High Resolution Image Synthesis with Latent Diffusion Models (LDM)

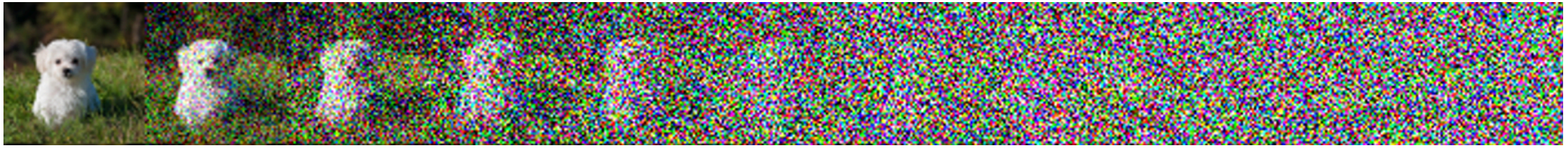
Rombach et al. 2022

Reviewed & Presented by Joon Hyeok Kim

Contents

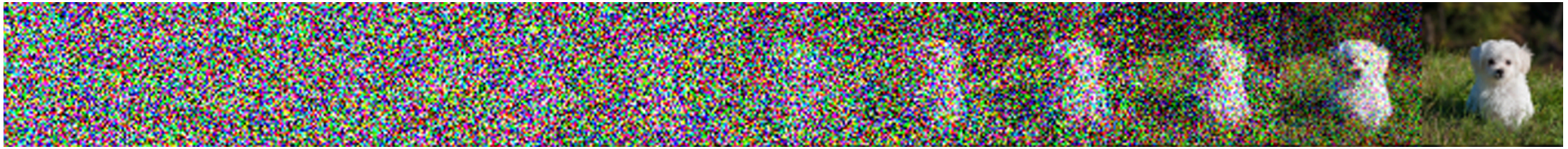
Recap : Denoising Diffusion Probabilistic Model (DDPM)

Forward Process $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$: $\underbrace{\mathbf{x}_0}_{\text{original image}} \rightarrow \mathbf{x}_1 \rightarrow \cdots \rightarrow \mathbf{x}_{T-1} \rightarrow \underbrace{\mathbf{x}_T}_{\text{pure noise!}}$



- What we choose by scheduling β_t

Reverse Process $p_\theta(\mathbf{x}_t \mid \mathbf{x}_{t-1})$: $\underbrace{\mathbf{x}_T}_{\text{pure noise}} \rightarrow \mathbf{x}_1 \rightarrow \cdots \rightarrow \mathbf{x}_{T-1} \rightarrow \underbrace{\mathbf{x}'_0}_{\text{synthetic image!}}$



- A neural network with parameters θ that we want to learn!

Optimizing DDPM

1. Use Bayes Rule to get posterior $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0) = \dfrac{q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) q(\mathbf{x}_{t-1} \mid \mathbf{x}_0)}{q(\mathbf{x}_t \mid \mathbf{x}_0)}$

2. Maximize ELBO : $\log p_\theta(\mathbf{x}0) \geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \dfrac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)} \right]$

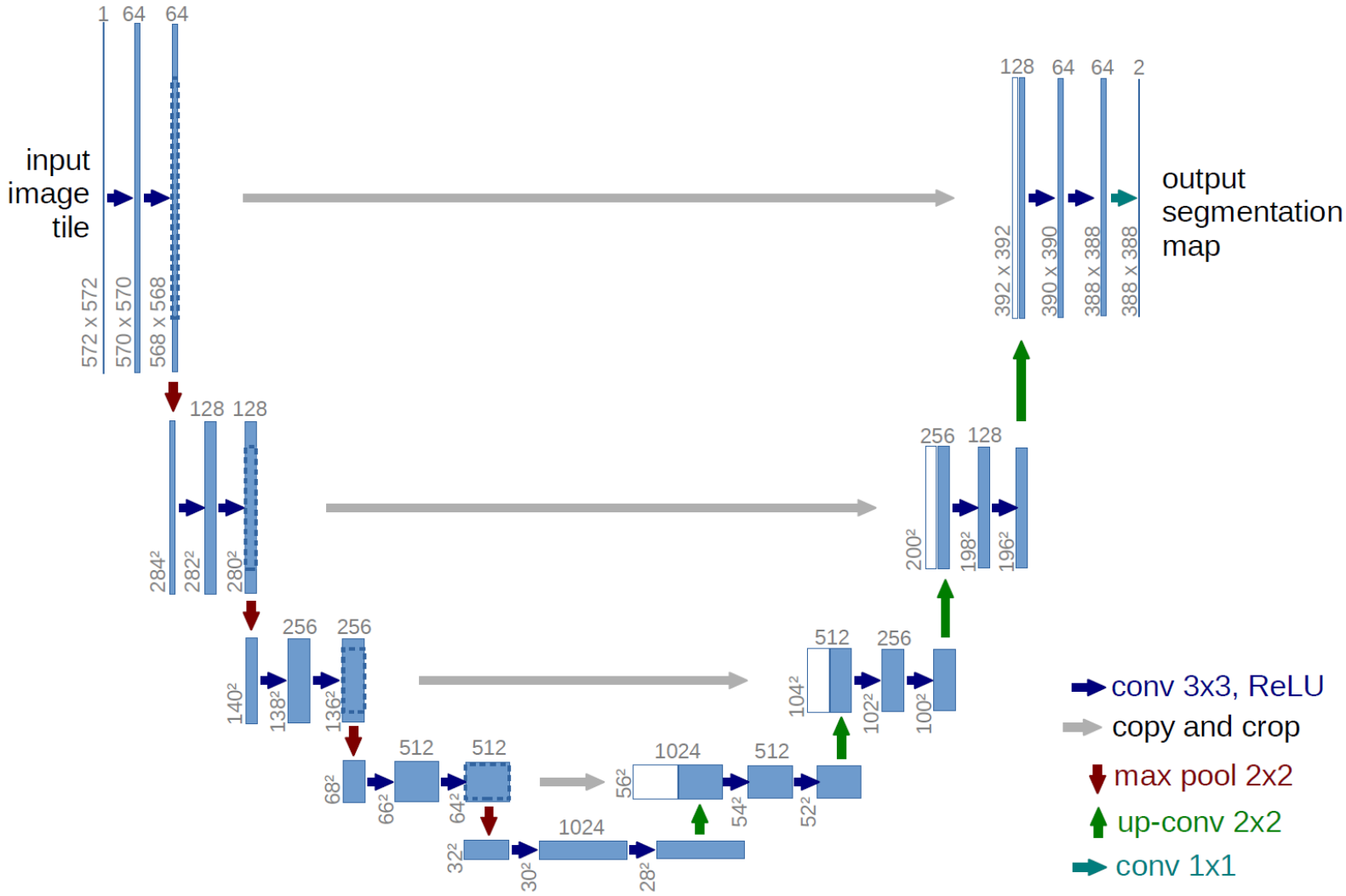
   - Equivalent to minimizing KL-Divergence (L_{vlb})

3. Authors found that this was equivalent to estimating the noise that we added

   - $L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}} \left[\left\| \underbrace{\boldsymbol{\epsilon}}_{\text{noise!}} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2 \right]$

cf.) Later improved to $L_{\text{hybrid}} = L_{\text{simple}} + \lambda L_{\text{vlb}}$ to achieve competitive log-likelihood!

Implementing DDPM : UNet Architecture



Weakness 1 : Computationally costly

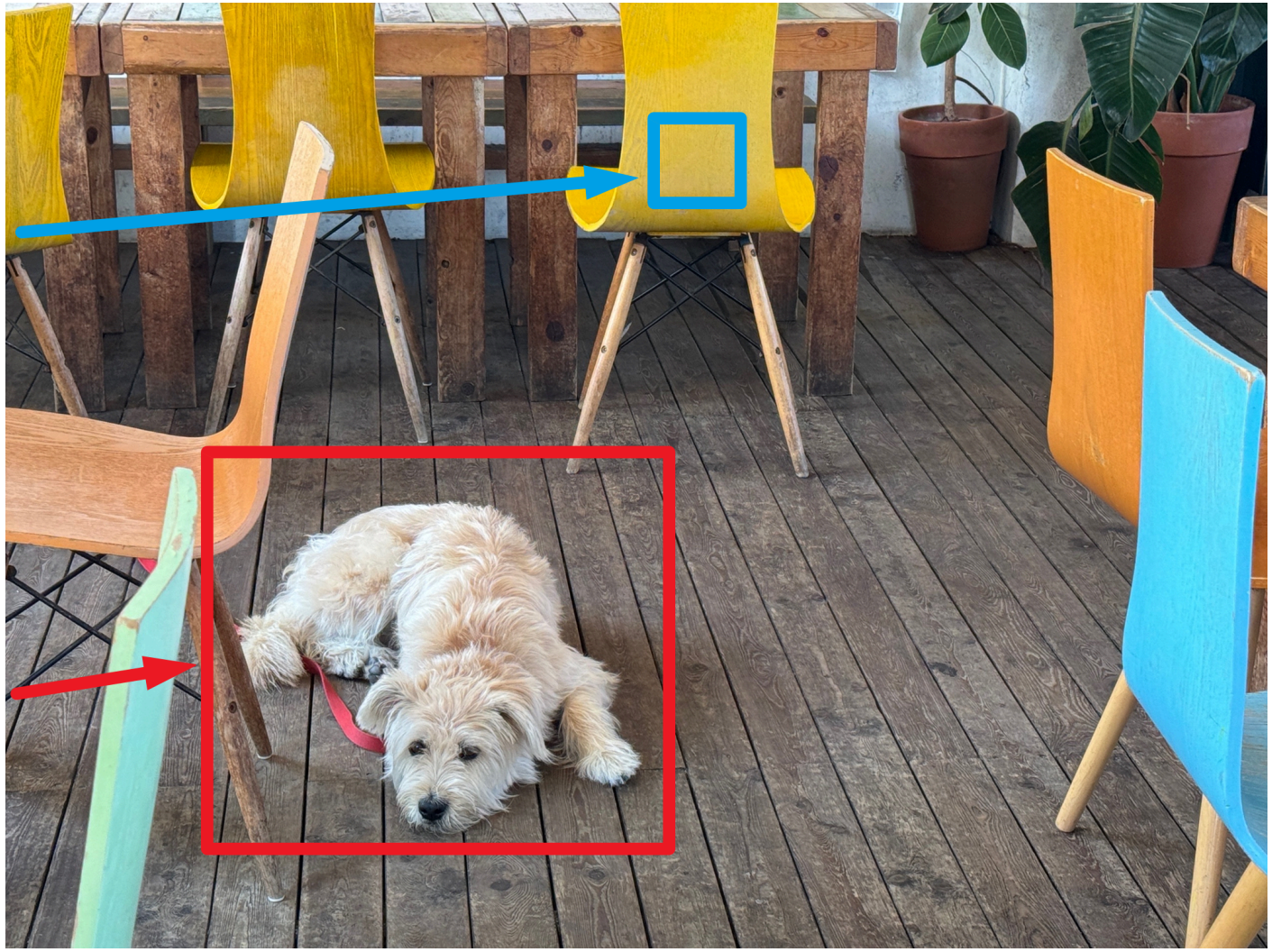
Pixel Space Dimension $D = H \times W \times \underbrace{3}_{\text{RGB}}$

$D = 512^2 \times 3 = 786,432$ for a single image!

Weakness 2 : Waste of resources on learning imperceptible details

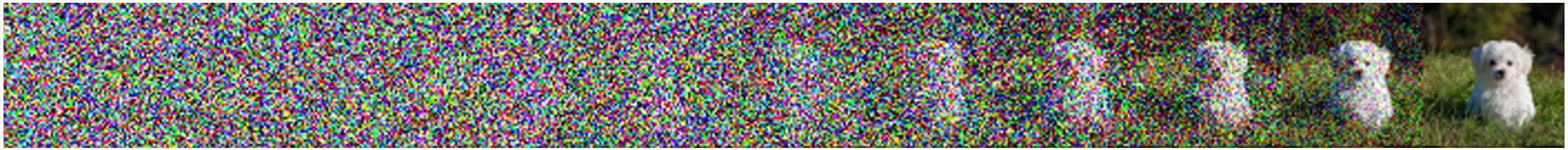
What model cares

What we care



Weakness 3 : Slow Sampling due to Markov Chain Structure

Reverse Process

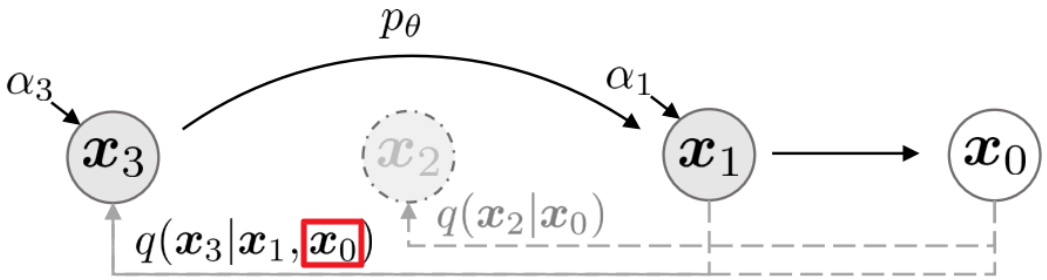


- $p_\theta(\mathbf{x}_t \mid \mathbf{x}_{t-1})$: $\underbrace{\underbrace{\mathbf{x}_T}_{\text{pure noise}} \rightarrow \mathbf{x}_1 \rightarrow \cdots \rightarrow \mathbf{x}_{T-1} \rightarrow \underbrace{\mathbf{x}'_0}_{\text{synthetic image!}}}_{T \text{ steps!}}$
  - At each step, we sample $\mathbf{x}_{t-1} \sim \mathcal{N}(\mathbf{x}_{t-1};\ \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$

Idea 1: What if we lower the dimension and perform Diffusion?

1. Encode image to the Latent Space (VAE)
2. Perform diffusion in the Latent Space (Diffusion Model)
3. Decode image back to the Pixel Space (VAE)

Idea 2: Skip some sampling steps using DDIM



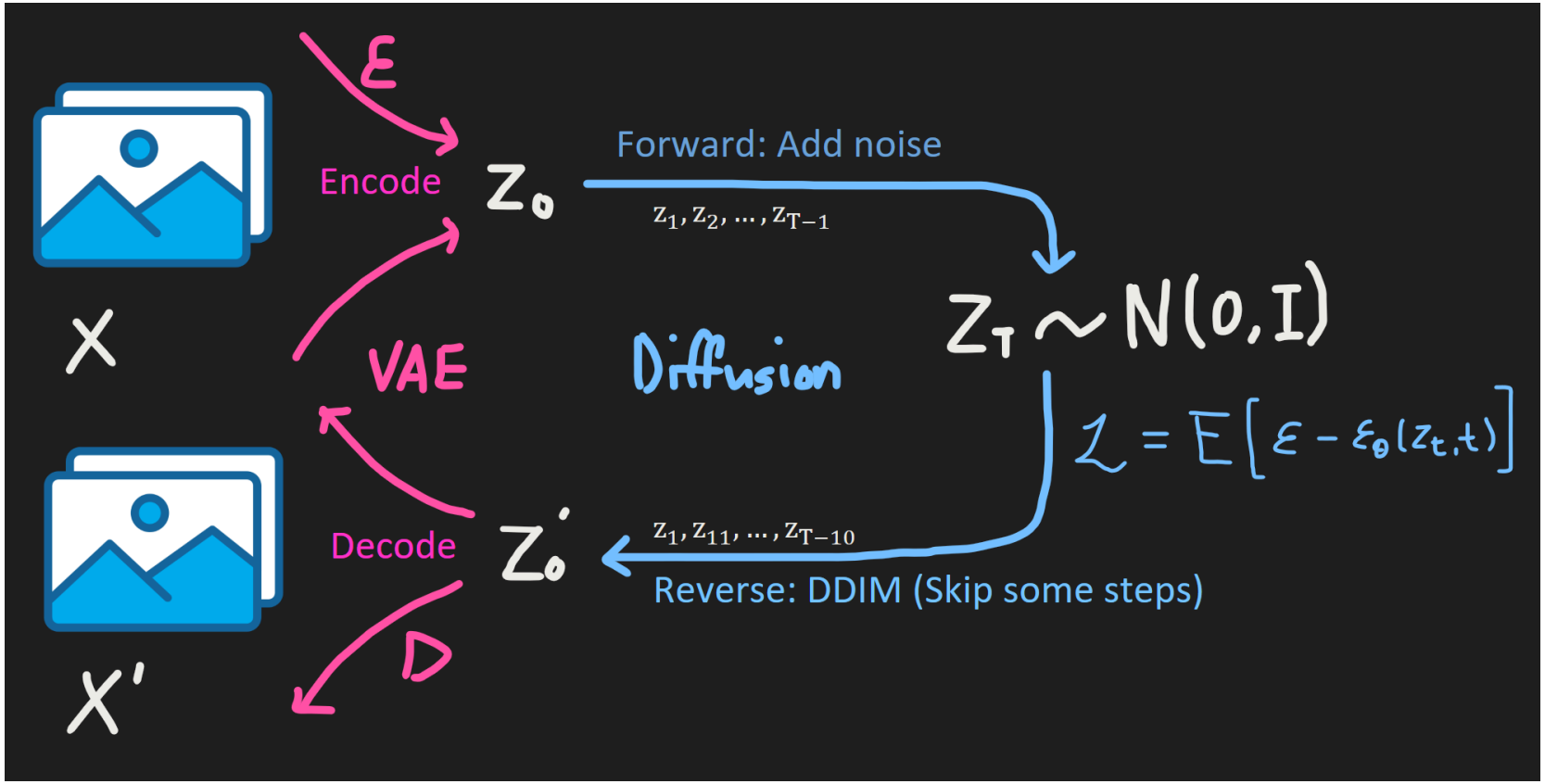
Use the directional info to the final answer!

Figure 2: Graphical model for accelerated generation, where $\tau = [1, 3]$.

Model Structure : Perceptual Compression × Latent Diffusion

VAE : Perceptual Compression Model

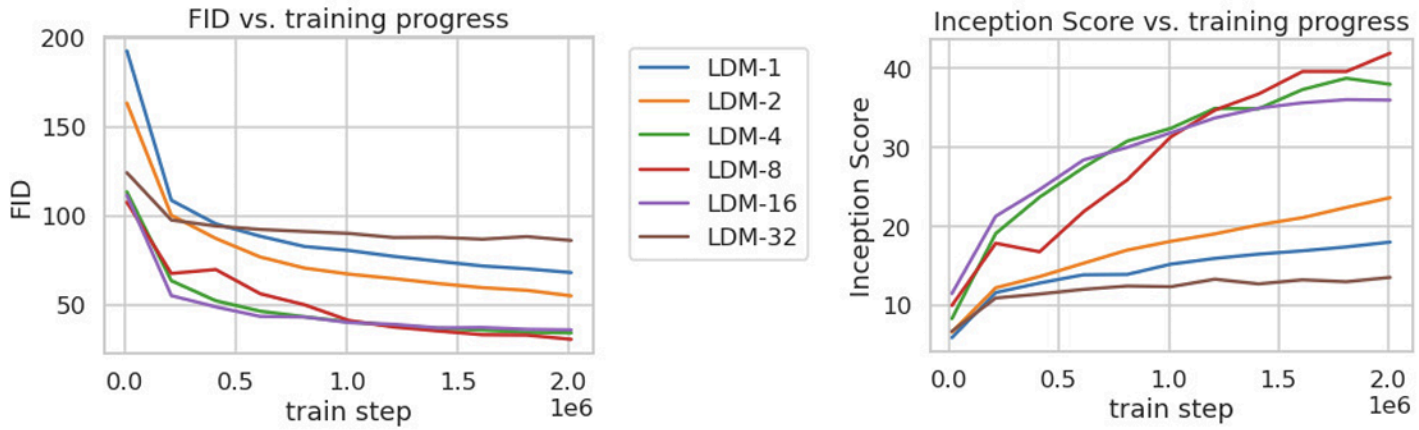
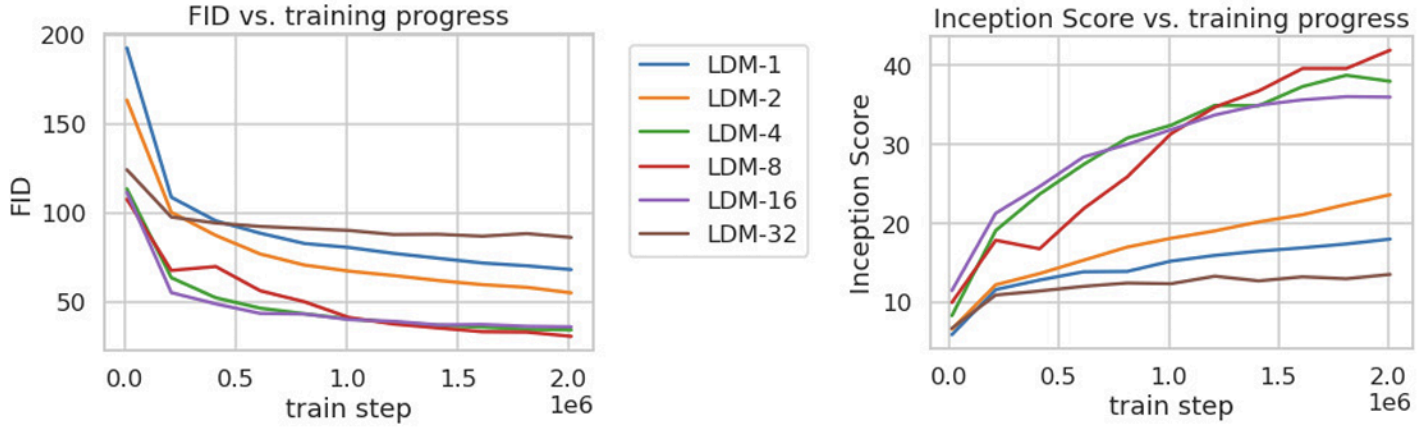
Diffusion : Latent Diffusion Model



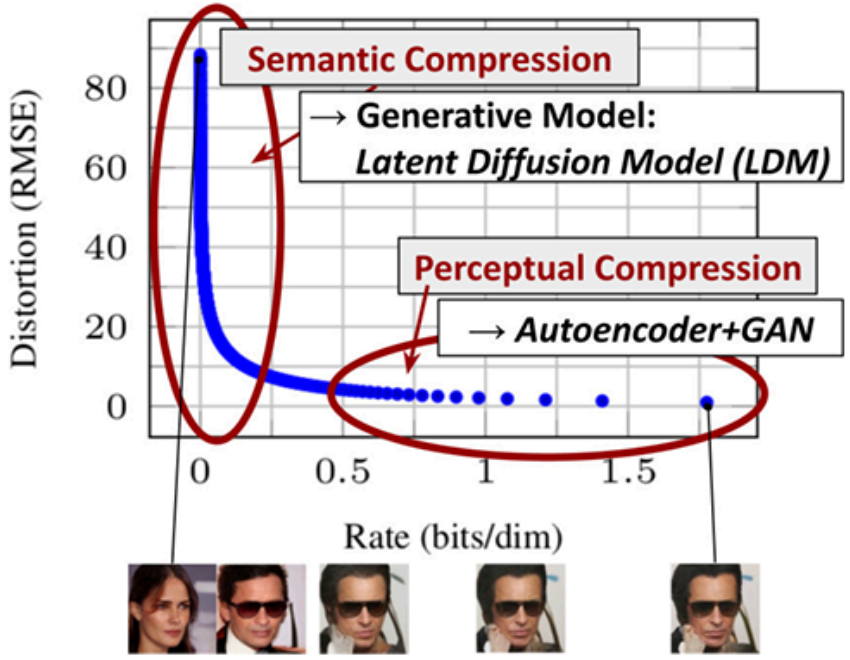
Result on Unconditional Image Generation

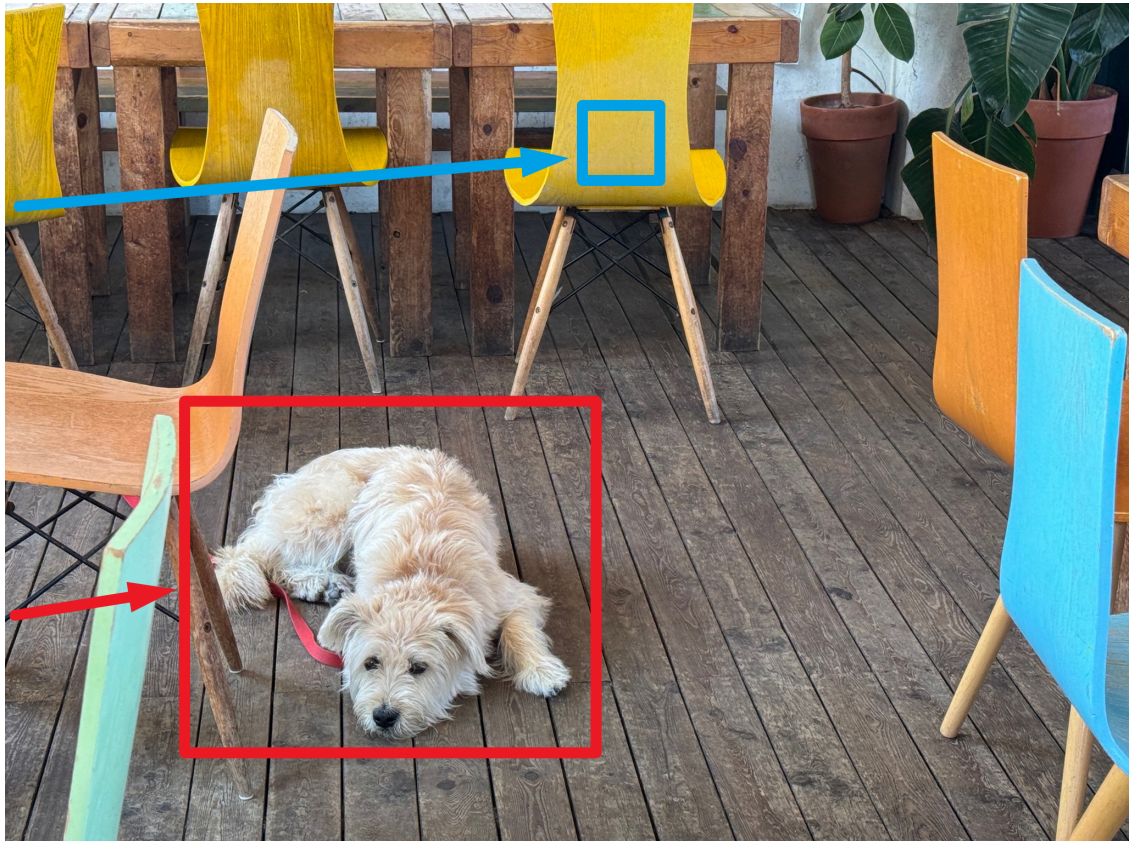
DDPM (2020)	LDM (2022)
	

Advantage 1 : Improved Image Quality & Faster Sampling

	LDM-fs with different downsampling factors $f \in \{1, 2, 4, 8, 16, 32\}$
Improved Quality	
Faster Sampling	

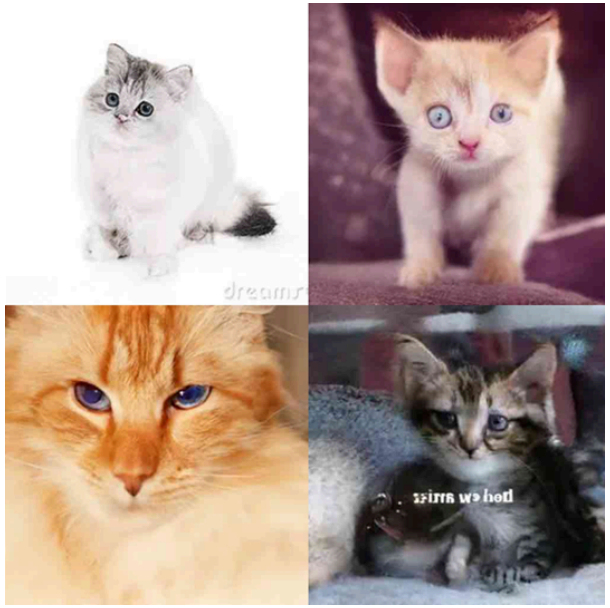
How? Teamwork maybe? 👫





Is that it? Not even close.

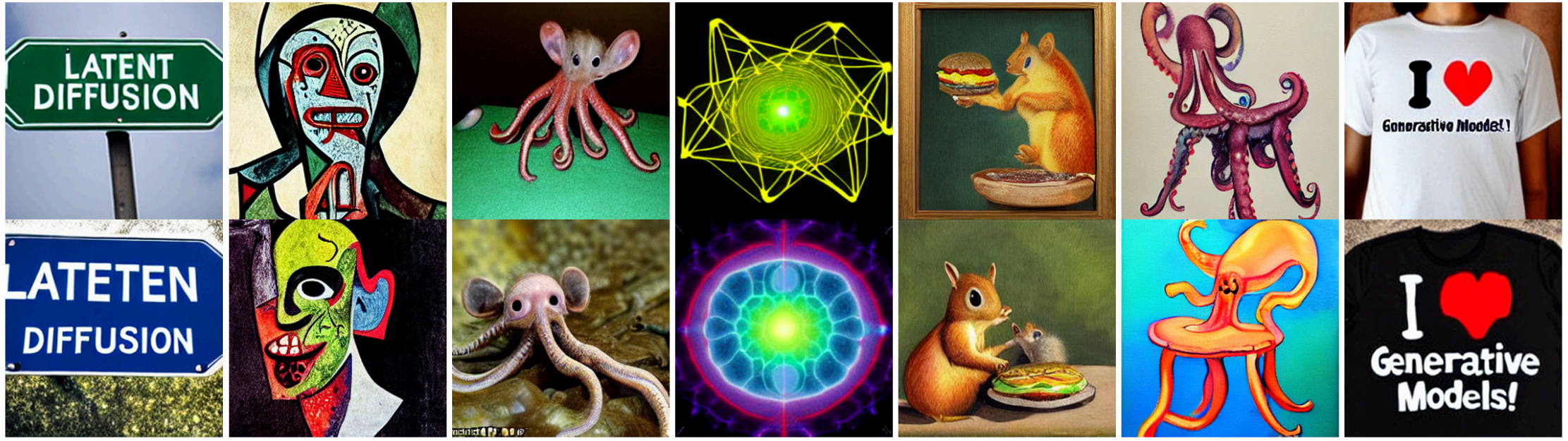
Conditional generation in previous diffusion models...

"Cat" in DDPM	"Cat" in CFG	
		Pre-defined **labels** on the dataset

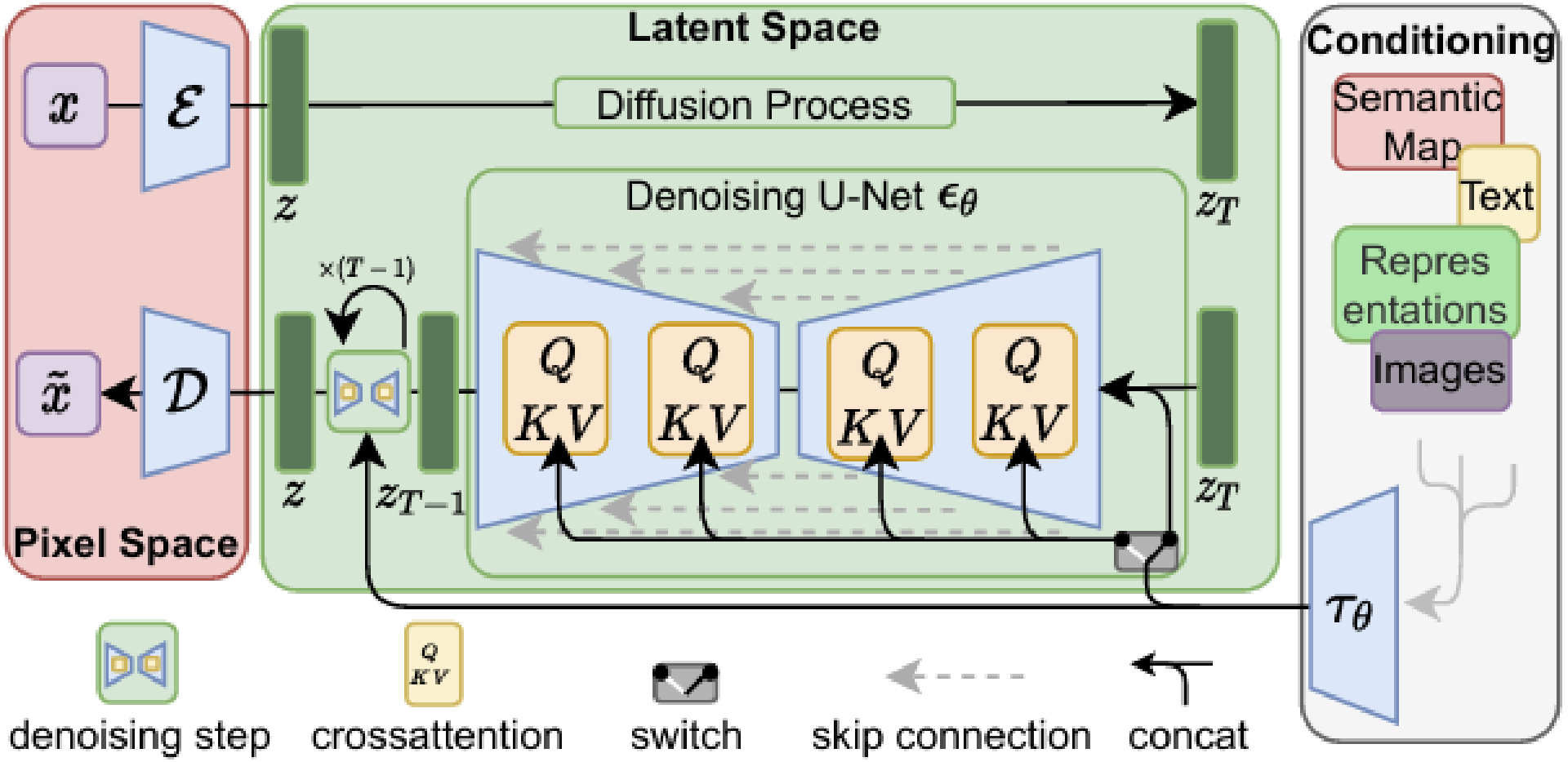
Phenomenal Shift : Expressive Conditional Image Generation

Text-to-Image Synthesis on LAION. 1.45B Model.

'A street sign that reads "Latent Diffusion" '	*'A zombie in the style of Picasso'*	*'An image of an animal half mouse half octopus'*	*'An illustration of a slightly conscious neural network'*	*'A painting of a squirrel eating a burger'*	*'A watercolor painting of a chair that looks like an octopus'*	*'A shirt with the inscription: "I love generative models!" '*

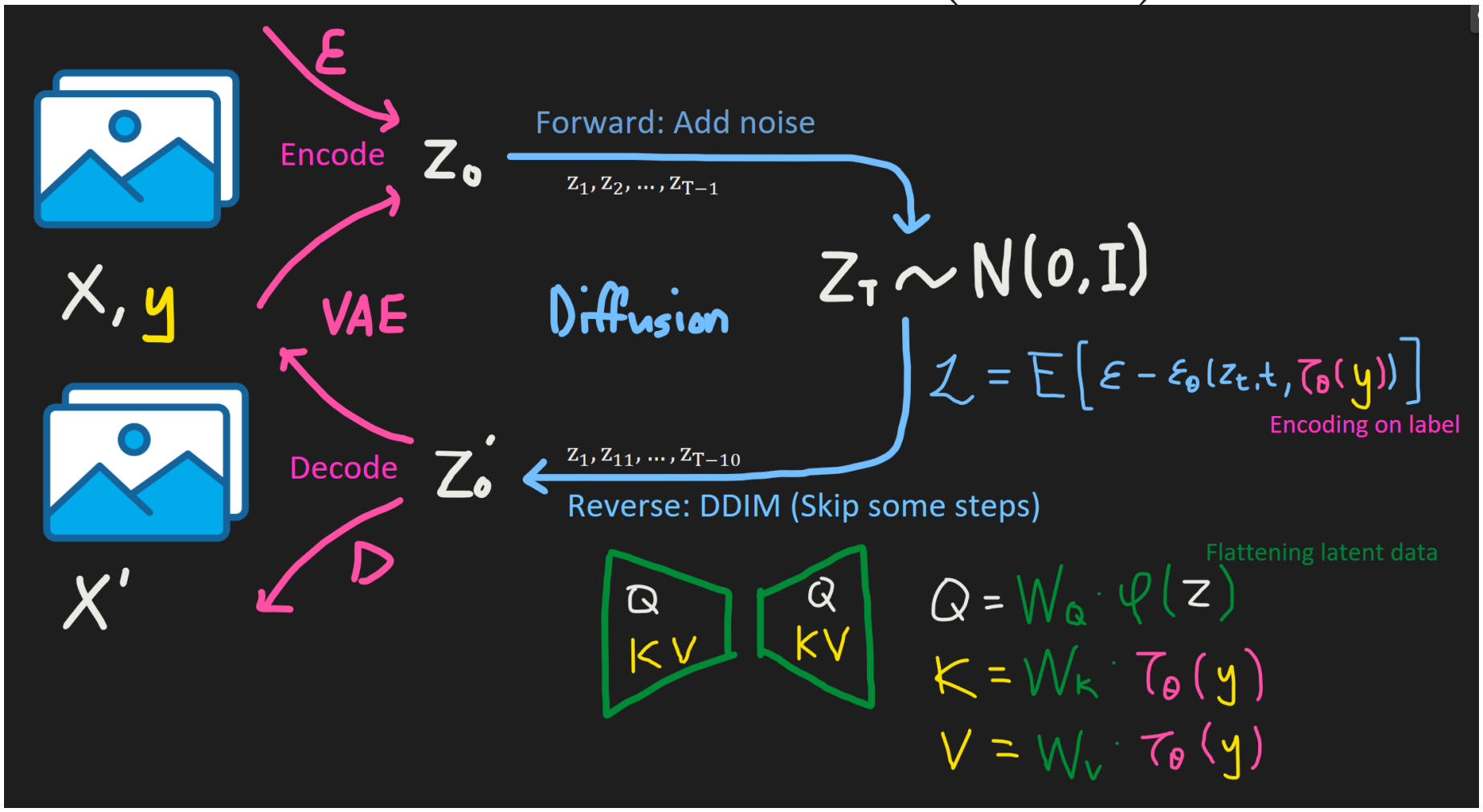


Upgrade : Conditional LDM = DDPM × VAE × (Cross) Attention



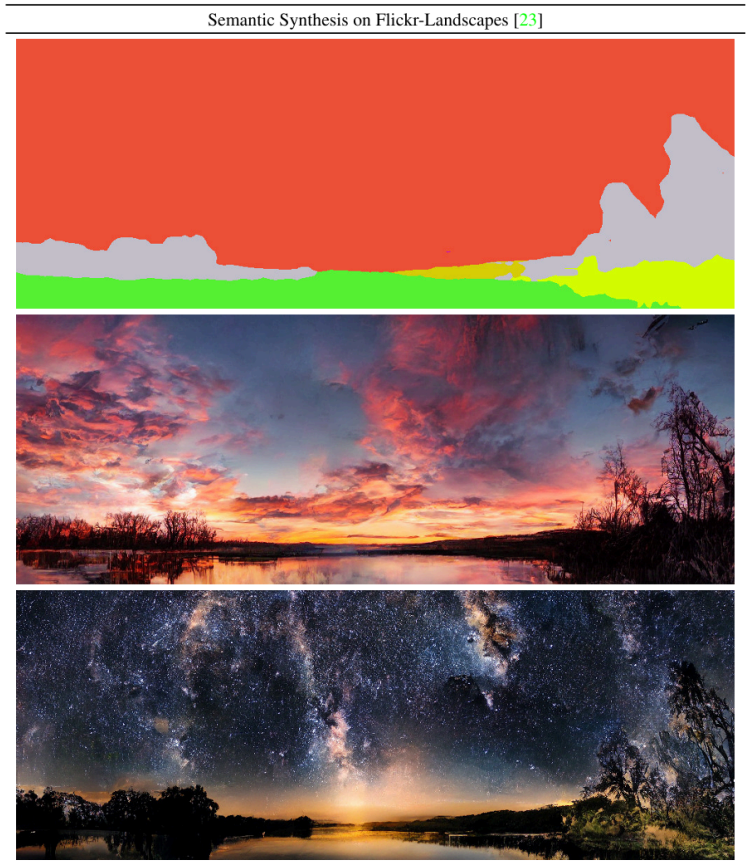
Cross Attention

$$\text{Attention}^{(i)}\left(Q^{(i)}, K^{(i)}, V^{(i)}\right) = \text{Softmax}\left(\frac{Q^{(i)}{K^{(i)}}^{\top}}{\sqrt{d}}\right) \cdot V^{(i)}$$



Other capabilities...

Semantic Synthesis



Upscaling

In-painting

Pros & Cons of LDM

Strengths

Improvement in image quality compared to previous Diffusion Models.

Faster sampling speed compared to previous Diffusion Models.

- Key bottleneck of the Diffusion Models

Expressive conditional image generation

- Thanks to the Cross-Attention Mechanism

Versatile capabilities

- Expressive T2I, Semantic Synthesis, Upscaling, In-painting

Pros & Cons of LDM

Weaknesses

Still, slow sampling speed compared to other models

- **GANs** are way faster. (But Diffusion Models are more stable!)

Questionable when high precision is required

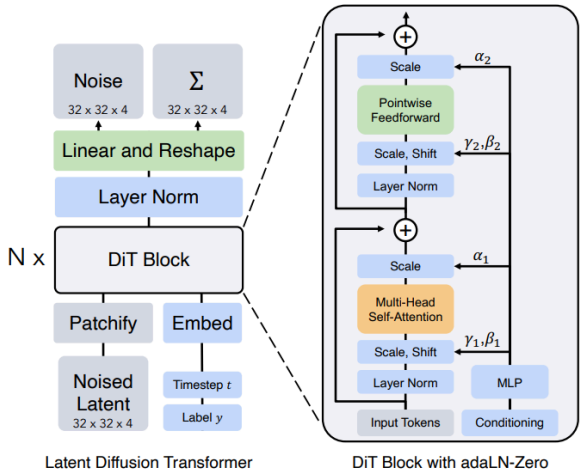
- Why?
  - Some information compressed by the VAE may not be recovered.
  - Stochastic nature of the Diffusion Models (Then, deterministic? Flow?)

High training cost

- Inference is relatively cheaper though...

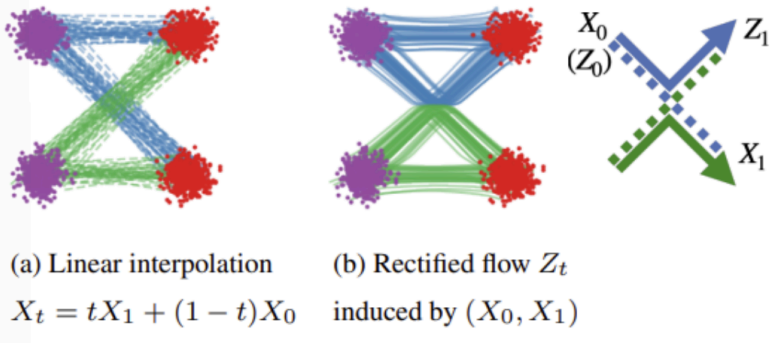
Updates: LDM → Stable Diffusion (SD) → SD2 →SD3

Spoiler : $\underbrace{\text{LDM}}_{\text{Now!}} + \underbrace{\text{DiT}}_{\text{Next}} + \underbrace{\text{Rectified Flow}}_{\text{Next week}} = \underbrace{\text{Stable Diffusion 3}}_{\text{3 weeks later}}$



Latent Diffusion Transformer

DiT Block with adaLN-Zero



(a) Linear interpolation $X_t = tX_1 + (1-t)X_0$

(b) Rectified flow Z_t induced by (X_0, X_1)

Dystopia of thousand of workers picking cherries and feeding them into a machine that runs on steam and is as large as a skyscraper. Written on the side of the machine: "SD3 Paper"

Questions

Thank you