



STAT 453: Introduction to Deep Learning and Generative Models

Ben Lengerich

Lecture 15: Diffusion Models

October 29, 2026

Reading: See course homepage

Project

- Midway report due next Friday
- 4 pages, more detailed expectations:
<https://adaptinfer.org/dgm-fall-2026/project/>

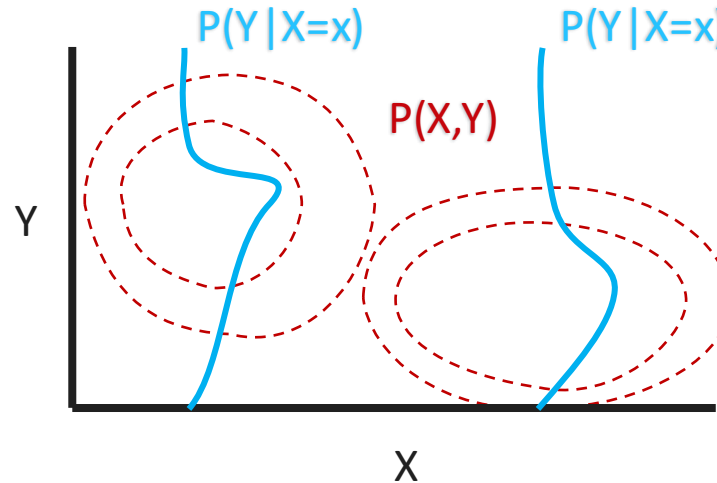
Today

- Diffusion Models



Recall: Generative and Discriminative Models

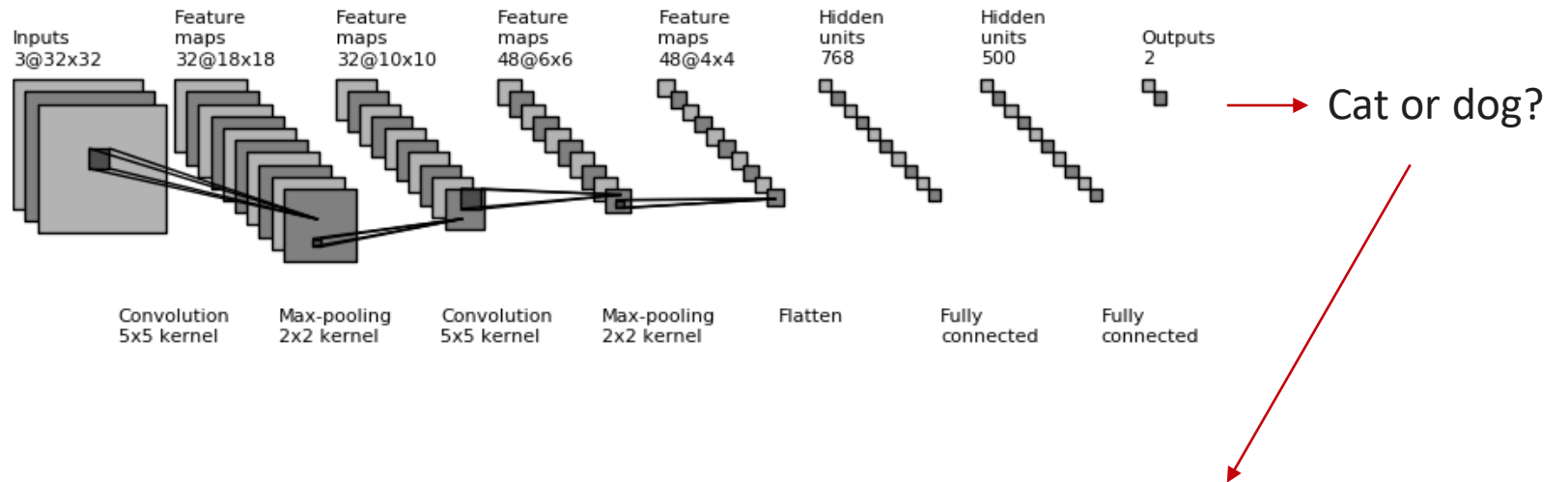
- **Generative:**
 - Models the joint distribution $P(X, Y)$.
- **Discriminative:**
 - Models the conditional distribution $P(Y|X)$.



Where we're going: Deep Generative Models



Discriminative Model (what we've seen so far)



Generative Model (what we're going to see)



Gemini

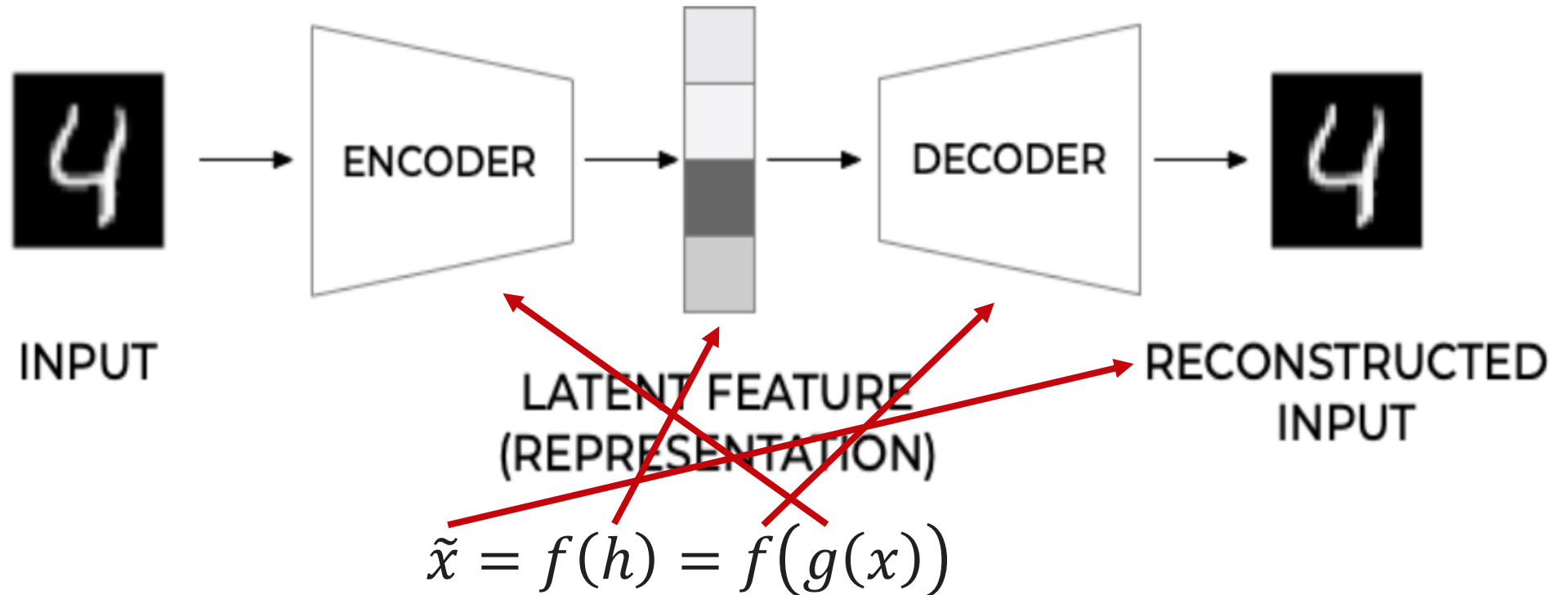


Grok



deepseek

Autoencoders

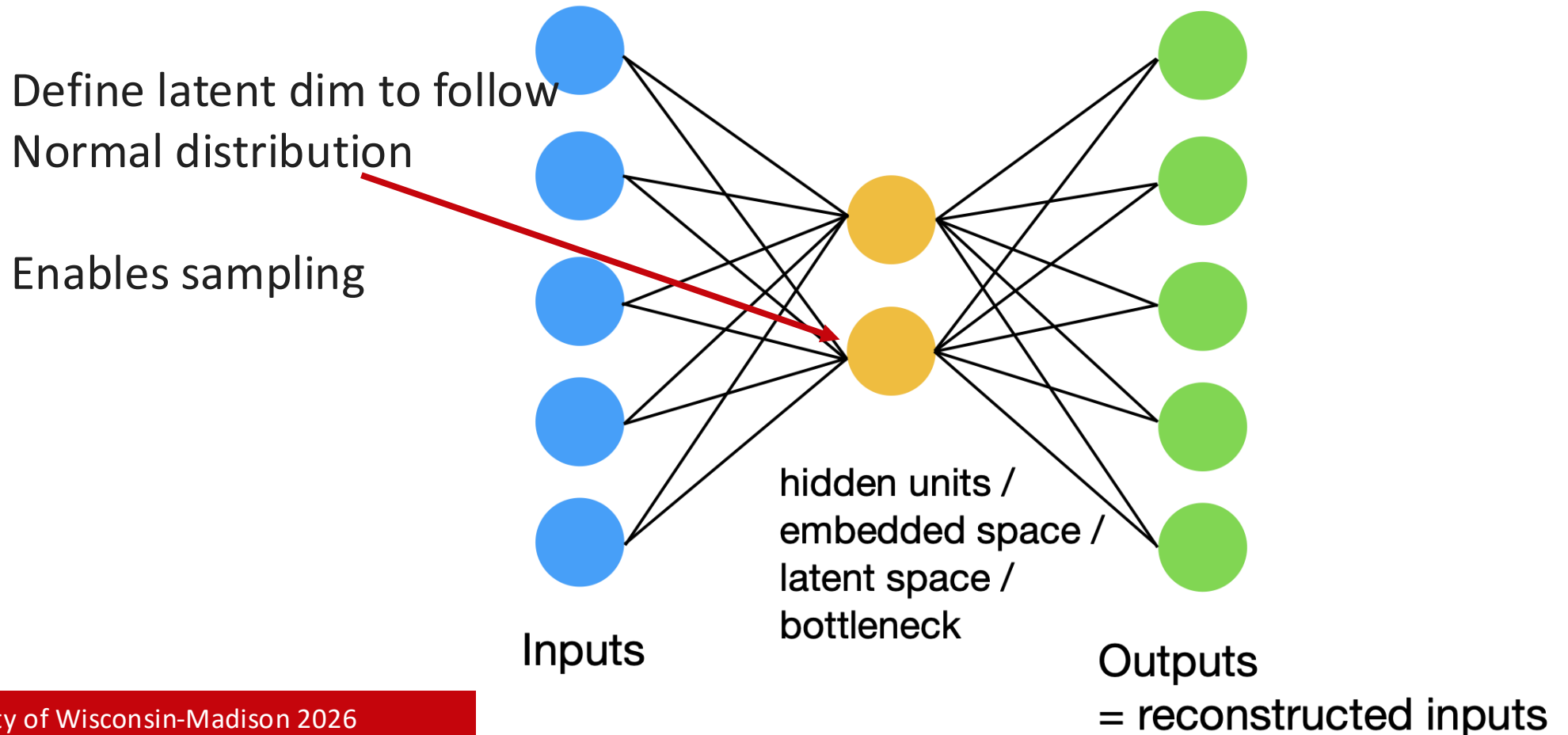


[[Michelucci 2022](#)]

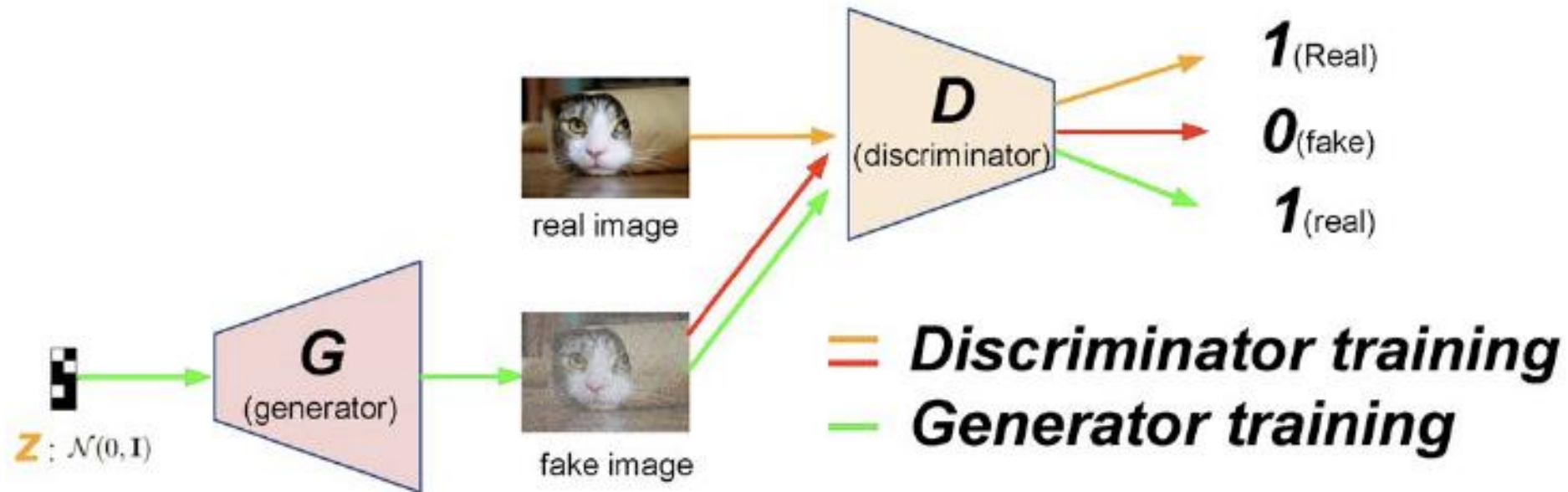
Variational Autoencoders

Kullback-Leibler divergence term
where $p(z) = \mathcal{N}(\mu = 0, \sigma^2 = 1)$

$$L^{[i]} = -\mathbb{E}_{z \sim q_w(z|x^{[i]})} [\log p_w(x^{[i]}|z)] + \text{KL}(q_w(z|x^{[i]}) || p(z))$$



Generative Adversarial Networks



Discriminator: $\max_D \mathcal{L}_D = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim G(\mathbf{z}), \mathbf{z} \sim p(\mathbf{z})} [\log(1 - D(\mathbf{x}))]$

Generator: $\min_G \mathcal{L}_G = \mathbb{E}_{\mathbf{x} \sim G(\mathbf{z}), \mathbf{z} \sim p(\mathbf{z})} [\log(1 - D(\mathbf{x}))]$.

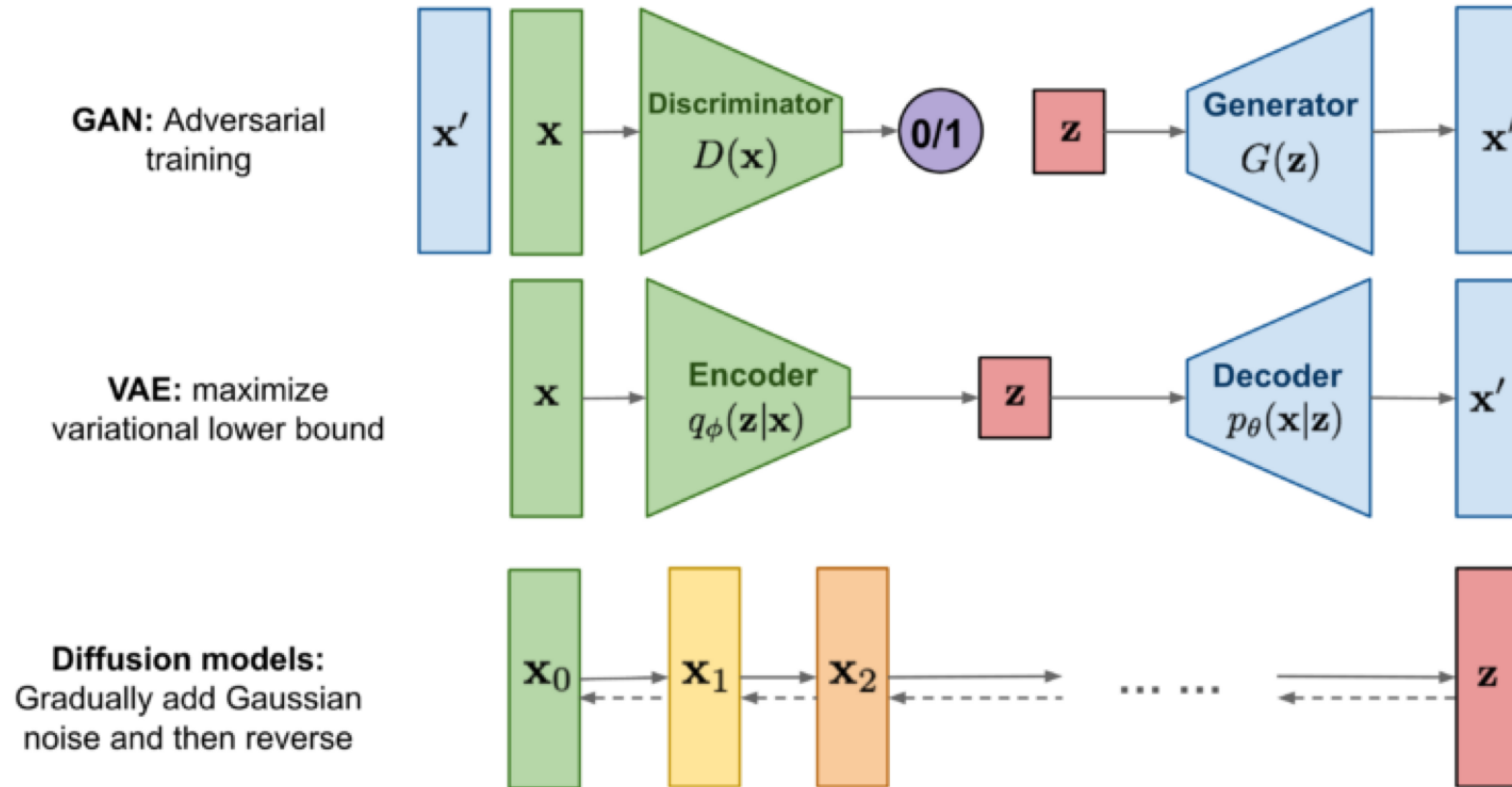
Summary

Property	VAE	GAN
What we specify	Prior $p(z)$, Likelihood $p_\theta(x z)$	Prior $p(z)$, Generator $G_\theta(z)$
Induced $p(x)$	$p_\theta(x) = \int_z p_\theta(x z) p(z) dz$	$p_\theta(x) = \int_z p_\epsilon(x - G_\theta(z)) p(z) dz$
Simplifying assumption	Choose a restricted variational posterior $q_\phi(z x)$	Replace NLL with a distributional discrepancy on samples (adversarial/IPM).
Training objective	ELBO: $E_q[\log p_\theta(x z)] - KL(q_\phi(z x) p(z))$	Minimax fooling discriminator
What's ignored from $p_\theta(x)$	$KL(q_\phi(z x) p_\theta(z x))$	All of NLL: $\log p_\theta(x)$ isn't evaluated or maximized.
Modes	Covering	Collapse
Generated Samples	Blurry	Realistic
Training	Relatively robust	Fragile



Diffusion Models

Overview and comparison of generative models

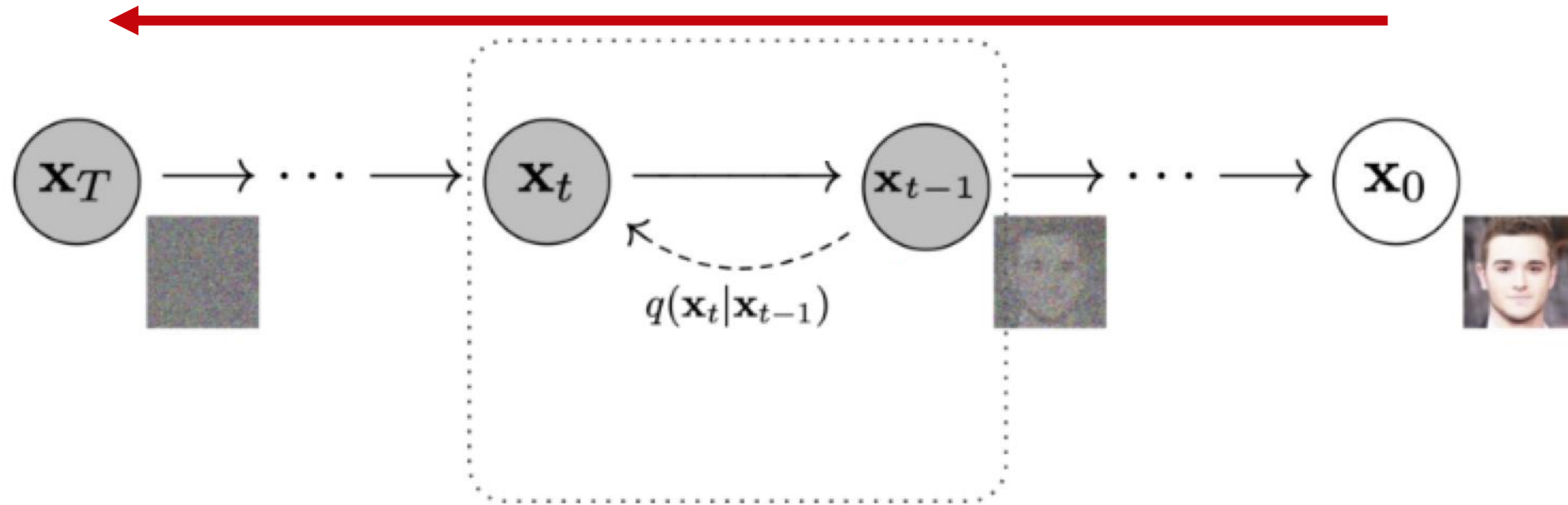


Diffusion



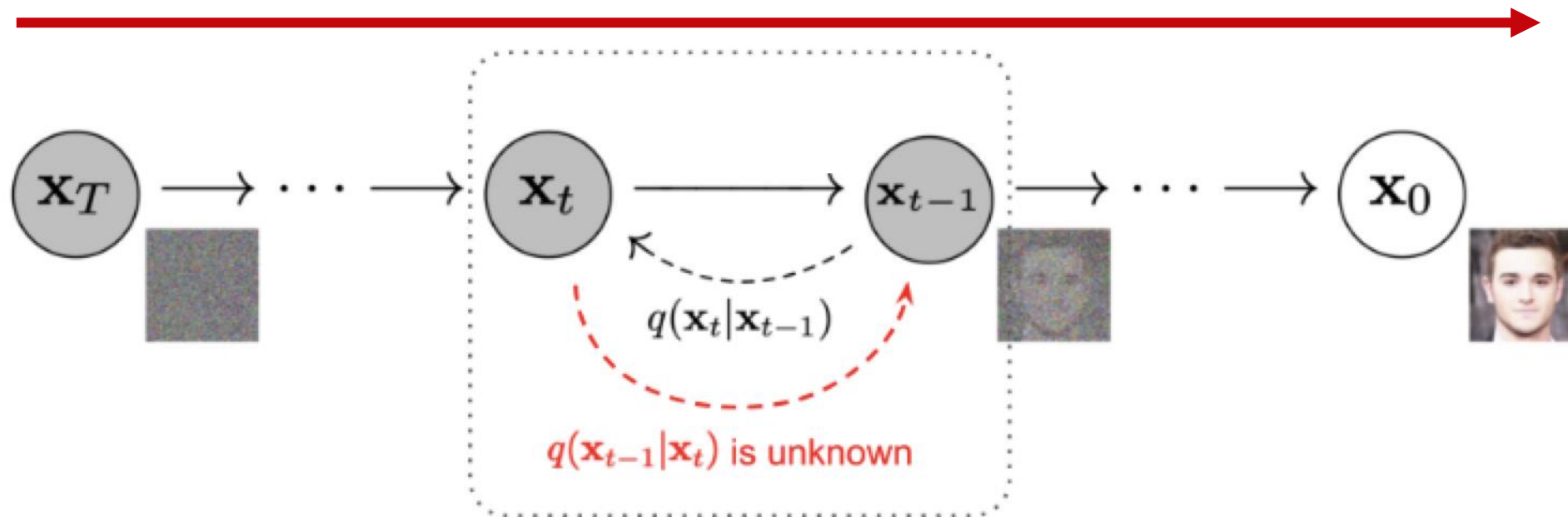
Diffusion models: forward pass

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})$$



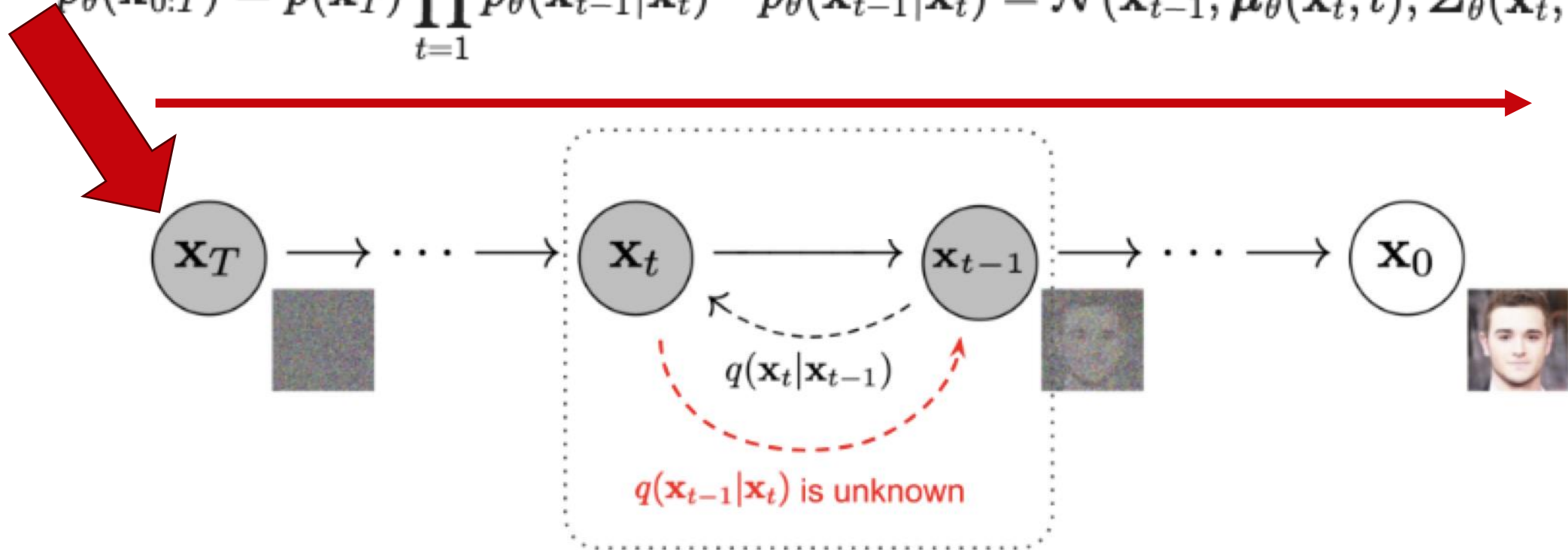
Diffusion models: reverse pass

$$p_{\theta}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\theta}(\mathbf{x}_t, t), \boldsymbol{\Sigma}_{\theta}(\mathbf{x}_t, t))$$

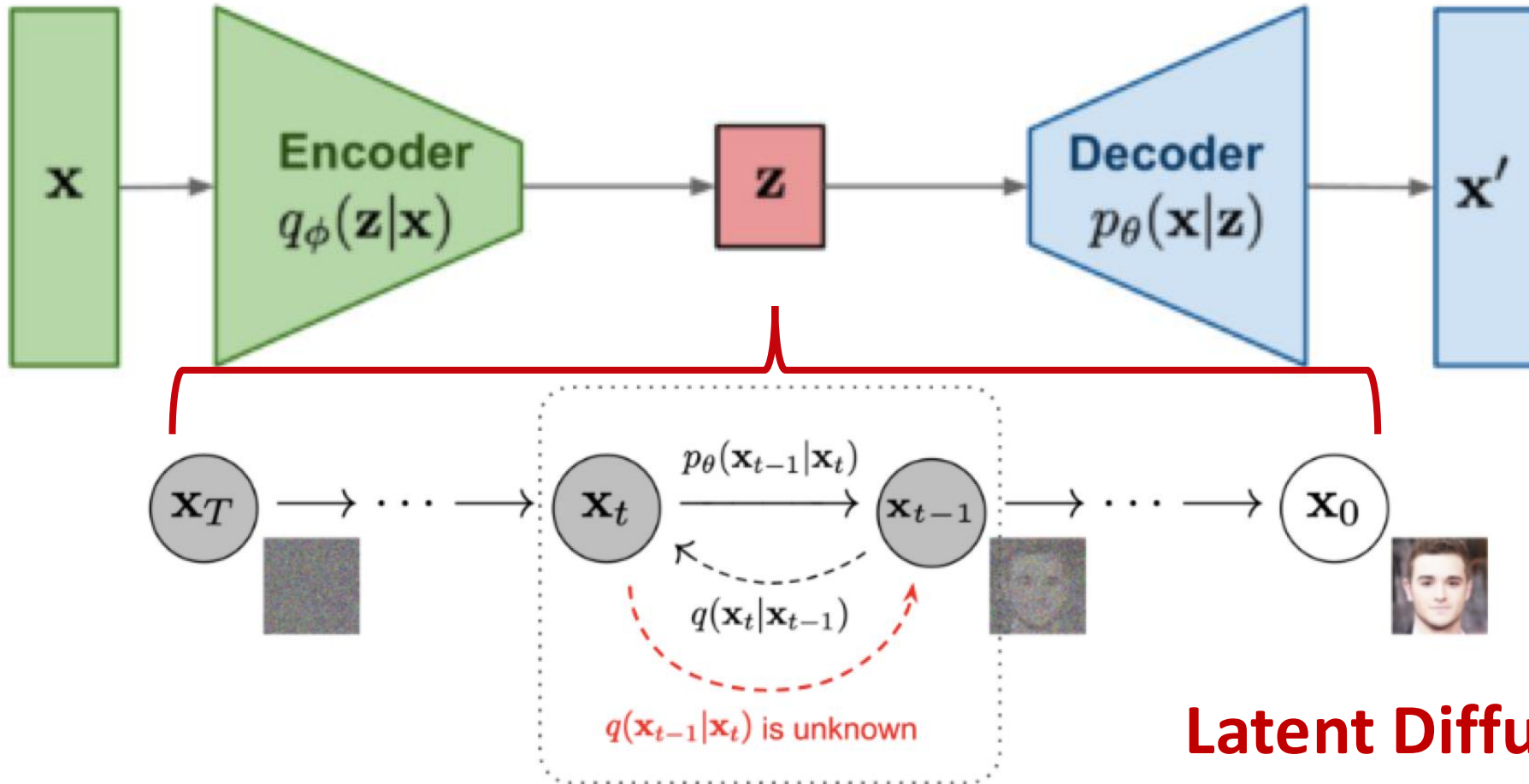


Diffusion models: generating a new sample

$$p_{\theta}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\theta}(\mathbf{x}_t, t), \boldsymbol{\Sigma}_{\theta}(\mathbf{x}_t, t))$$

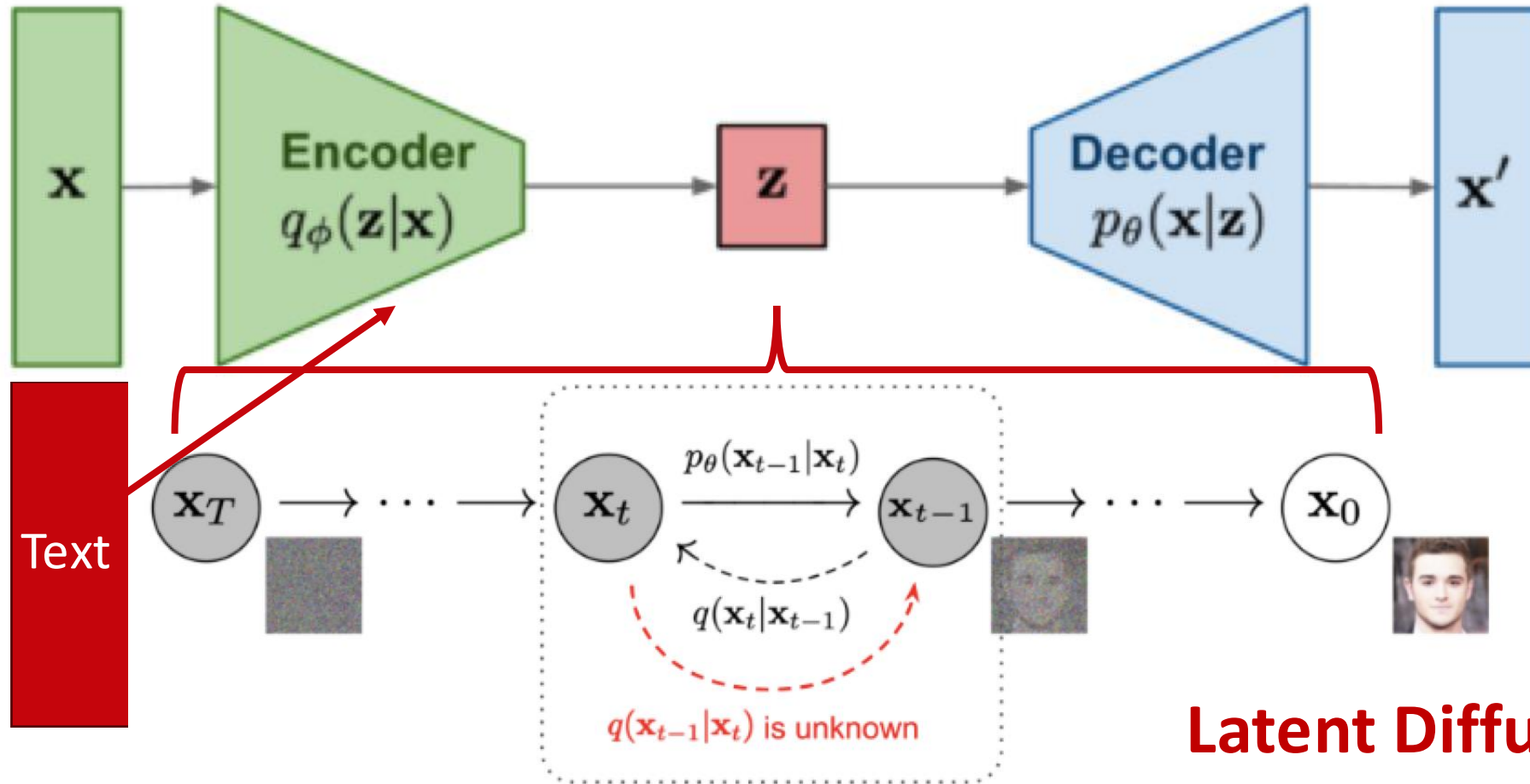


Should we really run this process on pixels?



Latent Diffusion!

Stable Diffusion: Add Text Conditioning



Latent Diffusion!

Stable Diffusion: Modern Image Generators



More reading

<https://lilianweng.github.io/posts/2021-07-11-diffusion-models/>

<https://lilianweng.github.io/posts/2021-07-11-diffusion-models/>

<https://theaisummer.com/diffusion-models/>



Property	VAE	GAN	Diffusion
What we specify	Prior $p(z)$, Family of likelihood $p_\theta(x z)$	Prior $p(z)$, Family of generator $G_\theta(z)$	Fixed forward noising processes $q(x_t x_{\{t-1\}})$
Induced $p(x)$	$p_\theta(x) = \int_z p_\theta(x z) p(z) dz$	$p_\theta(x) = \int_z p_\epsilon(x - G_\theta(z)) p(z) dz$	$p_\theta(x)$ $= \int p(x_T) \prod_t p_\theta(x_{t-1} x_t) dx$
Simplifying assumption	Restricted variational posterior $q_\phi(z x)$	Replace NLL with a distributional discrepancy on samples (adversarial/IPM).	Fix forward noise q ; and optimize a variational bound on $-\log p_\theta(x_0)$.
Training objective	ELBO: $E_q[\log p_\theta(x z)]$ $- KL(q_\phi(z x) p(z))$	Minimax fooling discriminator	VLB / score matching : with Gaussian schedules reduces to $\mathbb{E}_{t, x_0, \epsilon} [w(t) \ \epsilon - \epsilon_\theta(x_t, t) \ ^2]$
What's ignored from $p_\theta(x)$	$KL(q_\phi(z x) p_\theta(z x))$	All of NLL : $\log p_\theta(x)$ isn't evaluated or maximized.	Exact NLL not computed; optimize a variational upper bound on NLL (equivalently lower bound on $\log p$; (practical losses often reweight or drop constants from the exact VLB.
Modes	Covering	Collapse	Covering

Questions?

