STAT 453: Introduction to Deep Learning and Generative Models

Ben Lengerich

Lecture 18: Diffusion Models

November 5, 2025

Reading: See course homepage

Project

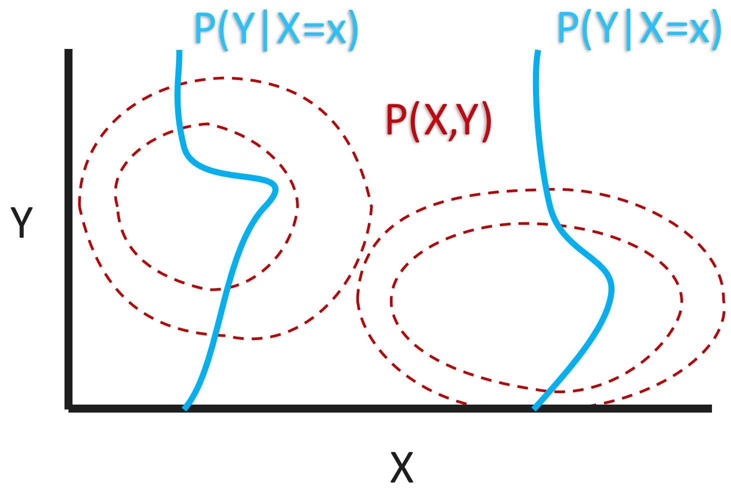
- https://adaptinfer.github.io/dgm-fall-2025/project/

Today

- Diffusion Models

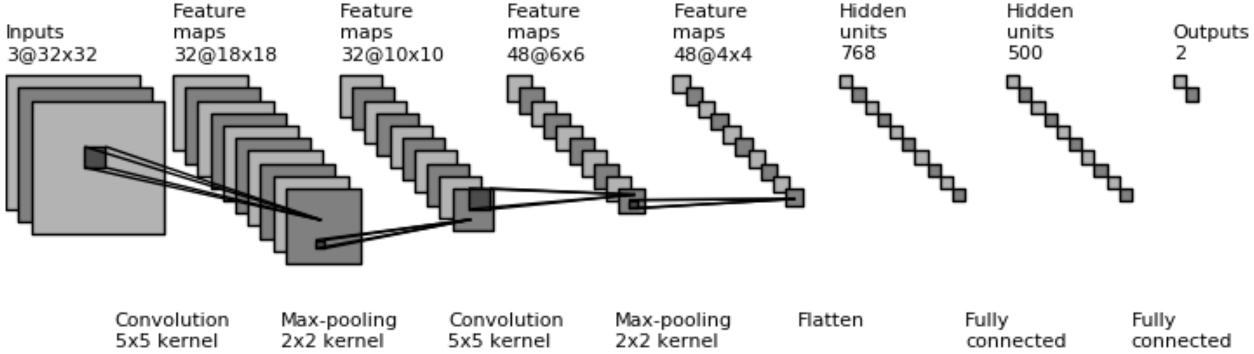
Generative and Discriminative Models

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



Where we're going: Deep Generative Models

Discriminative Model (what we've seen so far)



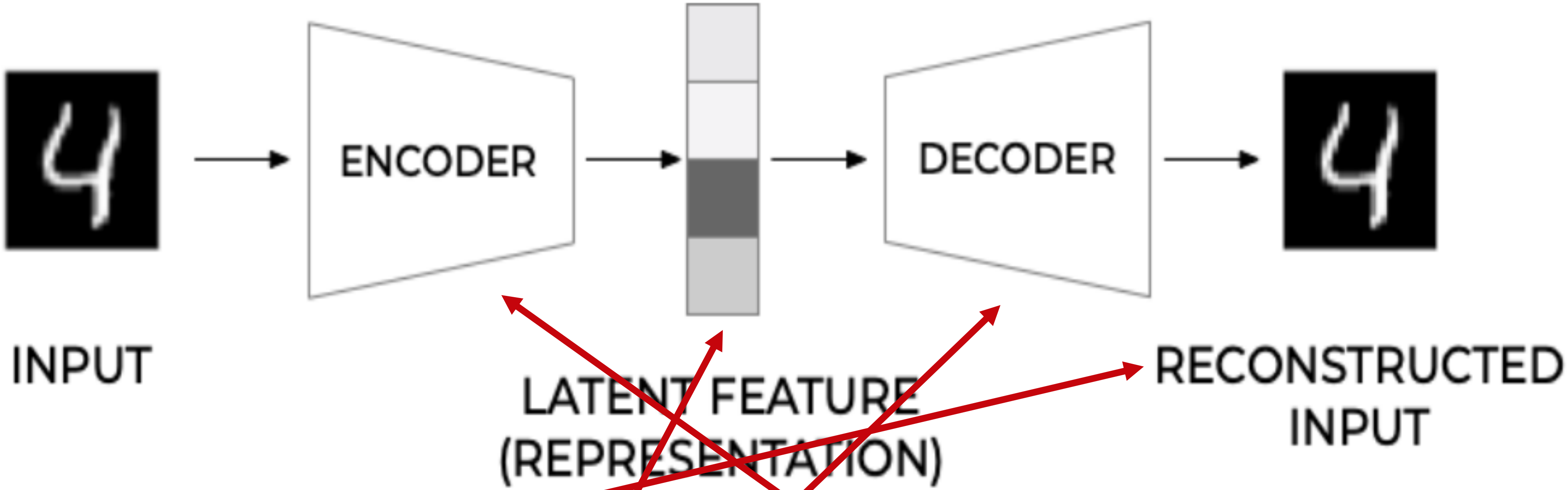
Cat or dog?

Generative Model (what we're going to see)

Modern Deep Generative Models (DGMs)

- Goal: Generative models of the form $\boldsymbol{P}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta})$ without strong simplifying assumptions.
- Hidden structure z that explains high-dim. x
- Fundamental challenge: We never observe z
- This makes two core computations difficult:
  - **Marginal likelihood:** $p_\theta(x) = \int p_\theta(x, z) dz$
  - **Posterior inference:** $p_\theta(z \mid x) \propto p_\theta(x \mid z) p(z)$
- Each type of DGM makes a tradeoff

Autoencoders



$$\tilde{x} = f(h) = f\big(g(x)\big)$$

[Michelucci 2022]

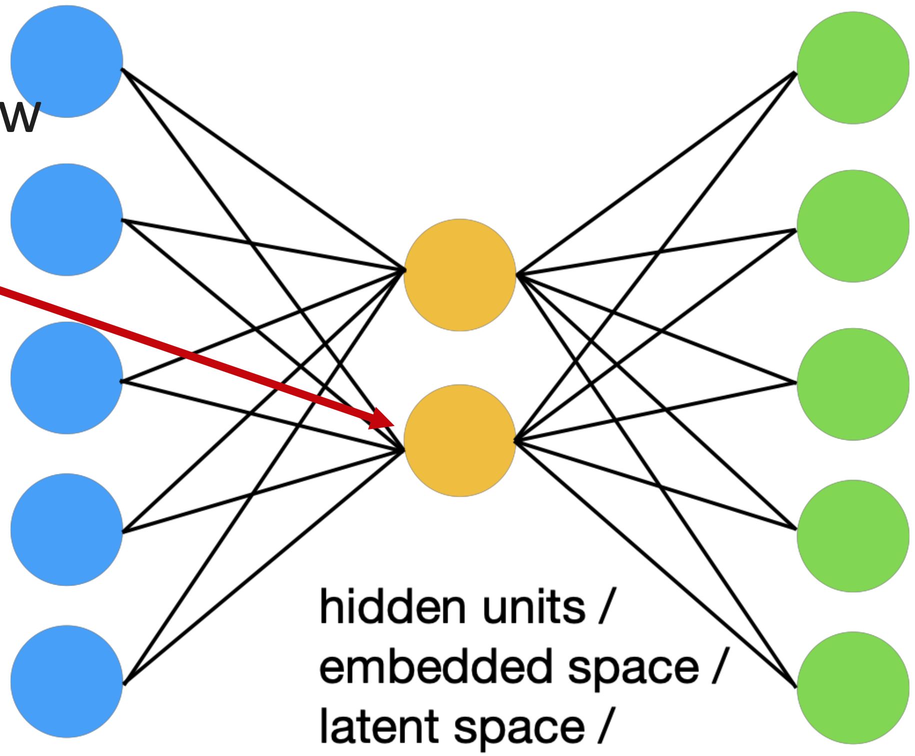
Variational Autoencoders

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

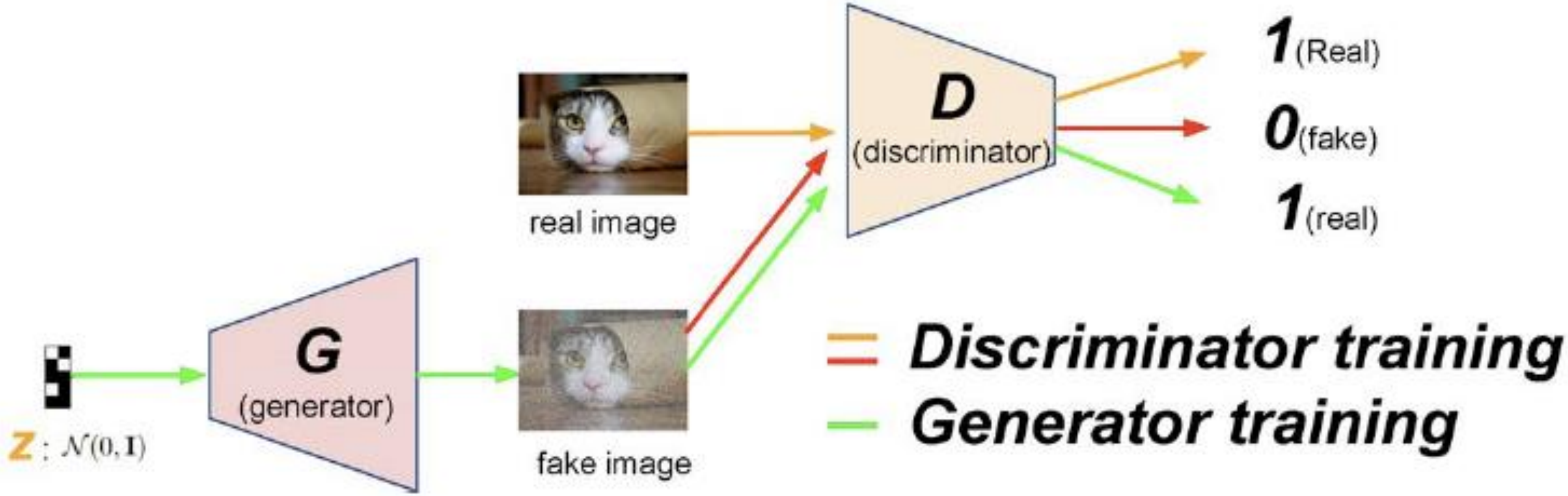
$$L^{[i]} = -\mathbb{E}_{z \sim q_w(z \mid x^{[i]})}\left[\log p_w\left(x^{[i]} \mid z\right)\right] + \mathbf{KL}\left(q_w\left(z \mid x^{[i]}\right) \| p(z)\right)$$

Define latent dim to follow Normal distribution

Enables sampling



Generative Adversarial Networks



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

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

Summary

Property	VAE	GAN
What we specify	Prior p(z), Likelihood $p_\theta(x \mid z)$	Prior p(z), Generator $G_\theta(z)$
Induced $p(x)$	$p_\theta(\mathrm{x}) = \int_z p_\theta(X \mid z)p(z)dz$	$p_\theta(\mathrm{x}) = \int_z p_\epsilon(x - \mathrm{G}_\theta(\mathrm{z}))\mathrm{p}(\mathrm{z})\mathrm{dz}$
Simplifying assumption	Choose a **restricted** variational posterior $q_\phi(z \mid x)$	**Replace NLL** with a **distributional discrepancy** on samples (adversarial/IPM).
Training objective	ELBO: $E_q[\log p_\theta(x \mid z)] - KL(q_\phi(z \mid x) \| p(z))$	Minimax fooling discriminator
What's ignored from $p_\theta(x)$	$KL(q_\phi(z \mid x) \| p_\theta(z \mid x))$	**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

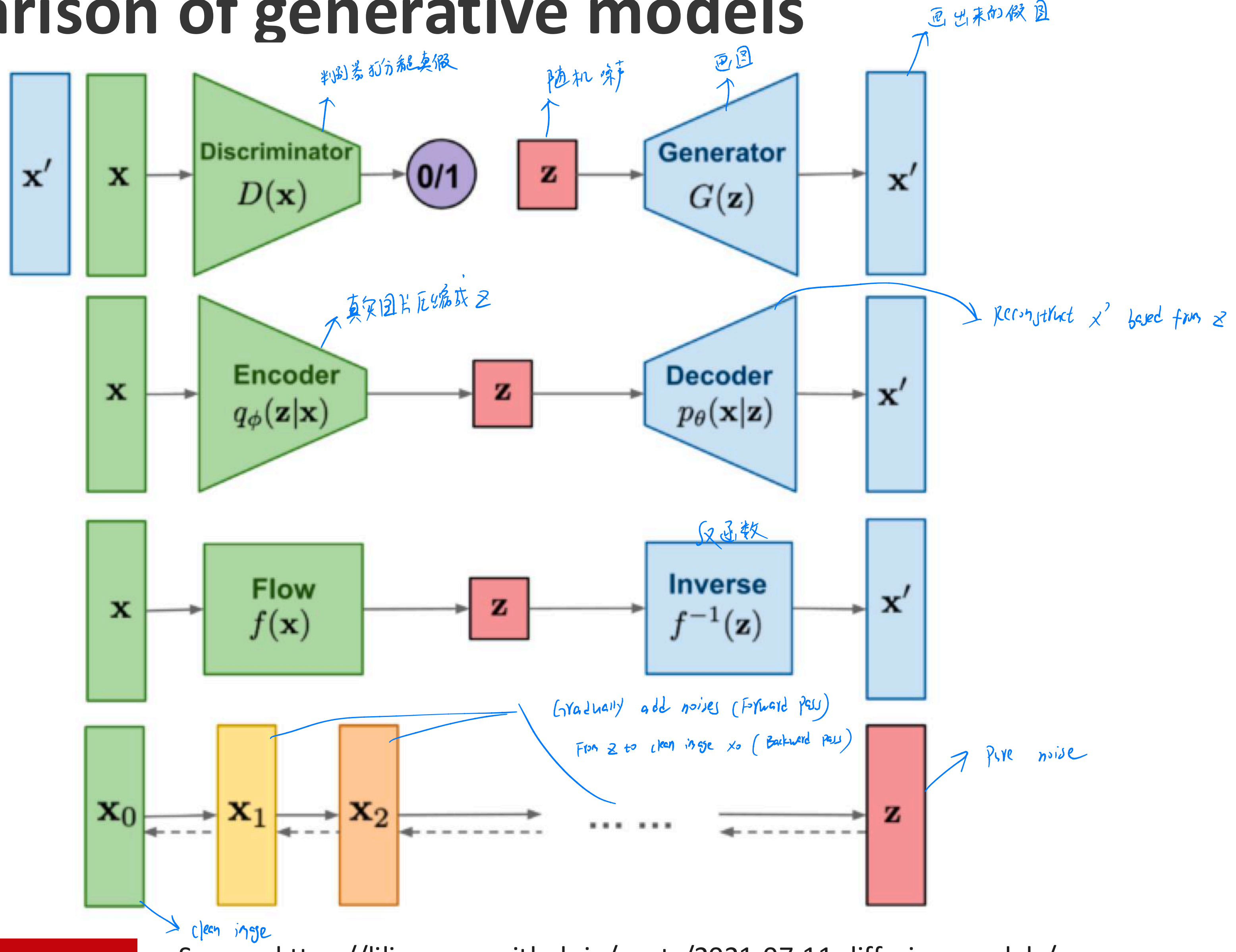
Overview and comparison of generative models

GAN: Adversarial training

VAE: maximize variational lower bound

Flow-based models: Invertible transform of distributions

Diffusion models: Gradually add Gaussian noise and then reverse



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

Diffusion

Diffusion models: forward pass

Turn a clear image into complete random noise

条件概率分布（给定上一刻的图，这一刻的图会变成啥样）

Normal distribution (Random noise)

For whole procedure of x_0 to x_T

Multiply by T times (Markov chain)

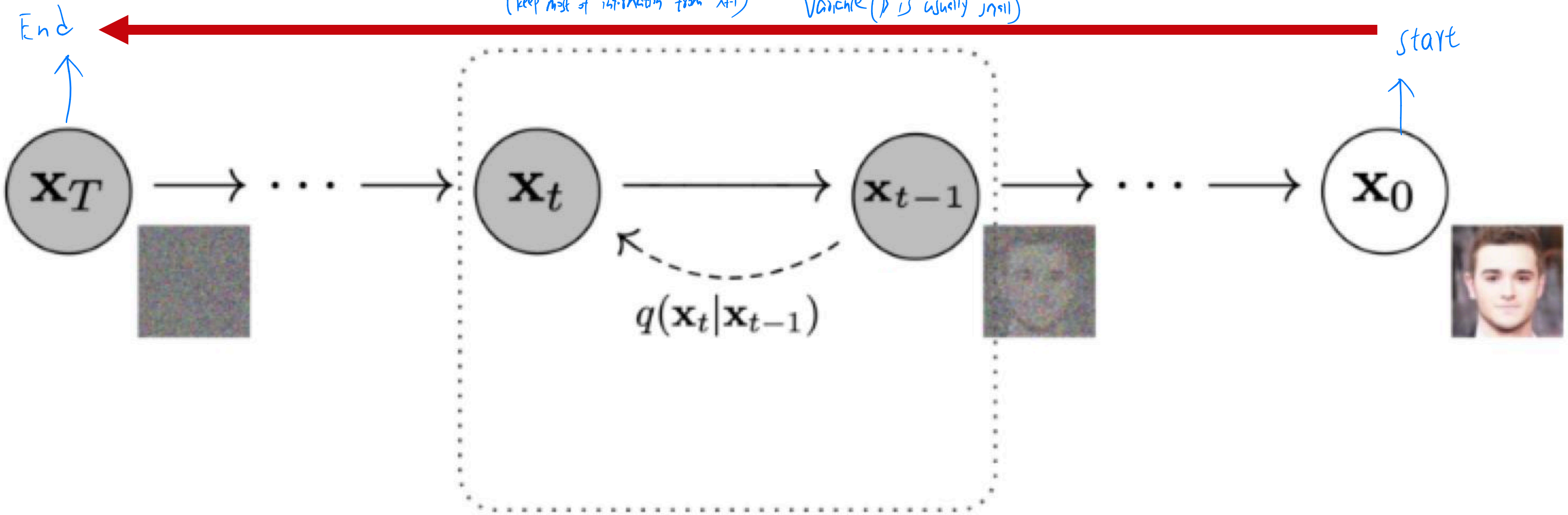
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \qquad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

Mean 中心

(keep most of information from x_{t-1})

波动

Variance (β is usually small)



Diffusion models: reverse pass

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

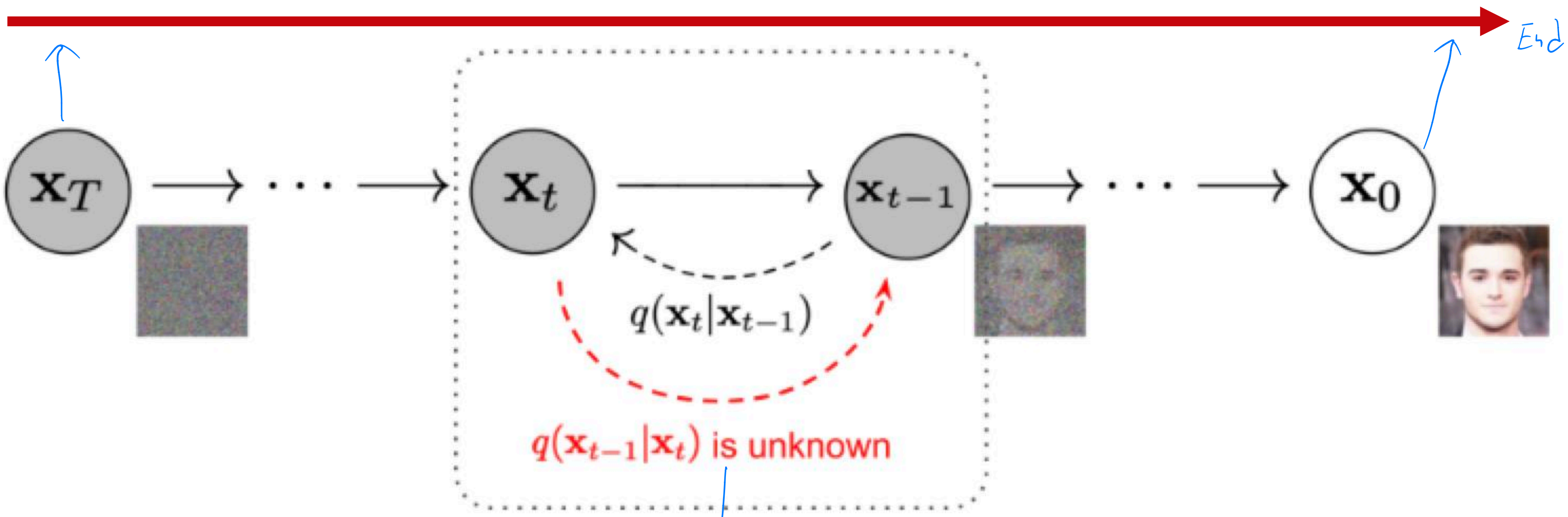
Whole generation chain from x_T to x_0

Assume also follow the normal distribution in backward pass

mean

variance

Start



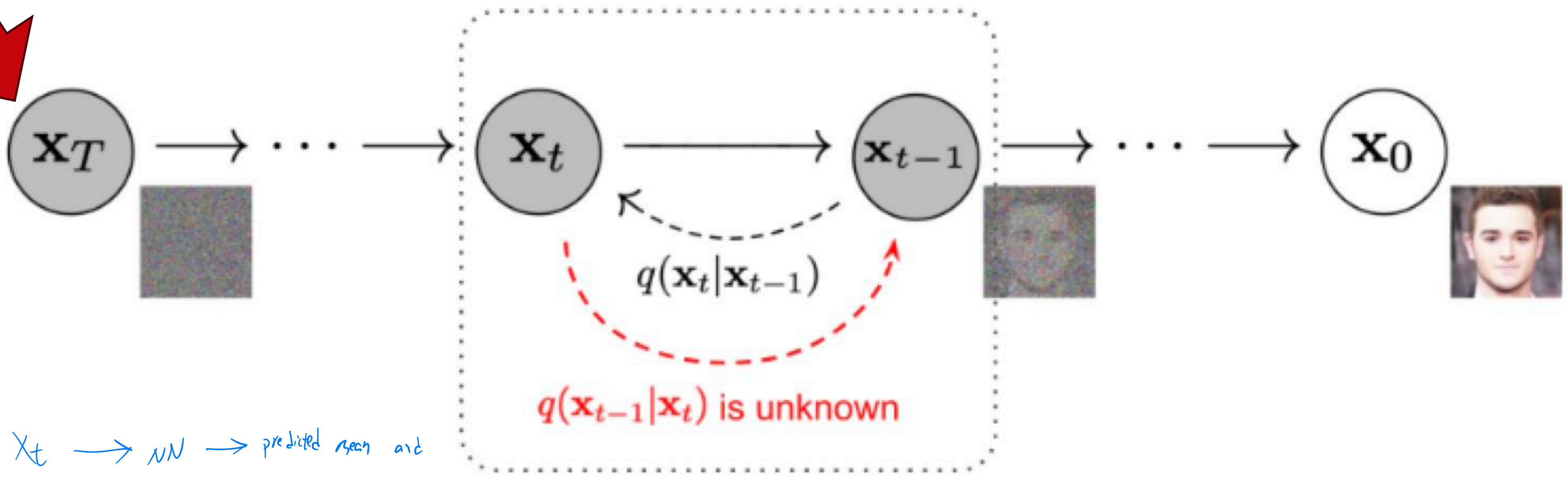
End

反向的是未知的

(Solution: guess by using neural network with parameters θ)

Diffusion models: generating a new sample

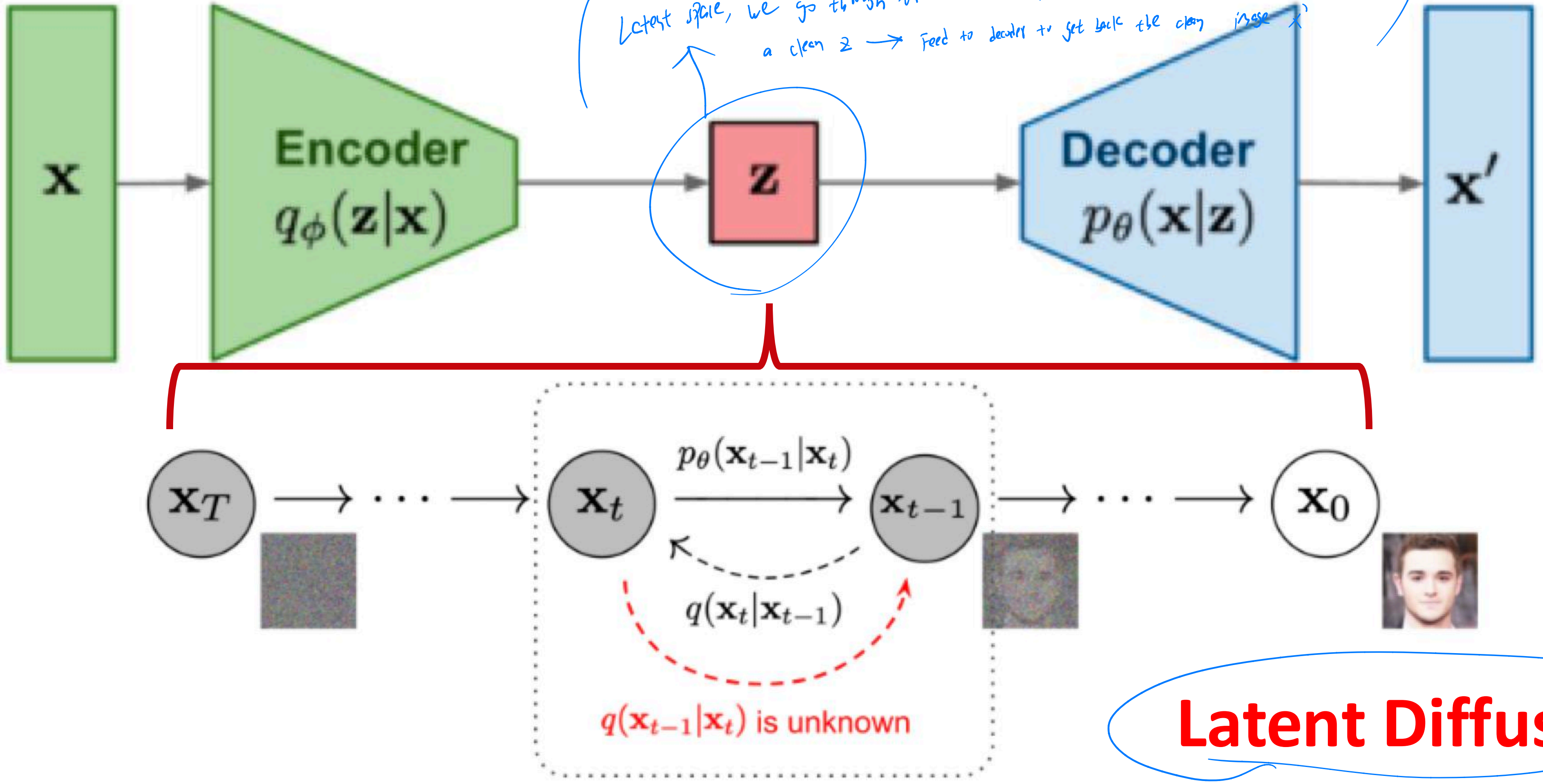
$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$



$X_t \rightarrow NN \rightarrow$ predicted mean and variance $\rightarrow$ Handle these and move to X_{t-1} ... $\rightarrow X_0$

Should we really run this process on pixels?

(Latent space, we go through the diffusion process on z (much smaller) and get a clean z → Feed to decoder to get back the clean image x')

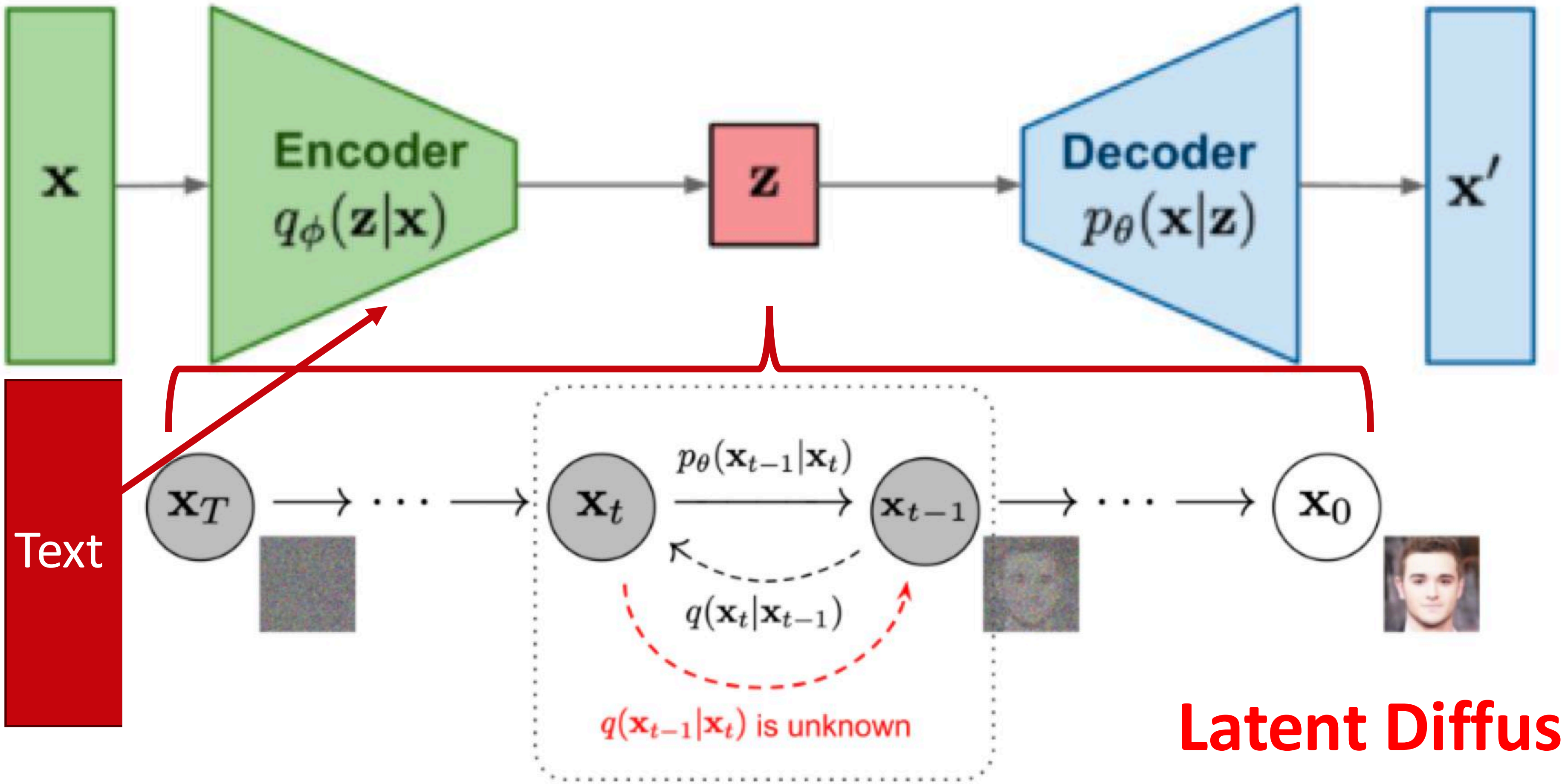


Latent Diffusion

☆ The previous pixel diffusion is too hard to compute for large images

→ (Latent diffusion combine the AE with diffusion model)

Stable Diffusion: Add Text Conditioning



Latent Diffusion

Previously, we have unconditional generation, you cannot what it generates

Now, with text, it things into command on how to denoise the picture and generate conditionally

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), Likelihood $p_\theta(x \mid z)$	Prior p(z), Generator $G_\theta(z)$	Fixed **forward noising** $q(x_t \mid x_{\{t-1\}})$; learn reverse $\mathrm{p}_\theta(x_{t-1} \mid x_t)$
Induced $p(x)$	$p_\theta(\mathrm{x}) = \int_z p_\theta(X \mid z)p(z)dz$	$p_\theta(\mathrm{x}) = \int_z p_\epsilon(x - \mathrm{G}_\theta(\mathrm{z}))\mathrm{p}(\mathrm{z})\mathrm{dz}$	$p_\theta(\mathrm{x}) = \int p(x_T) \prod_t p_\theta(x_{t-1} \mid x_t) dx$
Simplifying assumption	Choose a **restricted** variational posterior $q_\phi(z \mid x)$	**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: $E_q[\log p_\theta(x \mid z)] - KL(q_\phi(z \mid x) \| p(z))$	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 $p_\theta(x)$	$KL(q_\phi(z \mid x) \| p_\theta(z \mid x))$	**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?

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON