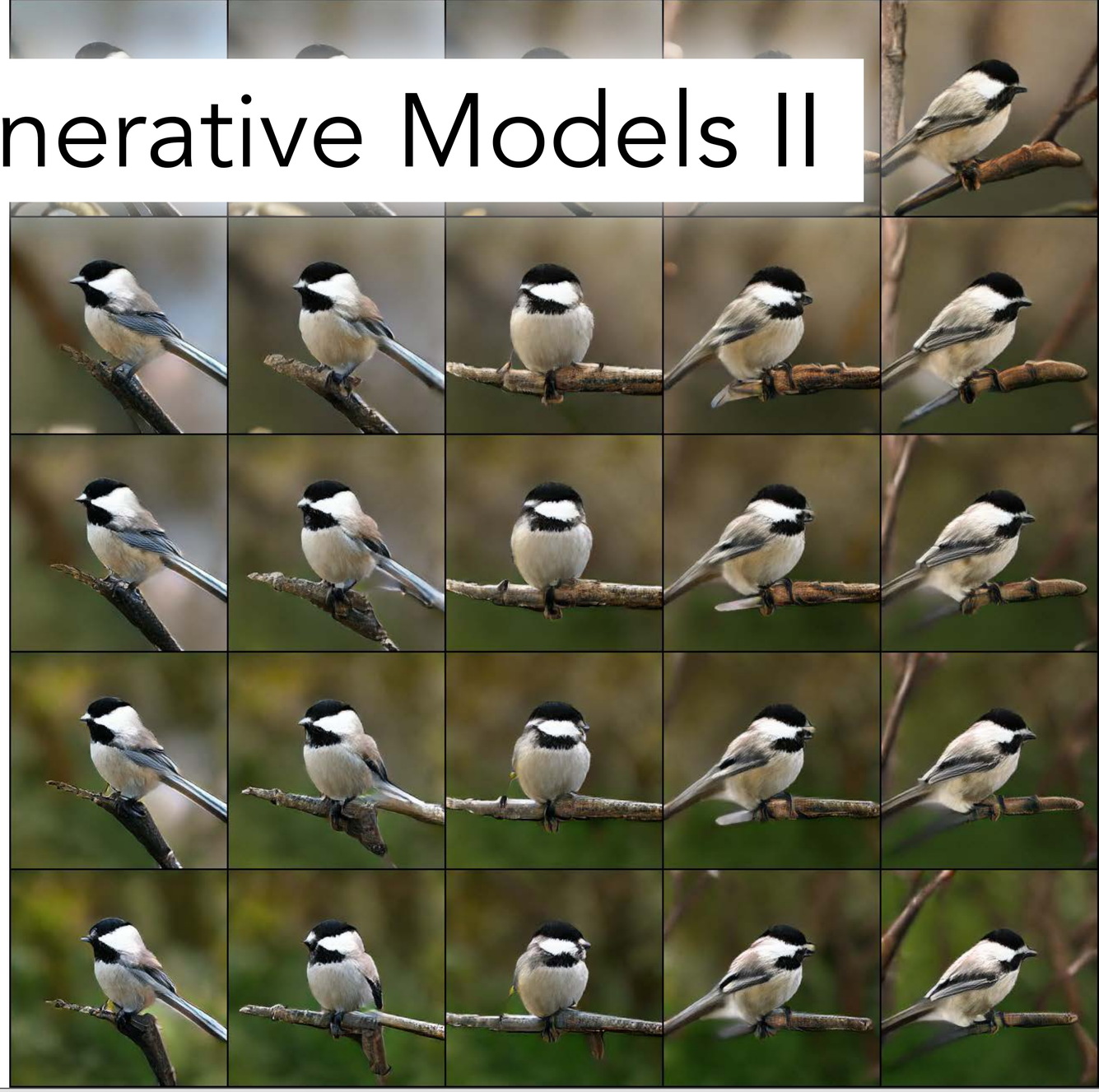
Lecture 15: Deep Generative Models II

Speaker: Phillip Isola



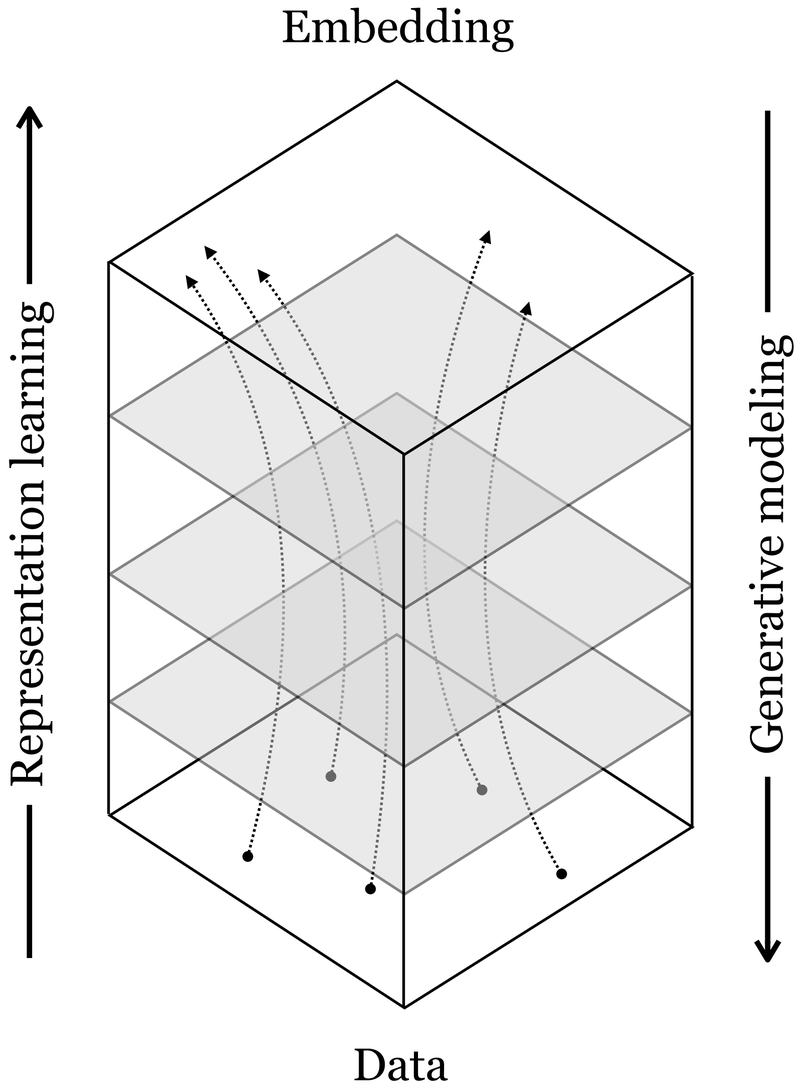
Deep generative models II

- Generative modeling as inverse representation learning
- Variational autoencoders
- Disentanglement

Generative modeling vs Representation learning

Representation learning:
mapping data to abstract representations (analysis)

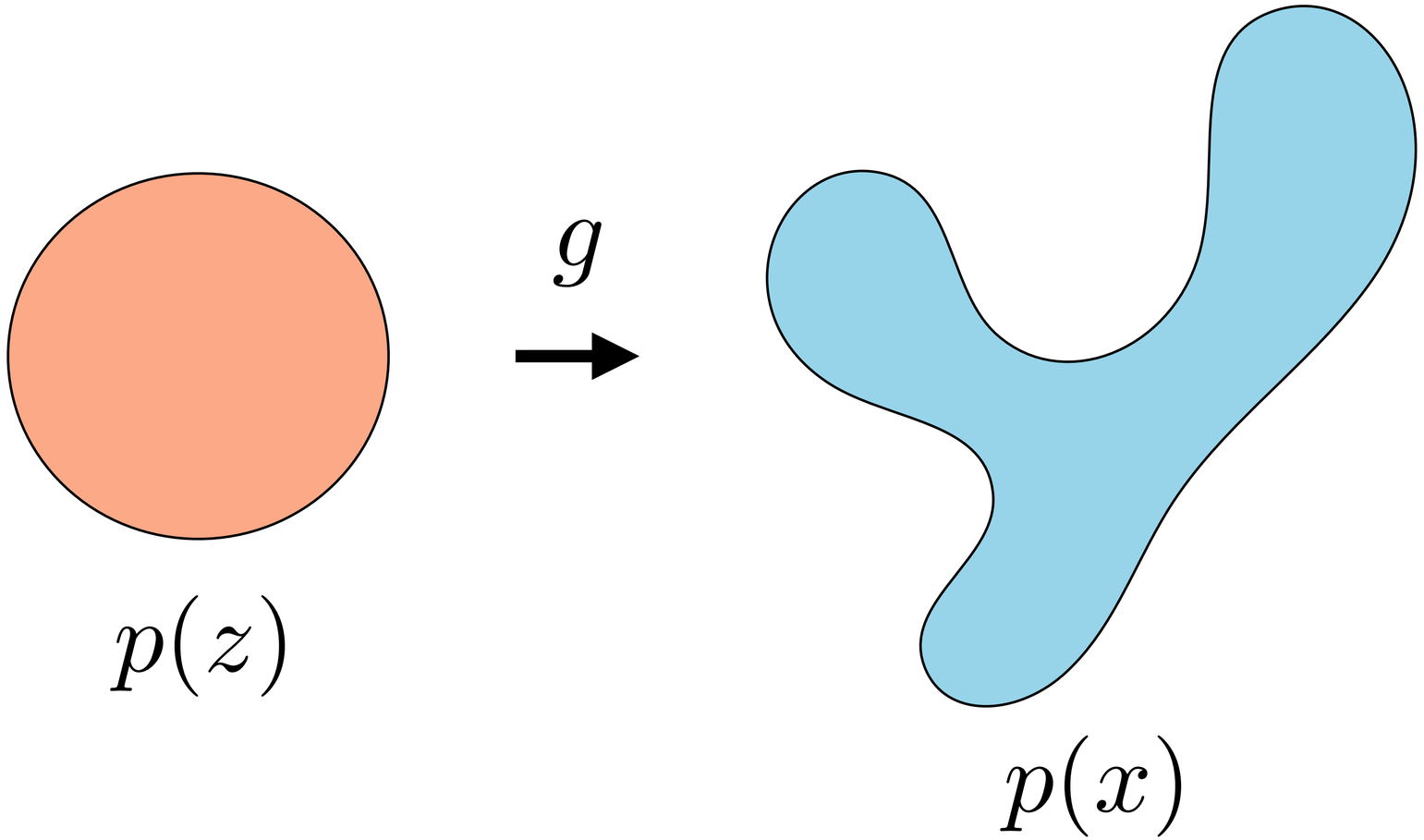
Generative modeling:
mapping abstract representations to data (*synthesis*)



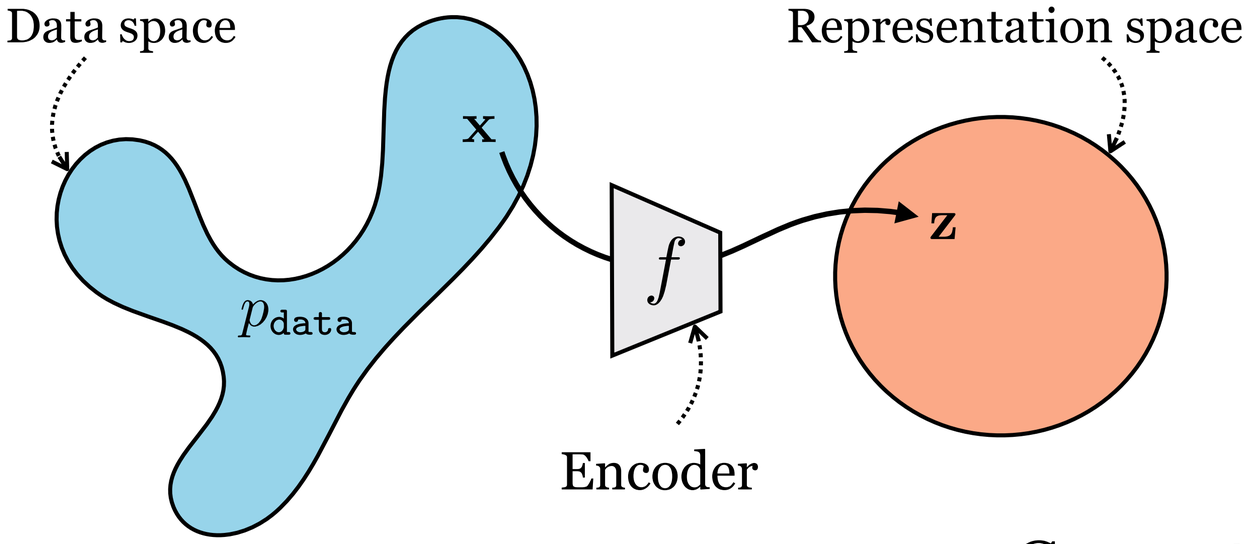
Deep generative models are distribution transformers

Prior distribution

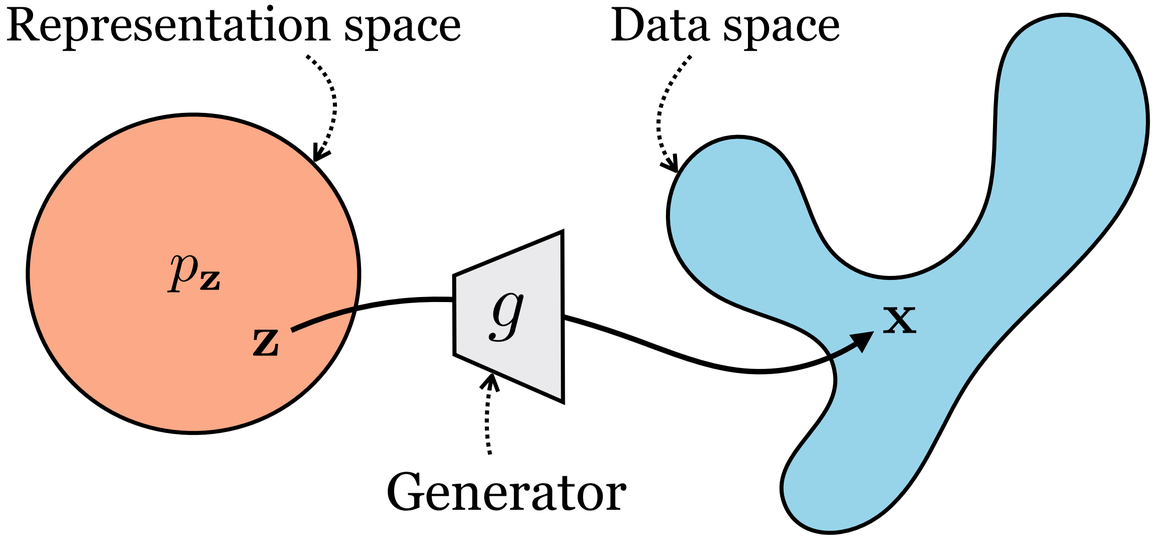
Target distribution



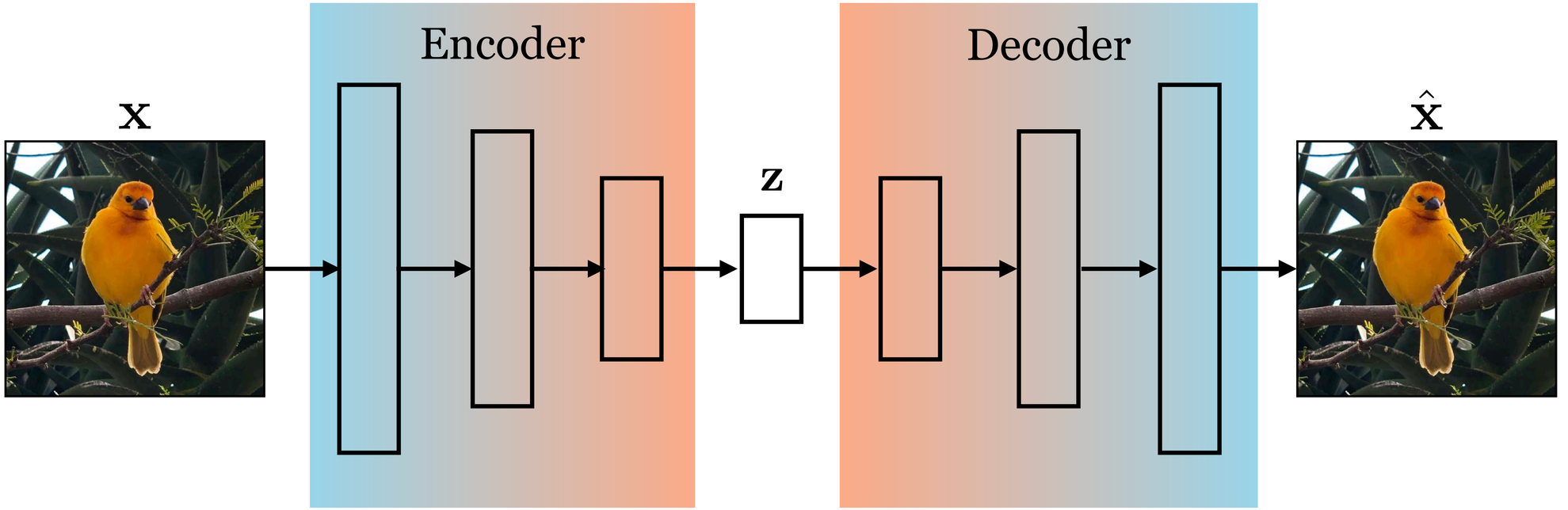
Representation learning



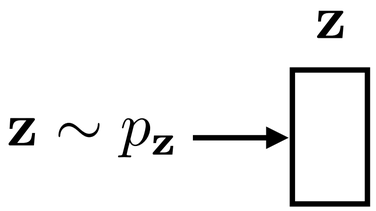
Generative modeling

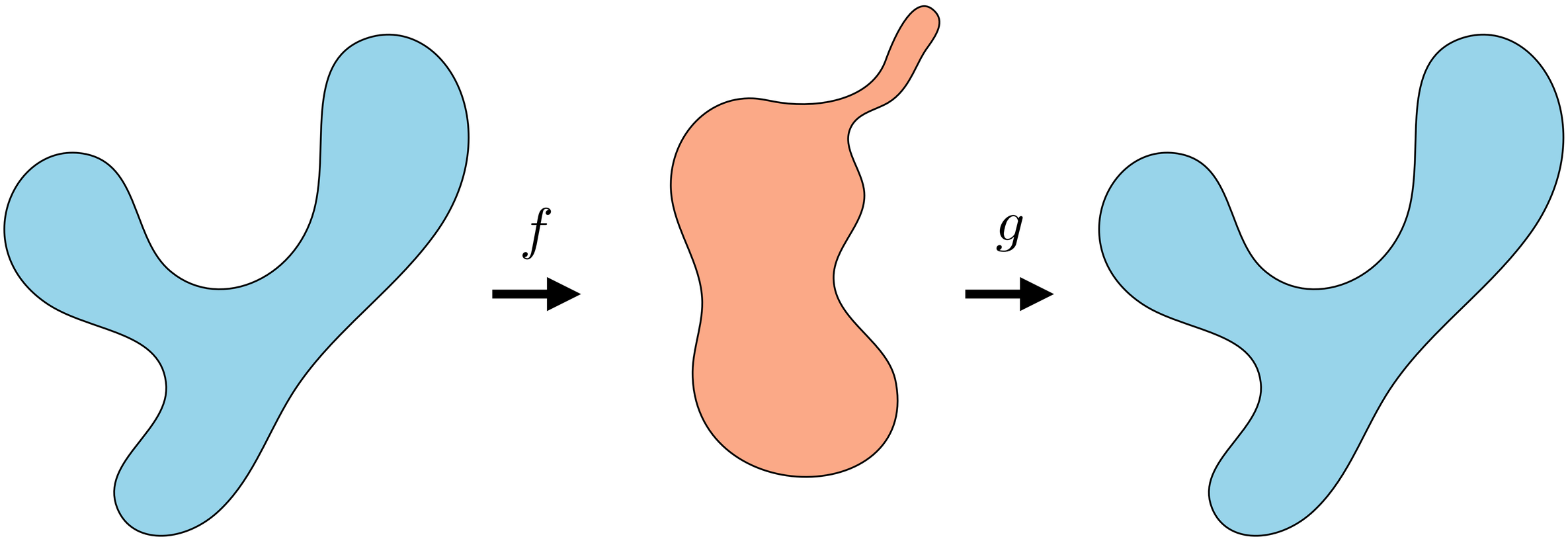


Autoencoder

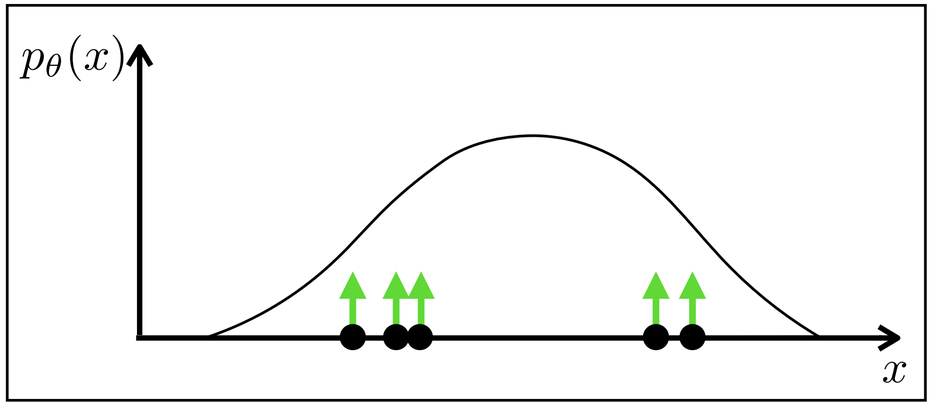


Generative model



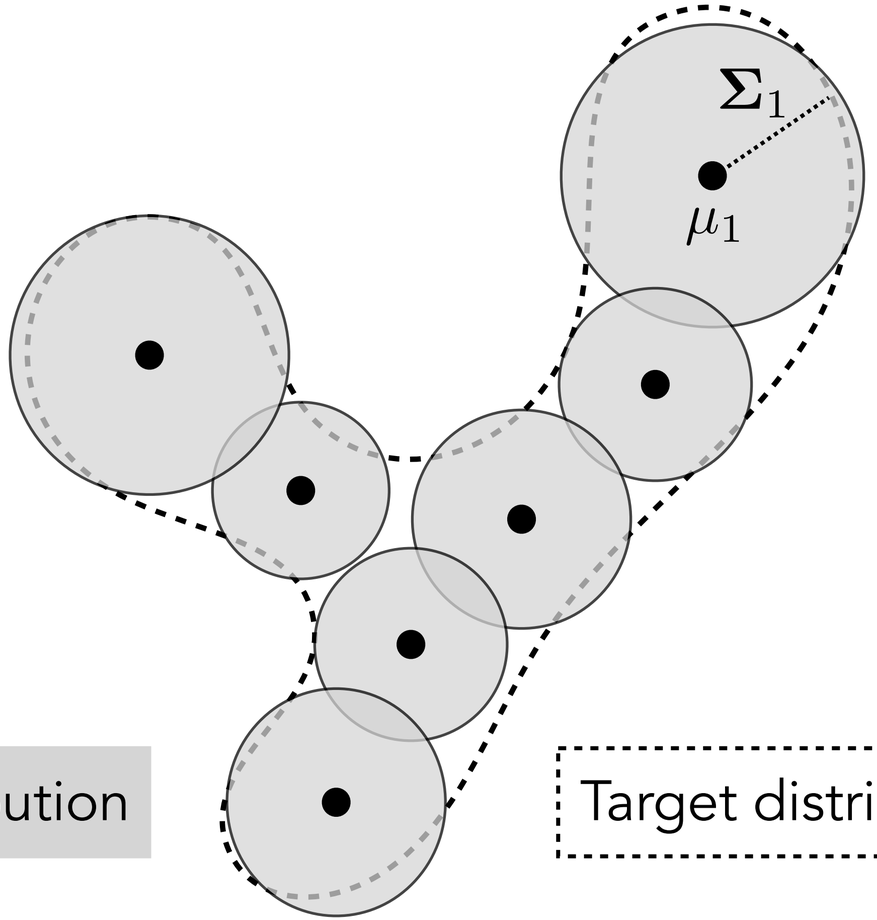

f
g

Mixture of Gaussians



Parameters: $\{w_i, \mu_i, \mathbf{\Sigma}_i\}_{i=1}^{k}$

$$p_\theta(\mathbf{x}) = \sum_{i=1}^{k} w_i \mathcal{N}(\mathbf{x}; \mu_i, \mathbf{\Sigma}_i)$$

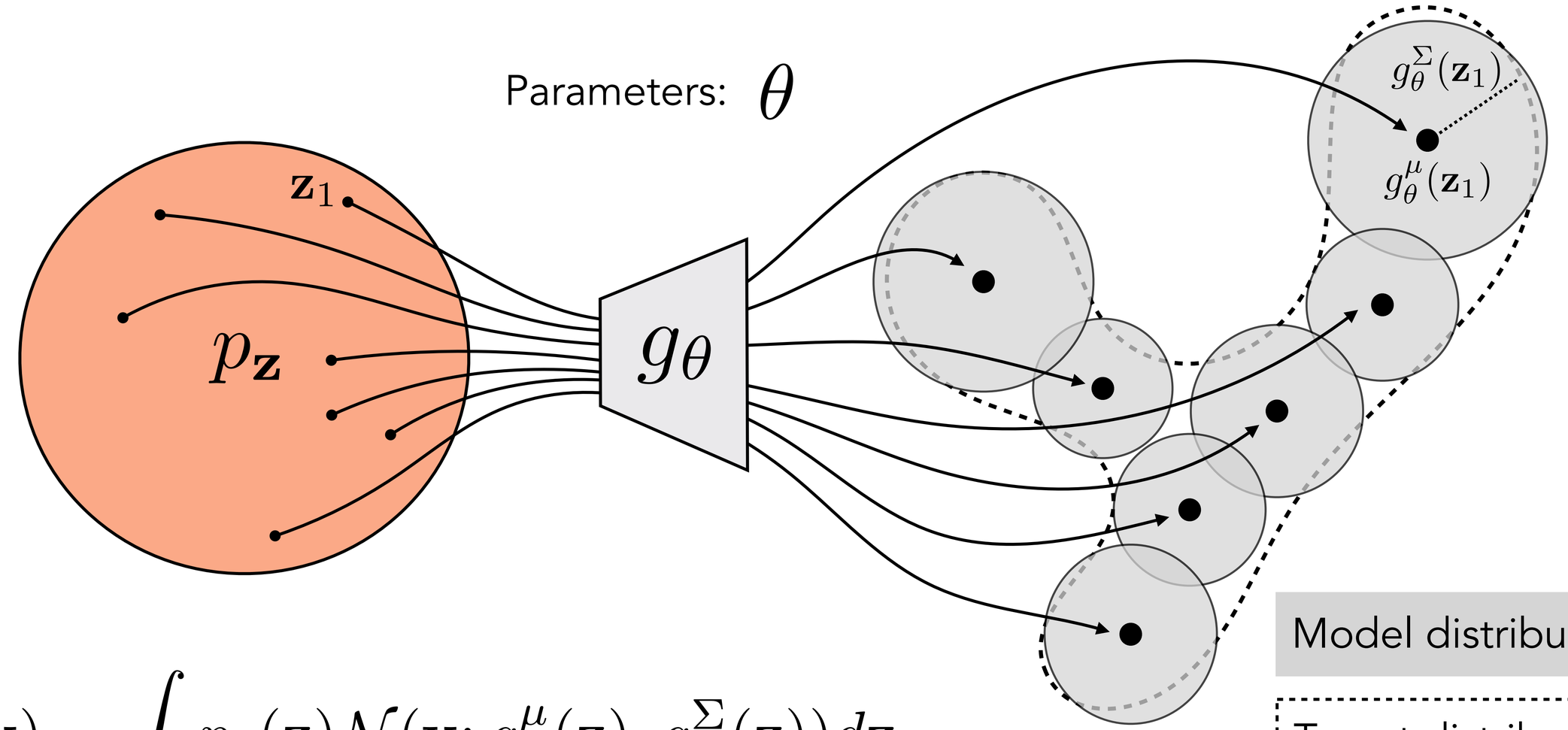


Model distribution

Target distribution

Infinite Mixture of Gaussians

Parameters: θ



$$p_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_{\mathbf{z}}(\mathbf{z}) \mathcal{N}(\mathbf{x}; g_\theta^{\mu}(\mathbf{z}), g_\theta^{\Sigma}(\mathbf{z})) d\mathbf{z}$$

Latent variable models

latent variables

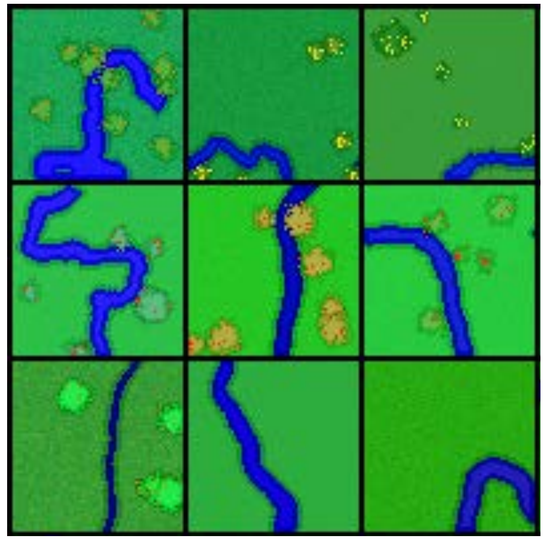
$\mathbf{z}_1 \sim \texttt{Bernoulli}(0.5)$ (River turns)

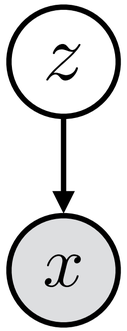
$\mathbf{z}_2 \sim \texttt{Normal}(\mu_1, \Sigma_1)$ (Grass color)

$\mathbf{z}_3 \sim \texttt{Unif}(0, 10)$ (Number trees)

$\vdots$

Generator





$$p_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z}) p_{\mathbf{z}}(\mathbf{z}) d\mathbf{z}$$

latent variables

Infinite GMM / VAE

Data $\rightarrow$

Learner

Objective

$$\max_{\theta} \mathbb{E}_{\mathbf{x} \sim p_{\texttt{data}}} [\log p_\theta(\mathbf{x})]$$

Hypothesis space

$$p_\theta(\mathbf{x}) = \int_{\mathbf{z}} \underbrace{\mathcal{N}(\mathbf{x}; g_\theta^\mu(\mathbf{z}), g_\theta^\Sigma(\mathbf{z}))}_{p_\theta(\mathbf{x} \mid \mathbf{z})} \underbrace{\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})}_{p_{\mathbf{z}}(\mathbf{z})} d\mathbf{z}$$

$$x = g_\theta^\mu(\mathbf{z}) + g_\theta^\sigma(\mathbf{z})\epsilon \qquad \epsilon \sim \mathcal{N}(0, 1)$$

"Reparameterization trick"

$\rightarrow$

Density

$$p_\theta : \mathcal{X} \rightarrow [0, \infty)$$

Sampler

$$g_\theta : \mathcal{Z} \rightarrow \mathcal{X}$$

VAEs — learning the model parameters

$$\theta^* = \arg\max_\theta L(\{\mathbf{x}^{(i)}\}_{i=1}^N, \theta)$$

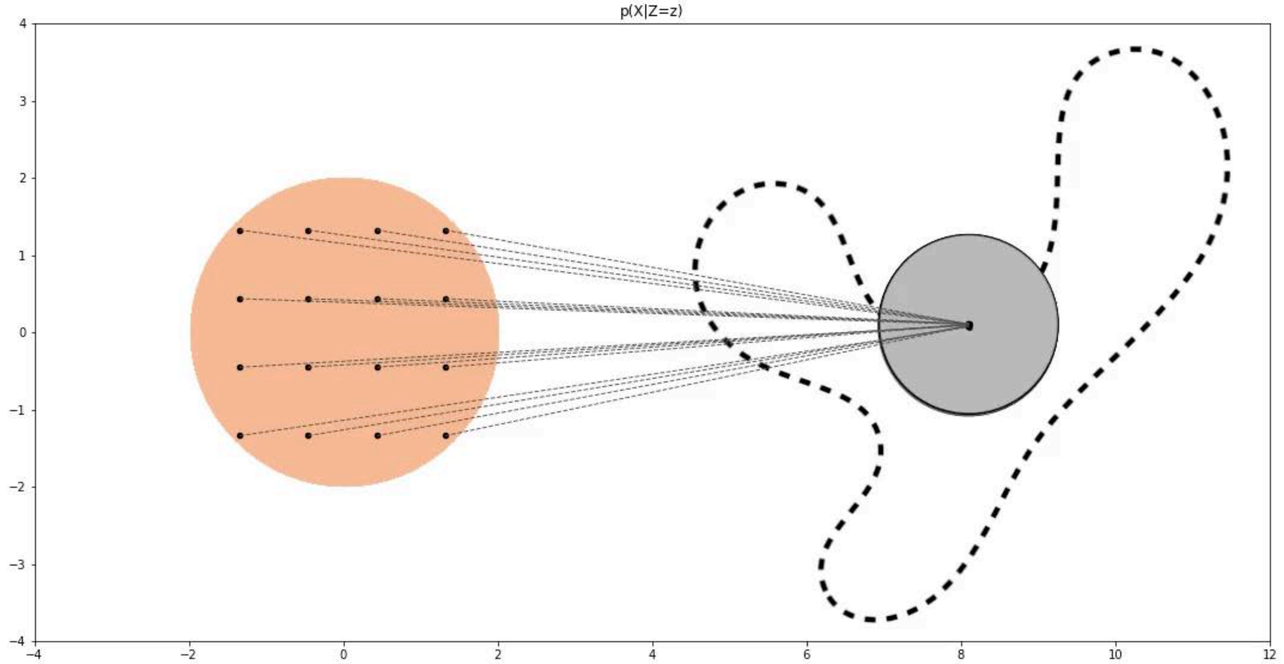
$$= \arg\max_\theta \sum_{i=1}^{N} \log \underbrace{\int_{\mathbf{z}} \overbrace{\mathcal{N}(\mathbf{x}^{(i)}; g_\theta^\mu(\mathbf{z}), g_\theta^\Sigma(\mathbf{z}))}^{p_\theta(\mathbf{x}^{(i)} \mid \mathbf{z})} \overbrace{\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})}^{p_\mathbf{z}(\mathbf{z})} d\mathbf{z}}_{p_\theta(\mathbf{x}^{(i)})}$$

Trick #1: Estimate the integral via sampling (Monte Carlo)

$$
\begin{aligned}
p_\theta(\mathbf{x}) &= \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z}) p_{\mathbf{z}}(\mathbf{z}) d\mathbf{z} \\
&= \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}\left[p_\theta(\mathbf{x}|\mathbf{z})\right] \\
&\approx \frac{1}{M}\sum_{i=1}^{M} p_\theta(\mathbf{x}|\mathbf{z}^{(j)}), \quad \mathbf{z}^{(j)} \sim p_{\mathbf{z}}
\end{aligned}
$$

Fitting an infinite mixture of Gaussians via Monte Carlo

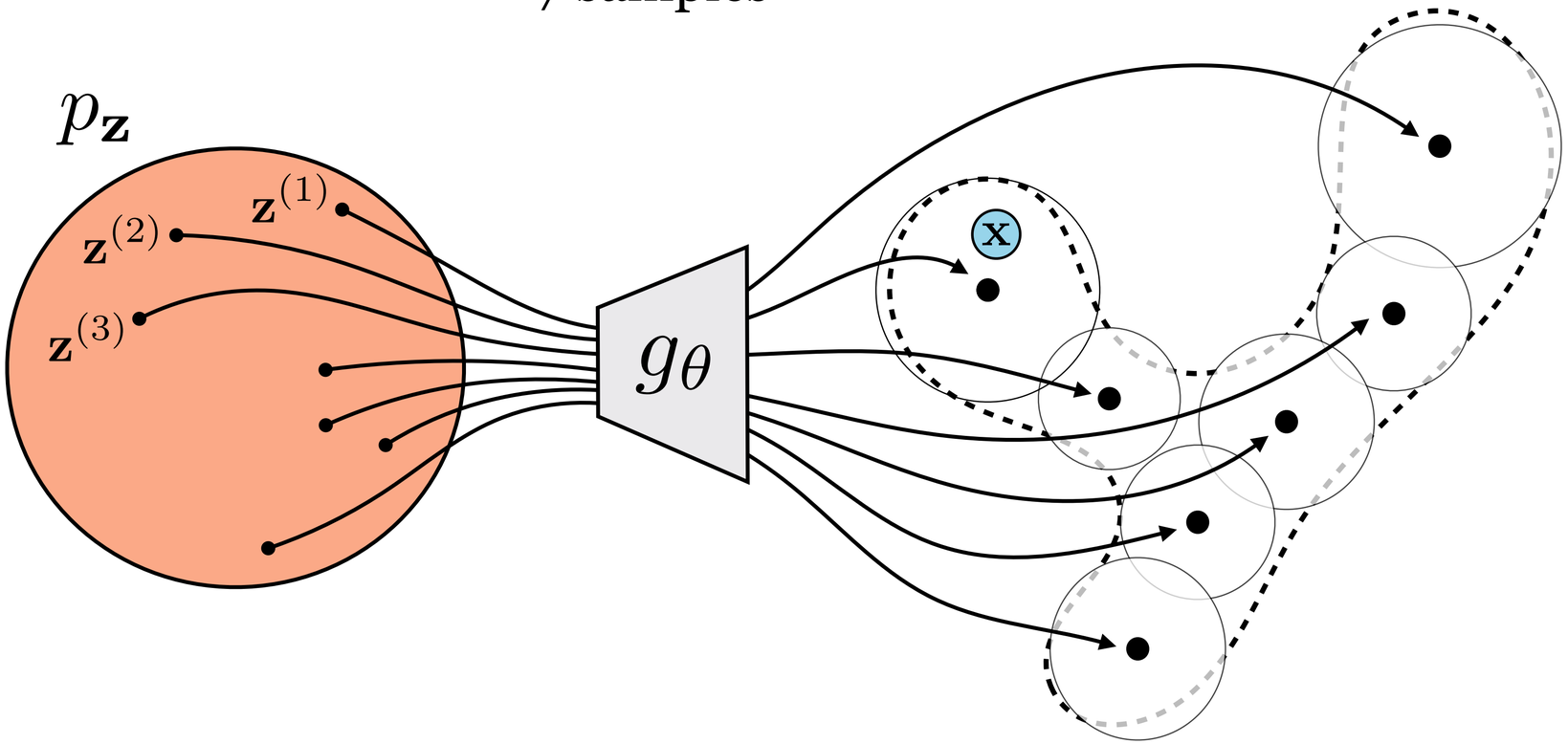
$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{N} \log \frac{1}{M} \sum_{j=1}^{M} \overbrace{\mathcal{N}(\mathbf{x}^{(i)}; g_\theta^\mu(\mathbf{z}^{(j)}), g_\theta^\Sigma(\mathbf{z}^{(j)}))}^{p_\theta(\mathbf{x}^{(i)} \mid \mathbf{z}^{(j)})}$$



Trick #2: Importance sampling

Random sampling $\mathbf{z}^{(j)} \sim p_{\mathbf{z}}$

7 samples

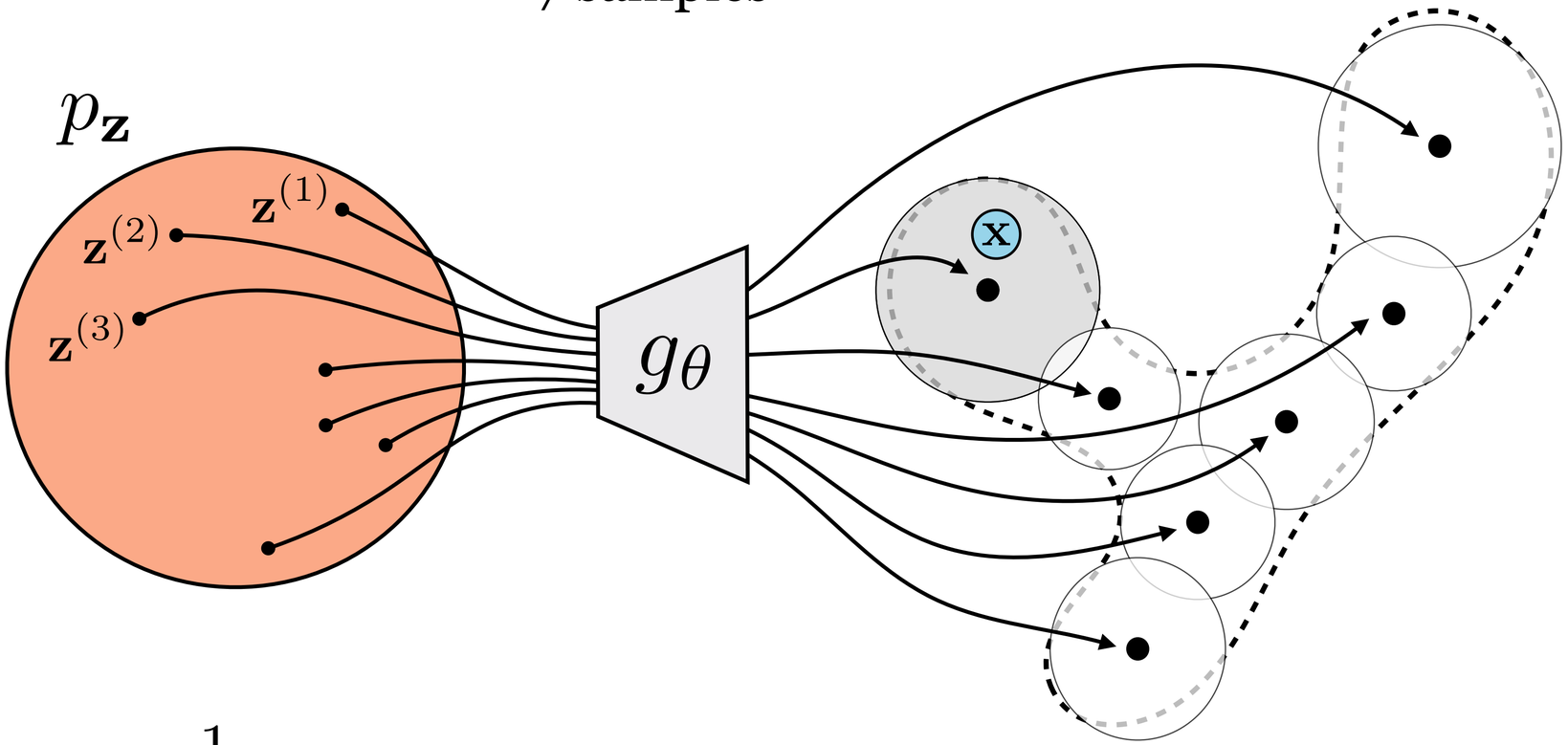


$$p_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}\left[p_\theta(\mathbf{x}|\mathbf{z})\right]$$

Trick #2: Importance sampling

Random sampling $\mathbf{z}^{(j)} \sim p_{\mathbf{z}}$

7 samples



$$p_\theta(\mathbf{x}) \approx \frac{1}{M}(p_\theta(\mathbf{x}|\mathbf{z}^{(1)}) + p_\theta(\mathbf{x}|\mathbf{z}^{(2)}) + p_\theta(\mathbf{x}|\mathbf{z}^{(3)}) + \ldots)$$

$$p_\theta(\mathbf{x}) \approx \frac{1}{M}(0 + p_\theta(\mathbf{x}|\mathbf{z}^{(2)}) + 0 + \ldots)$$

Trick #2: Importance sampling

$$p_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim p_\mathbf{z}}\left[p_\theta(\mathbf{x}\,|\,\mathbf{z})\right] = \int_\mathbf{z} p_\mathbf{z}(\mathbf{z}) p_\theta(\mathbf{x}\,|\,\mathbf{z}) d\mathbf{z} = \int_\mathbf{z} q_\mathbf{z}(\mathbf{z}) \frac{p_\mathbf{z}(\mathbf{z})}{q_\mathbf{z}(\mathbf{z})} p_\theta(\mathbf{x}\,|\,\mathbf{z}) d\mathbf{z}$$

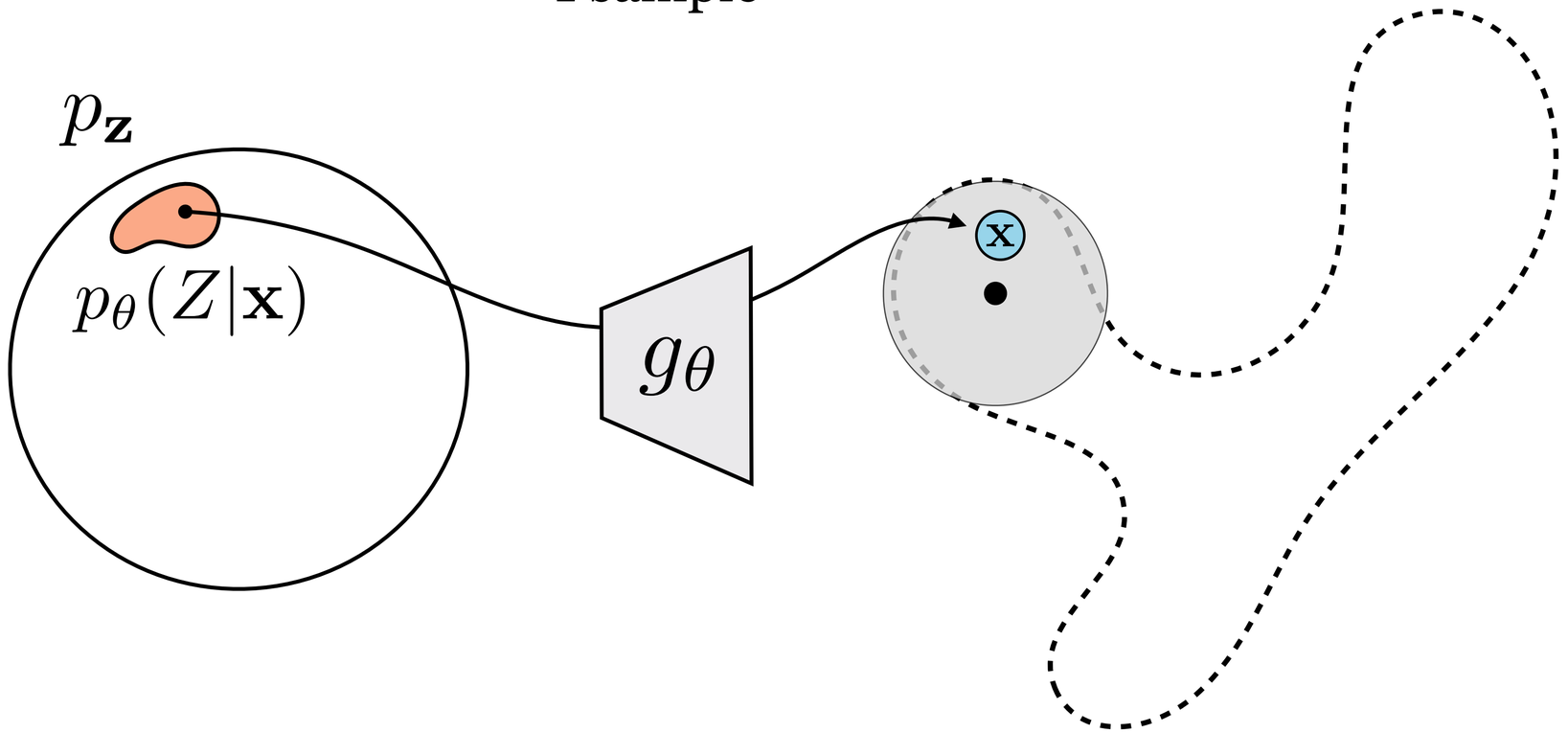
$$= \mathbb{E}_{\mathbf{z}\sim q_\mathbf{z}}\left[\frac{p_\mathbf{z}(\mathbf{z})}{q_\mathbf{z}(\mathbf{z})} p_\theta(\mathbf{x}\,|\,\mathbf{z})\right]$$

Set $q_\mathbf{z} = p_\theta(Z|\mathbf{x})$

Trick #2: Importance sampling

Importance sampling $\quad \mathbf{z}^{(j)} \sim p_\theta(Z|\mathbf{x})$

1 sample

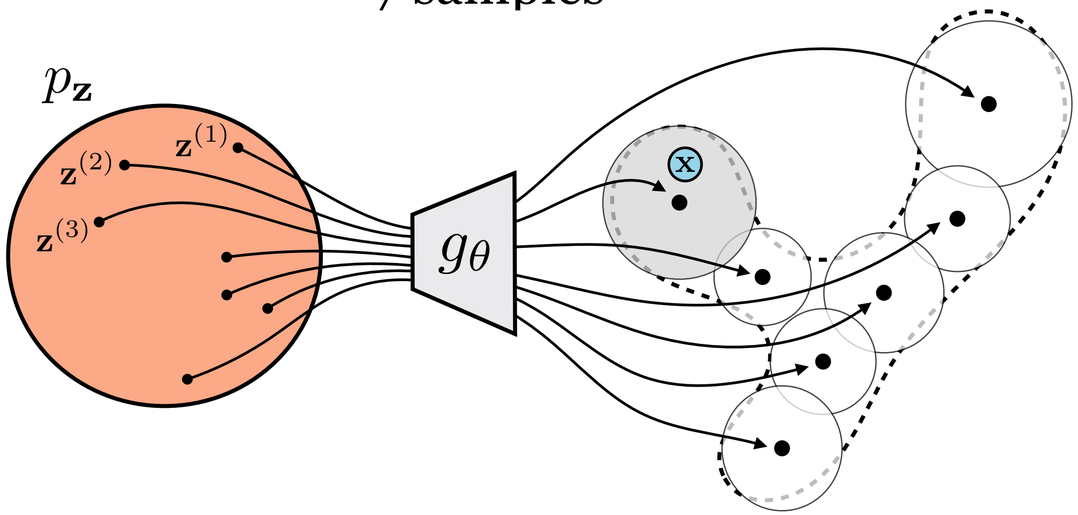


$$p_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim p_\mathbf{z}}\left[p_\theta(\mathbf{x}|\mathbf{z})\right] \longrightarrow p_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim p_\theta(Z|\mathbf{x})}\left[\frac{p_\mathbf{z}(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z})\right]$$

Trick #2: Importance sampling

Random sampling

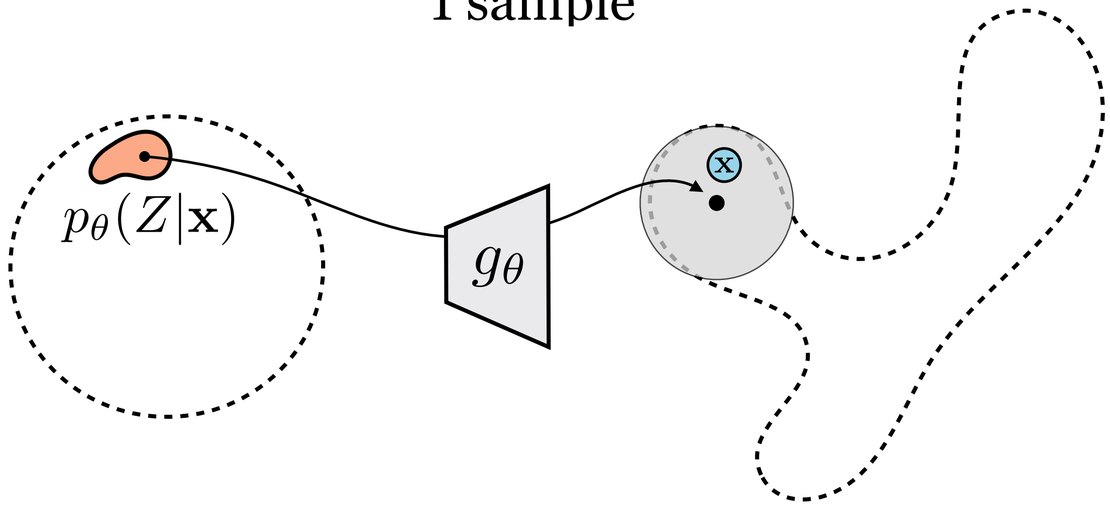
7 samples



$$p_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}\left[p_\theta(\mathbf{x}|\mathbf{z})\right]$$

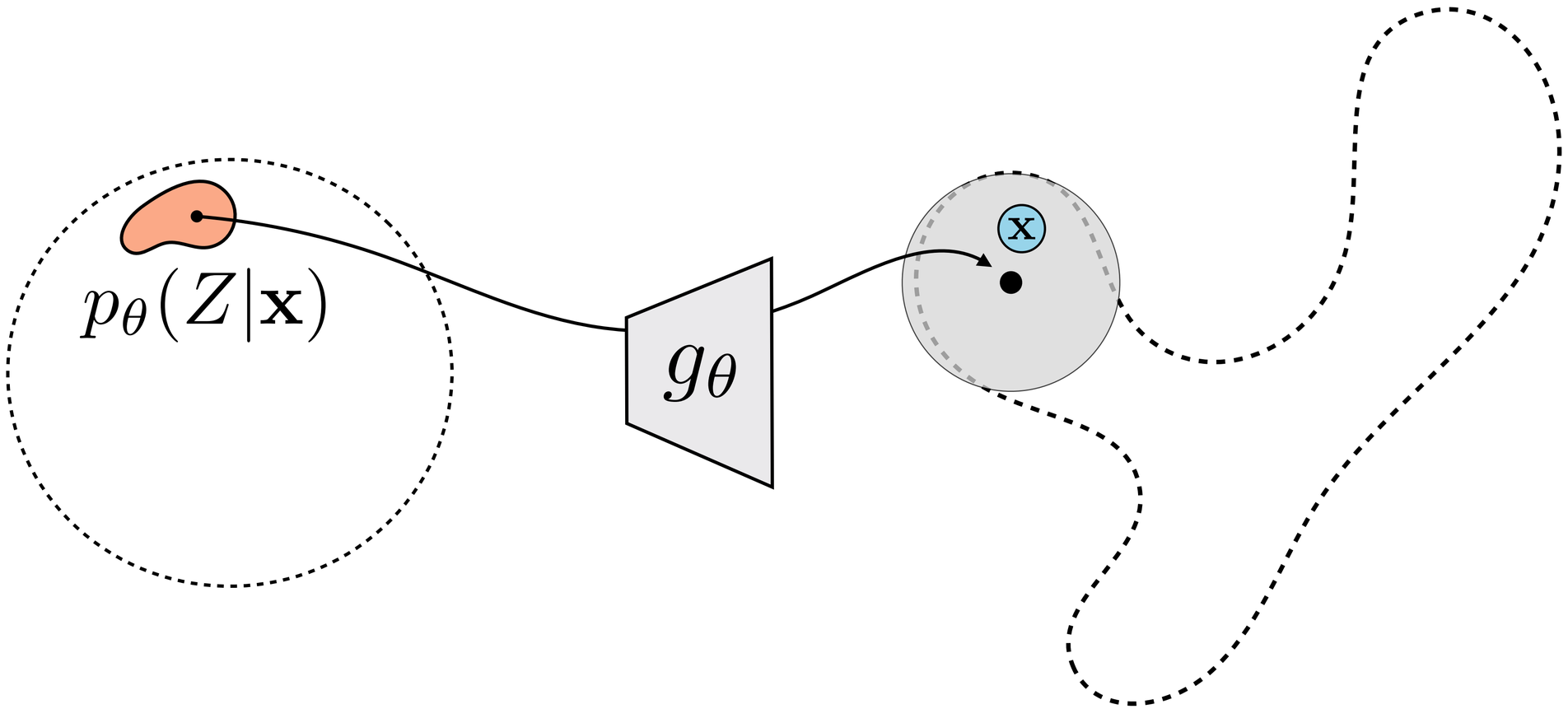
Importance sampling

1 sample

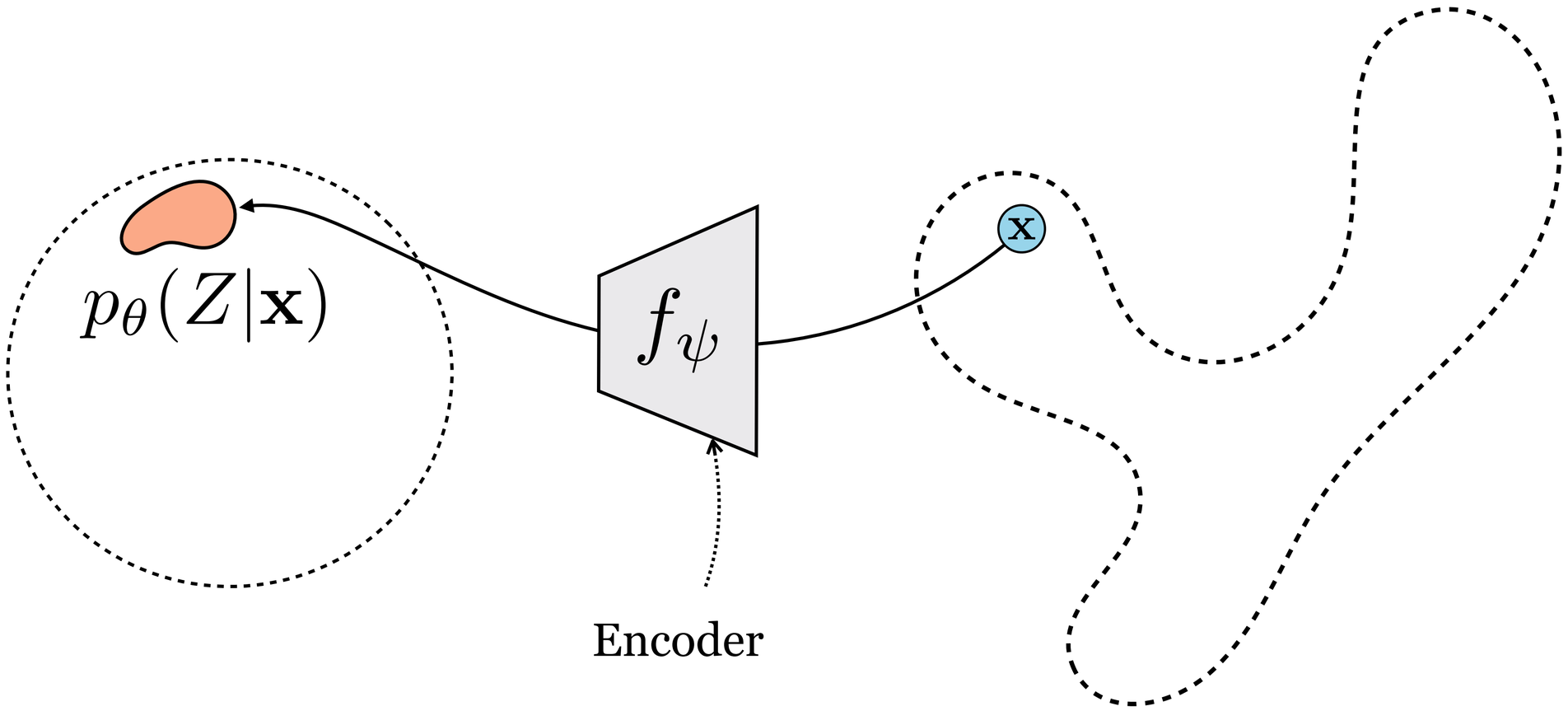


$$p_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim p_\theta(Z|\mathbf{x})}\left[\frac{p_{\mathbf{z}}(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})}p_\theta(\mathbf{x}|\mathbf{z})\right]$$

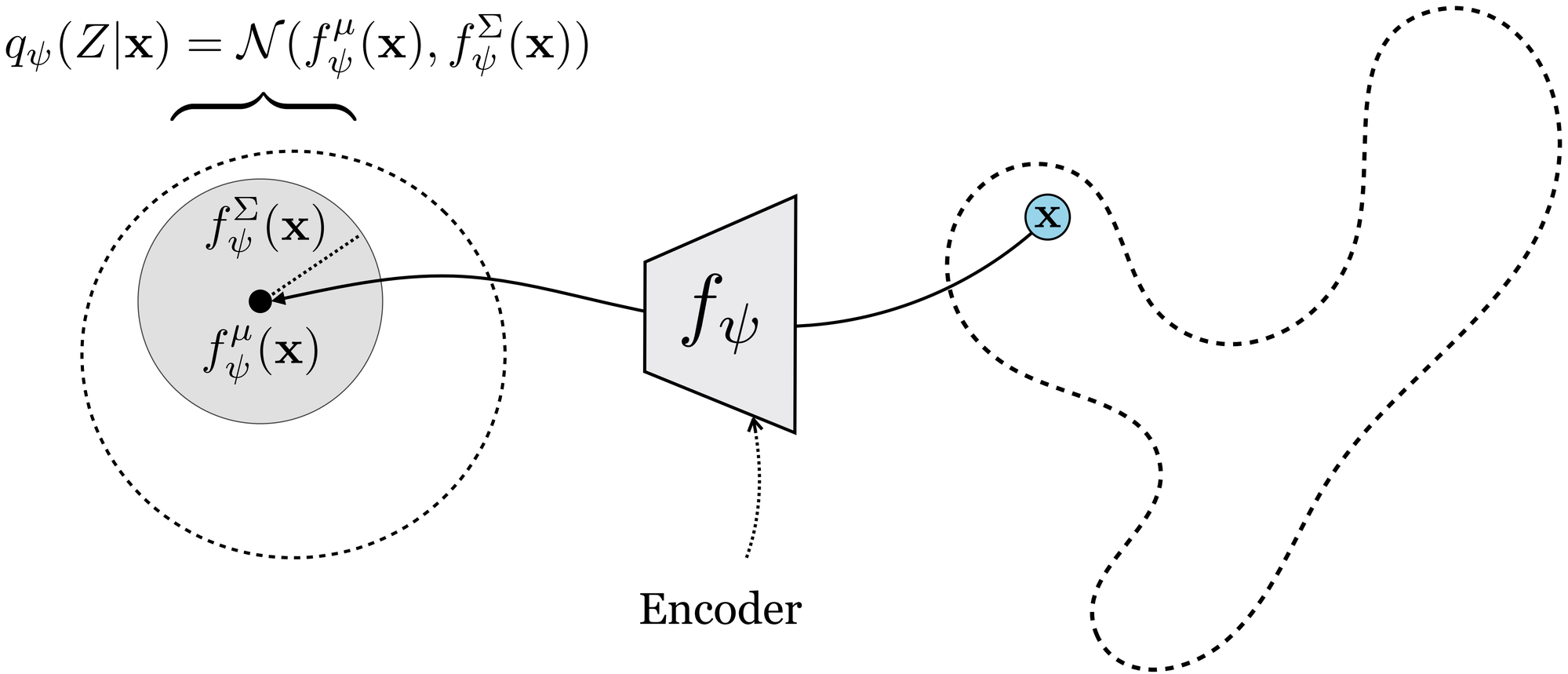
Trick #3: Predict the optimal sampling distribution for each given datapoint



Trick #3: Predict the optimal sampling distribution for each given datapoint



Trick #3: Predict the optimal sampling distribution for each given datapoint



Trick #3: Predict the optimal sampling distribution for each given datapoint

We are fitting this ... to model this

$$J_q(\mathbf{x}, \psi) = -\mathrm{KL}(q_\psi(Z \mid \mathbf{x}) \,\|\, p_\theta(Z \mid \mathbf{x}))$$

$$= \mathbb{E}_{\mathbf{z} \sim q_\psi(Z \mid \mathbf{x})}[-\log q_\psi(\mathbf{z} \mid \mathbf{x}) + \log p_\theta(\mathbf{x} \mid \mathbf{z}) + \log p_{\mathbf{z}}(\mathbf{z})] - \log p_\theta(\mathbf{x})$$

Our learning problem for q:

$$\psi^* = \arg\max_\psi \frac{1}{N} \sum_{i=1}^{N} J_q(\mathbf{x}^{(i)}, \psi)$$

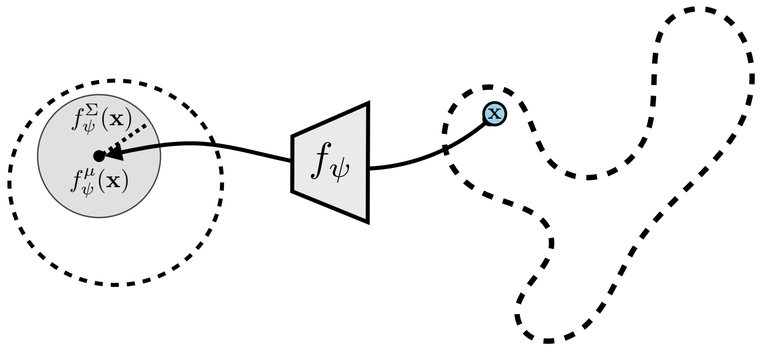
$$= \arg\max_\psi \frac{1}{N} \sum_{i=1}^{N} \underbrace{\mathbb{E}_{\mathbf{z} \sim q_\psi(Z \mid \mathbf{x}^{(i)})}[-\log q_\psi(\mathbf{z} \mid \mathbf{x}^{(i)}) + \log p_\theta(\mathbf{x}^{(i)} \mid \mathbf{z}) + \log p_{\mathbf{z}}(\mathbf{z})]}_{J}$$

Putting everything together

Our learning problem for q:

$$\psi^* = \arg\max_{\psi} \frac{1}{N} \sum_{i=1}^{N} J_q(\mathbf{x}^{(i)}, \psi)$$

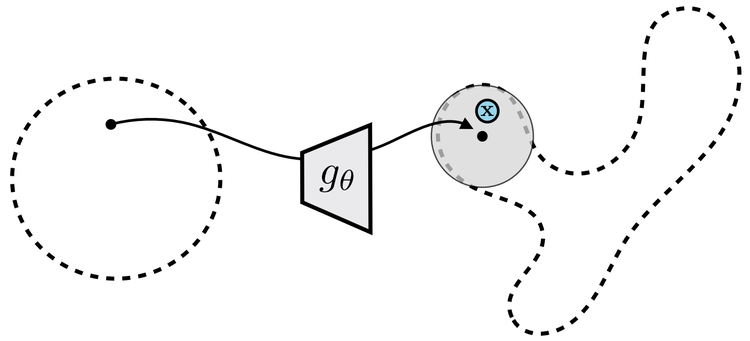
$$= \arg\max_{\psi} \frac{1}{N} \sum_{i=1}^{N} \underbrace{\mathbb{E}_{\mathbf{z} \sim q_\psi(Z \mid \mathbf{x}^{(i)})}[-\log q_\psi(\mathbf{z} \mid \mathbf{x}^{(i)}) + \log p_\theta(\mathbf{x}^{(i)} \mid \mathbf{z}) + \log p_{\mathbf{z}}(\mathbf{z})]}_{J}$$



Our learning problem for p:

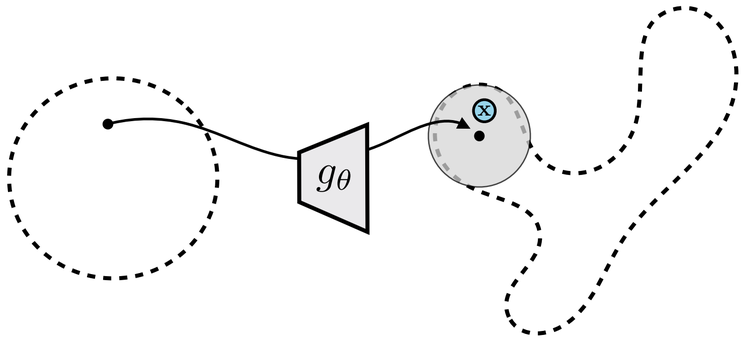
$$J_p(\mathbf{x}, \theta) = \log \mathbb{E}_{\mathbf{z} \sim q_\psi(Z \mid \mathbf{x})} \left[\frac{p_{\mathbf{z}}(\mathbf{z})}{q_\psi(\mathbf{z} \mid \mathbf{x})} p_\theta(\mathbf{x} \mid \mathbf{z}) \right]$$

$$\theta^* = \arg\max_{\theta} \frac{1}{N} \sum_{i=1}^{N} J_p(\mathbf{x}^{(i)}, \theta)$$



Putting everything together

Improving our learning problem for p:



$$J_p(\mathbf{x},\theta) = \log \mathbb{E}_{\mathbf{z}\sim q_\psi(Z|\mathbf{x})}\left[\frac{p_\mathbf{z}(\mathbf{z})}{q_\psi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z})\right]$$

$$\geq \mathbb{E}_{\mathbf{z}\sim q_\psi(Z|\mathbf{x})}\left[\log\left(\frac{p_\mathbf{z}(\mathbf{z})}{q_\psi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z})\right)\right] \qquad \triangleleft \quad \text{Jensen's inequality}$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\psi(Z|\mathbf{x})}\left[-\log q_\psi(\mathbf{z}|\mathbf{x}) + \log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\mathbf{z}(\mathbf{z})\right]$$

$$= J \qquad \triangleleft \quad \text{VAE objective}$$

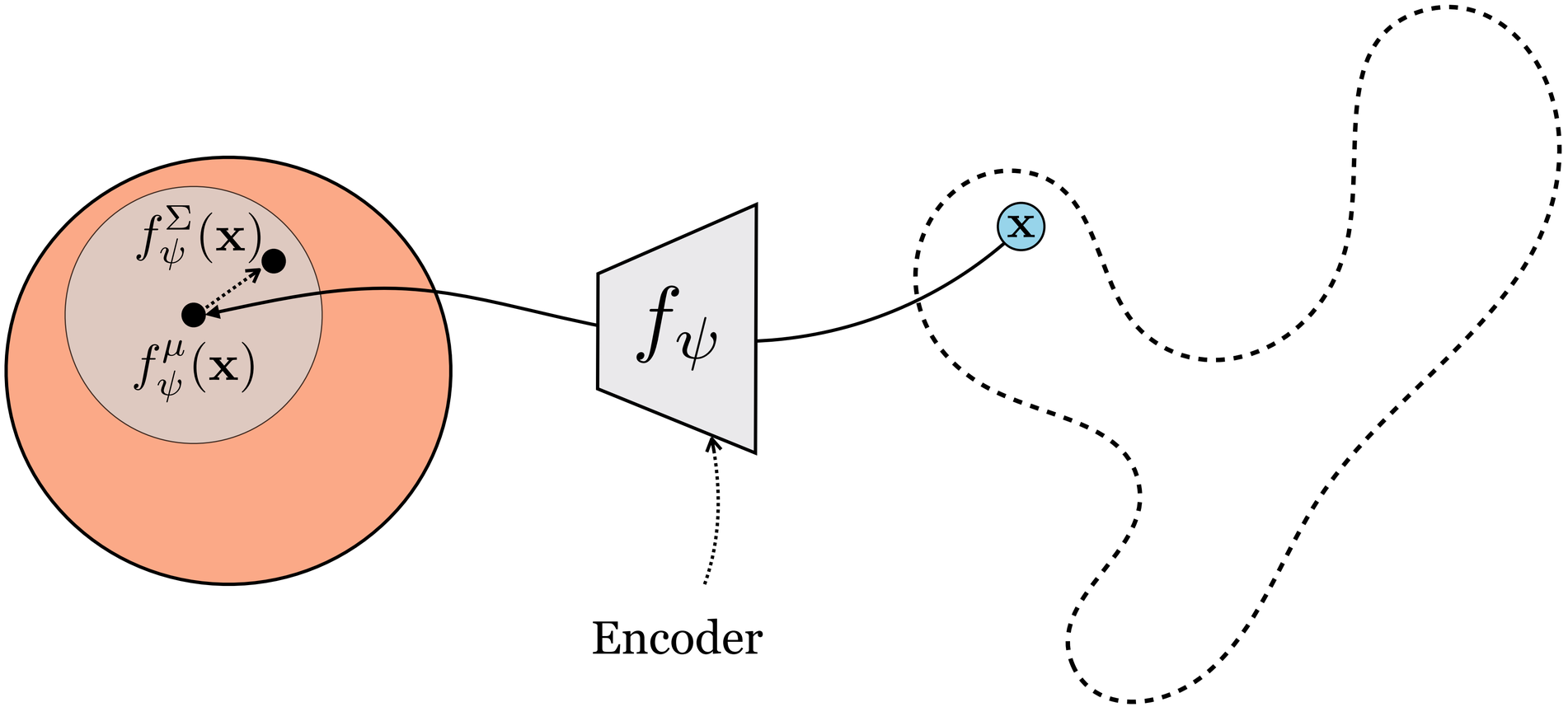
$$= \mathbb{E}_{\mathbf{z}\sim q_\psi(Z|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}(q_\psi(Z|\mathbf{x}) \,\|\, p_\mathbf{z})$$

← **J = Evidence Lower-Bound (ELBO)**

Now, p and q share the same exact objective!

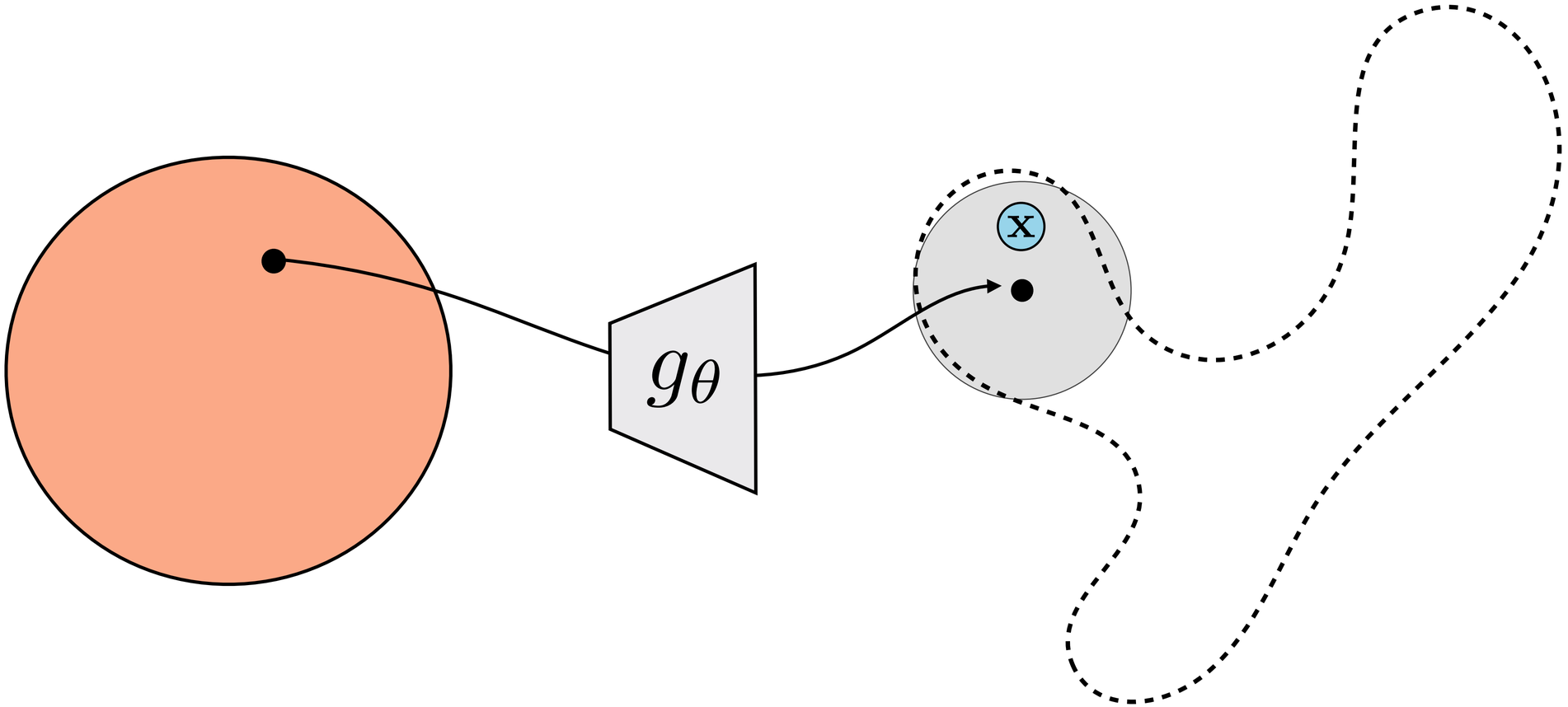
$$\theta^*, \psi^* = \arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} J(\mathbf{x}^{(i)}, \theta, \phi)$$

Trick #3: Predict the optimal sampling distribution for each given datapoint



$$p_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\psi(Z|\mathbf{x})}\left[\frac{p_\mathbf{z}(\mathbf{z})}{q_\psi(\mathbf{z}|\mathbf{x})}p_\theta(\mathbf{x}|\mathbf{z})\right]$$

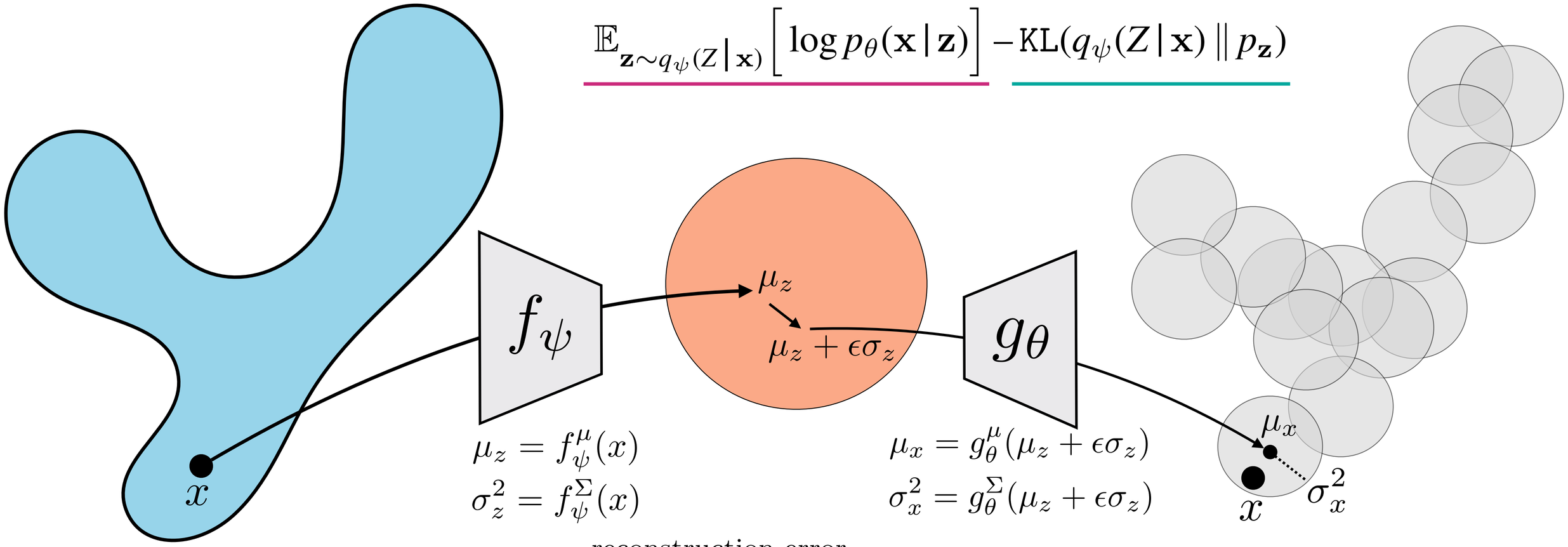
Trick #3: Predict the optimal sampling distribution for each given datapoint



$$p_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim q_\psi(Z|\mathbf{x})} \left[\frac{p_\mathbf{z}(\mathbf{z})}{q_\psi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z}) \right]$$

Looks like an autoencoder!

$$\mathbb{E}_{\mathbf{z}\sim q_\psi(Z|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] - \mathrm{KL}(q_\psi(Z|\mathbf{x}) \,\|\, p_\mathbf{z})$$



$$\mu_z = f_\psi^\mu(x)$$
$$\sigma_z^2 = f_\psi^\Sigma(x)$$

$$\mu_x = g_\theta^\mu(\mu_z + \epsilon\sigma_z)$$
$$\sigma_x^2 = g_\theta^\Sigma(\mu_z + \epsilon\sigma_z)$$

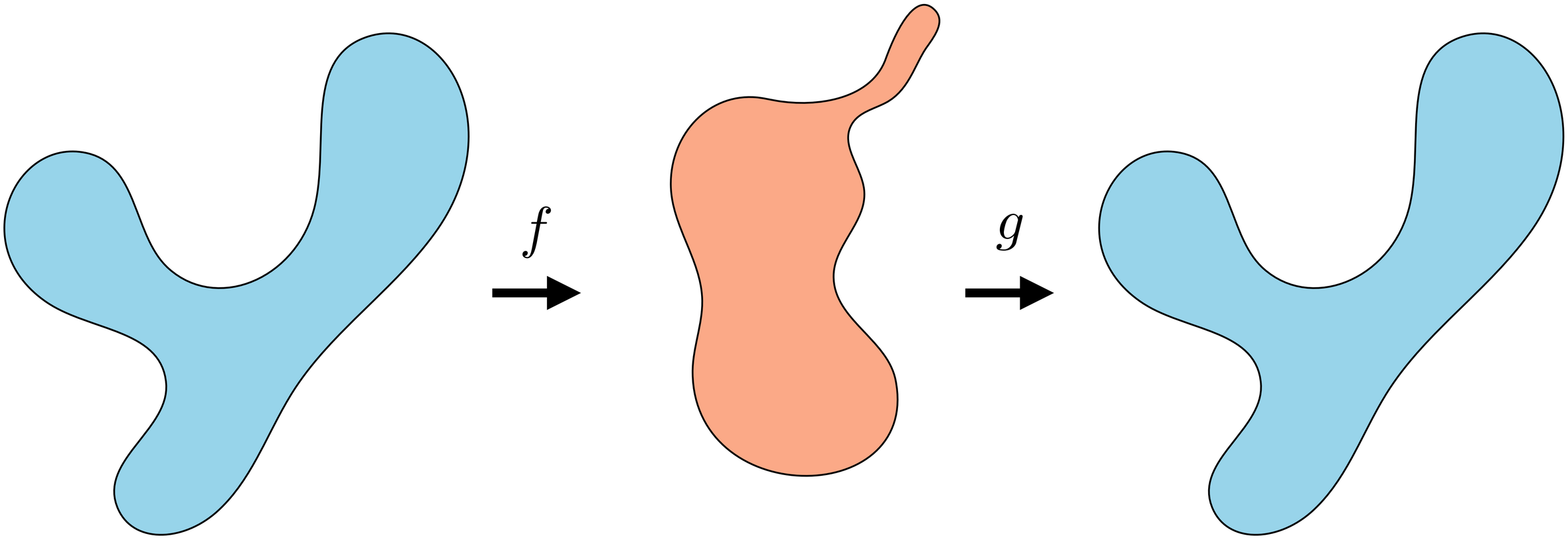
$$\log \frac{1}{\sigma_x\sqrt{2\pi}} - \frac{\overbrace{(x-\mu_x)^2}^{\text{reconstruction error}}}{2\sigma_x^2} \quad - \quad \frac{1}{2}(\mu_z^2 + \sigma_z^2 - \log(\sigma_z^2) - 1)$$

(eqns for 1D x and z)

data likelihood under 1 importance sample

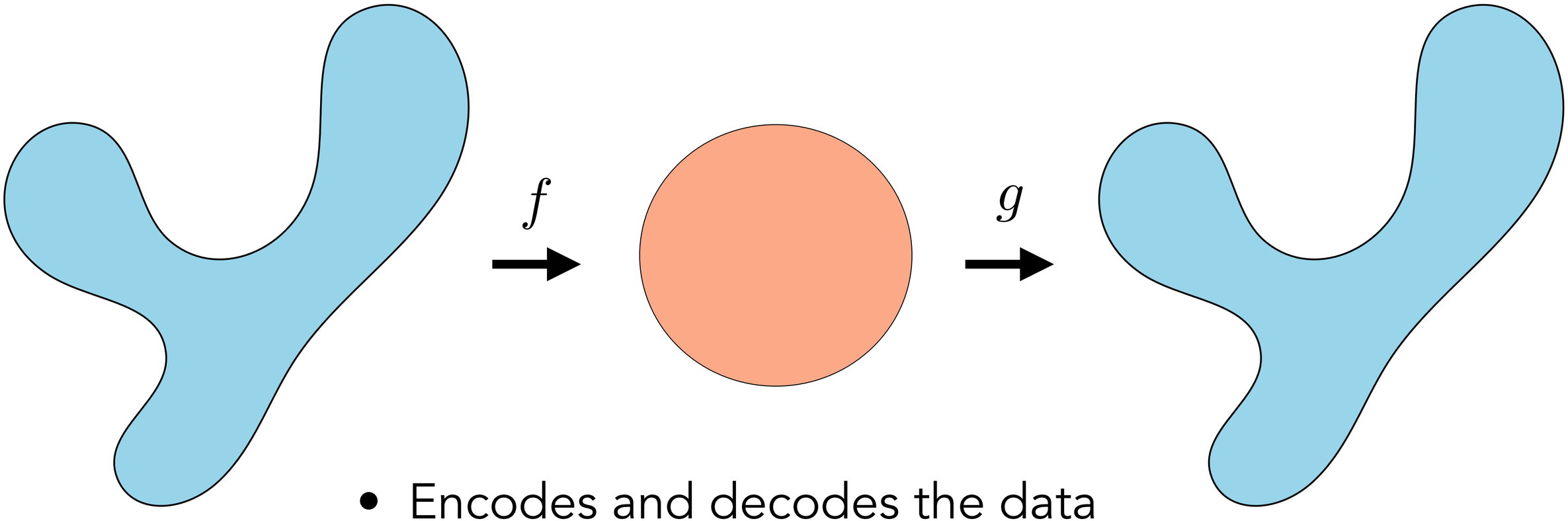
squash encodings toward origin

Autoencoder

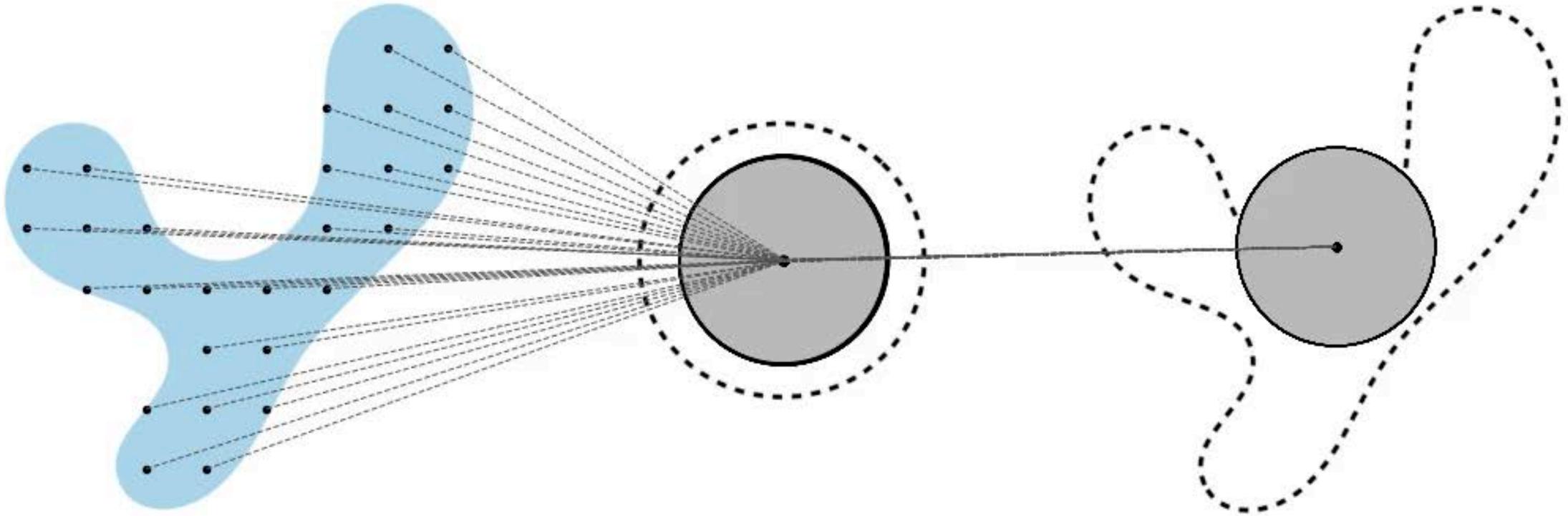


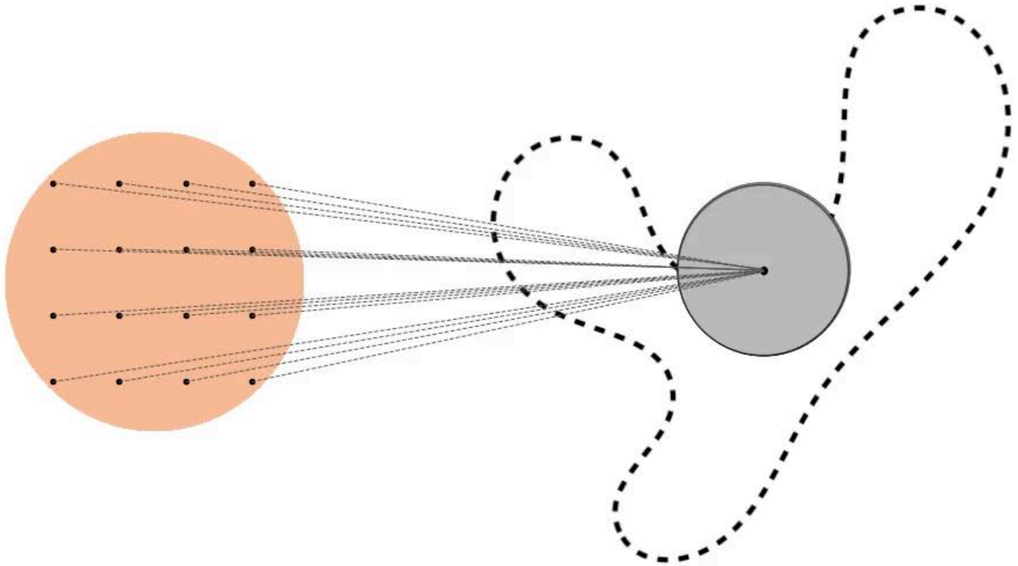
- Encodes and decodes the data
- Low-dimensional bottleneck

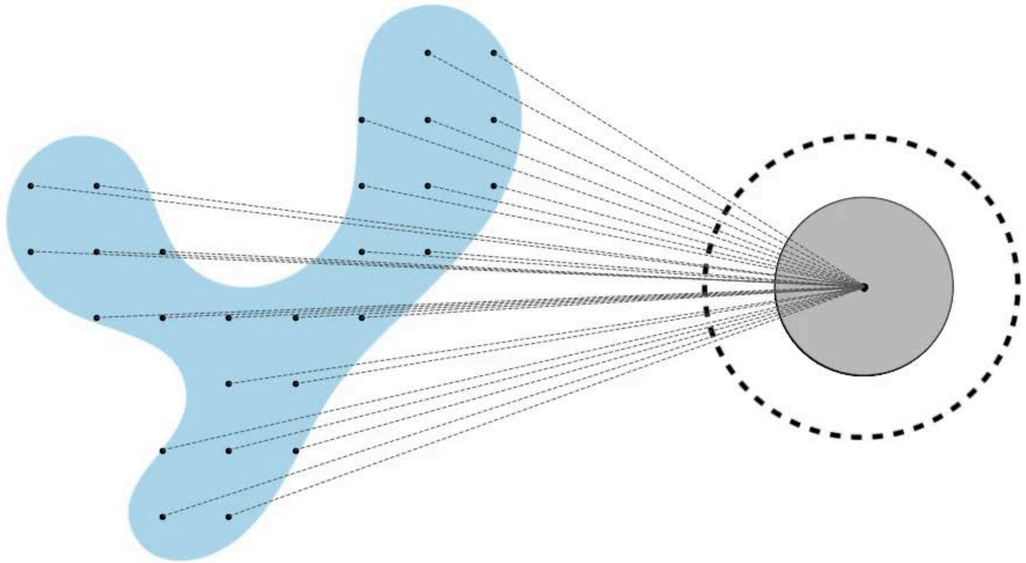
Variational Autoencoder

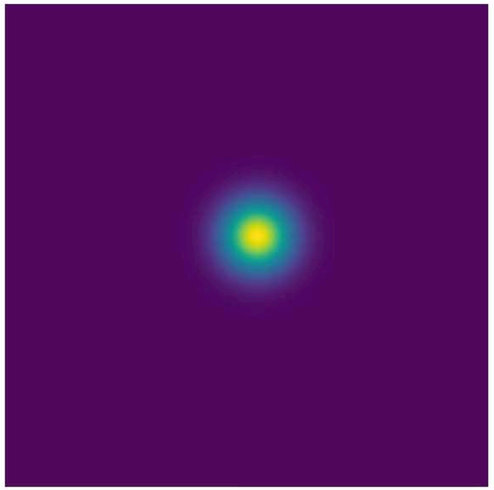


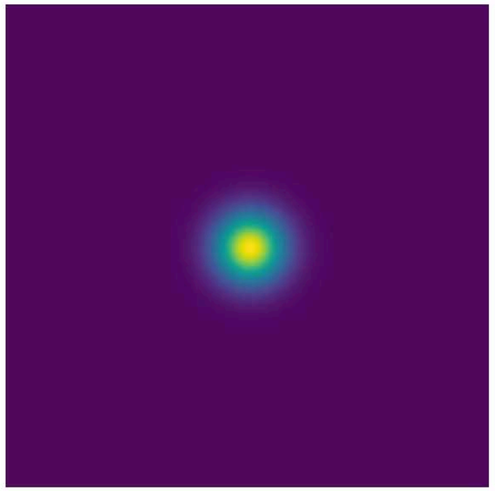
- Encodes and decodes the data
- Low-dimensional bottleneck
- Gaussian bottleneck (**can sample**; **disentangled**)











VAE — three tricks

- Trick #1: approximate an infinite mixture with samples
- Trick #2: sample efficiently via importance sampling
- Trick #3: predict the optimal sampling distribution for each datapoint

Training a VAE — Step by step

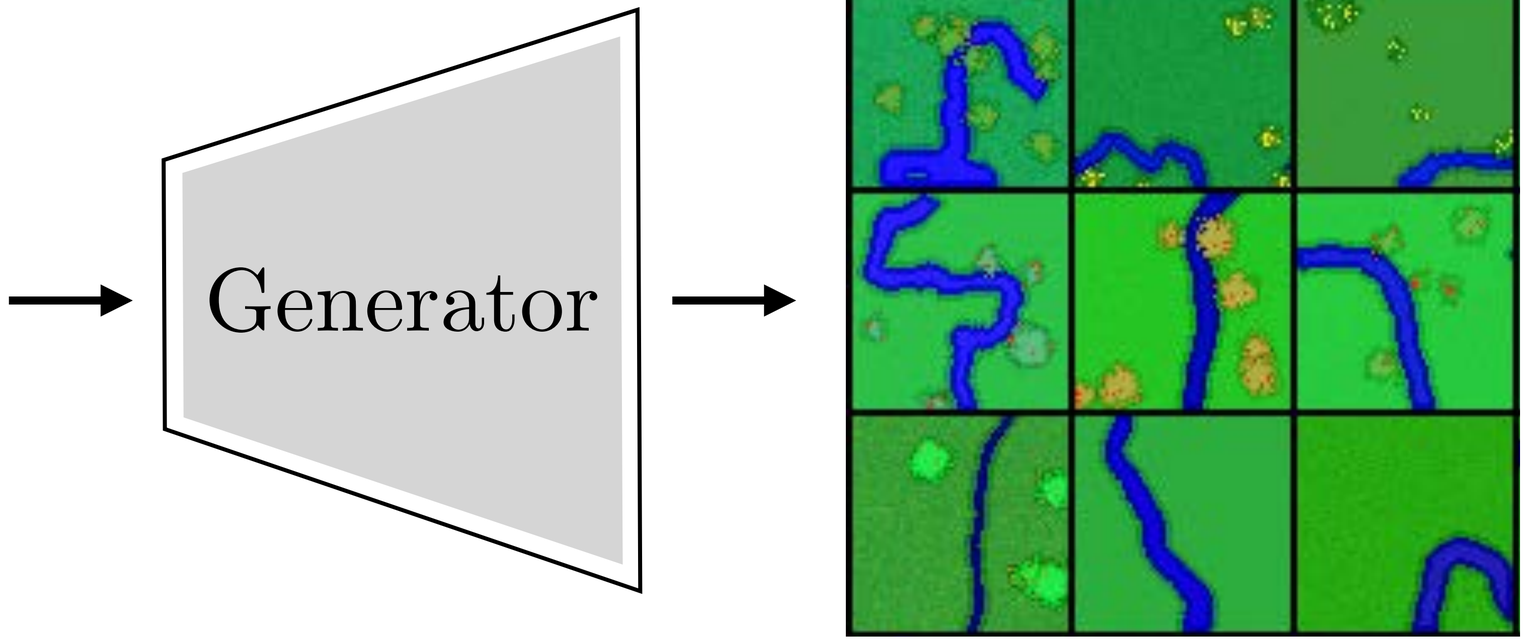
- Sample one or more $\mathbf{x} \sim \{\mathbf{x}^{(i)}\}_{i=1}^{N}$
- *Encode* the data with a forward pass through f_ψ
- For each datapoint, create one or more noisy latent codes using the distribution parameterized by the encoder
- *Decode* the data by passing the noisy latent codes through the g_θ
- Compute the losses and backprop to update θ and ψ

$\mathbf{z}_1 \sim \texttt{Bernoulli}(0.5)$ (River turns)

$\mathbf{z}_2 \sim \texttt{Normal}(\mu_1, \mathbf{\Sigma}_1)$ (Grass color)

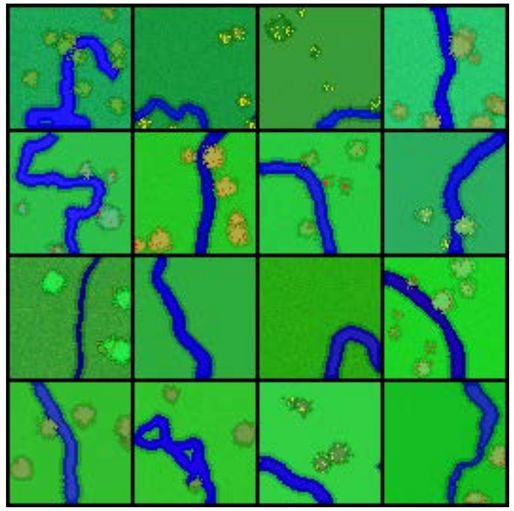
$\mathbf{z}_3 \sim \texttt{Unif}(0, 10)$ (Number trees)

$\vdots$



Training

Data

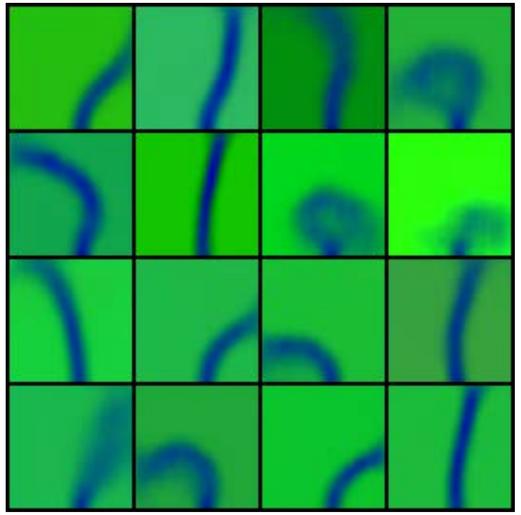


$\rightarrow$ Learner $\rightarrow f_\psi\ ,\ g_\theta$

Sampling

$\mathbf{z} \rightarrow g_\theta \rightarrow$

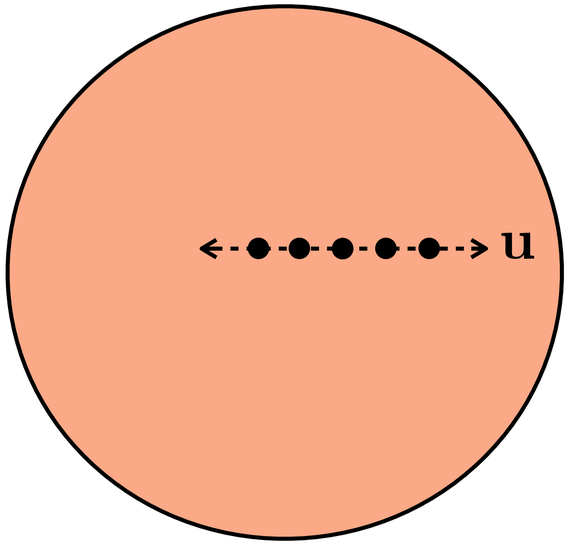
Samples

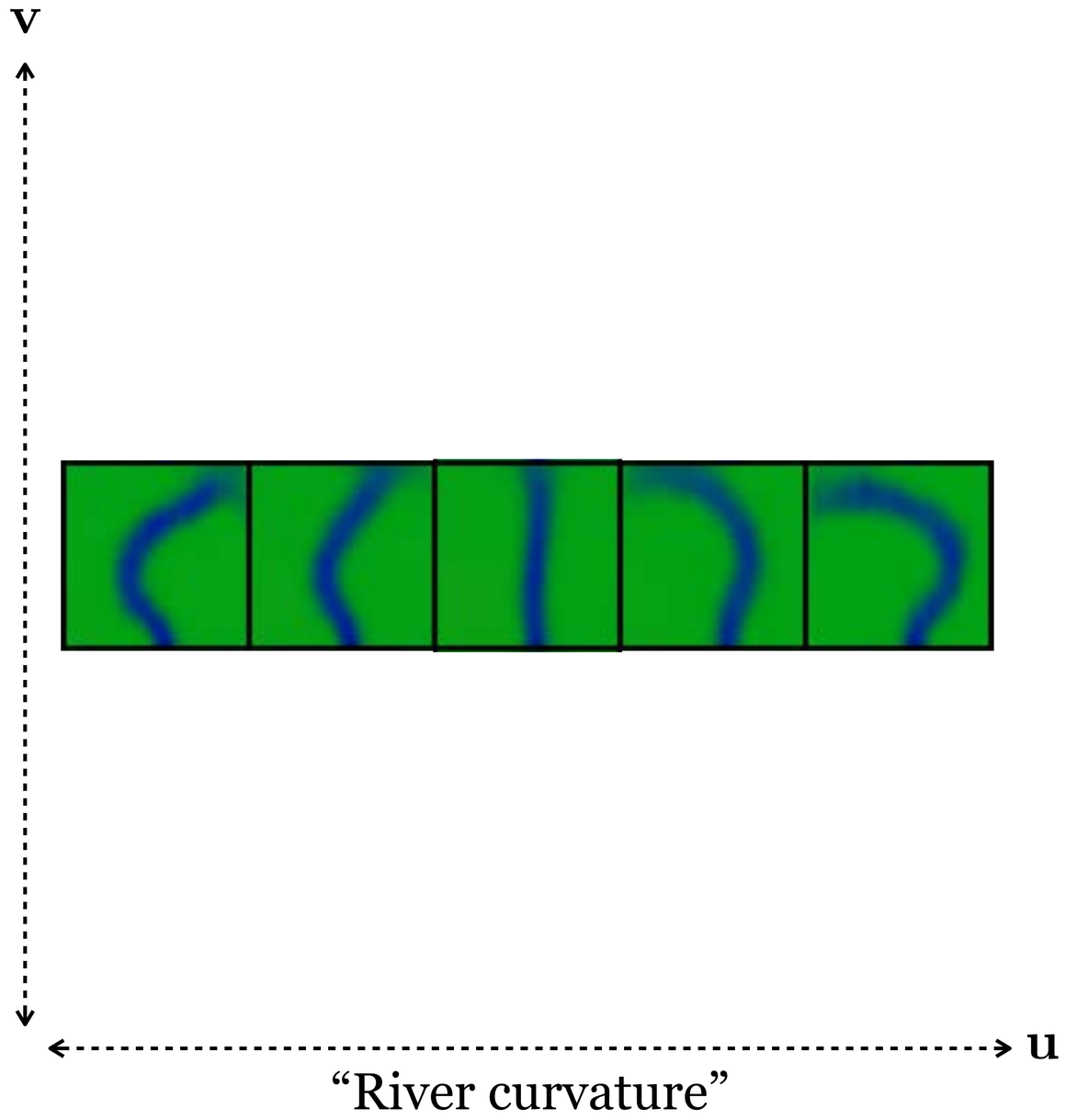


Synthesized Images

Latent variables

(controls)

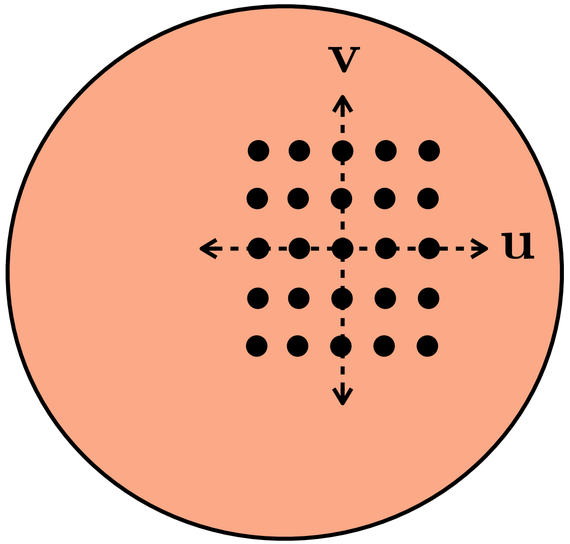


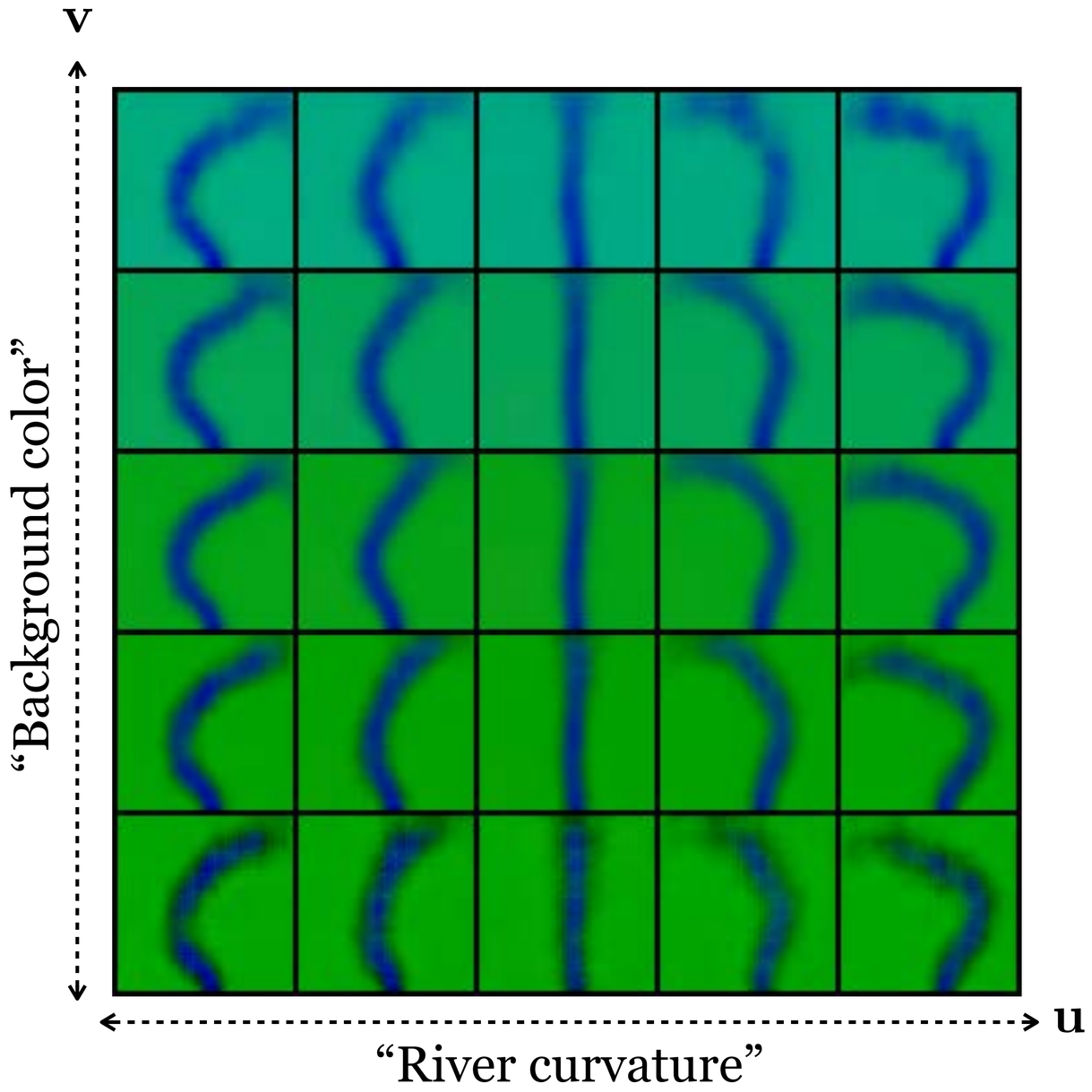


Synthesized Images

Latent variables

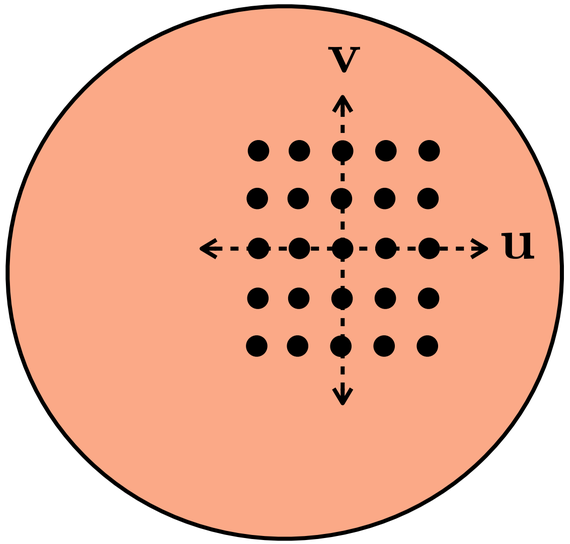
(controls)



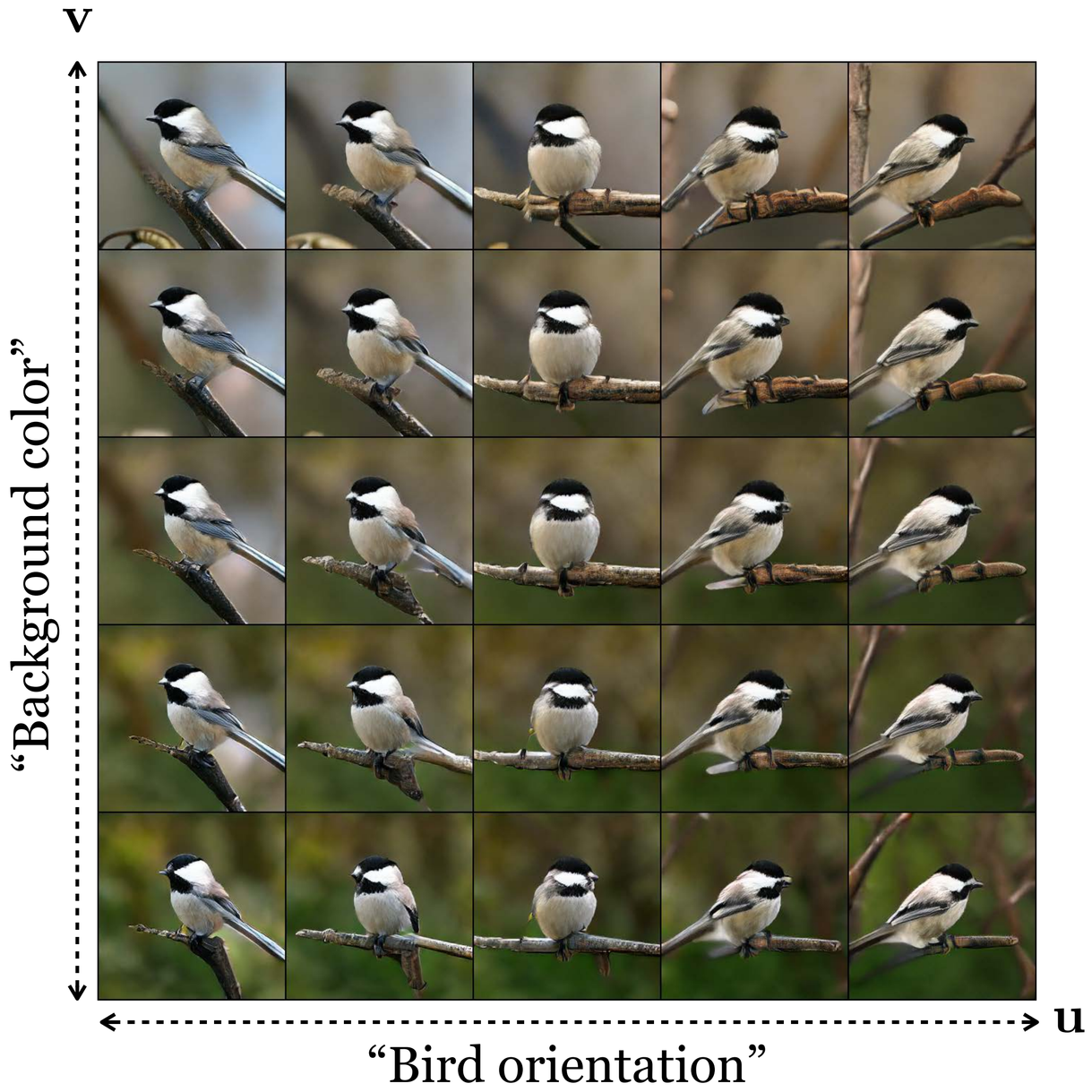


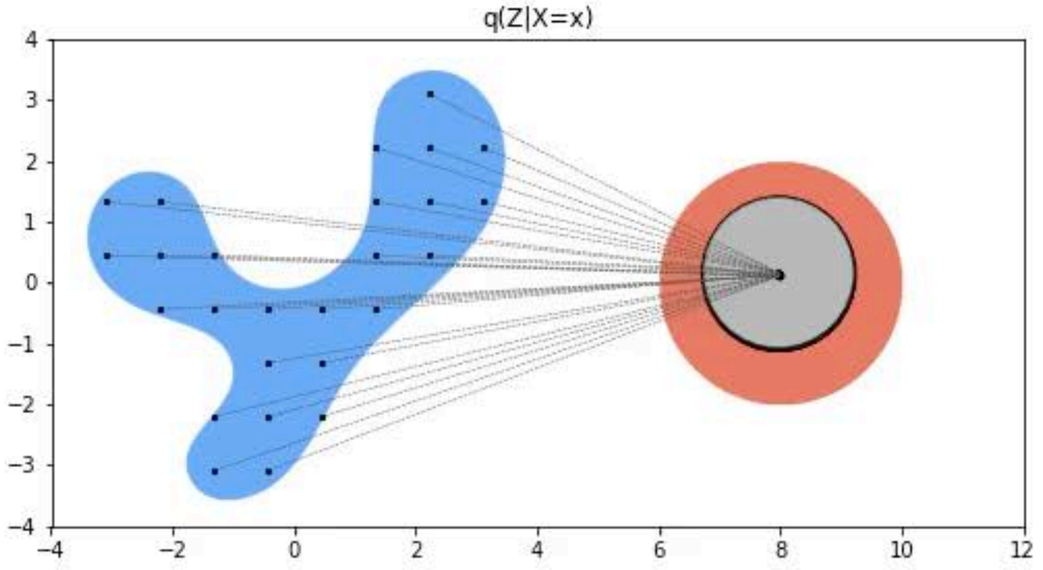
BigGAN [Brock et al. 2018]

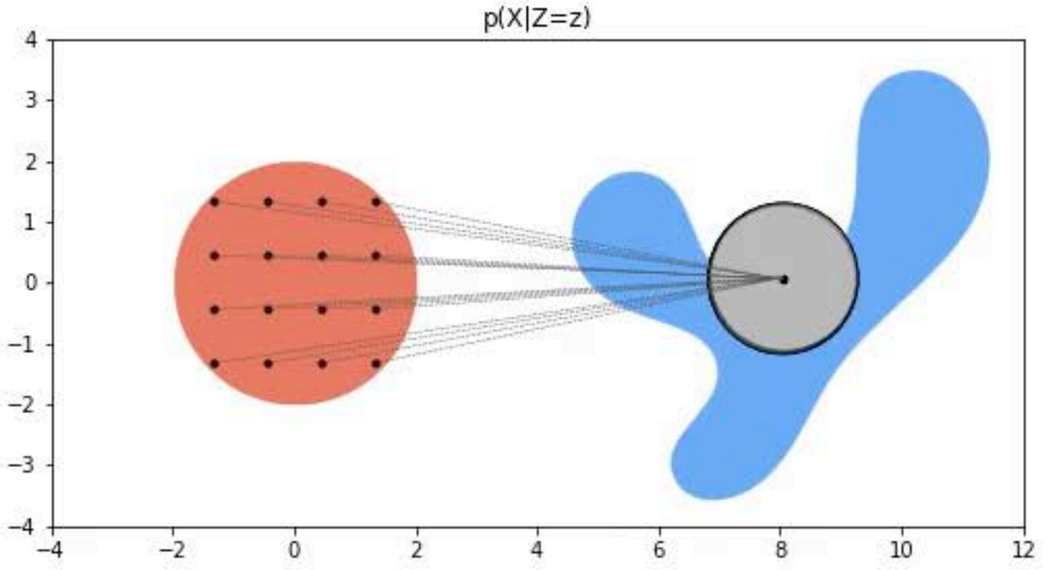
Latent variables
(controls)

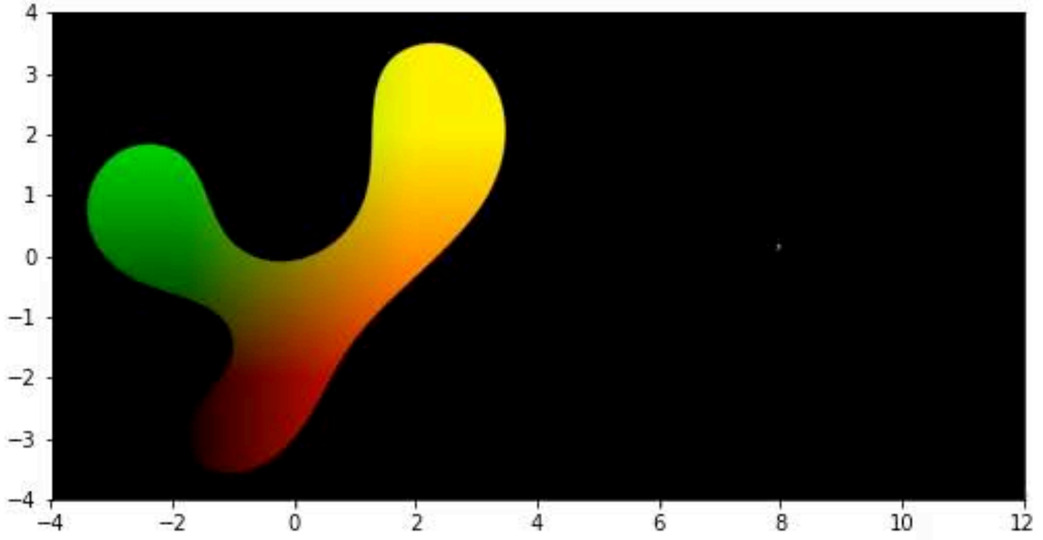


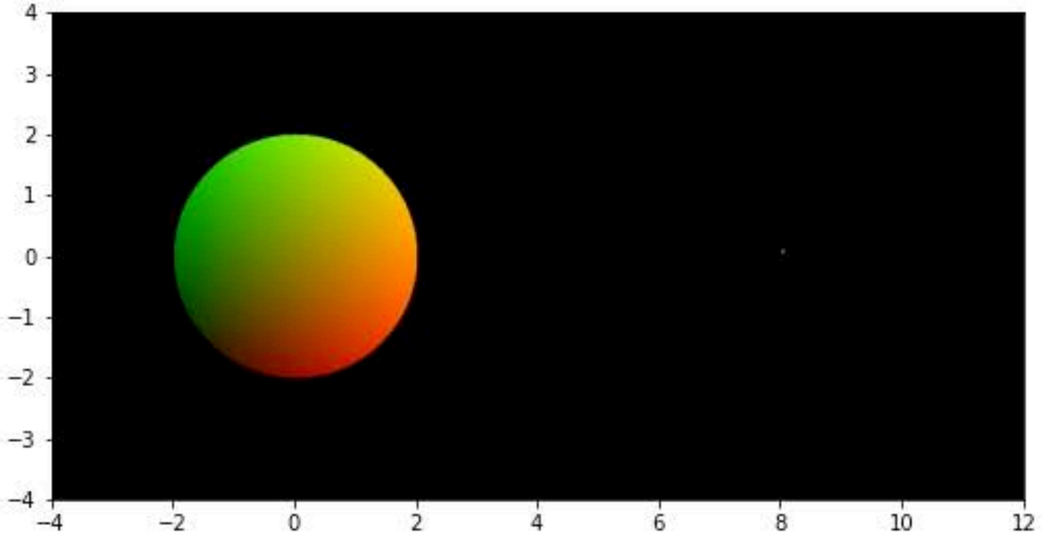
Synthesized Images




q(Z|X=x)


p(X|Z=z)





Comparing popular model types

Method	Latents?	Density/Energy?	Generator?
Energy-based models	✗	✓(energy only)	✗
Gaussian	✗	✓	✗
Autoregressive models	✗	✓	✓(slow)
Diffusion models	✓(high-dimensional)	✗	✓(slow)
GANs	✓	✗	✓
VAEs	✓	✗	✓

* rough categorization of vanilla versions of each model