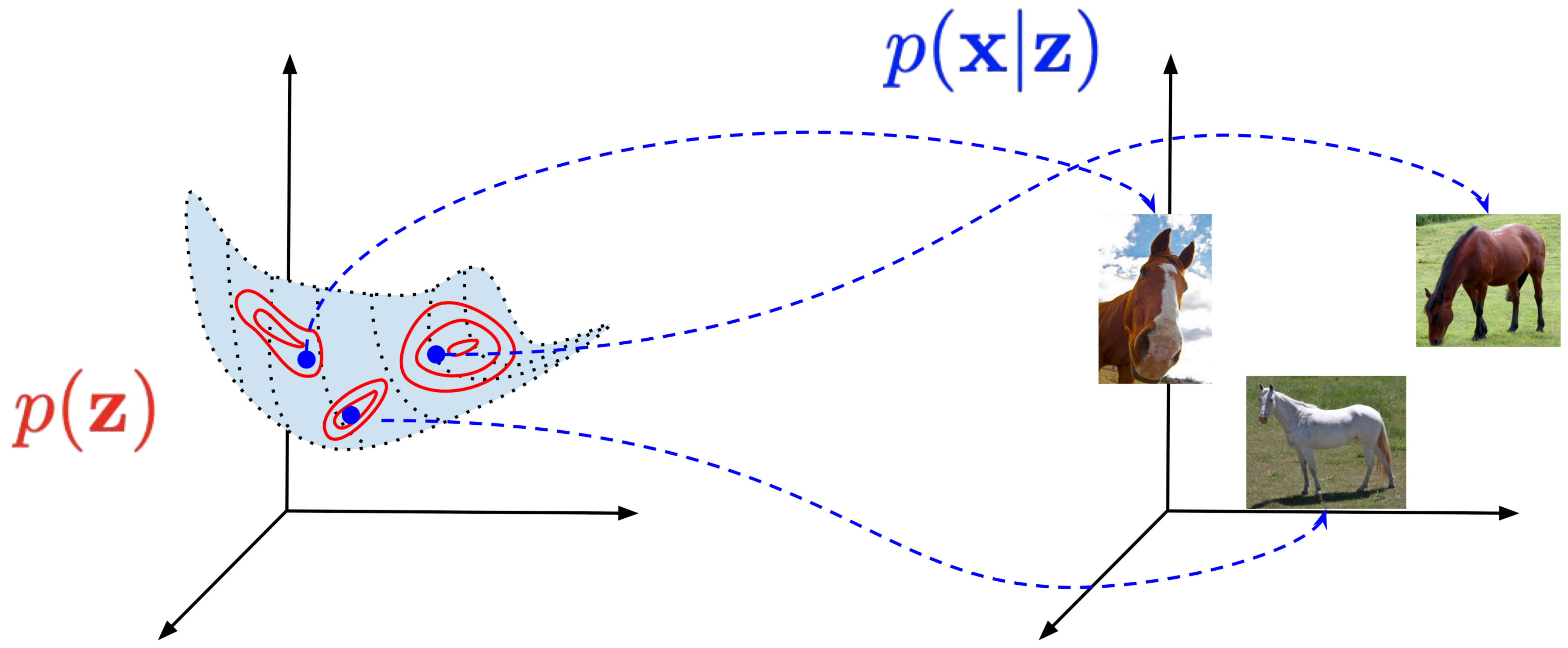
Analysing Adversarial Robustness of VAE and Denoising Abilities of DGM

Anna Kuzina
Vrije Universiteit Amsterdam

Latent Variable Models

Observing a finite sample $x_1, \ldots, x_N$ we are interested in the underlying distribution $p(x)$



Latent Variable Models

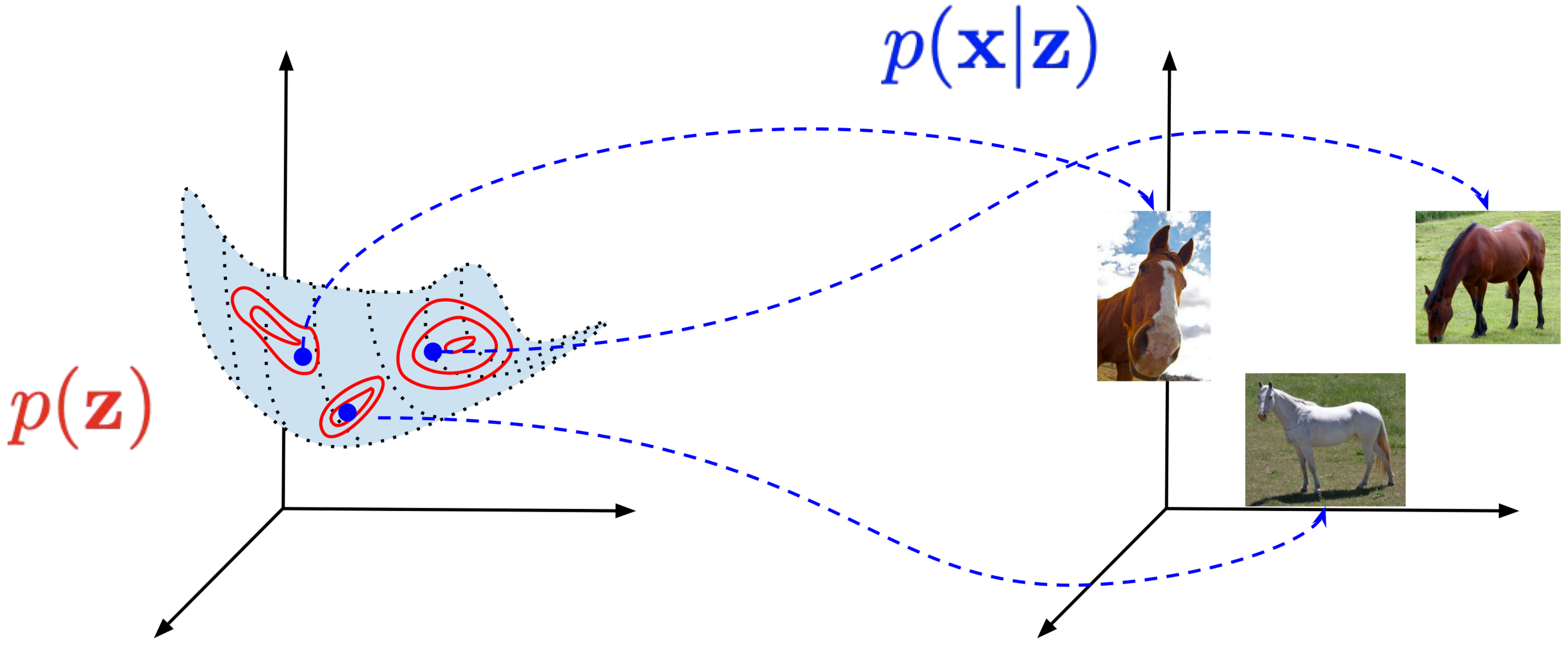
Observing a finite sample $x_1, \ldots, x_N$ we are interested in the underlying distribution $p(x)$

Assume the model

$$p_\theta(x) = \int p_\theta(x \mid z) p_\theta(z) dz$$

aka generative process:

$$z \sim p_\theta(z)$$
$$x \sim p_\theta(x \mid z)$$



Latent Variable Models

Observing a finite sample $x_1, \ldots, x_N$ we are interested in the underlying distribution $p(x)$

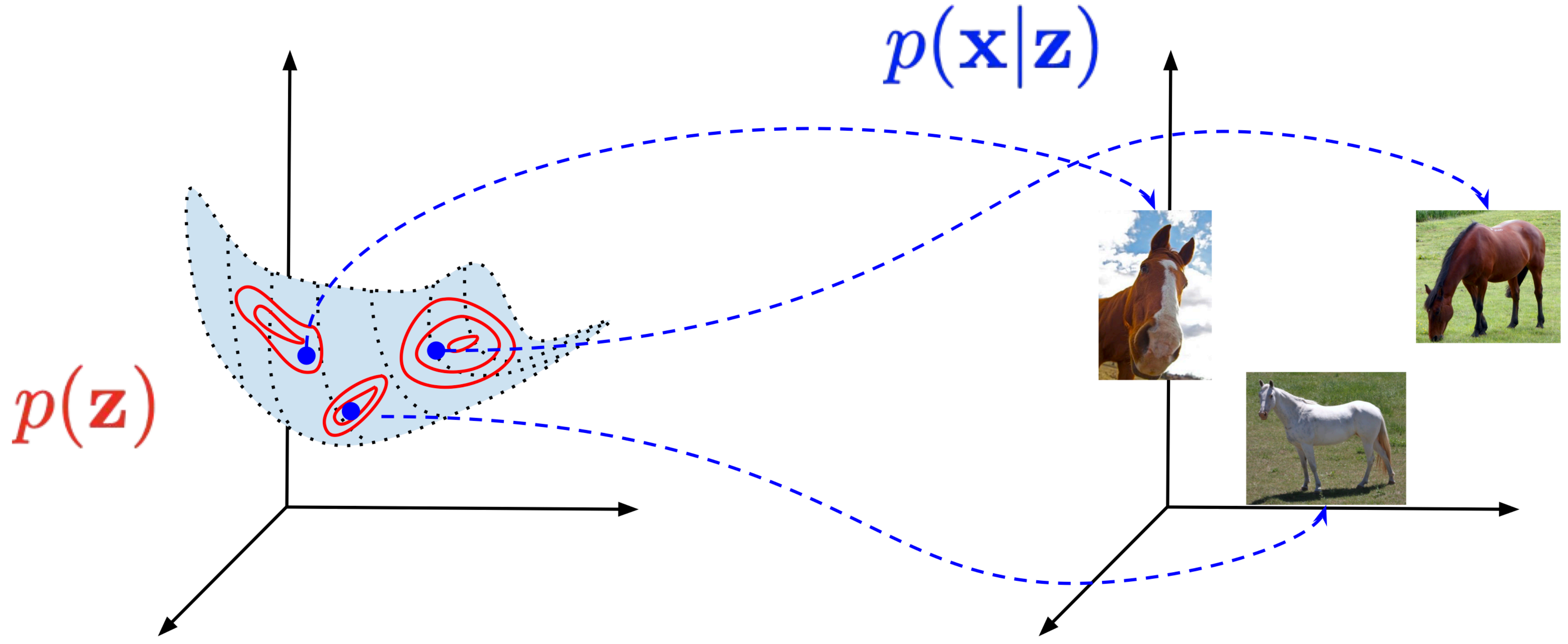
Assume the model

$$p_\theta(x) = \int p_\theta(x \mid z) p_\theta(z) dz$$

Unknown parameters

aka generative process:

$$z \sim p_\theta(z)$$
$$x \sim p_\theta(x \mid z)$$



Latent Variable Models

Observing a finite sample $x_1, \ldots, x_N$ we are interested in the underlying distribution $p(x)$

Assume the model

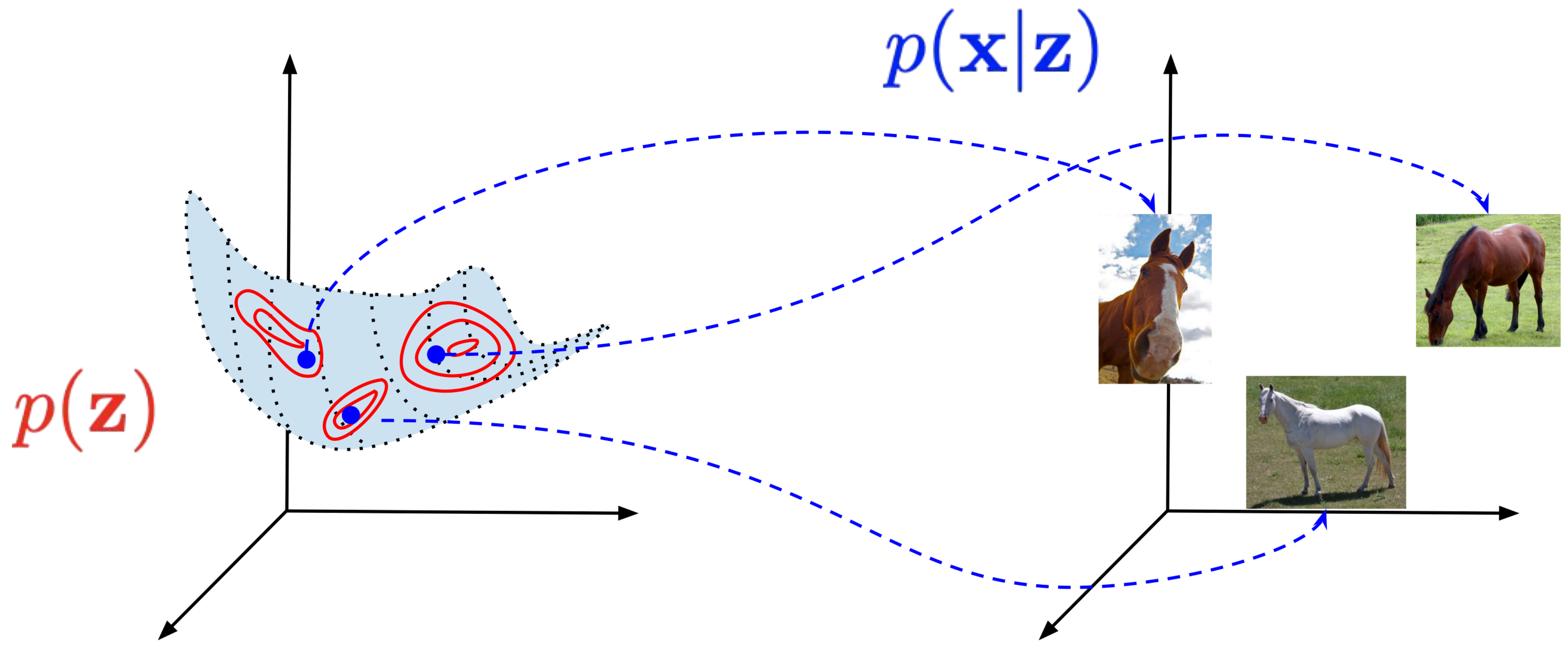
$$p_\theta(x) = \int p_\theta(x \mid z) p_\theta(z) dz$$

aka generative process:

$$z \sim p_\theta(z)$$
$$x \sim p_\theta(x \mid z)$$

MLE objective:

$$\max_\theta \sum_n \ln p_\theta(x_n)$$



Latent Variable Models

Observing a finite sample $x_1, \ldots, x_N$ we are interested in the underlying distribution $p(x)$

Assume the model

$$p_\theta(x) = \int p_\theta(x \mid z) p_\theta(z) dz$$

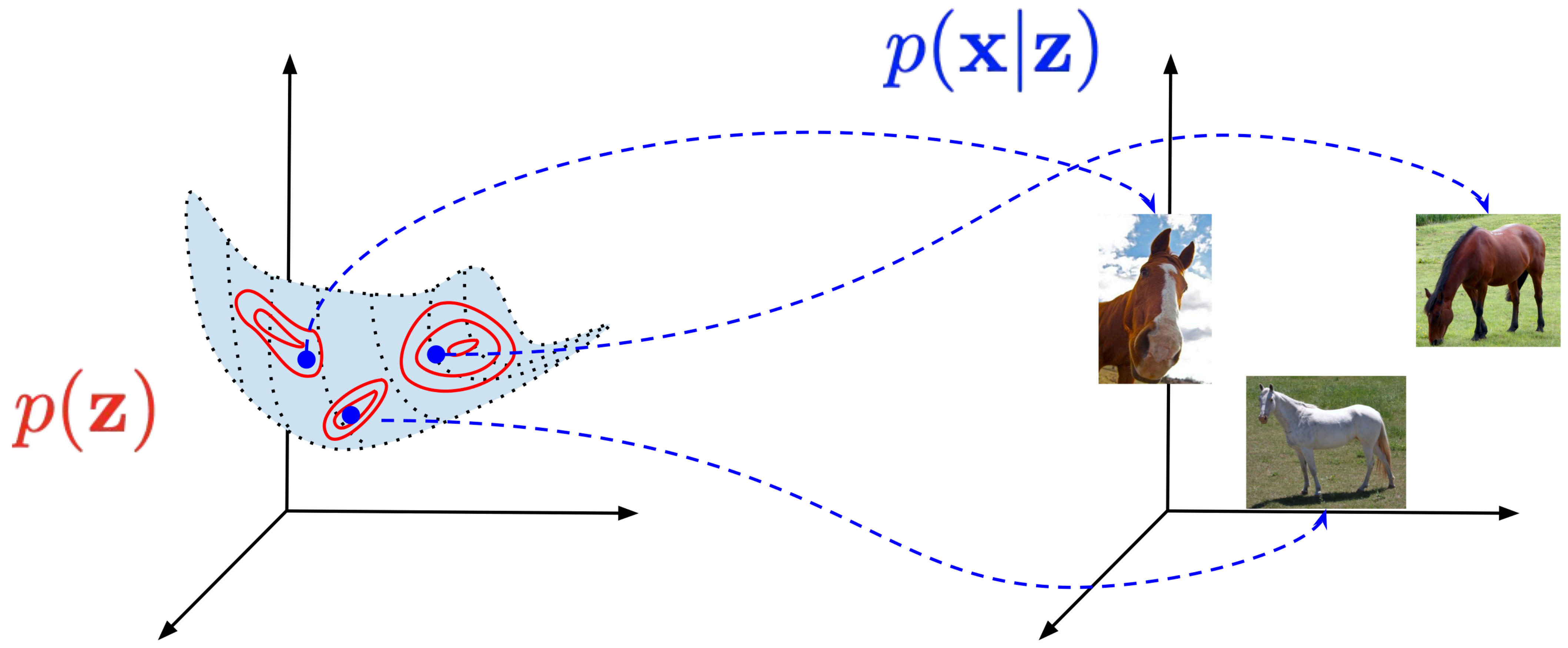
aka generative process:

$$z \sim p_\theta(z)$$
$$x \sim p_\theta(x \mid z)$$

MLE objective:

$$\max_\theta \sum_n \ln p_\theta(x_n)$$

intractable



Variational Autoencoder

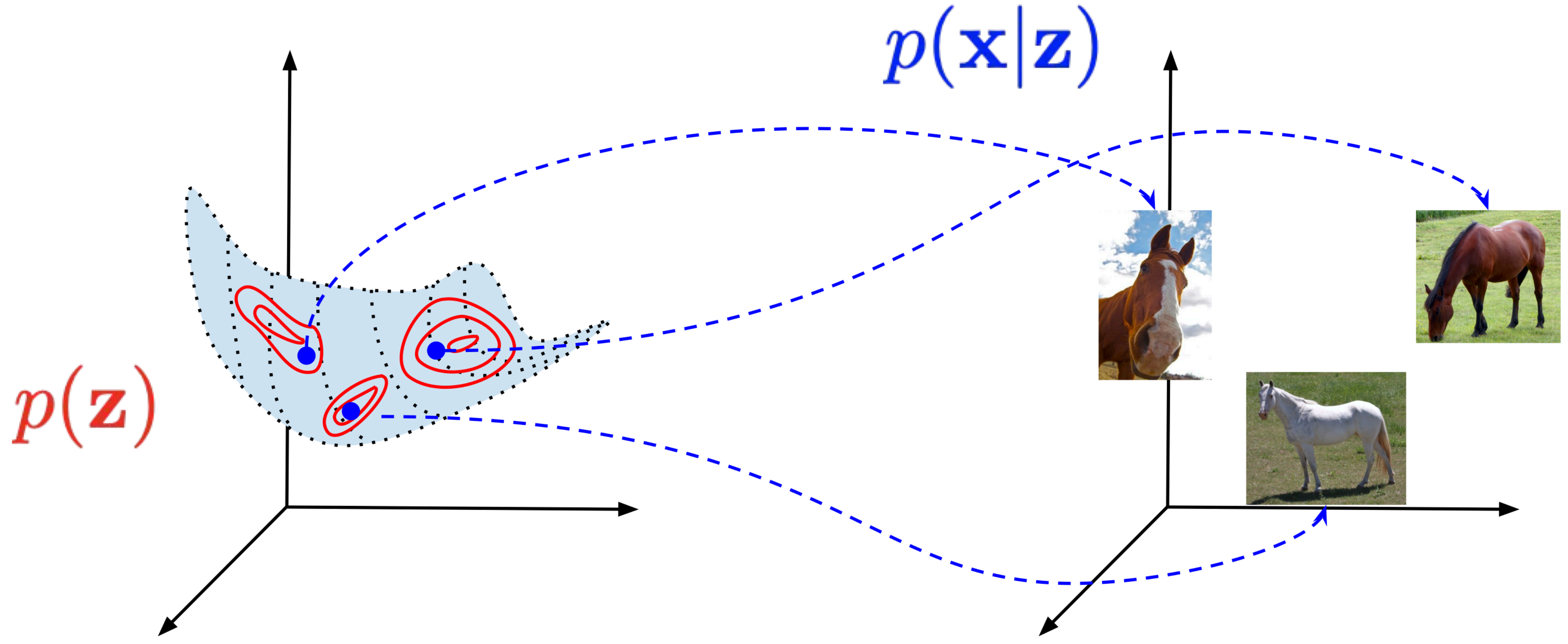
Generative Model:

$$p_\theta(x \mid z)p_\theta(z)$$

Let us define one more model:

$$q_\phi(z \mid x)$$

More
unknown
parameters



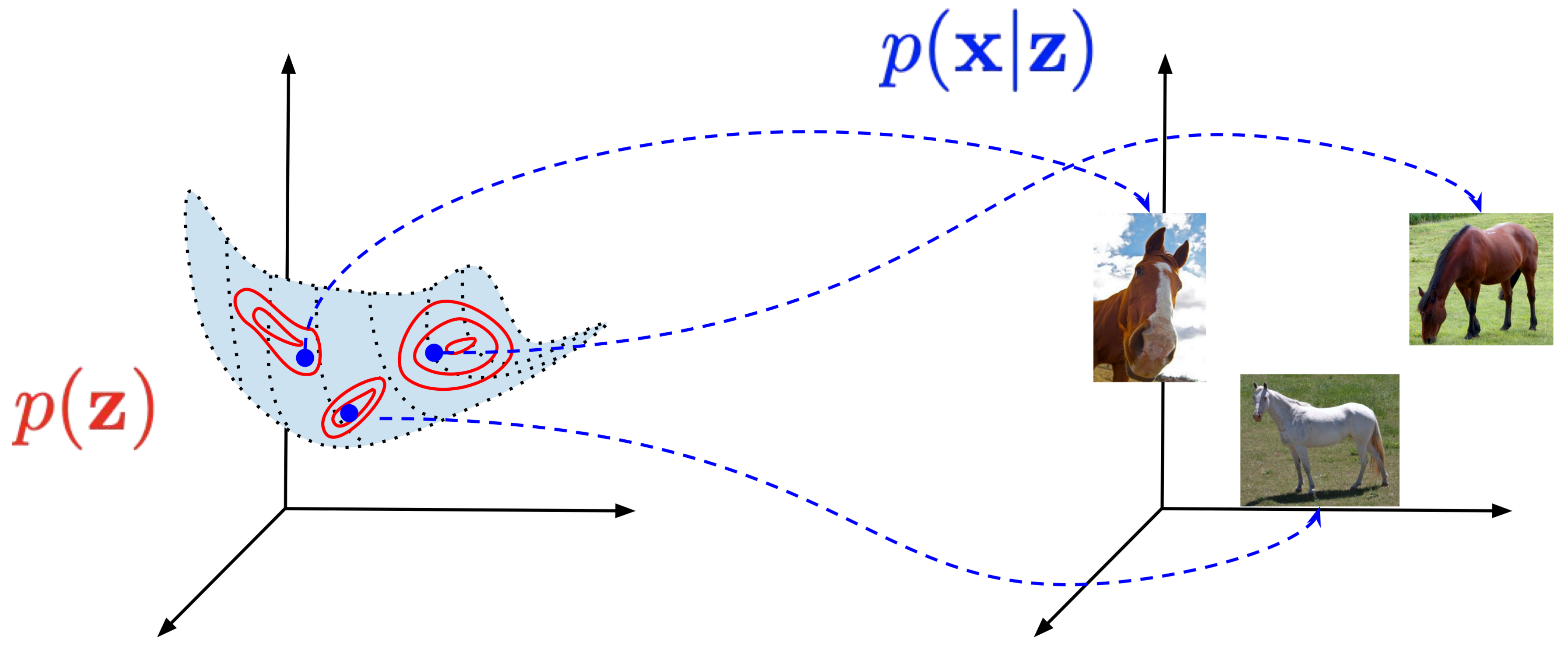
Variational Autoencoder

Generative Model:

$$p_\theta(x \mid z)p_\theta(z)$$

Let us define one more model:

$$q_\phi(z \mid x)$$



Tractable objective:

$$\ln p_\theta(x) \geq \underbrace{\mathbb{E}_{q_\phi(z|x)} \ln p_\theta(x \mid z) - \mathrm{KL}[q_\phi(z \mid x) \| p_\theta(z)]}_{\text{ELBO}}$$

Kingma, D.P., and Welling, M. "Auto-encoding variational bayes." ICLR 2014.

Variational Autoencoder

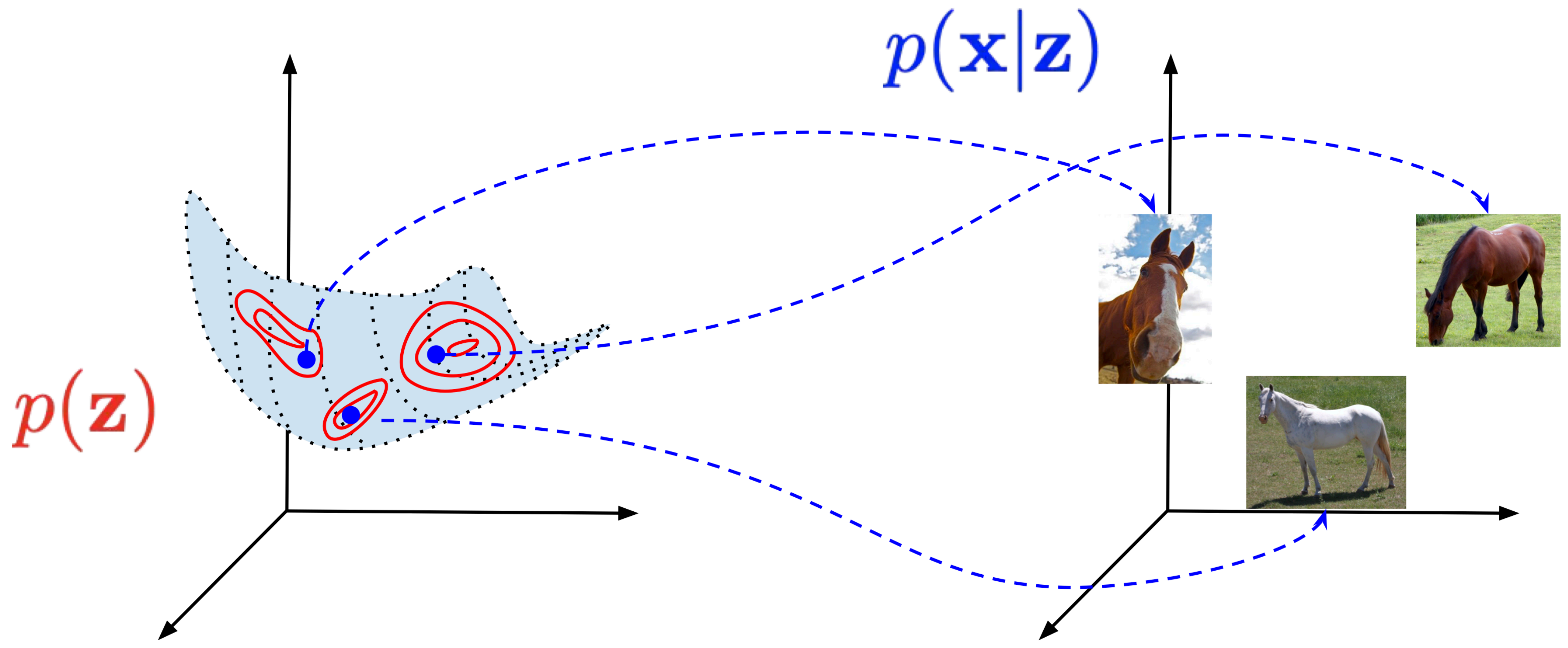
Generative Model:

$$p_\theta(x \mid z)p_\theta(z)$$

Let us define one more model:

$$q_\phi(z \mid x)$$

- Inference Model
- Variational Posterior
- Encoder



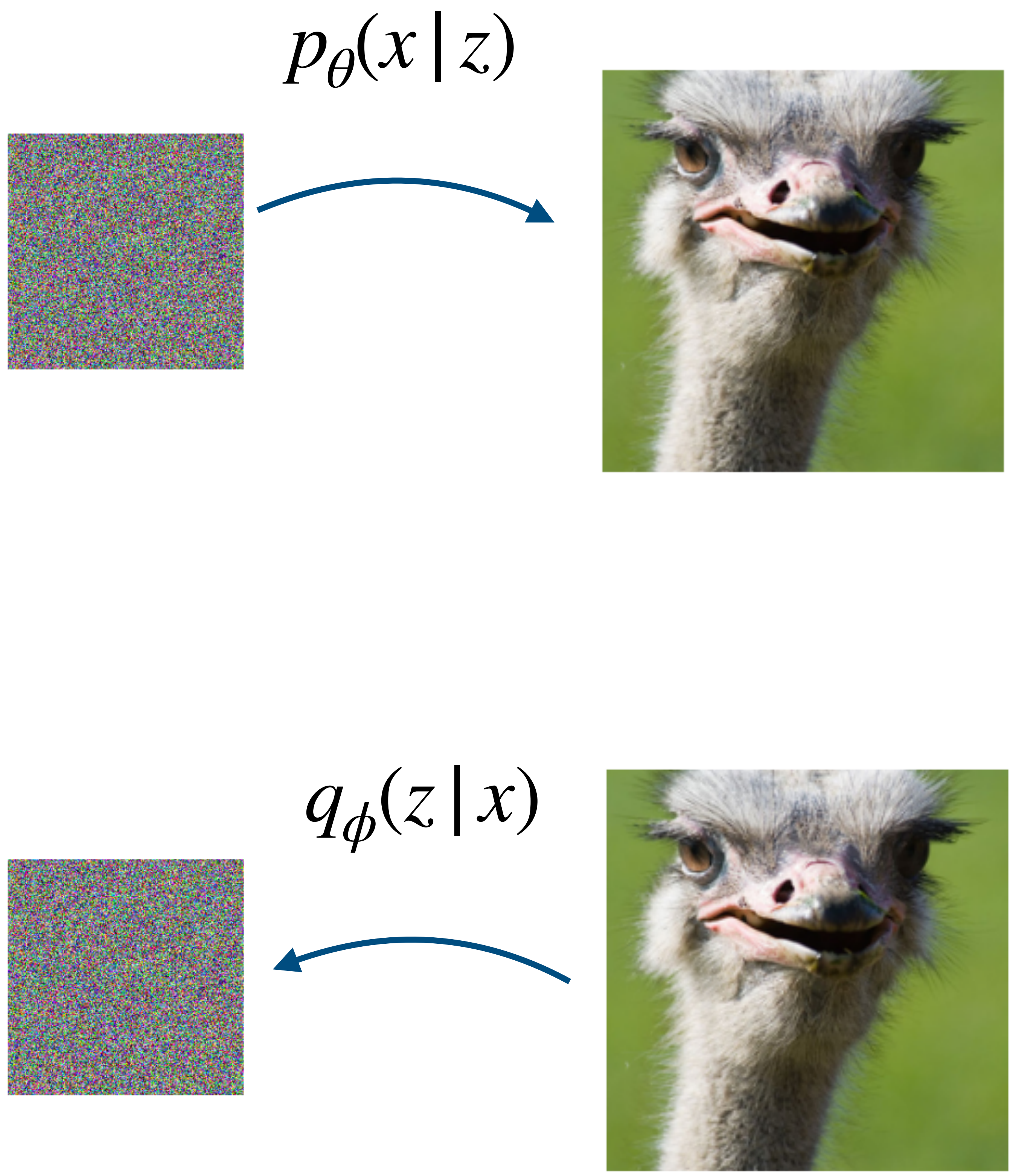
Turns out it can help us to get a tractable objective:

$$\ln p_\theta(x) \geq \underbrace{\mathbb{E}_{q_\phi(z|x)} \ln p_\theta(x \mid z) - \mathrm{KL}[q_\phi(z \mid x) \| p_\theta(z)]}_{\text{ELBO}}$$

Kingma, D.P., and Welling, M. "Auto-encoding variational bayes." ICLR 2014.

Variational Autoencoder

Latent
Space



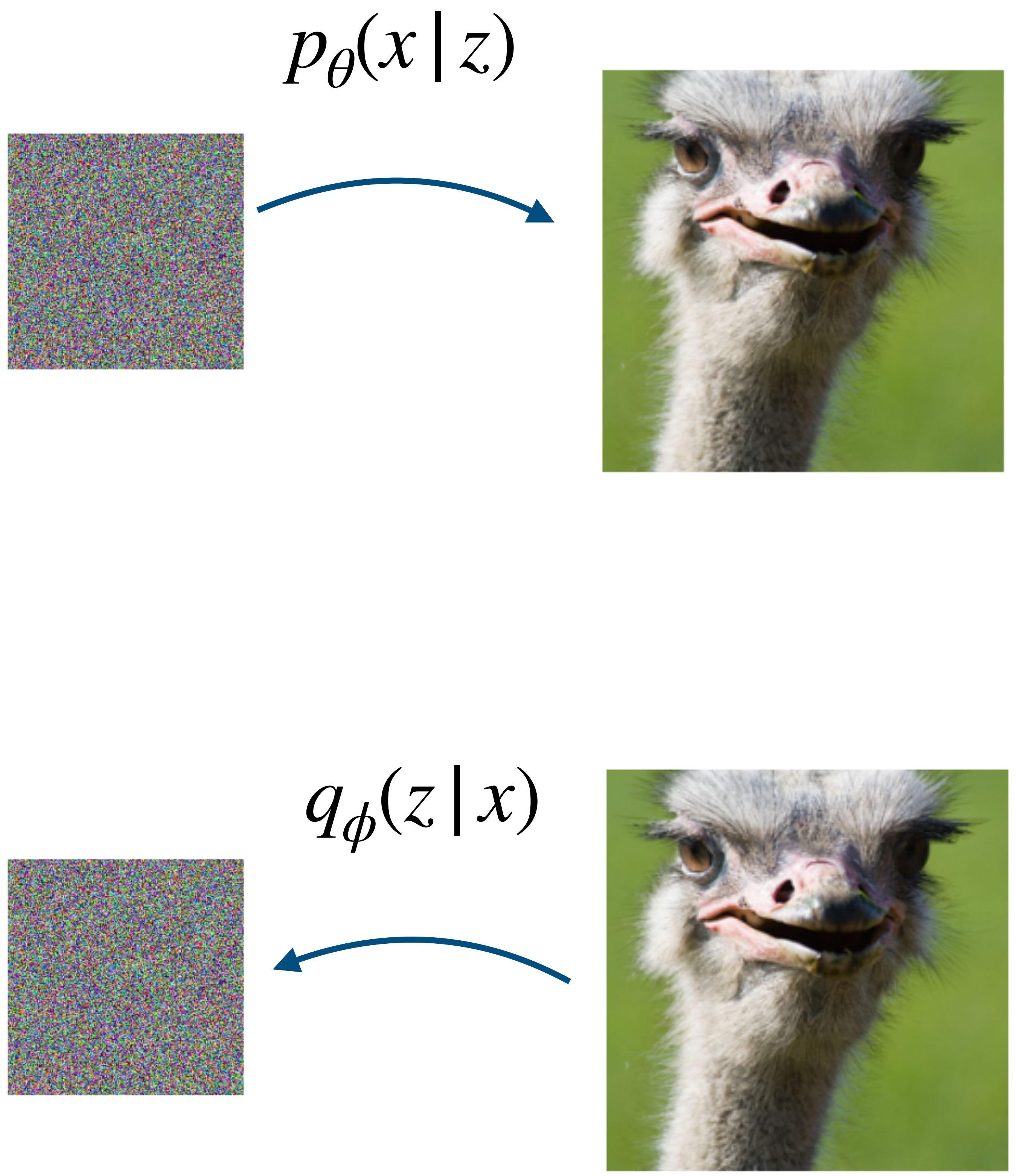
Generative Model
(or decoder)

Inference Model
(or encoder)
(or variational posterior)

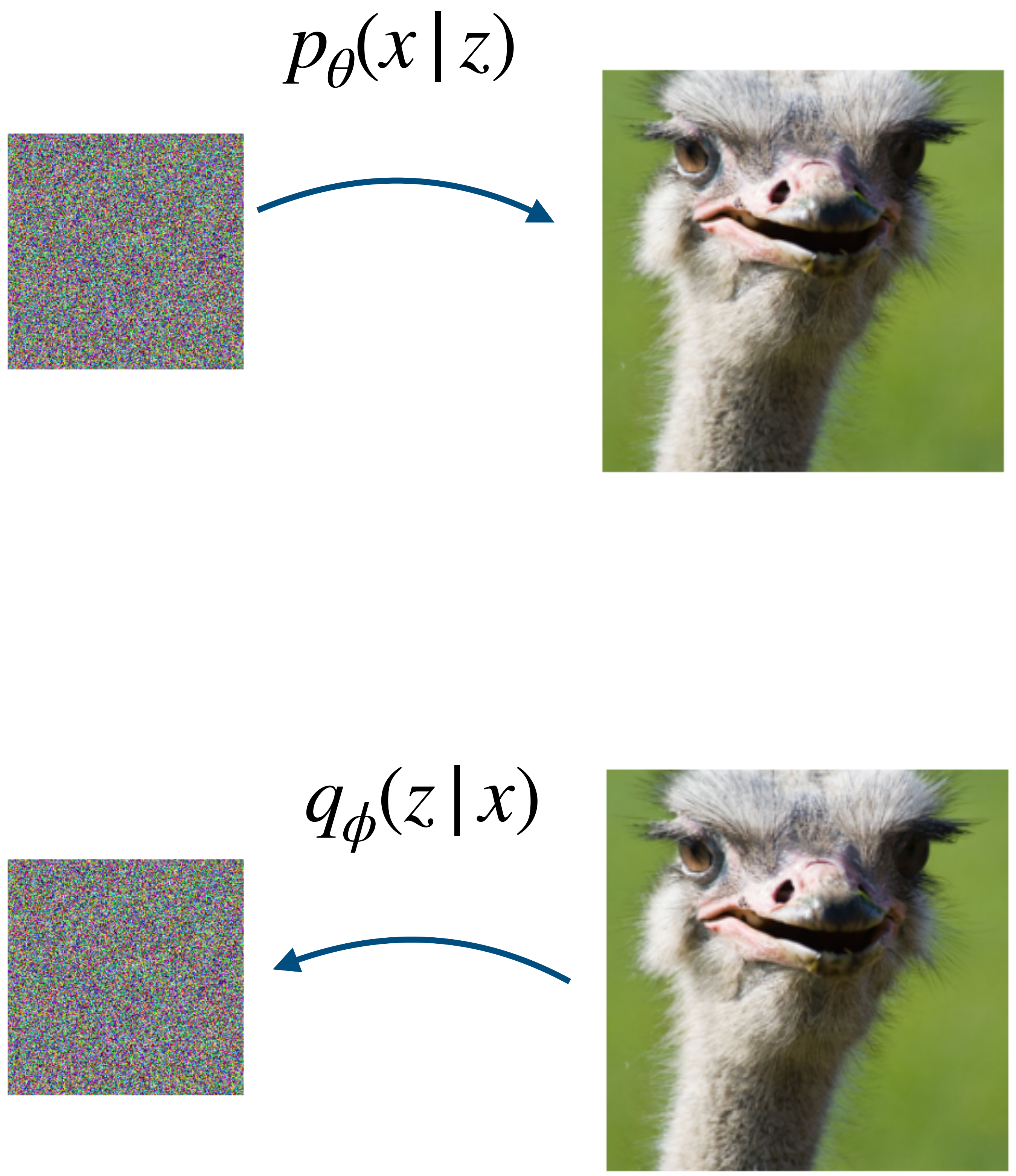
Kingma, D.P., and Welling, M. "Auto-encoding variational bayes." ICLR 2014.

Are VAEs robust to Adversarial Attacks?

Latent
Space



Are VAEs robust to Adversarial Attacks?



Latent Space

No :(

But there is something we can do with that

Adversarial Attack

$$x^a = x^r + \varepsilon, \quad \|\varepsilon\| < \delta$$

x^a "looks" like reference x^r, but is "perceived" differently

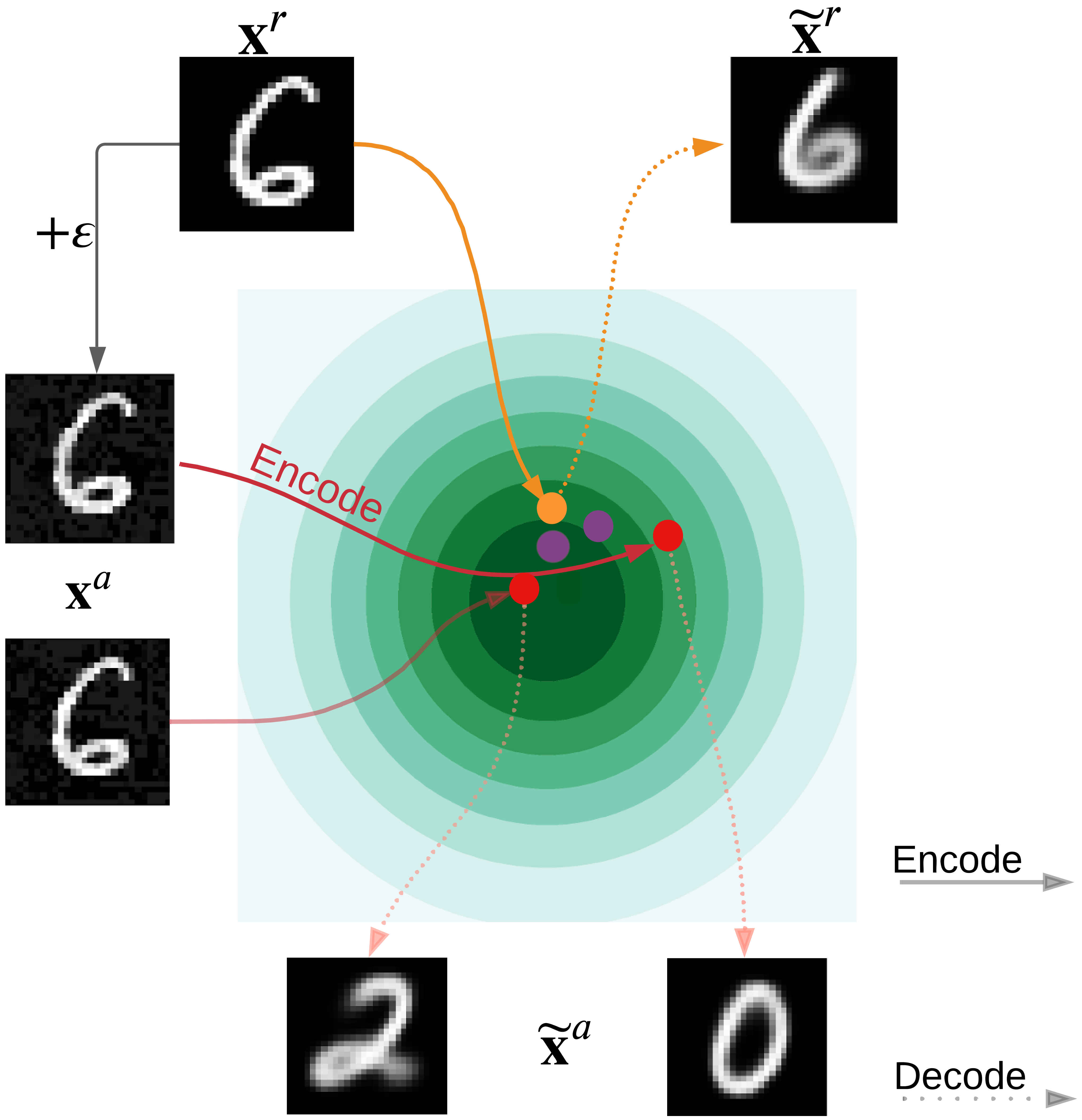
Attacker solves optimisation problem:

$$\varepsilon = \underset{\|\varepsilon\|<\delta}{\arg\max}\ \Delta \left[f(x^r + \varepsilon), f(x^r)\right]$$

Distance metric

What is being attacked

Adversarial Attack on VAEs



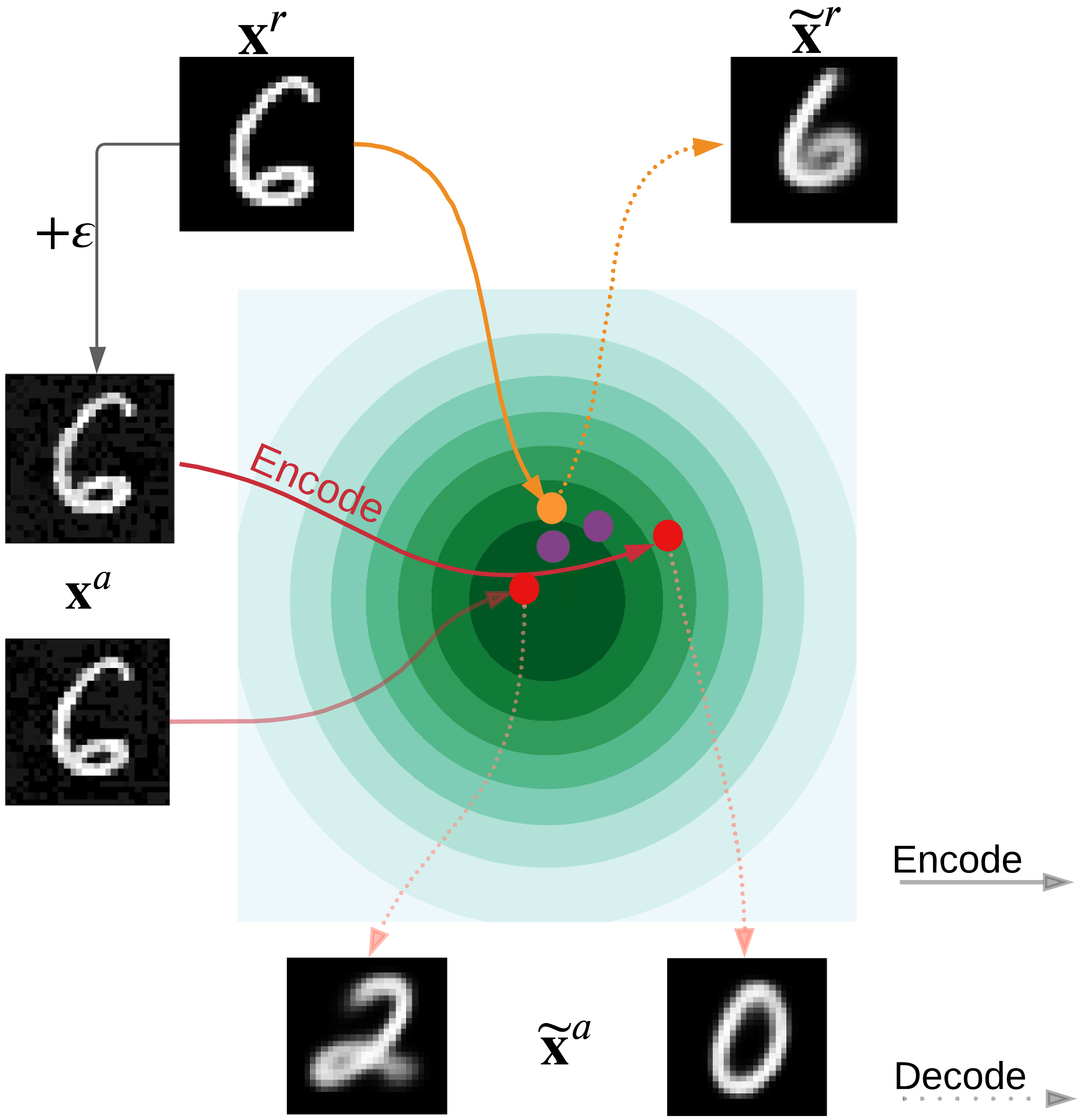
$$x^a = x^r + \varepsilon, \quad \|\varepsilon\| < \delta$$

x^a "looks" like reference x^r, but is "perceived" differently

Attacker solves optimisation problem:

$$\varepsilon = \arg\max_{\|\varepsilon\|<\delta} \mathrm{SKL}\left[q_\phi(z\,|\,x^r + \varepsilon), q_\phi(z\,|\,x^r)\right]$$

Defence Strategy



$z^a \sim q_\phi(z \mid x^a)$ **vs** $z^r \sim q_\phi(z \mid x^r)$

Let's use true posterior instead:

Target density:

$$p_\theta(z \mid x^a) \propto p(z) p_\theta(x^a \mid z)$$

Make T steps of MCMC (starting from z^a)

$$z^{(T)} \sim q^{(T)}(z \mid x^a)$$

Kuzina, A. et al, "Alleviating Adversarial Attacks on Variational Autoencoders with MCMC", NeurIPS 2022

Final Algorithm

1. *(Defender)*

Train a VAE:

$$q_\phi(z \mid x), p(z), p_\theta(x \mid z)$$

2. *(Attacker)*

For a given x^r, construct the attack x^a

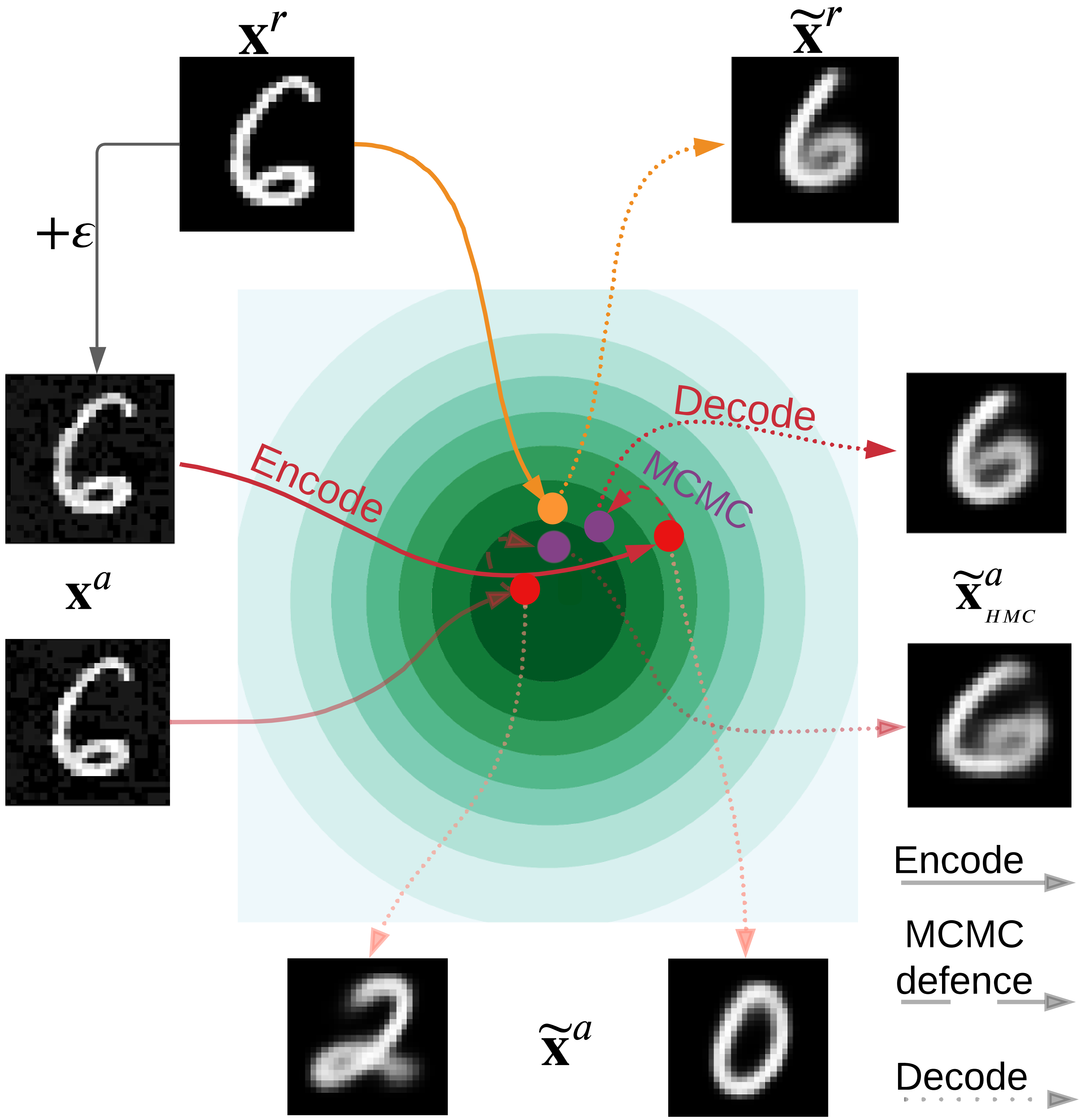
$$x^a = x^r + \varepsilon, \quad \|\varepsilon\| < \delta$$

s.t $q_\phi(z \mid x^a)$ is "far enough" from $q_\phi(z \mid x^r)$

3. *(Defender)*

Run T steps of HMC with the target $\propto p(z)p_\theta(x^a \mid z)$

Use $z := z^{(T)}$ to decode / in downstream task



* Note that q_ϕ and p_θ can be of any form, e.g., hierarchical VAEs

Why it works?

Theoretical justification

t steps of MCMC

"desired" distribution

$$\mathrm{TV}[q^{(t)}(z \mid x^a) \| q_\phi(z \mid x^r)] \leq \sqrt{\frac{1}{2}\mathrm{KL}\left[q^{(t)}(z \mid x^a) \| p_\theta(z \mid x^a)\right]} + \sqrt{\frac{1}{2}\mathrm{KL}\left[q_\phi(z \mid x^r) \| p_\theta(z \mid x^r)\right]} + o(\sqrt{\|\varepsilon\|})$$

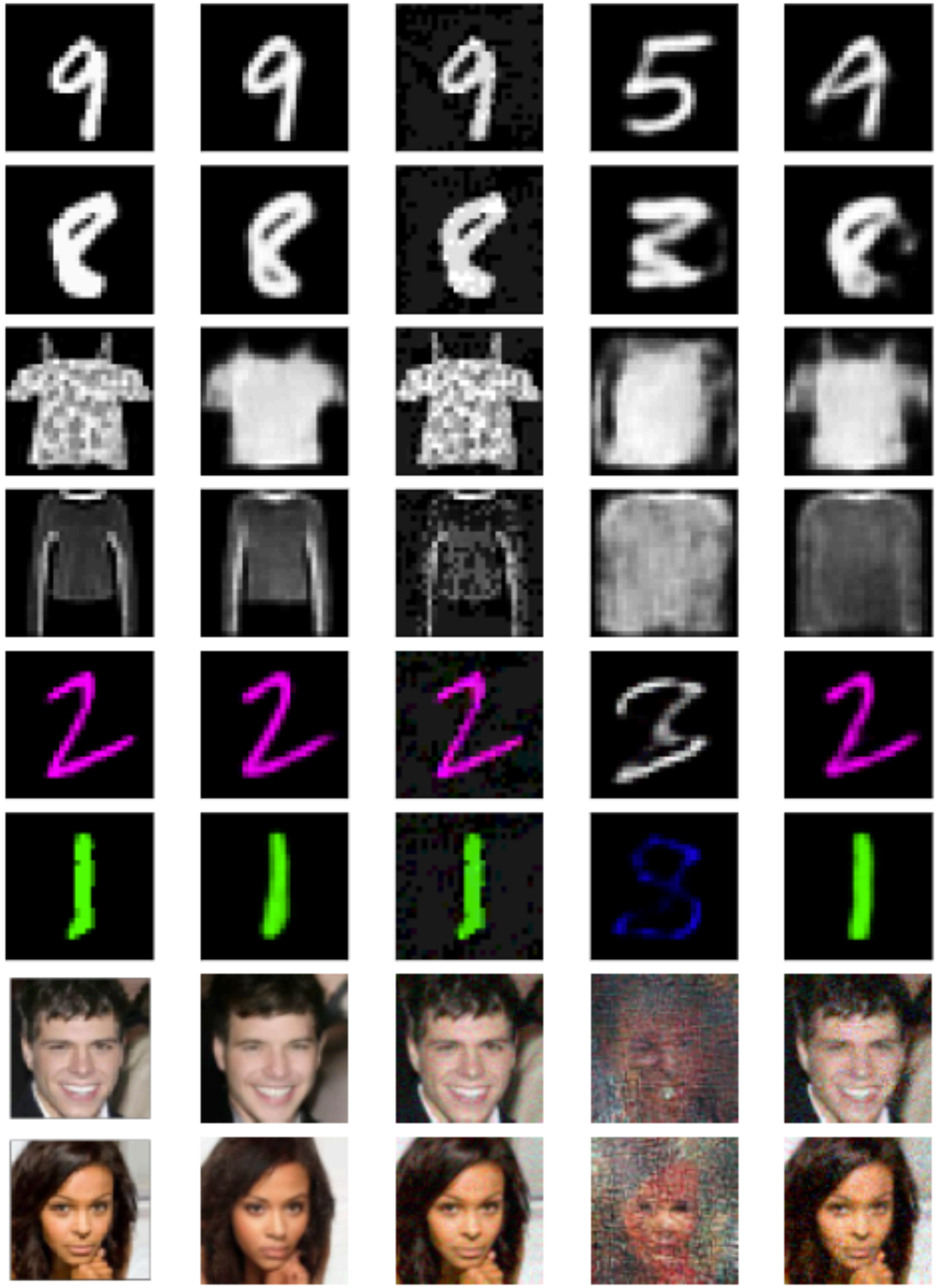
How good is defence

Goes to 0 with $t \to \inf$

How good VAE is (*approximation gap*)

Attack radius

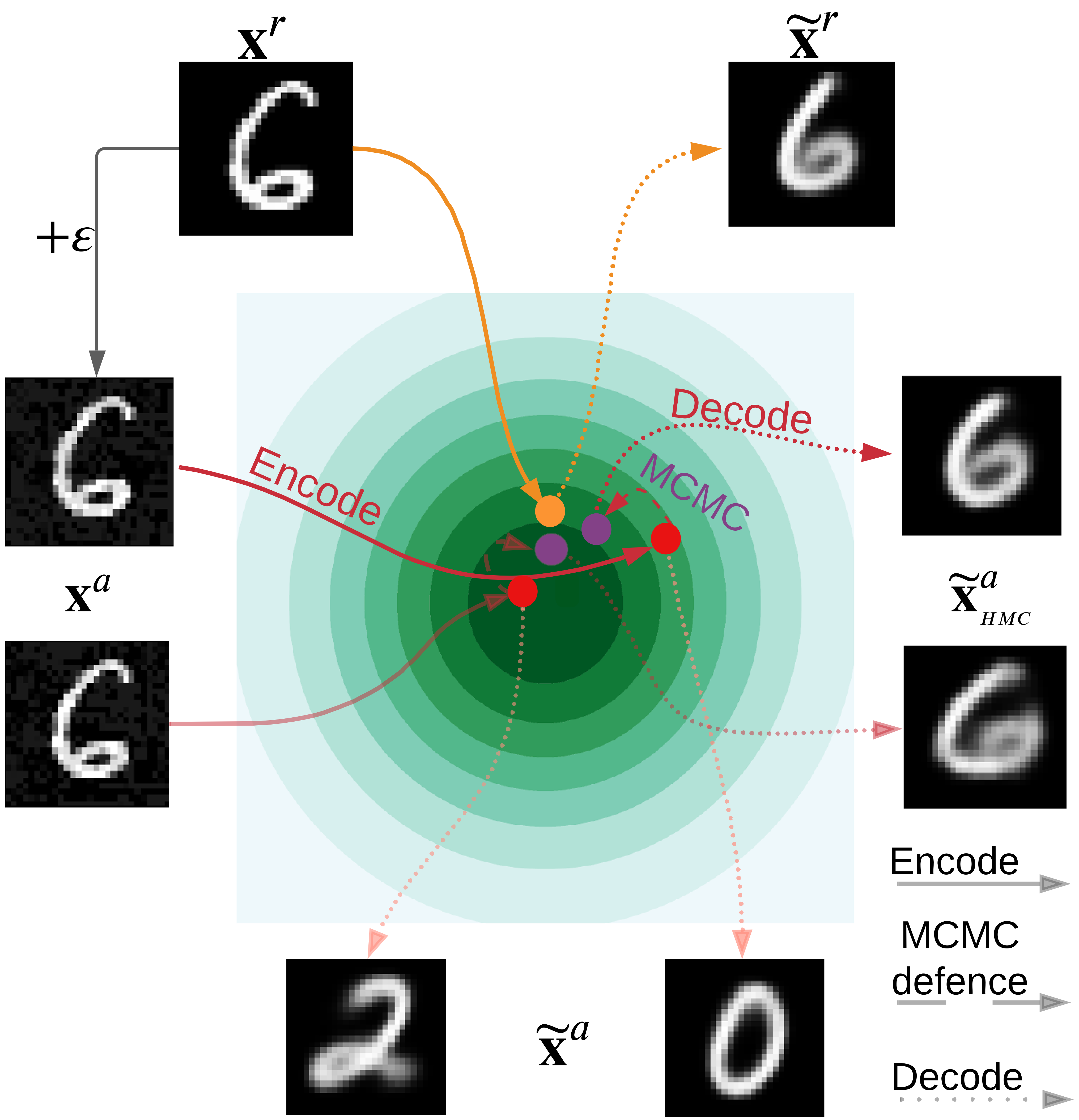
Empirical Results



(a) $\mathbf{x}^r$ (b) $\widetilde{\mathbf{x}}^r$ (c) $\mathbf{x}^a$ (d) $\widetilde{\mathbf{x}}^a$ (e) $\widetilde{\mathbf{x}}^{\mathrm{a}}_{\mathrm{HMC}}$

NVAE:
deep hierarchical VAE

Take Home Message



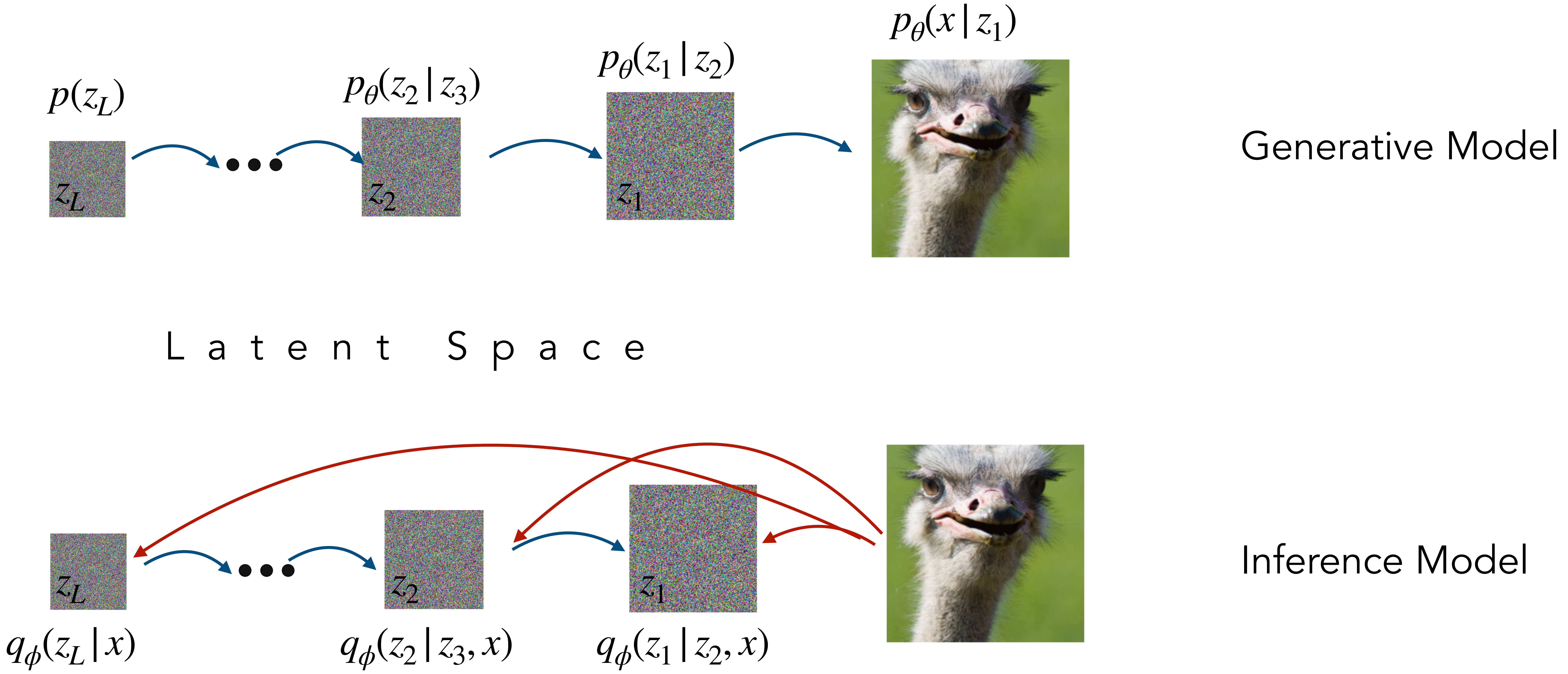
Latent representations of the data learned by VAE are vulnerable to adversarial attacks

We can use decoder to alleviate the effect of the attack

We "pay" with the inference time for the increased robustness

Hierarchical Variational Autoencoder

What if $z = (z_1, \ldots, z_L)$?

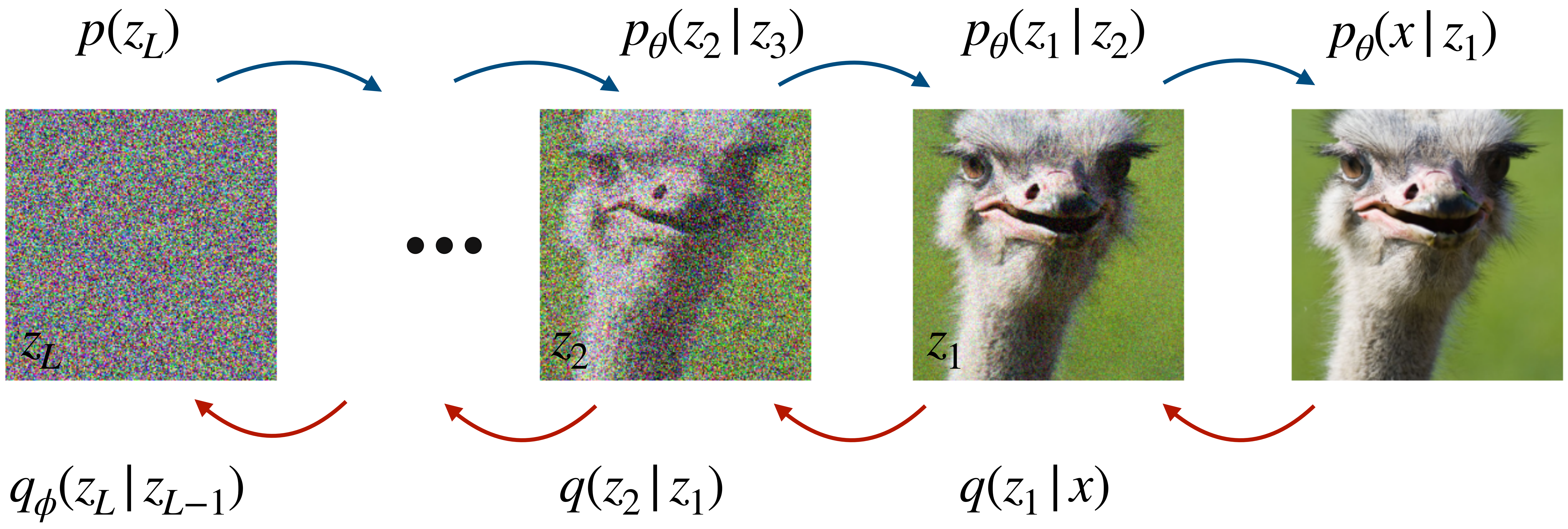


Sønderby, Casper Kaae, et al. "Ladder variational autoencoders." *NeurIPS* 2016.

Hierarchical Variational Autoencoder

Child, Rewon. "Very deep vaes generalize autoregressive models and can outperform them on images." *ICLR* 2021.

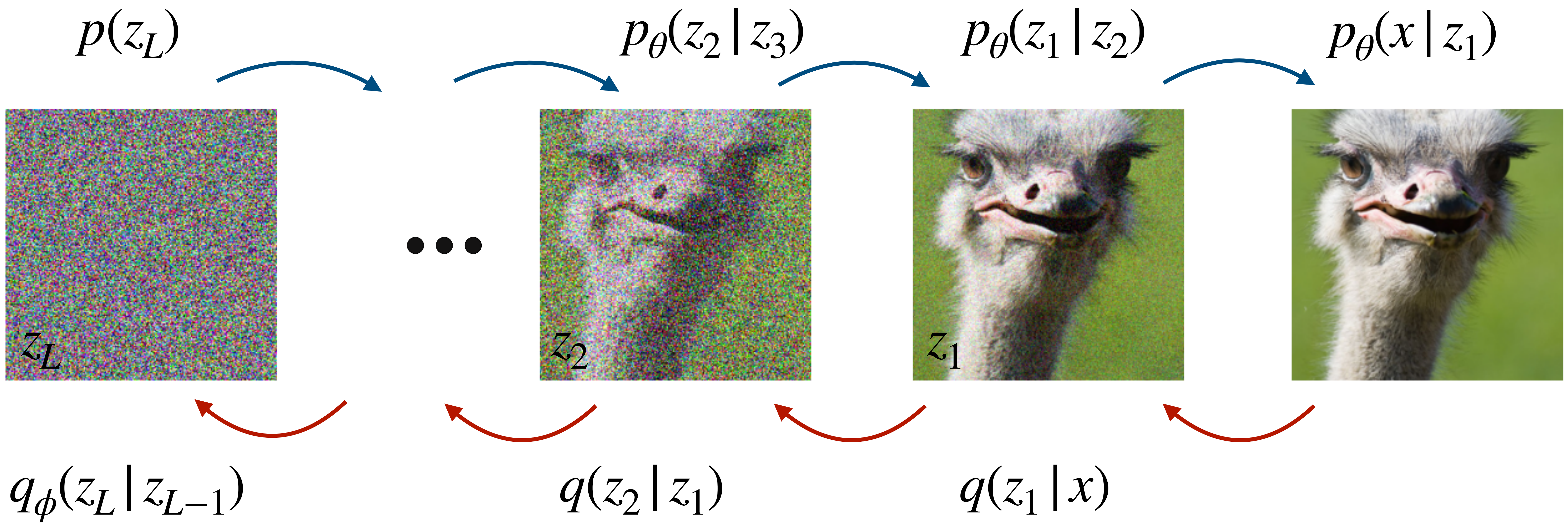
Diffusion-based Generative Models



Generative Model
(backward diffusion process)

all latent variables have the same dimensionality as the input

Diffusion-based Generative Models



Generative Model
(backward diffusion process)

all latent variables have the same dimensionality as the input

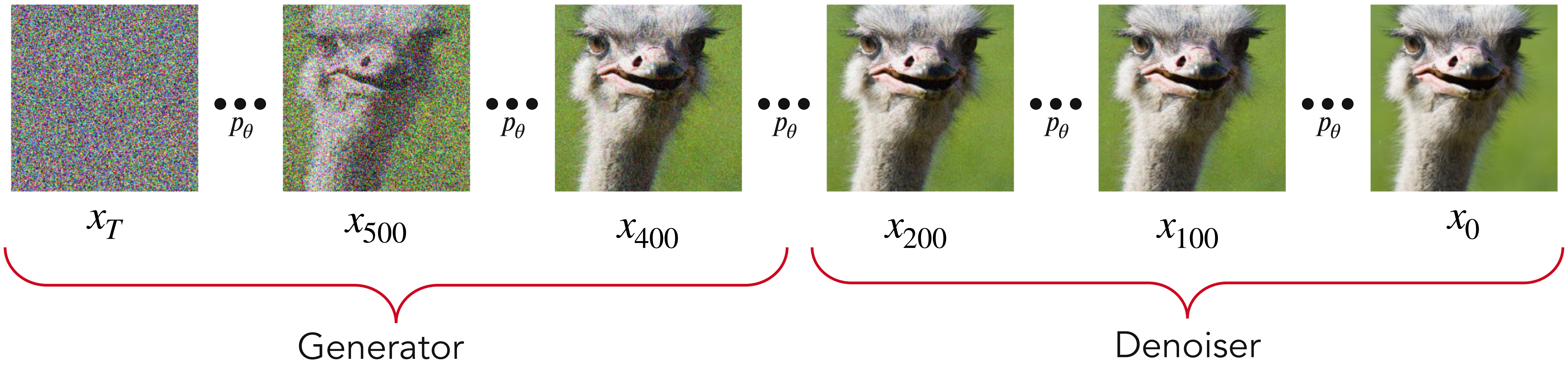
Fixed Inference Model
(or diffusion process)

$$q(z_l | z_{l-1}) = \mathcal{N}(z_l | \sqrt{1 - \beta_l} z_{l-1}, \beta_l I)$$

add gaussian noise according to pre-defined schedule $\beta_1, \ldots, \beta_L$

Diffusion-base Generative Models

Can we split the DGM into "generator" and "denoiser"?



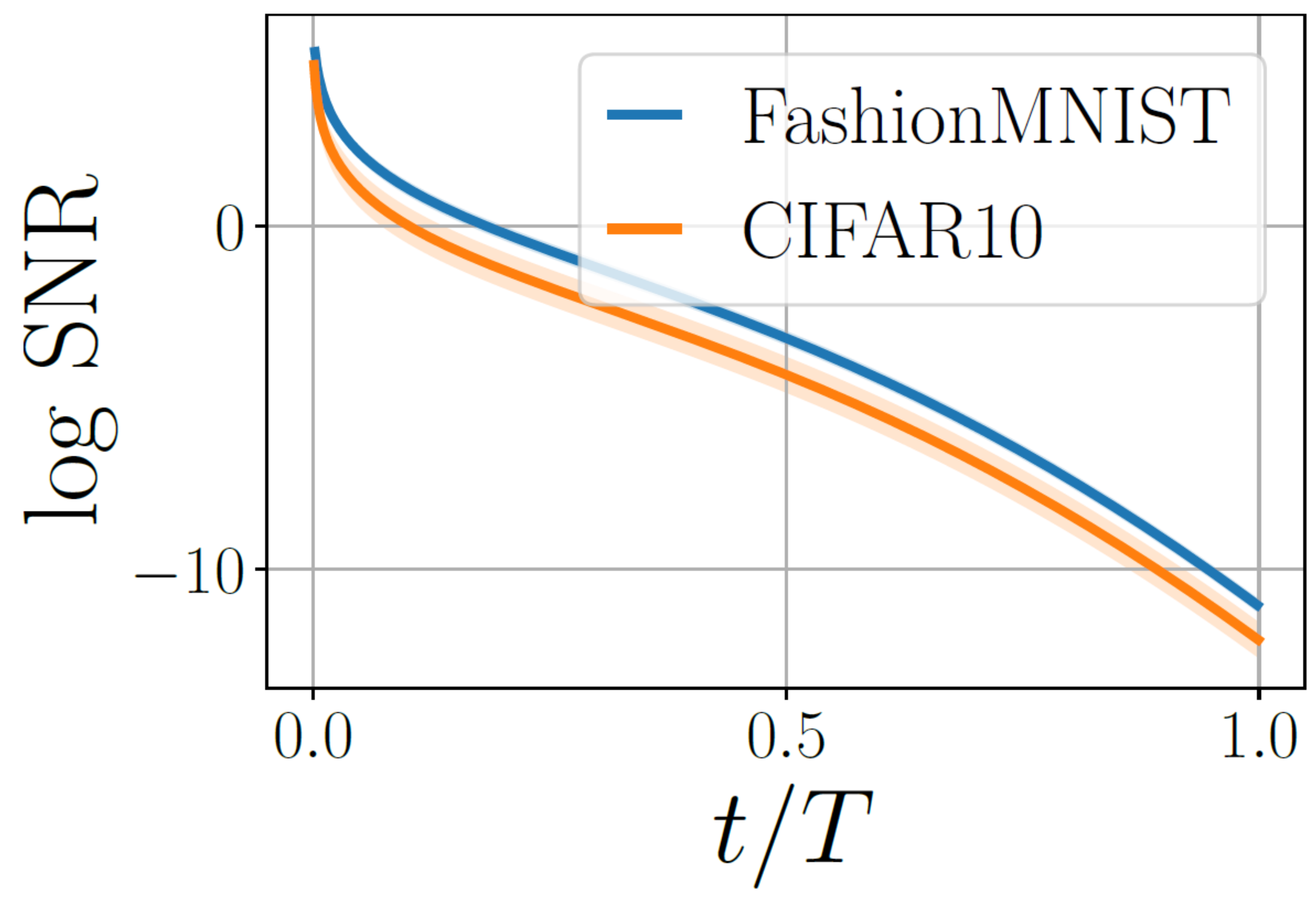
Deja, K. et al, "On Analyzing Generative and Denoising Capabilities of Diffusion-based Generative Models", NeurIPS 2022

Signal-to-noise ratio

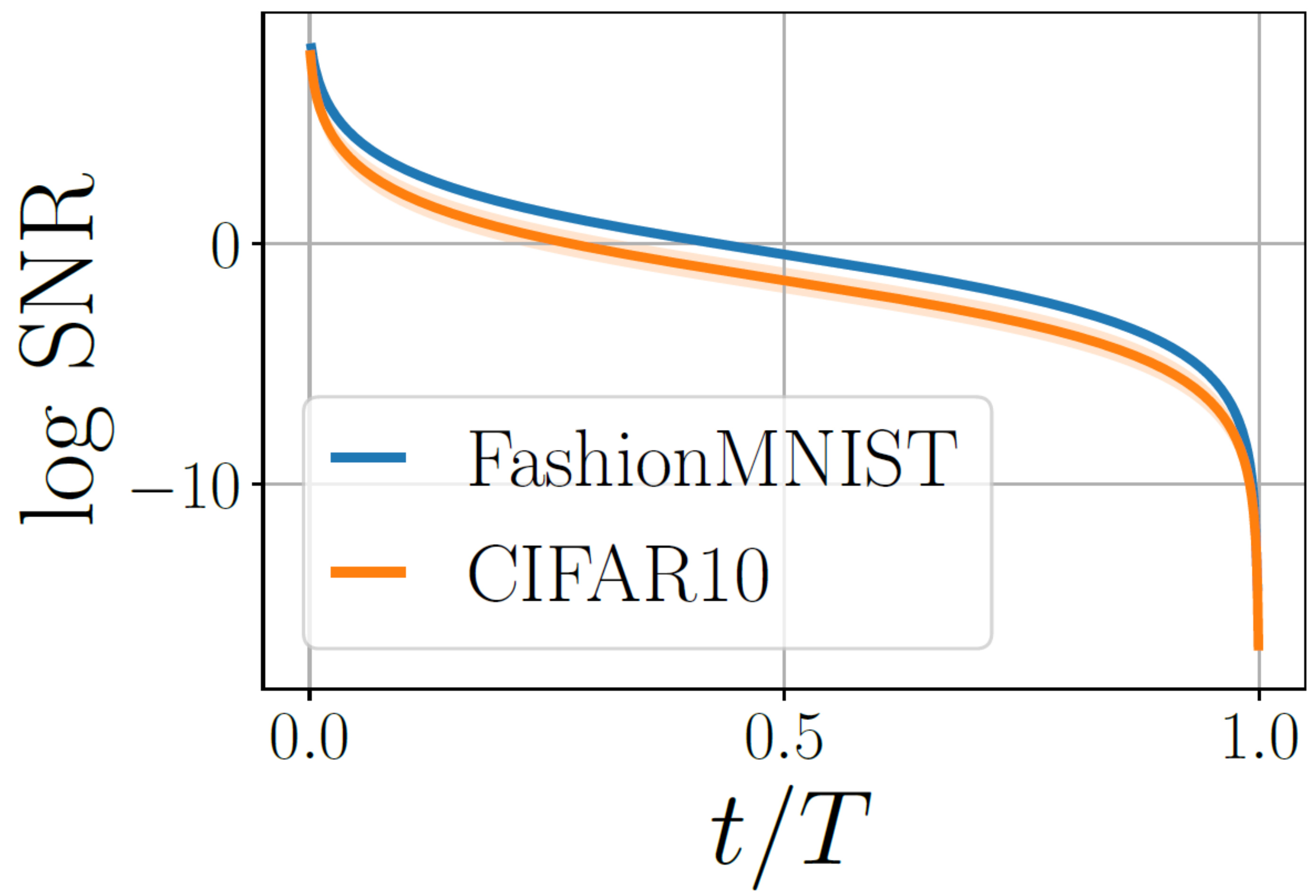
Observation 1: We have strong signal within first 10-20% steps of the forward process

$$q(z_l | z_{l-1}) = \mathcal{N}(z_l | \sqrt{1-\beta_l} z_{l-1}, \beta_l I)$$

Linear β schedule



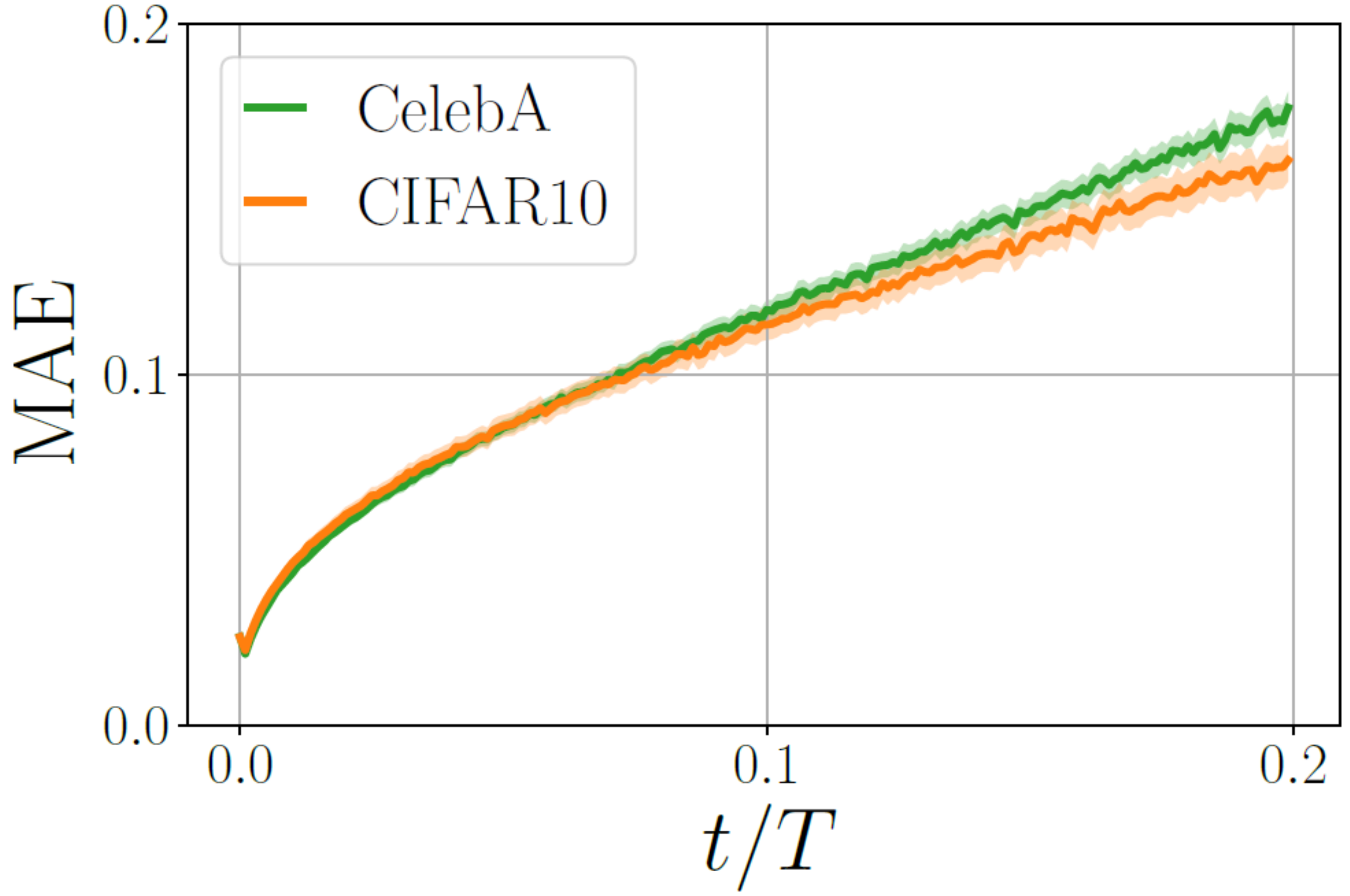
Cosine β schedule



How good is denoising?

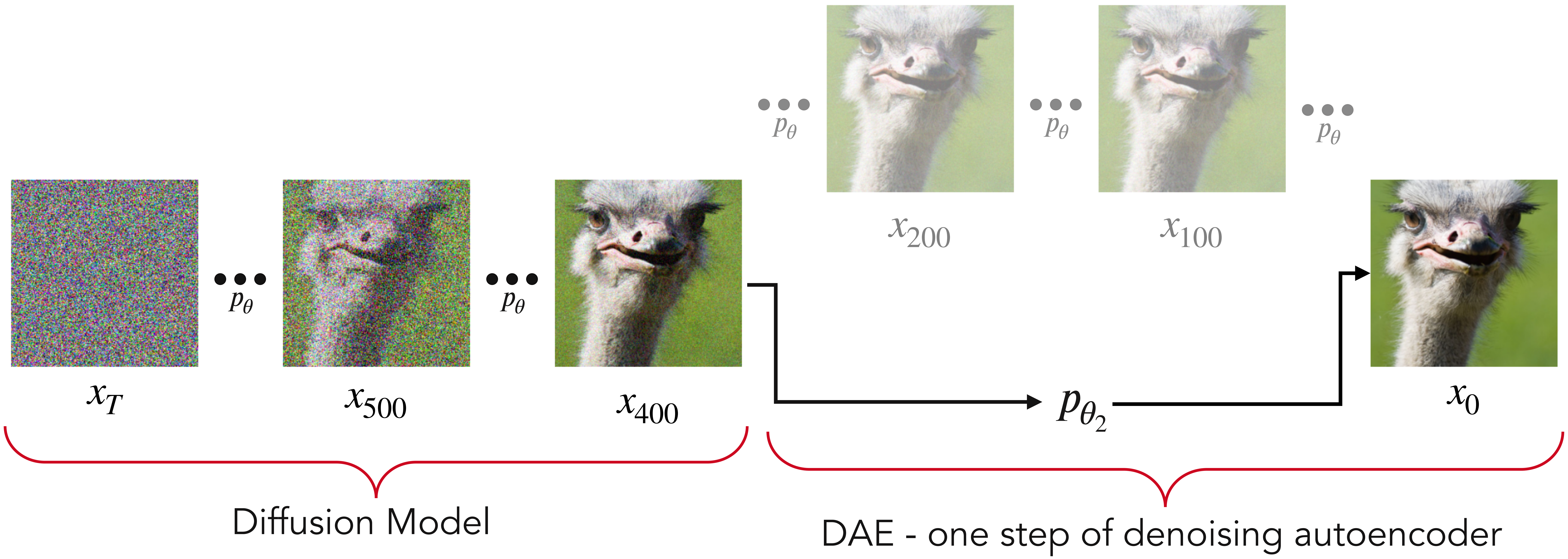
Observation 2: Backward process is capable of denoising the out-of-distribution data

DGM trained on CIFAR 10



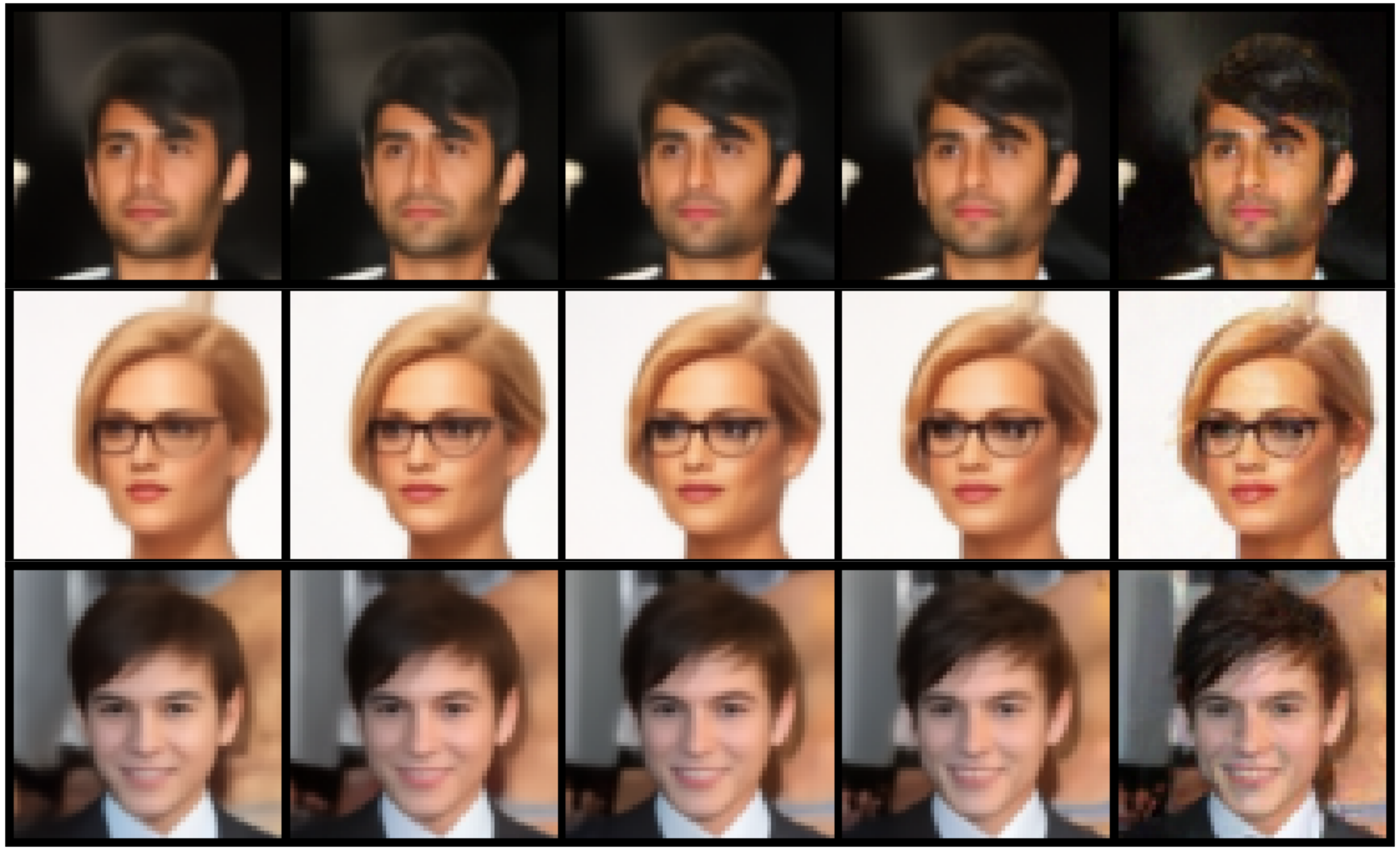
DAED: DGM + Denoising Autoencoder

Let's make denoising step **explicit**



DAED: DGM + Denoising Autoencoder

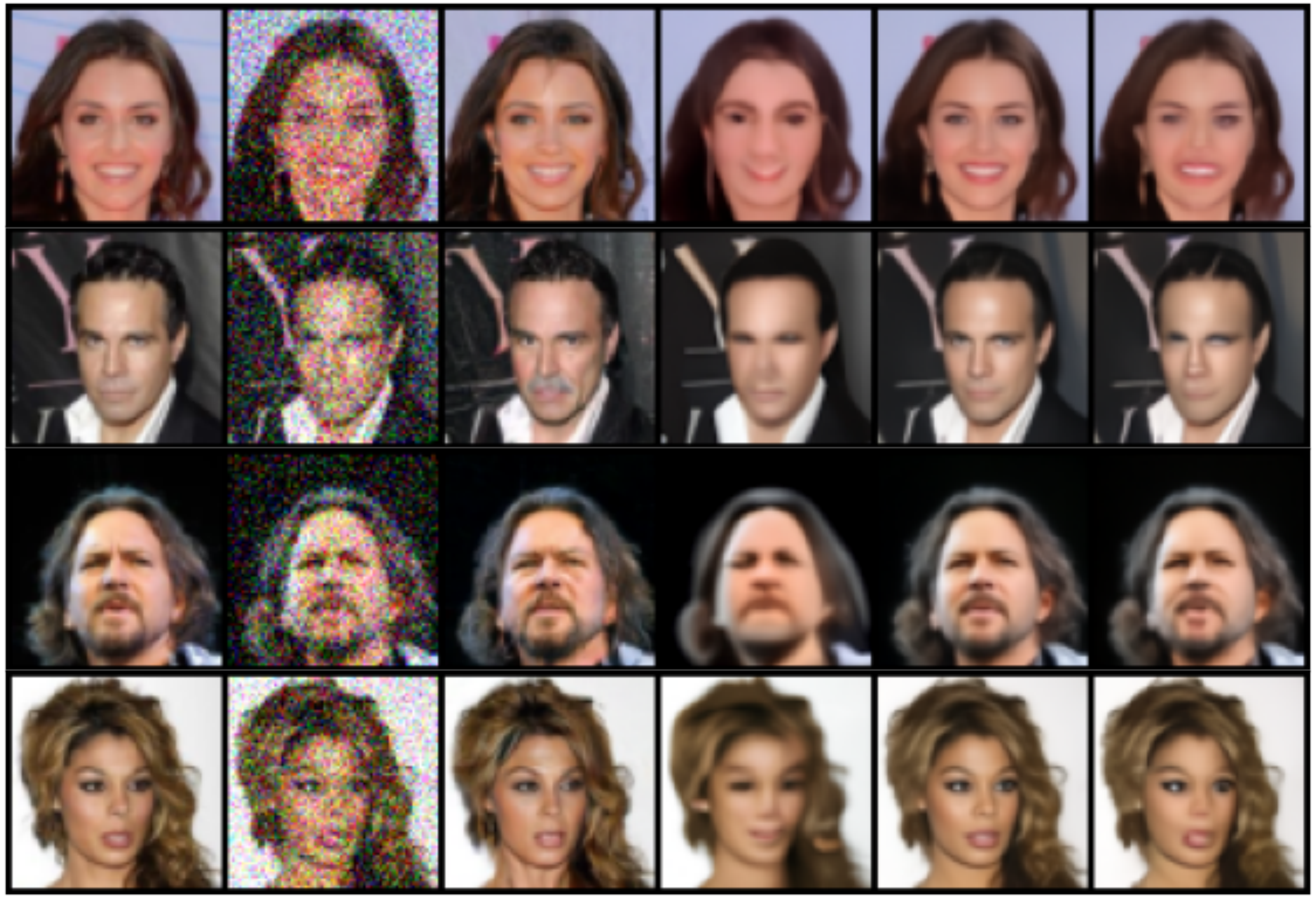
We replace up to 10% of DGM steps with a single DAE without significant drop in model's performance



$\beta_1 = 0.2$ $\beta_1 = 0.1$ $\beta_1 = 0.05$ $\beta_1 = 0.025$ $\beta_1 = 0.001$

Transferability of noise removal between data distributions

DAED is better in removing noise from unseen data distribution



Alleviating Adversarial Attacks on Variational Autoencoders with MCMC

NeurIPS 2022

the best co-authors:

Max Welling

Jakub M. Tomczak

On Analyzing Generative and Denoising Capabilities of Diffusion-based Generative Models

NeurIPS 2022

the best co-authors:

Kamil Deja

Tomasz Trzciński

Jakub M. Tomczak

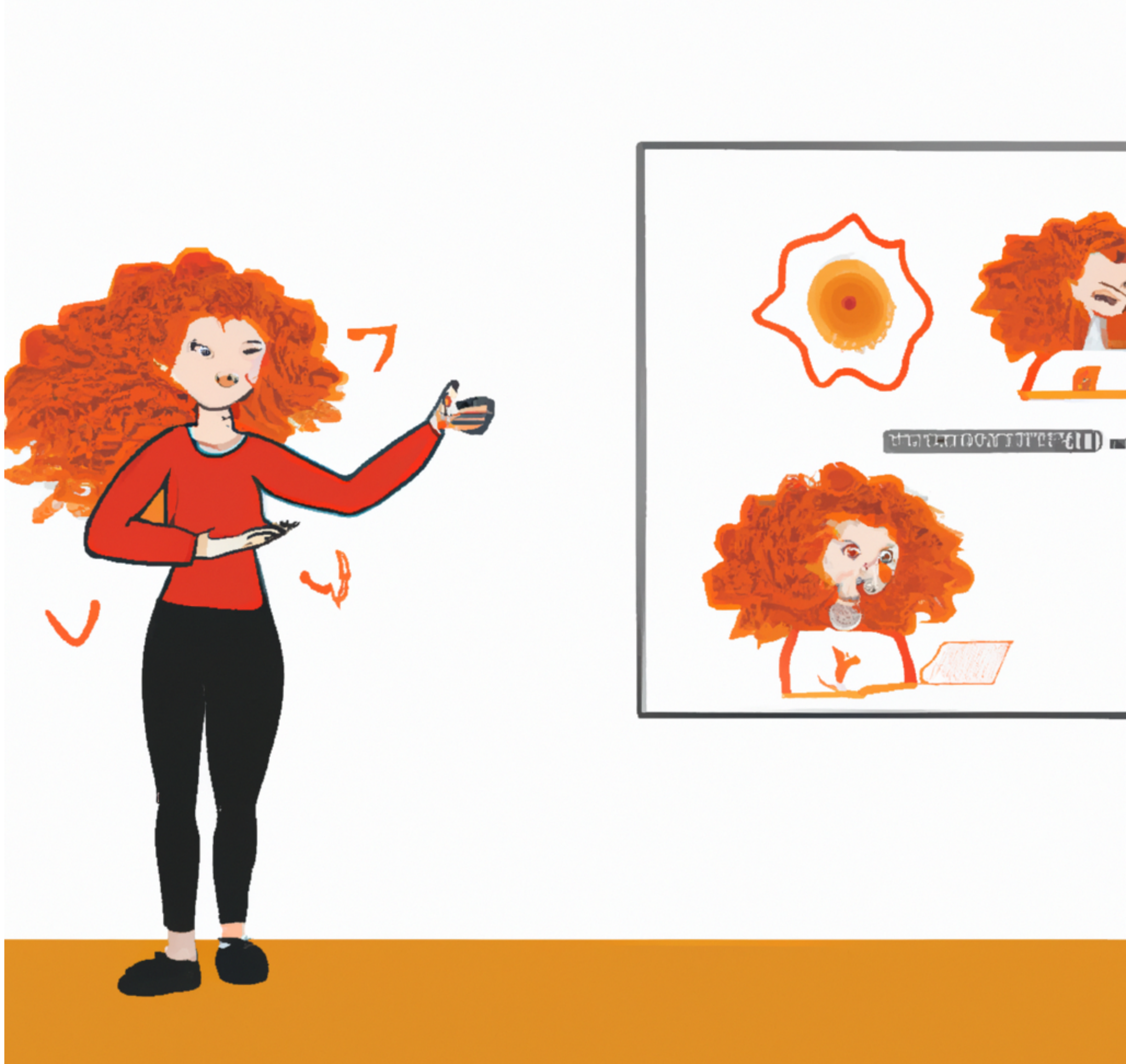
Thank you

Anna Kuzina

PhD student
Vrije Universiteit Amsterdam

a.kuzina@vu.nl

akuzina.github.io/



Generated with DALL·E2
Ramesh et al., “Hierarchical Text-Conditional Image Generation with CLIP Latents”

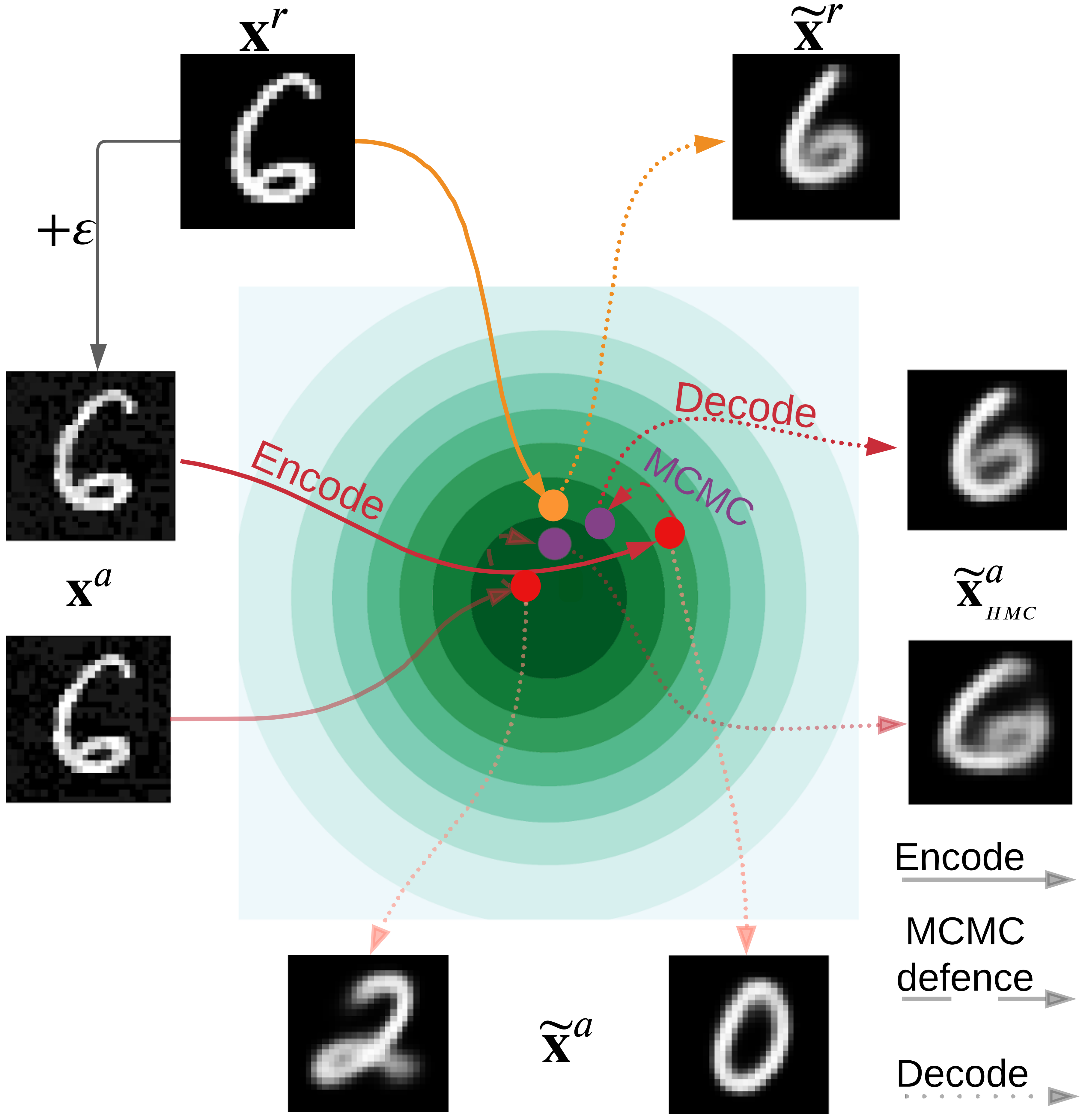
Open questions

Can we use adversarial attacks as a diagnostic tool?

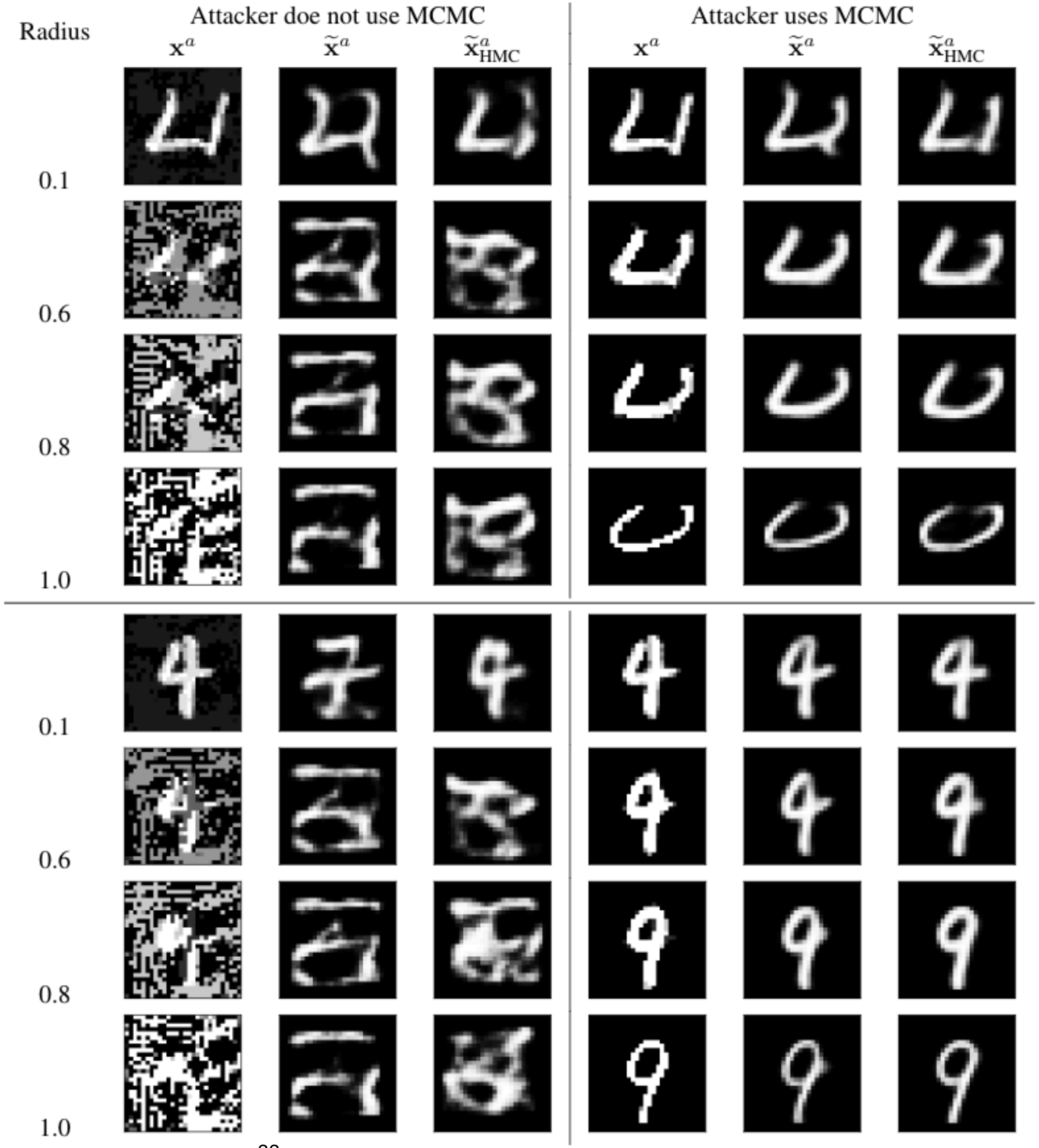
Is this problem relevant to diffusion models?

Can we use samples from the posterior for OOD detection?

Can we attack decoder in a more efficient way?



What if attacker knowns the defence strategy?



Why it works?

Empirical Evidence

Given a reference point, one can evaluate posterior ratio for two latent codes:

$$\mathrm{PR}(z_1, z_2) = \frac{p_\theta(z_1 \mid x^r)}{p_\theta(z_2 \mid x^r)}$$

Blue: reference latent code VS adversarial latent code

Orange: reference latent code VS adversarial latent code after HMC