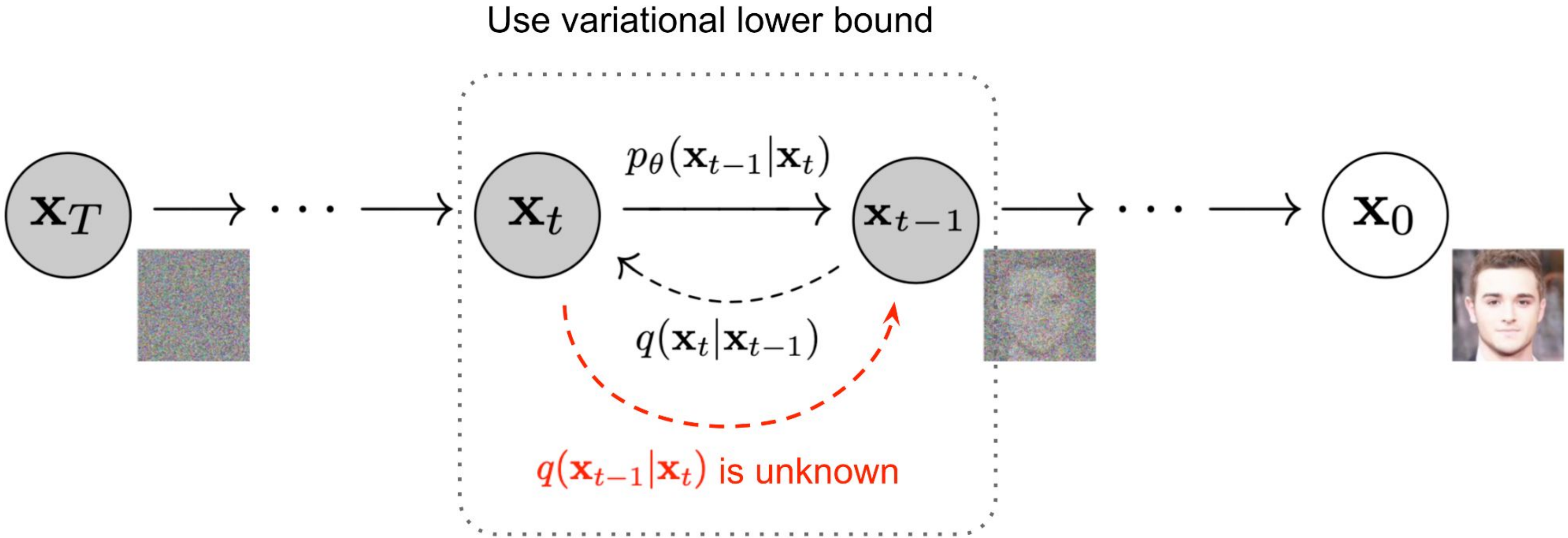
Agenda

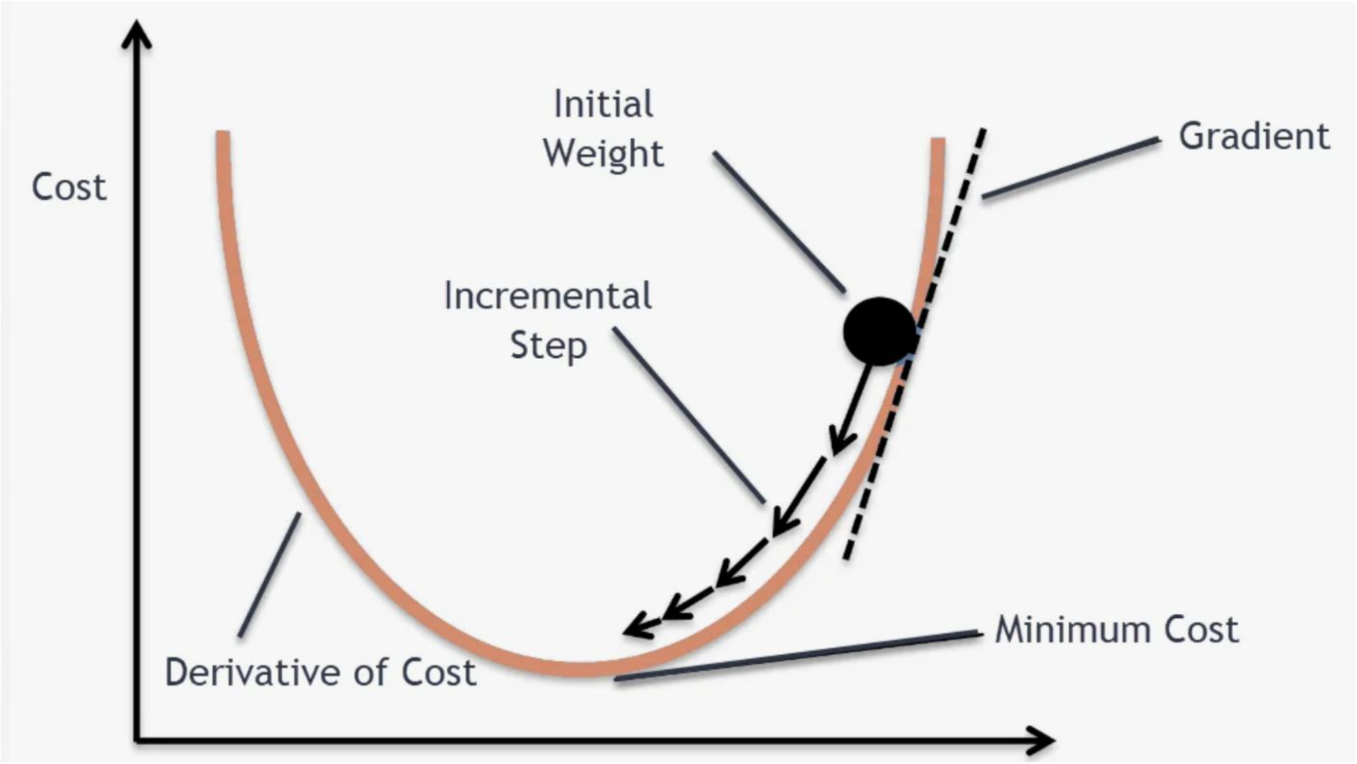
1. What are Diffusion Models?



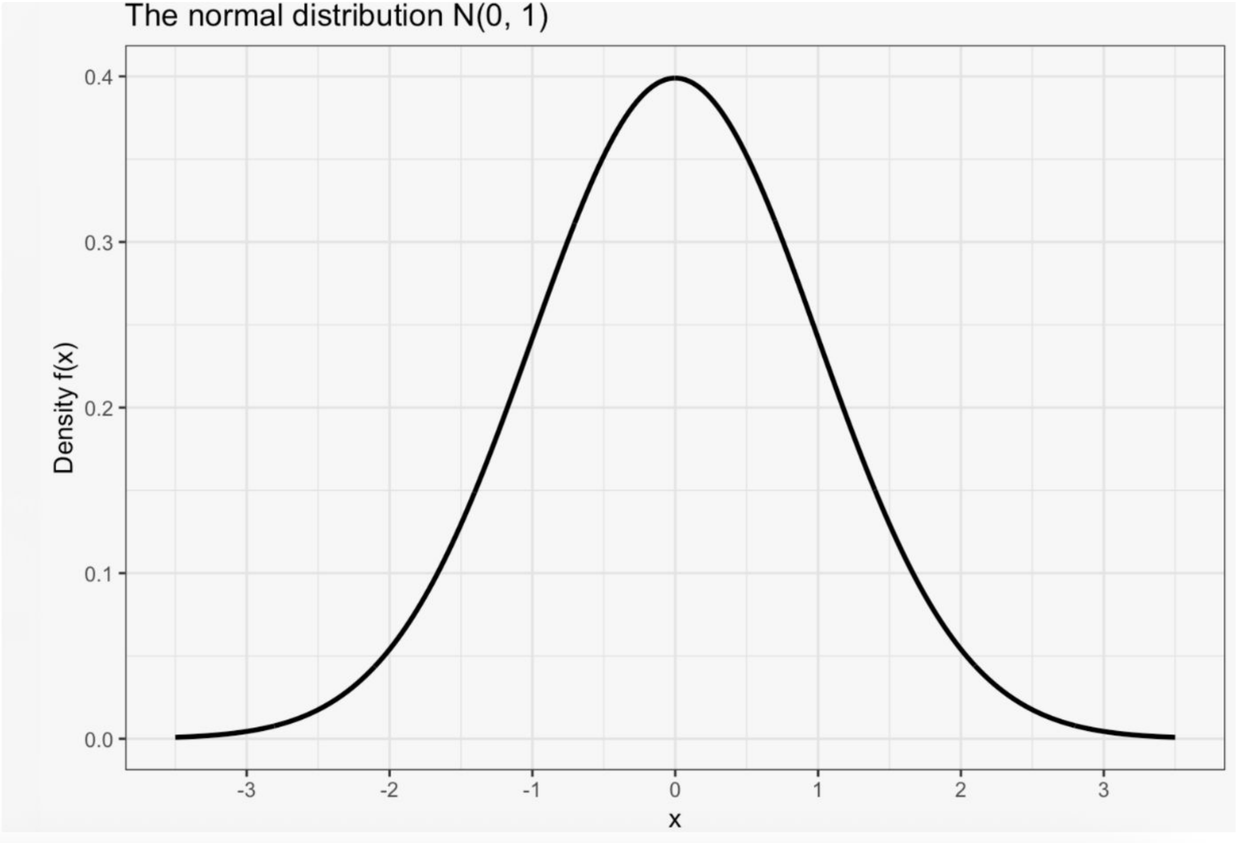
- Diffusion models transform noise into sample from target data distribution.
- Model outputs are of the same shape (B, H, W, C) as inputs.

The origins of Diffusion Models comes from Langevin Dynamics.

Stochastic Gradient Descent $x_{j+1} = x_j - \alpha \nabla_x L(x),\ x_0 \sim \mathcal{N}(x)$



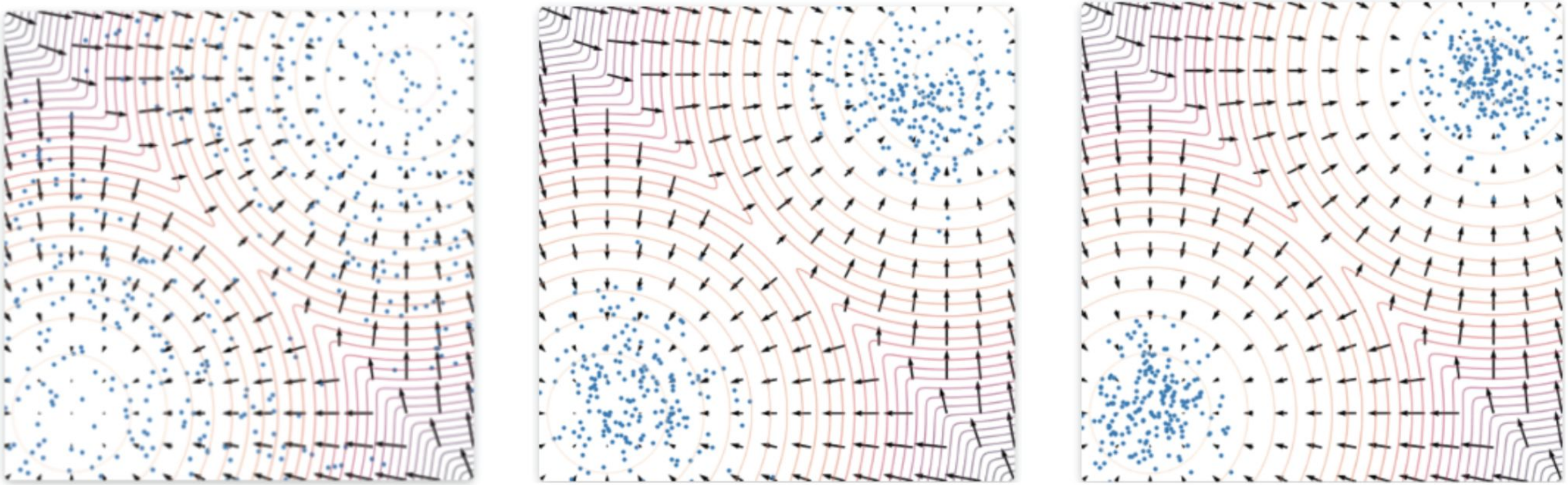
We can consider the density function as an optimization function



The goal is to find x that maximizes $p_{data}(x)$, exactly the same as in SGD. But for high dimensional data, like images.

$$x_{j+1} = x_j + \alpha \nabla_x \log p(x), \; x_0 \sim \mathcal{N}(x|0, 1).$$

$\nabla_x \log p(x)$ – score function



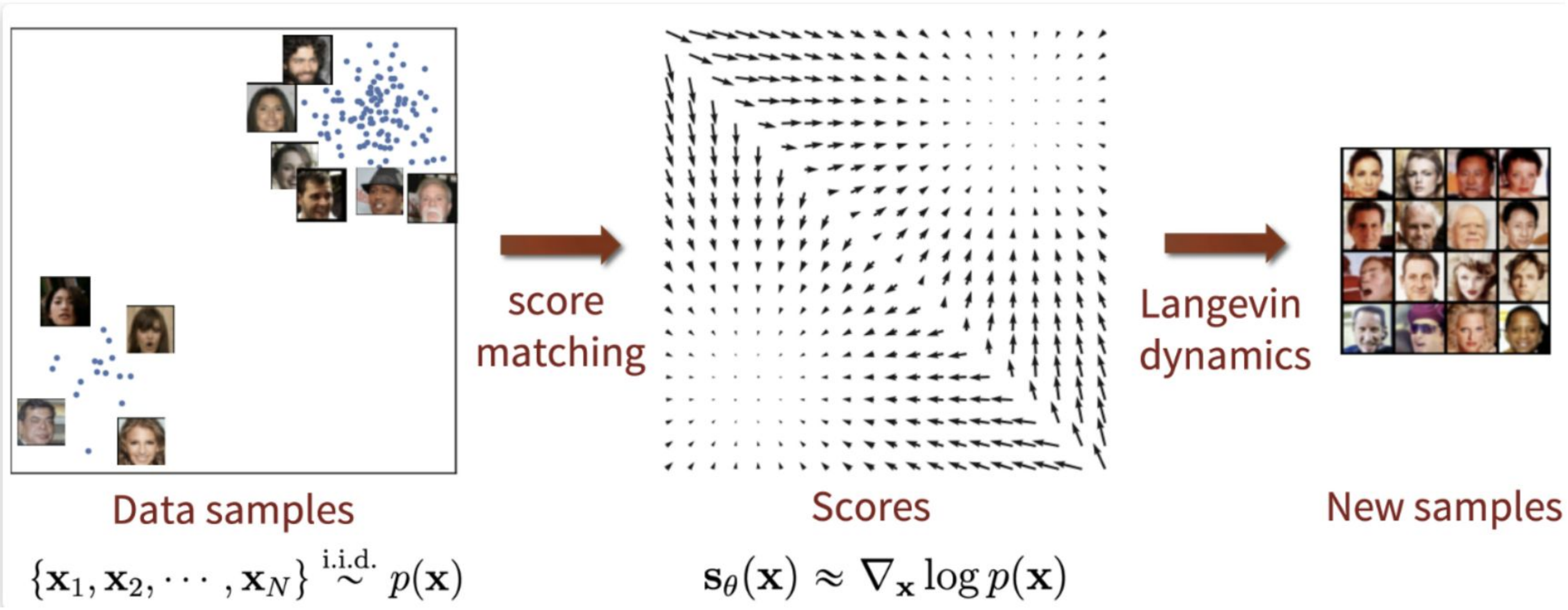
But how to get a score function? We only have a dataset.

This is **the most important question** in diffusion models.

$$\|s_\theta(x) - \nabla_x \log p_{\text{data}}(x)\| - ?$$

We will not dive deeper into this question

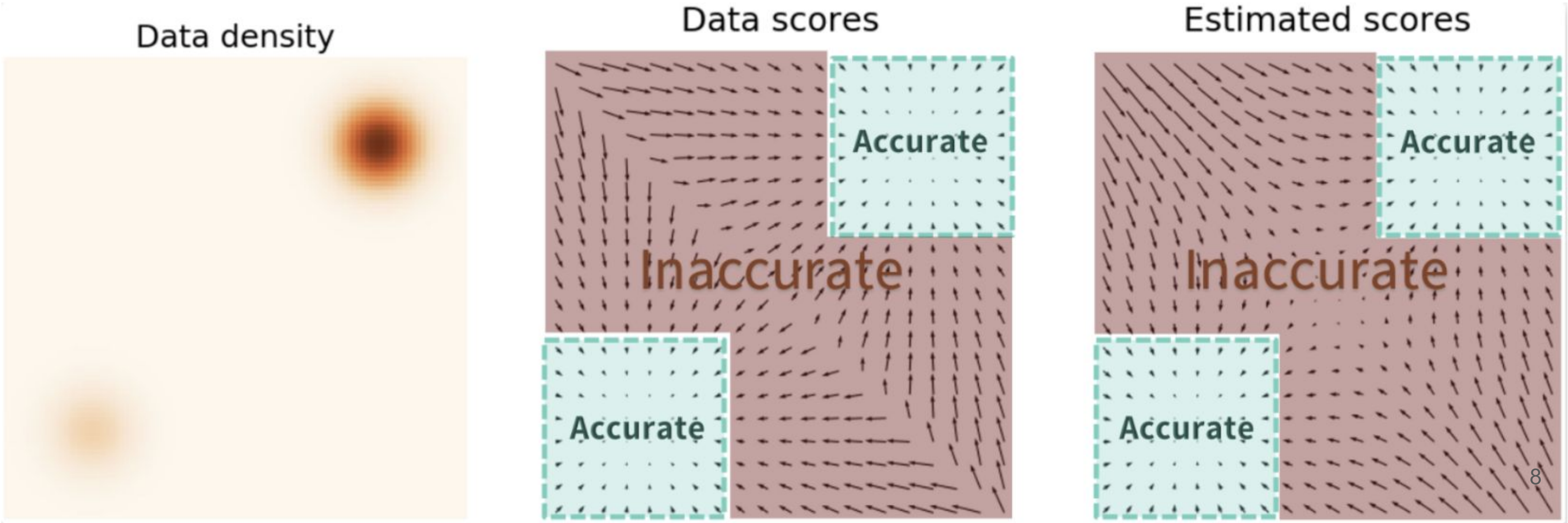
$$\|s_\theta(x) - \nabla_x \log p_{\text{data}}(x)\|$$
$$\sim \operatorname{tr}(\nabla_x s_\theta(x)) + \frac{1}{2}\|s_\theta(x)\|^2$$


score
matching
Langevin
dynamics
Data samples
$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$
Scores
$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_\mathbf{x} \log p(\mathbf{x})$
New samples

When sampling with Langevin dynamics, our initial sample is likely in low density regions

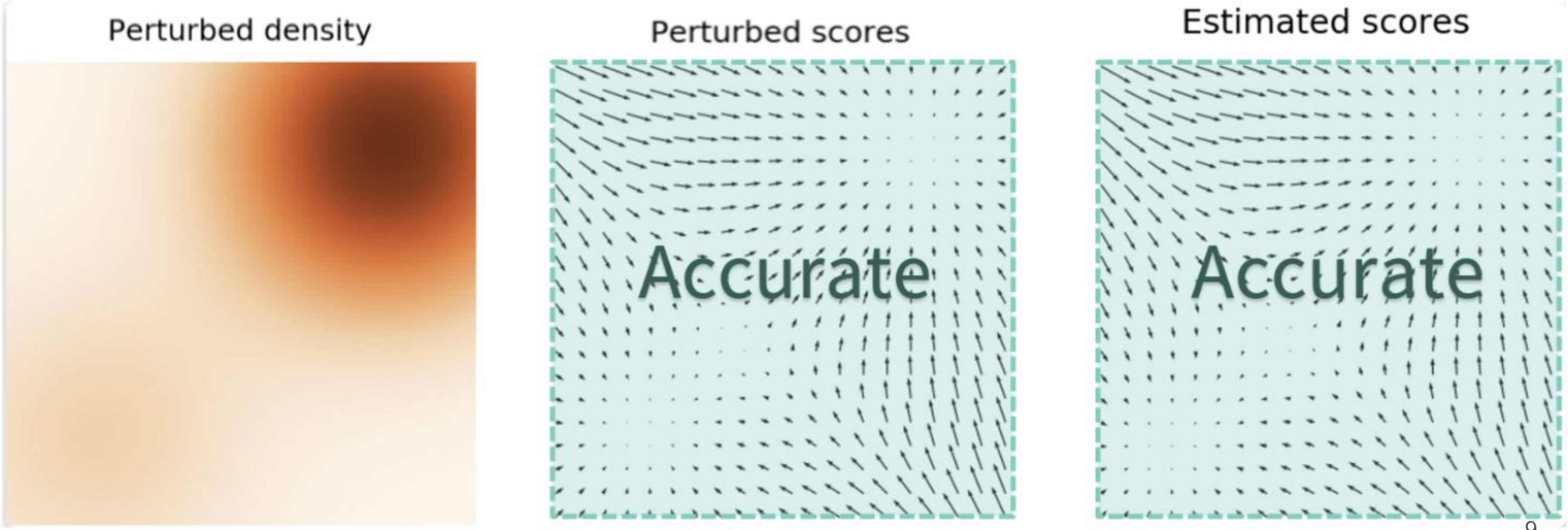
$$\mathbb{E}_{x \sim p_{\text{data}}} \| s_\theta(x) - \nabla_x \log p_{\text{data}}(x) \| \to \min_\theta$$

$$x_{j+1} = x_j + \alpha \nabla_x \log p(x), \quad \boxed{x_0 \sim \mathcal{N}(x|0,1).}$$

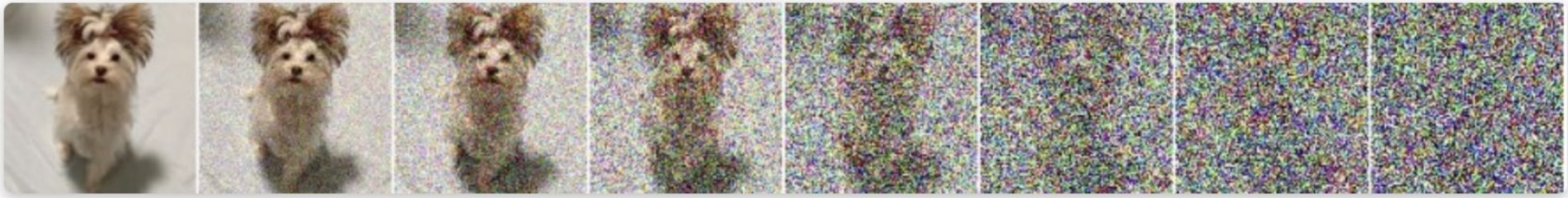


Solution: Lets noise our data and learn the score for noised samples

$$\nabla_x \log p(x) \to \nabla_x \log p(x, \sigma),$$



We noise our data for different σ.



$$x_{j+1} = x_j + \alpha s_\theta(x_j, \sigma_j),$$
$$x_0 \sim \mathcal{N}(x|0,1), \sigma_0 = 1, \sigma_N = 0.001$$

2. Diffusion Models Architectures

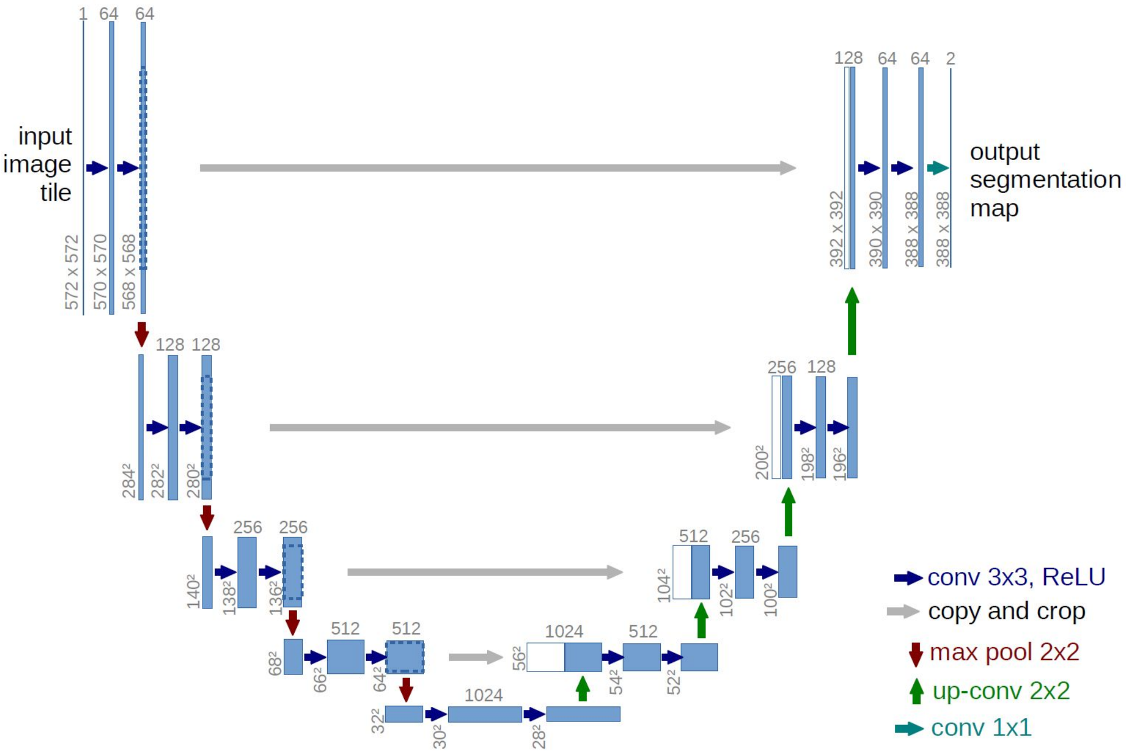
UNet?

Transformer?

SSM?

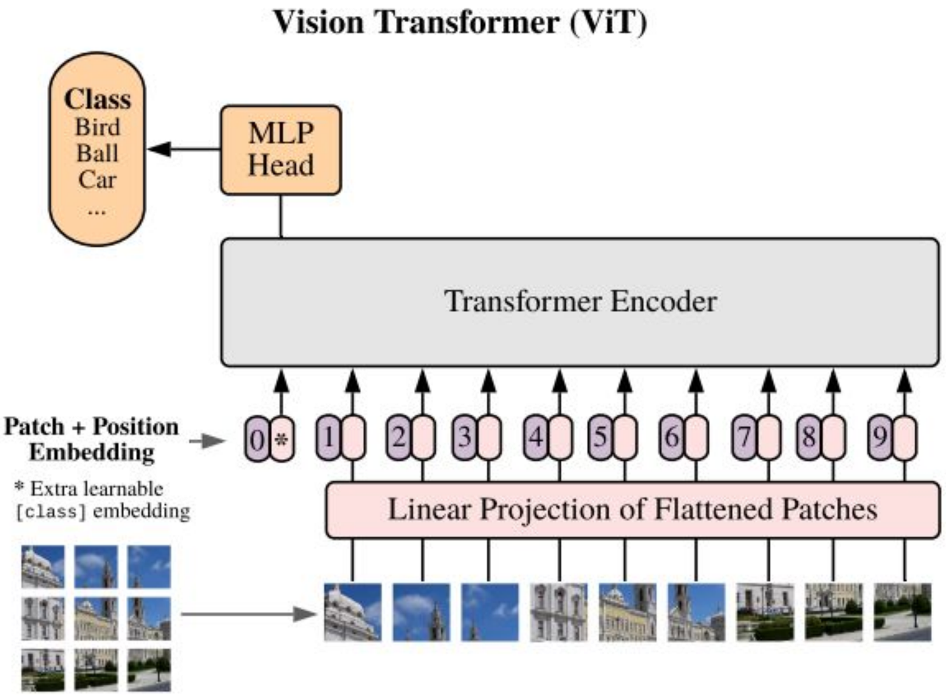
UNet

- The most popular architecture is old fellow **UNet**, familiar from the segmentation task.
- UNet is a stack of Residual Convolutional Blocks operating on different resolutions.
- Most diffusion models also include Attention blocks.
- Deeper layers have more channels and operate on smaller features maps.

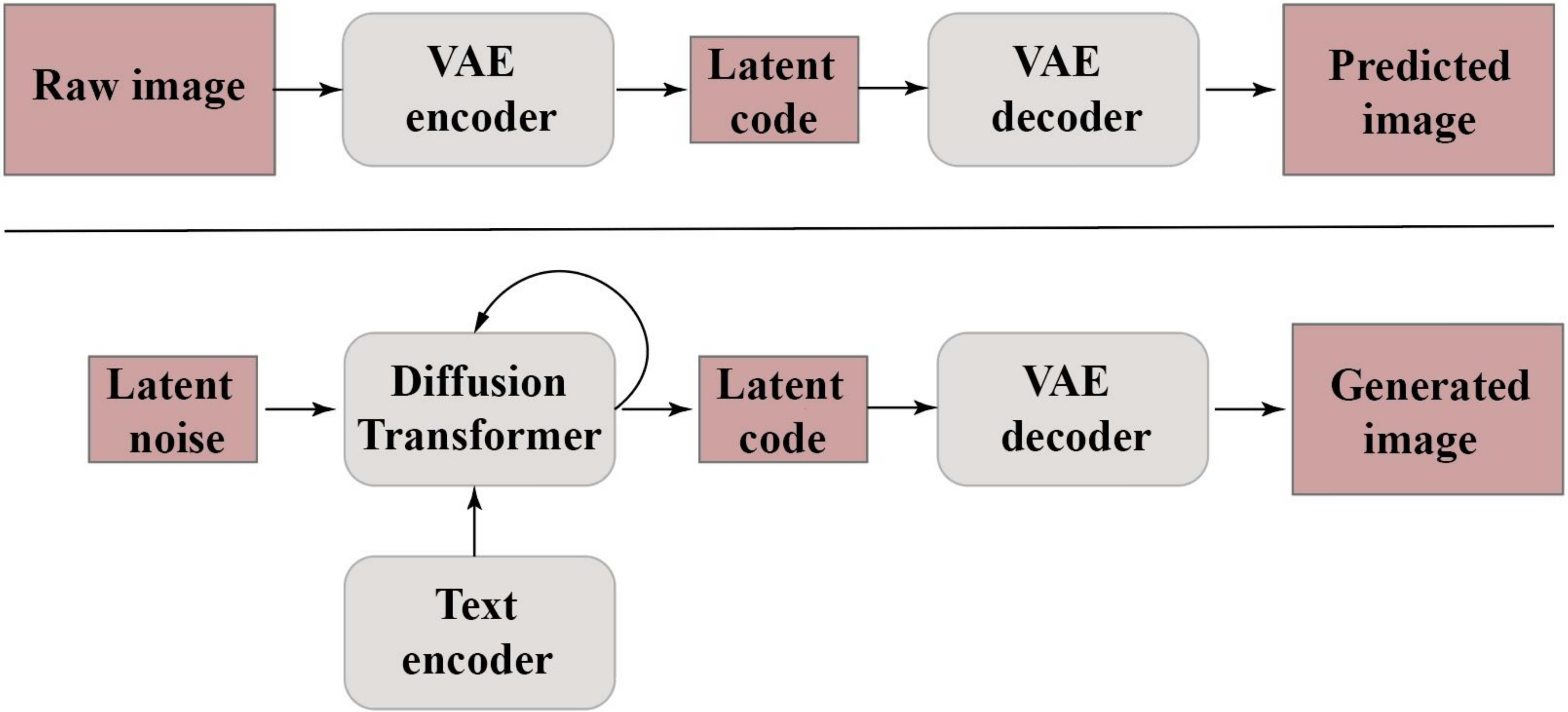


What about transformer?

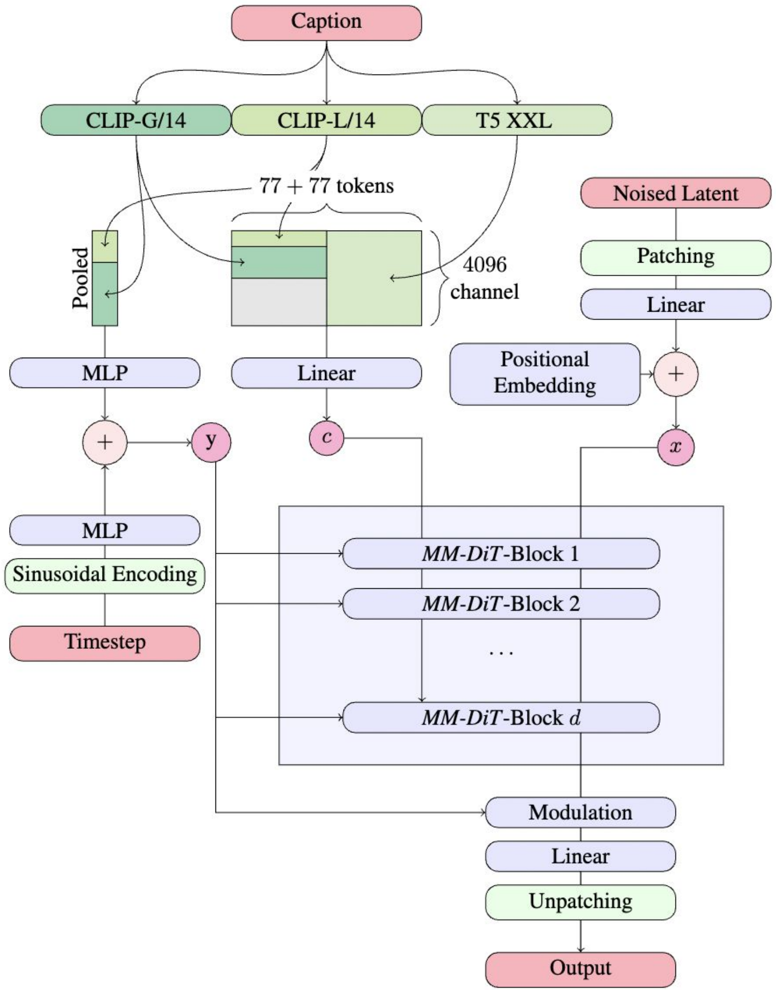
- A Diffusion model architecture could be built entirely from Transformer blocks.
- Vision Transformer (ViT) divides the input image into small patches, applies convolution, and processes the sequence like Language Models.
- Token positions are encoded using 2D positional embeddings.



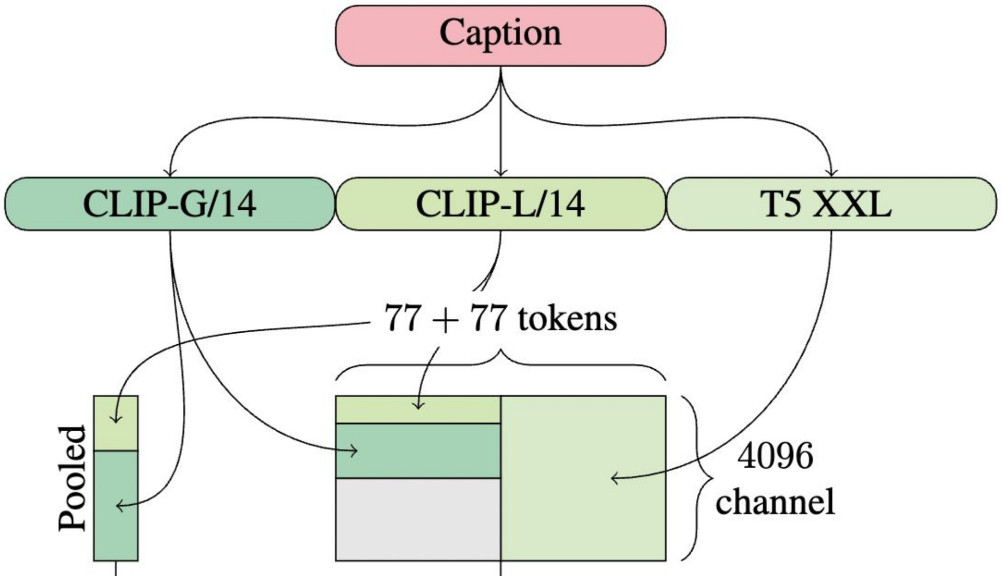
Current State



Diffusion Transformer SD3/FLUX/…



2.1 Text encoding



- B: batch size
- $L_C = 77$: CLIP token length
- L_T: T5 token length
- d_0, d_1, d_T: embedding dims of CLIP-G, CLIP-L, and T5
- $d_T = 4096$

1. **Text embedding**

1.1 CLIP encoding

$$E_G \in \mathbb{R}^{B \times L_C \times d_0},$$
$$E_L \in \mathbb{R}^{B \times L_C \times d_1},$$

1.2 Concat CLIP tokens

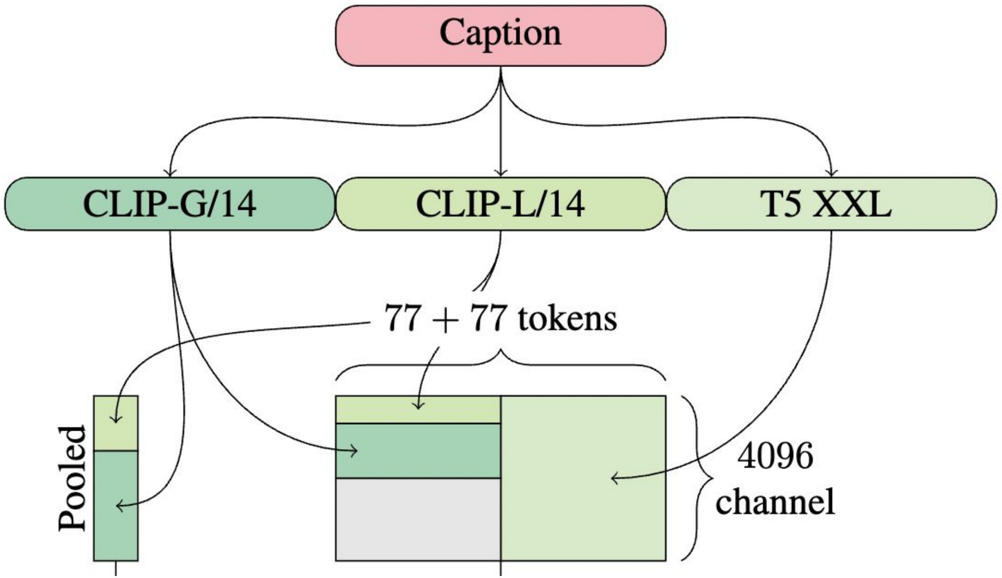
$$E_{\text{CLIP}} = [E_G; E_L] \quad \in \mathbb{R}^{B \times L_C \times (d_0 + d_1)}$$

1.3 T5 Embeddings

$$E_{T5} \in \mathbb{R}^{B \times L_T \times d_T}$$

2.1 Text encoding

- B: batch size
- $L_C = 77$: CLIP token length
- L_T: T5 token length
- d_0, d_1, d_T: embedding dims of CLIP-G, CLIP-L, and T5
- $d_T = 4096$



1. **Text embedding**

1.4 Pad CLIP Channels to 4096

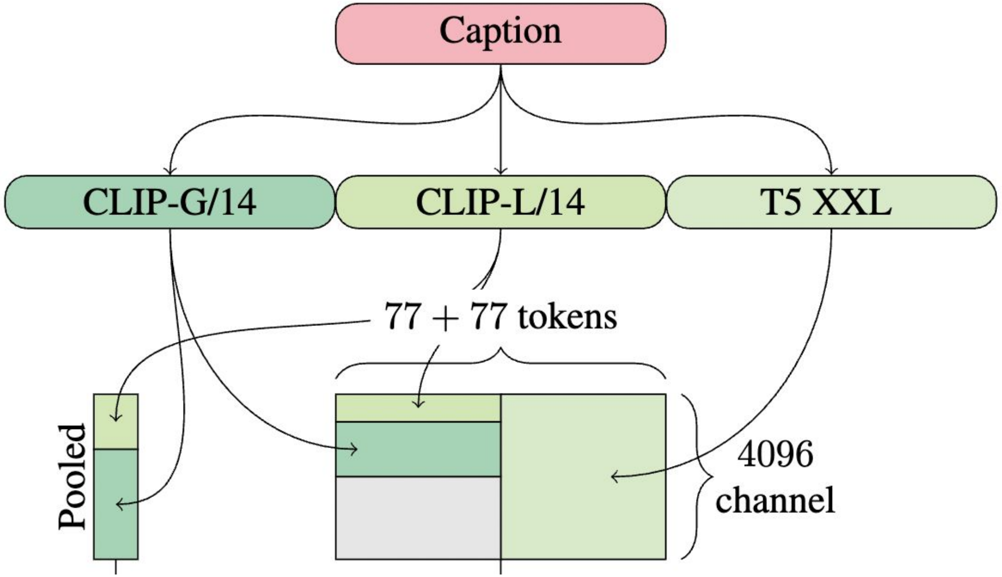
$$E'_{\text{CLIP}} = \text{Pad}\big(E_{\text{CLIP}}, (0,\ d_T - (d_0 + d_1))\big) \quad \in \mathbb{R}^{B \times L_C \times d_T}$$

1.5 T5 Embeddings

$$E_{\text{final}} = [E'_{\text{CLIP}}; E_{T5}] \quad \in \mathbb{R}^{B \times (L_C + L_T) \times d_T}$$

2.1 Text encoding

- B: batch size
- $L_C = 77$: CLIP token length
- L_T: T5 token length
- d_0, d_1, d_T: embedding dims of CLIP-G, CLIP-L, and T5
- $d_T = 4096$



2. Pooled text embedding

2.1 CLIP encodings

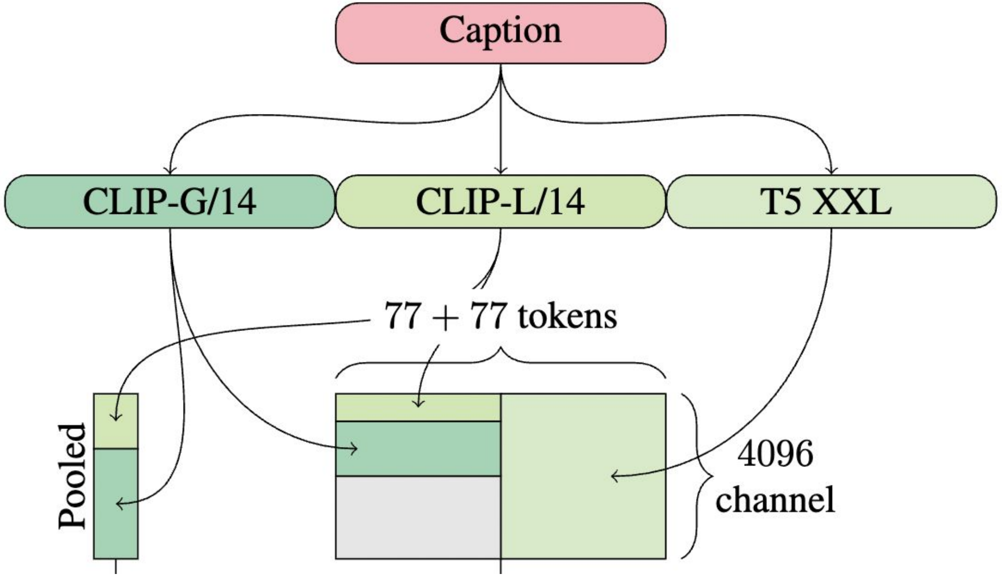
$$p_G \in \mathbb{R}^{B \times d_0}$$
$$p_L \in \mathbb{R}^{B \times d_1}$$

2.2 Concatenate Pooled CLIP Globals

$$p_{\text{CLIP}} = [p_G; p_L] \quad \in \mathbb{R}^{B \times (d_0 + d_1)}$$

2.1 Text encoding

- B: batch size
- $L_C = 77$: CLIP token length
- L_T: T5 token length
- d_0, d_1, d_T: embedding dims of CLIP-G, CLIP-L, and T5
- $d_T = 4096$



$$\boldsymbol{c} = \boldsymbol{E}_{\text{final}} \in \mathbb{R}^{B \times (L_C + L_T) \times 4096}$$

$$\boldsymbol{y} = \boldsymbol{p}_{\text{CLIP}} \in \mathbb{R}^{B \times (d_0 + d_1)}$$

2.2 Noised Latent

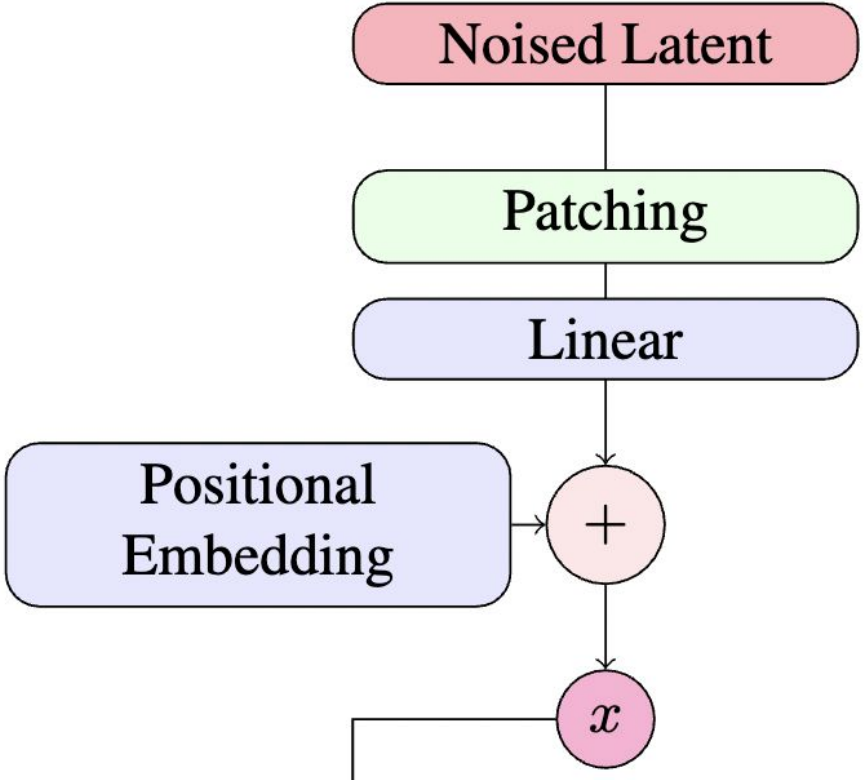
$$z_t \in \mathbb{R}^{B \times C \times H \times W}$$

1. **Patching**

$$z_t^{(p)} \in \mathbb{R}^{B \times N \times (C \cdot P^2)}$$

$$N = \frac{H \cdot W}{P^2}$$

2. **Linear Projection**



$$h = z_t^{(p)} W_\ell + b_\ell, \quad W_\ell \in \mathbb{R}^{(C \cdot P^2) \times d_x}, \quad h \in \mathbb{R}^{B \times N \times d_x}$$

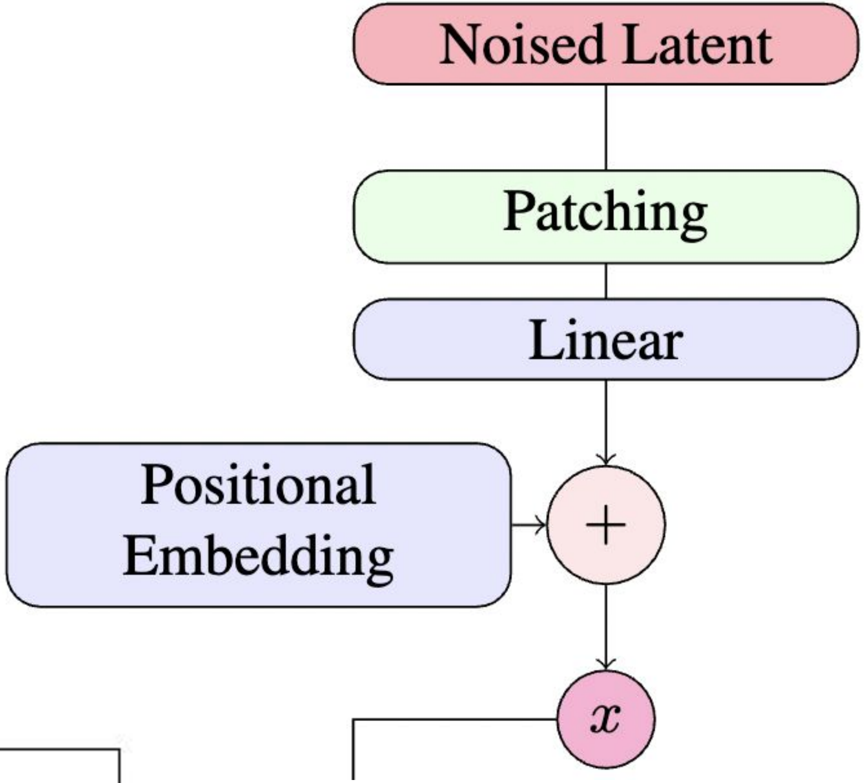
2.2 Noised Latent

3. Positional Embeddings

$$x = h + e_{\text{pos}} \qquad \in \mathbb{R}^{B \times N \times d_x}$$

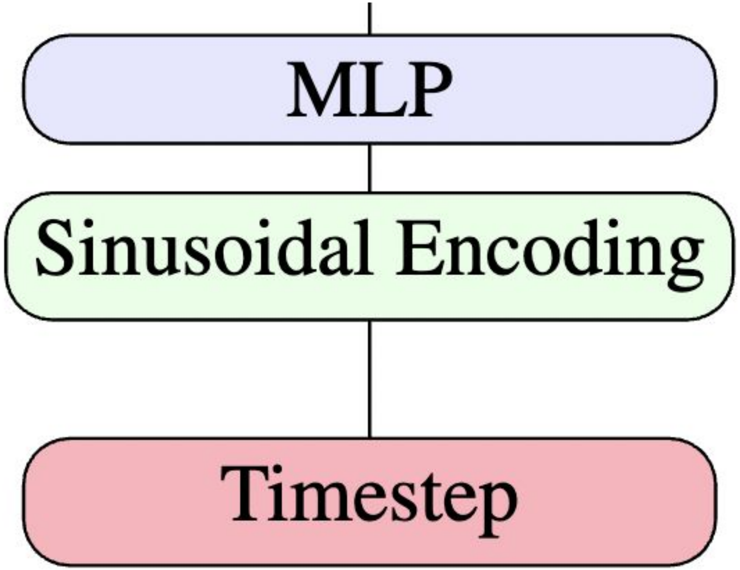
$$e_{\text{pos}} \in \mathbb{R}^{N \times d_x}$$

$$\boxed{x = z_t^{(p)} W_\ell + b_\ell + e_{\text{pos}}}$$



2.3 Timestep

t: the diffusion timestep (scalar, e.g. $t \in [0, 1000]$)



1. **Sinusoidal Encoding**

$$\mathrm{Sinusoidal}(t) = \Big[\sin\big(t \cdot \omega_0\big),\ \cos\big(t \cdot \omega_0\big),\ \sin\big(t \cdot \omega_1\big),\ \cos\big(t \cdot \omega_1\big),\ \ldots\Big]$$

$$\omega_k = \frac{1}{10000^{2k/d_t}}, \quad k = 0, 1, \ldots, \tfrac{d_t}{2} - 1$$

2. **MLP**

$$\boxed{e_t = W_2\,\mathrm{SiLU}\big(W_1\,\mathrm{Sinusoidal}(t) + b_1\big) + b_2} \qquad e_t \in \mathbb{R}^{d_t}$$

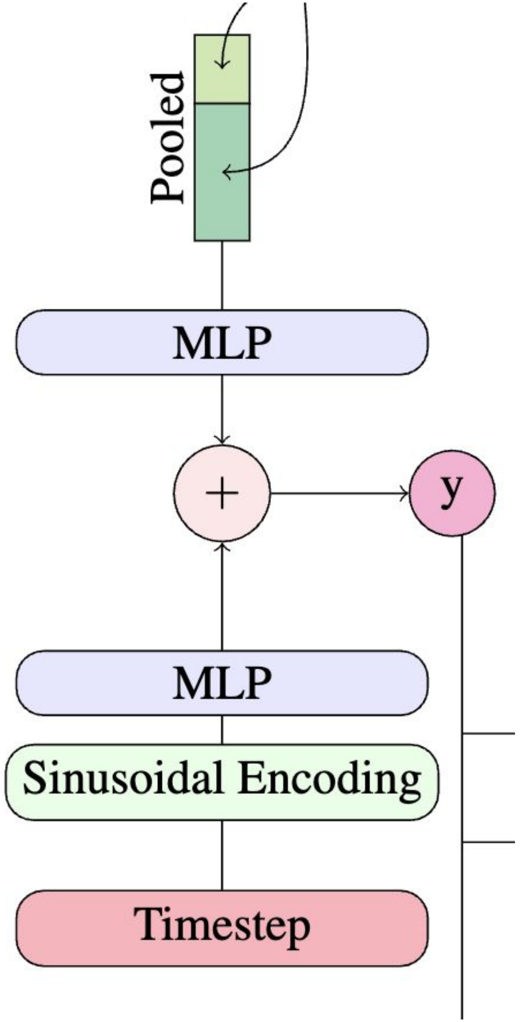
2.3 Final pooled embedding

$$\boxed{\begin{aligned} c &= E_{\text{final}} \in \mathbb{R}^{B\times(L_C+L_T)\times 4096} \\ y &= p_{\text{CLIP}} \in \mathbb{R}^{B\times(d_0+d_1)} \end{aligned}}$$

$$e_t = W_2\,\mathrm{SiLU}(W_1 \sin(t) + b_1) + b_2 \quad \in \mathbb{R}^{B\times d_y}$$

$$h_y = W_y\, p_{\text{CLIP}} + b_y \quad \in \mathbb{R}^{B\times d_y}$$

$$\boxed{y = e_t + h_y \quad \in \mathbb{R}^{B\times d_y}}$$



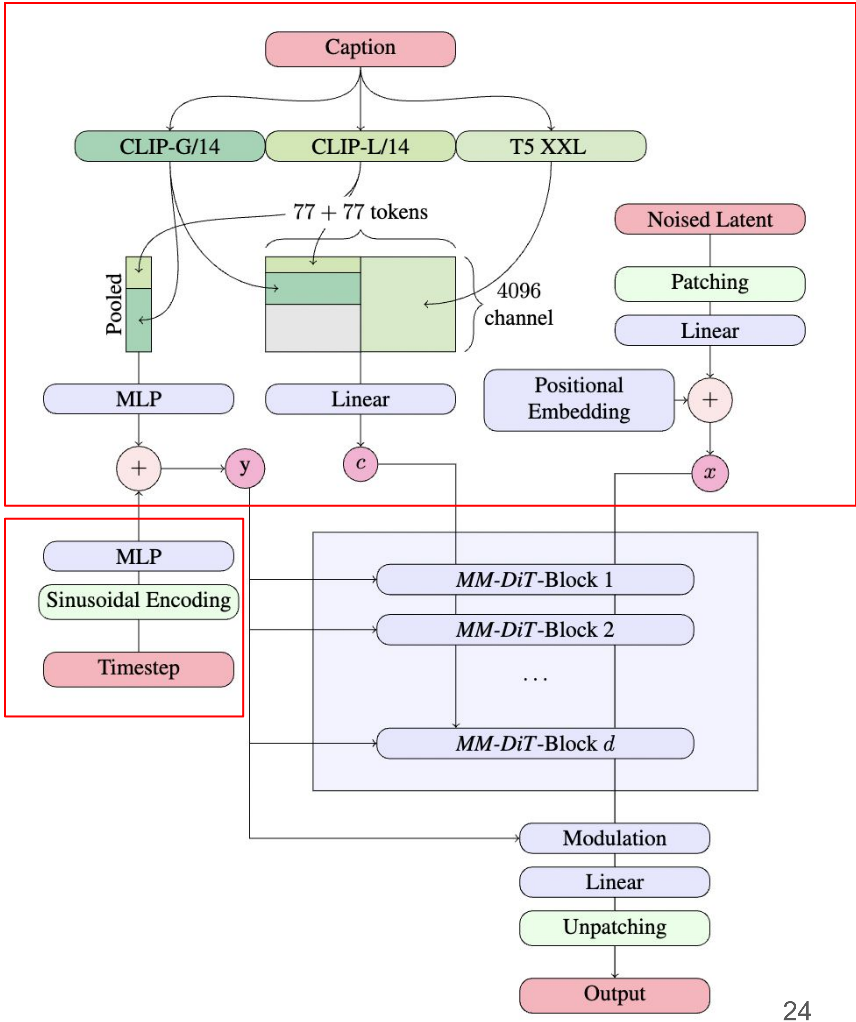
2 Summary

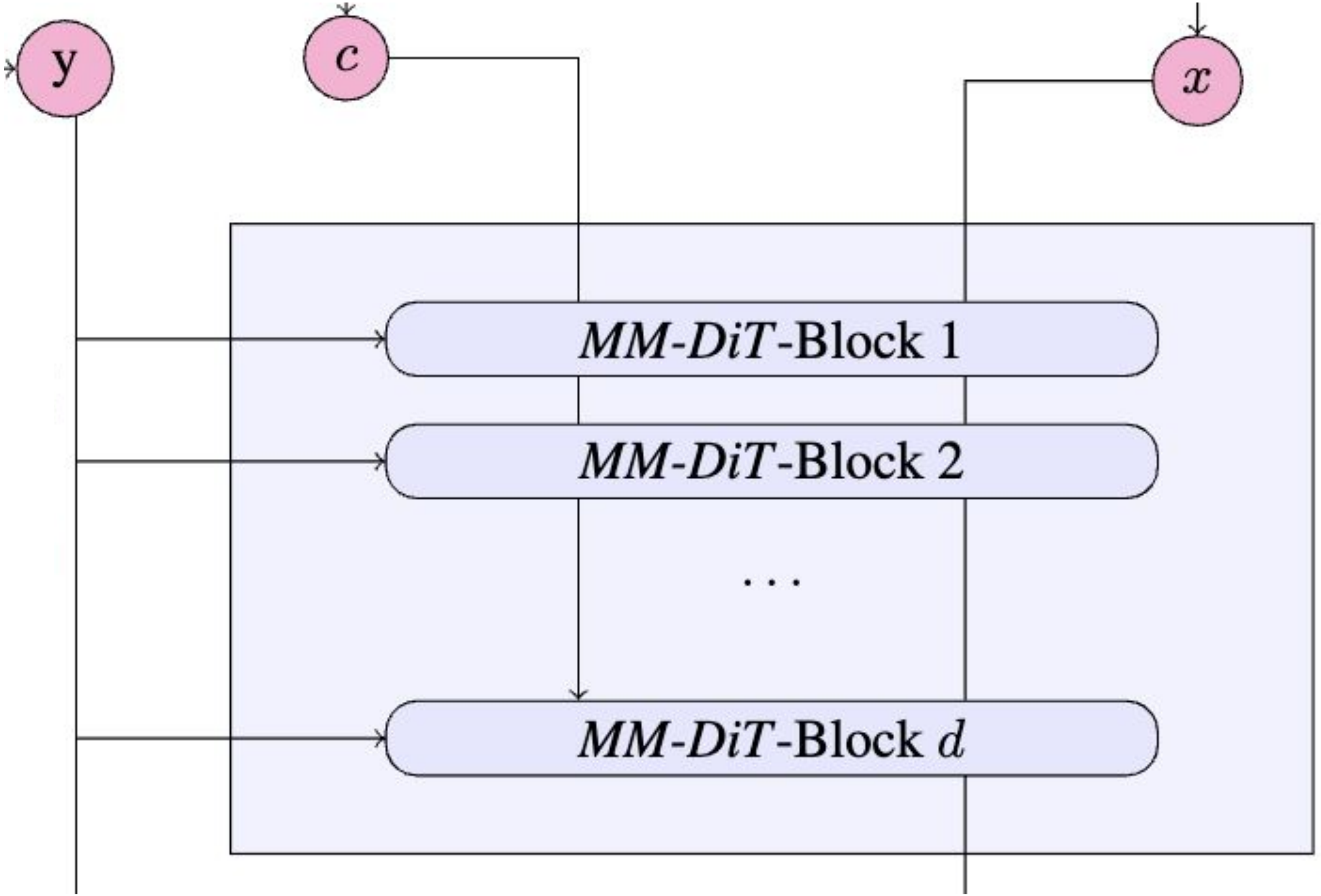
$$(x,\ c,\ y)$$

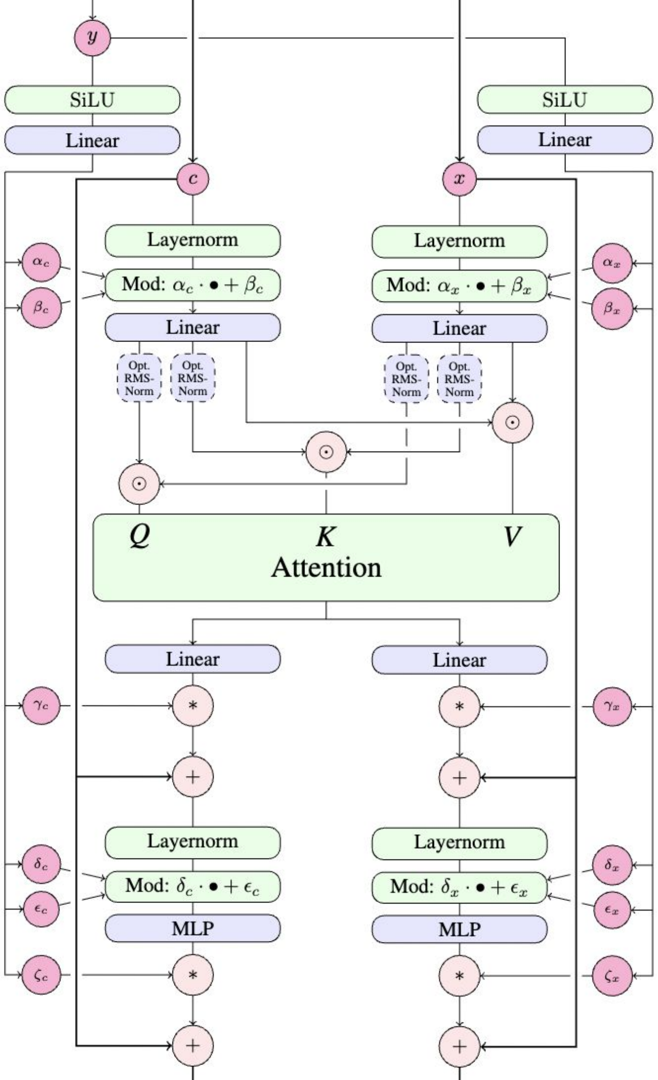
$$x = \text{Patchify}(z_t)W_\ell + e_{\text{pos}} \quad \in \mathbb{R}^{B\times N\times d_x}$$

$$c = [E_G; E_L; E_{T5}] \in \mathbb{R}^{B\times(L_C+L_T)\times 4096}$$

$$y = h_y + e_t$$



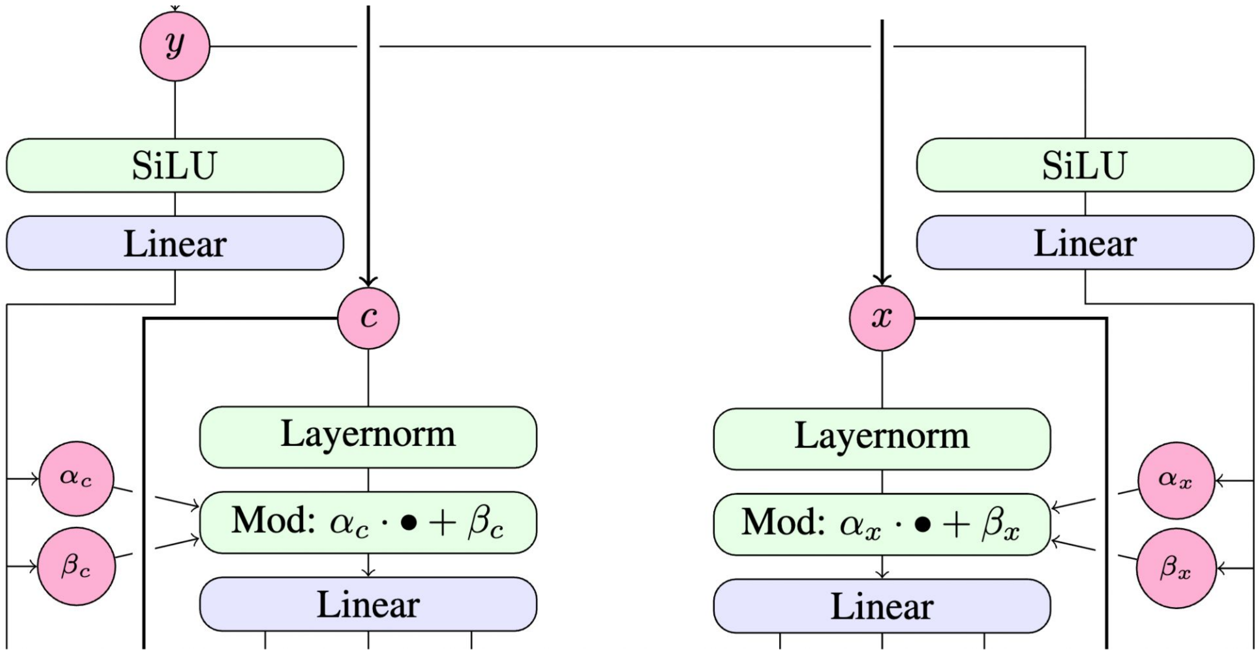

y
c
x
MM-DiT-Block 1
MM-DiT-Block 2
...
MM-DiT-Block d



- **Normalization**
- **Modulation layers**
- **Linear layers**
- **Attention**

2.4 Modulation Layers

$$y \in \mathbb{R}^{B \times d_y}, \quad x \in \mathbb{R}^{B \times N \times d_x}.$$

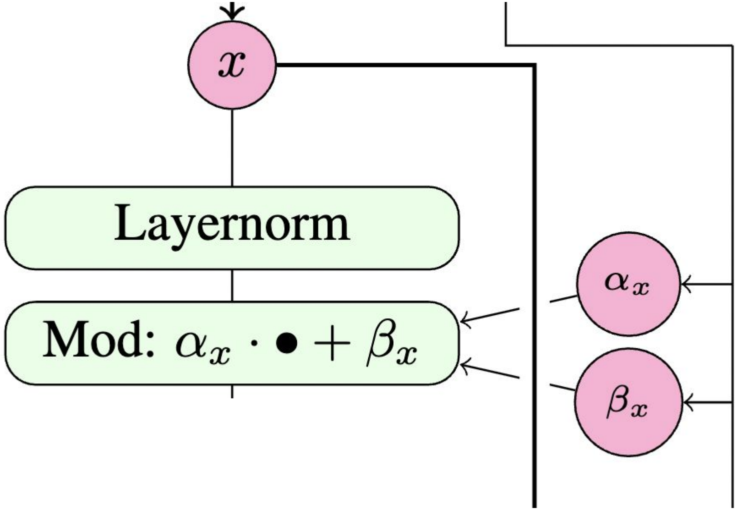


$$[\alpha_x, \beta_x, \gamma_x, \delta_x, \epsilon_x, \zeta_x] = W_y \,\mathrm{SiLU}(y) + b_y, \qquad \alpha_x, \beta_x, \gamma_x, \delta_x, \epsilon_x, \zeta_x \in \mathbb{R}^{B \times d_x}.$$

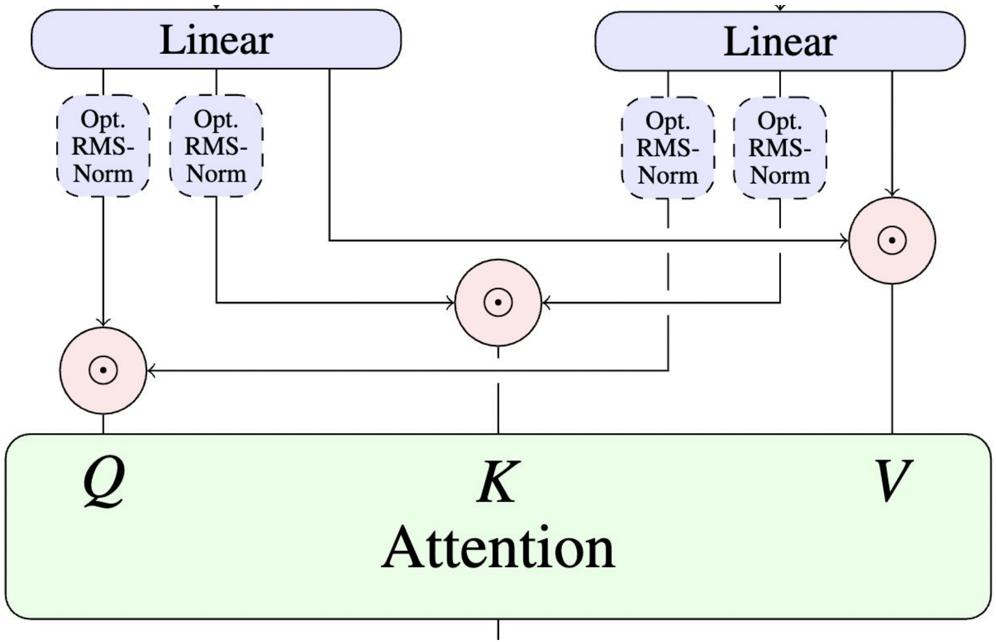
2.4 Modulation Layers

$$[\alpha_x, \beta_x, \gamma_x, \delta_x, \epsilon_x, \zeta_x] = W_y \,\mathrm{SiLU}(y) + b_y,$$

$$\tilde{x} = \alpha_x \odot \mathrm{LayerNorm}(x) + \beta_x$$



2.4 Attention Layer



1. Linear

$$Q_x = x_{\text{mod}} W_Q^{(x)}, \quad K_x = x_{\text{mod}} W_K^{(x)}, \quad V_x = x_{\text{mod}} W_V^{(x)}$$

$$Q_c = c_{\text{mod}} W_Q^{(c)}, \quad K_c = c_{\text{mod}} W_K^{(c)}, \quad V_c = c_{\text{mod}} W_V^{(c)}$$

2. RMS Norm

$$Q = g_Q \odot \frac{Q'}{\sqrt{\frac{1}{d}\sum_k (Q'_k)^2 + \epsilon}}, \quad K = g_K \odot \frac{K'}{\sqrt{\frac{1}{d}\sum_k (K'_k)^2 + \epsilon}},$$

3. Concatenation

$$Q = [Q_x; Q_c], \quad K = [K_x; K_c], \quad V = [V_x; V_c].$$

4. Self-attention

$$A = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_h}}\right) V,$$

2.5 Again modulation

1. **Attention residual scaling**

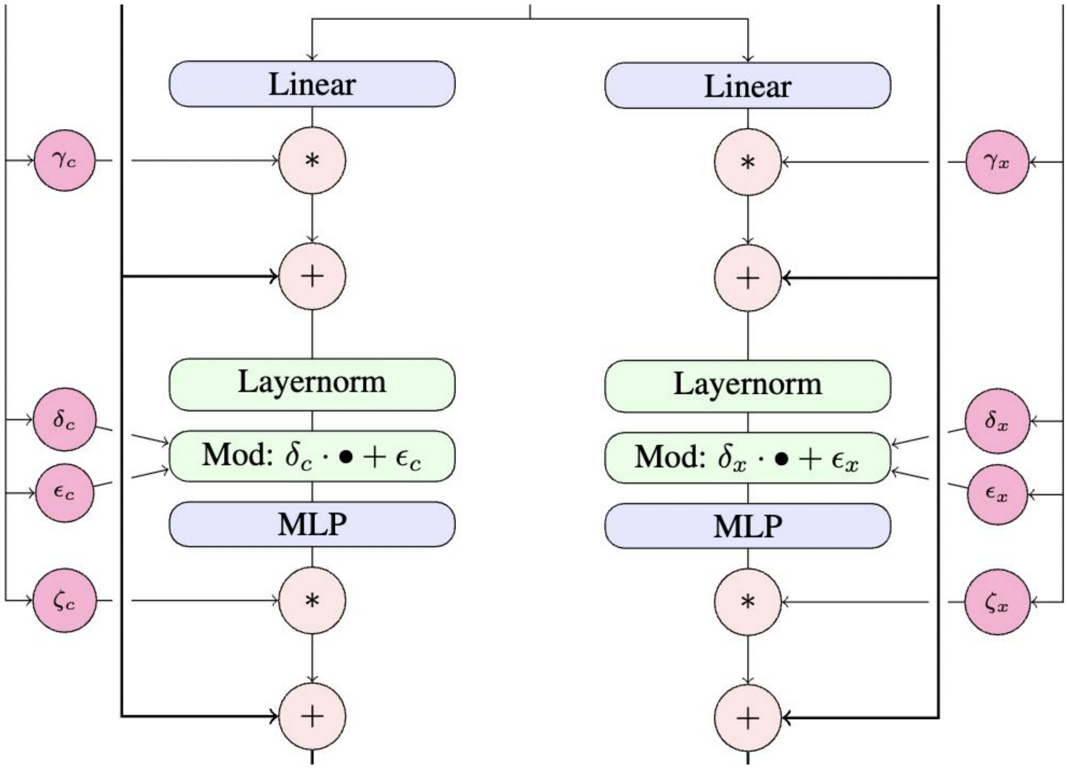
$$x_{\text{att}} = x + \gamma_x \odot A_x$$

2. **LayerNorm + modulation**

$$x_{\text{mlp-in}} = \delta_x \odot \text{LayerNorm}(x_{\text{att}}) + \epsilon_x$$

3. **MLP**

$$m_x = W_2^{(x)}\,\text{SiLU}(W_1^{(x)} x_{\text{mlp-in}} + b_1^{(x)}) + b_2^{(x)}$$



4. **Gated residual update**

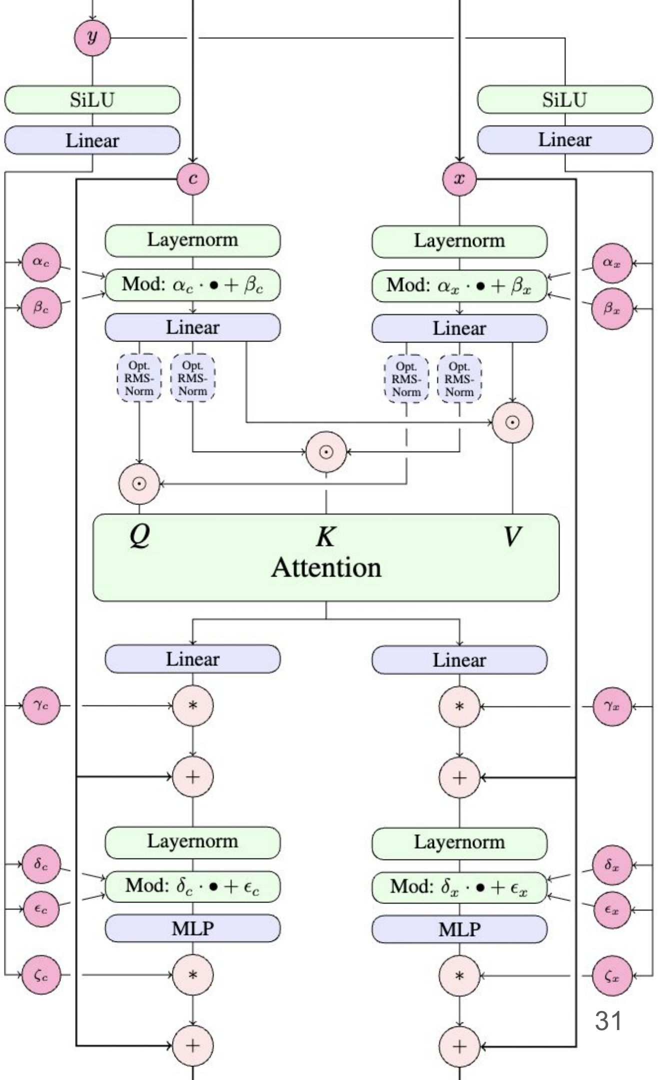
$$x' = x_{\text{att}} + \zeta_x \odot m_x$$

$$[\alpha_x, \beta_x, \gamma_x, \delta_x, \epsilon_x, \zeta_x] = W_x(\mathrm{SiLU}(W_y y))$$
$$[\alpha_c, \beta_c, \gamma_c, \delta_c, \epsilon_c, \zeta_c] = W_c(\mathrm{SiLU}(W_y y))$$

$$x_m = \alpha_x \odot \mathrm{LayerNorm}(x) + \beta_x,$$
$$c_m = \alpha_c \odot \mathrm{LayerNorm}(c) + \beta_c$$

$$Q = [x_m W_Q;\ c_m W_Q], \quad K = [x_m W_K;\ c_m W_K],$$
$$V = [x_m W_V;\ c_m W_V] \qquad A = \mathrm{softmax}\!\left(\frac{QK^\top}{\sqrt{d_h}}\right)V$$

$$x' = x + \gamma_x \odot \mathrm{Linear}(A_x) + \zeta_x \odot \mathrm{MLP}\big(\delta_x \odot \mathrm{LayerNorm}(A_x) + \epsilon_x\big)$$
$$c' = c + \gamma_c \odot \mathrm{Linear}(A_c) + \zeta_c \odot \mathrm{MLP}\big(\delta_c \odot \mathrm{LayerNorm}(A_c) + \epsilon_c\big)$$



3. Image editing

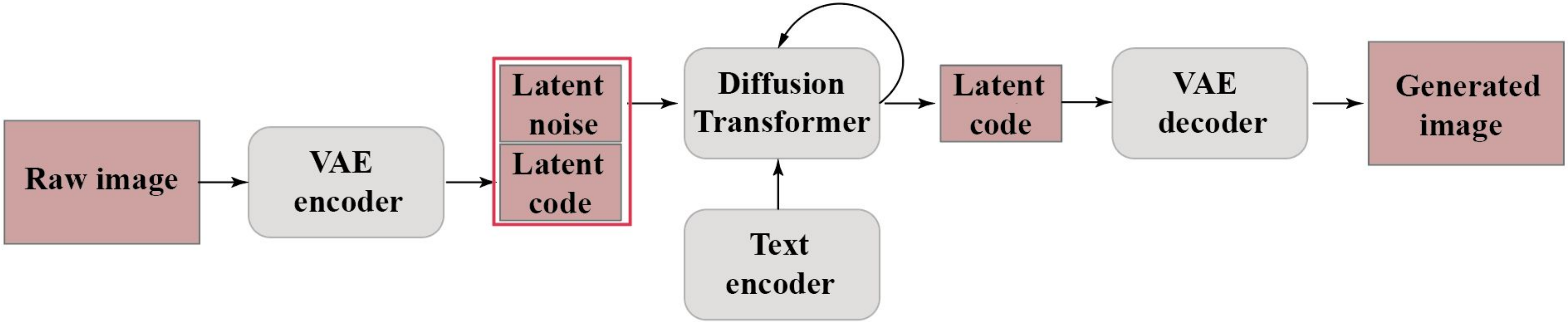
(a) Input image

(b) *"remove the thing from her face"*

(c) *"she is now taking a selfie in the streets of Freiburg, it's a lovely day out."*

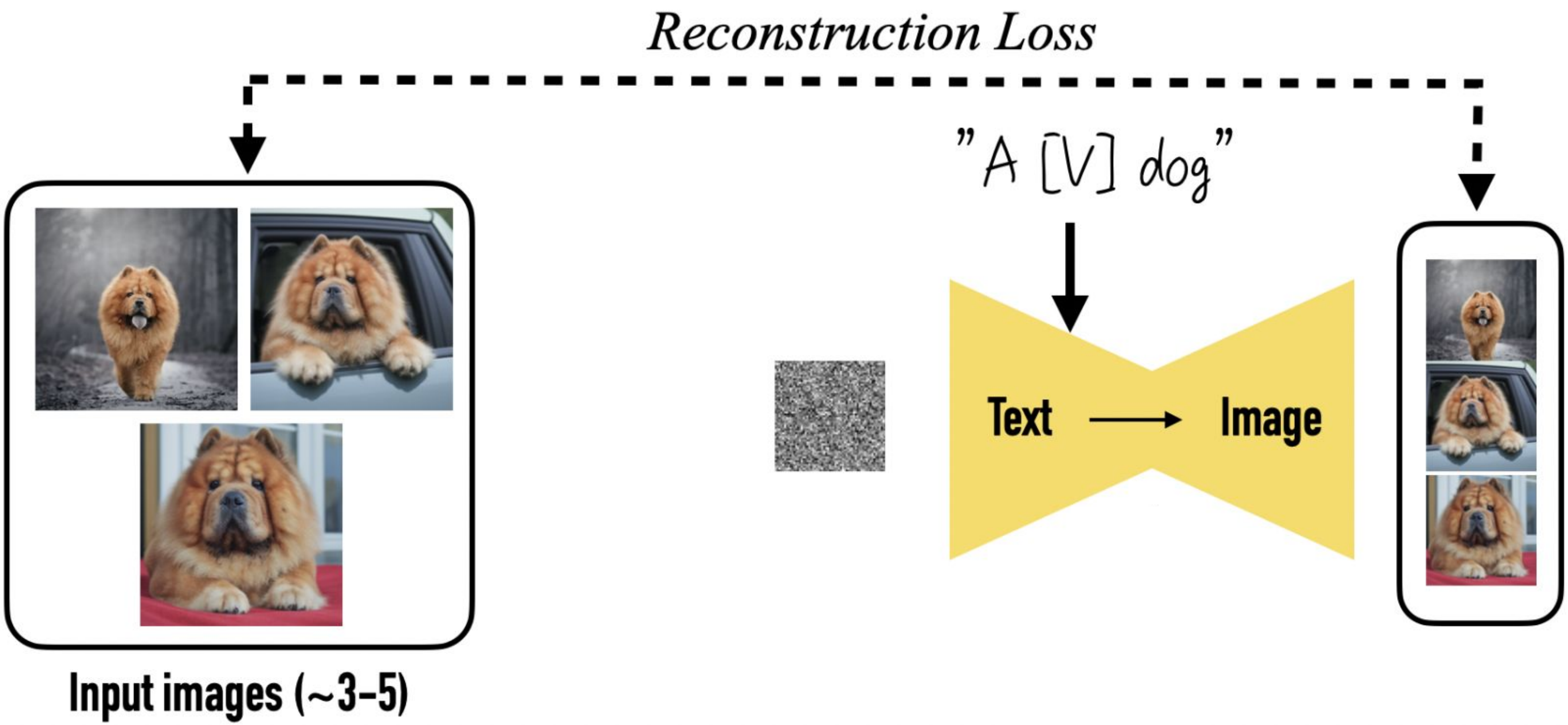
(d) *"it's now snowing, everything is covered in snow."*

3. Image editing



3. Personalization

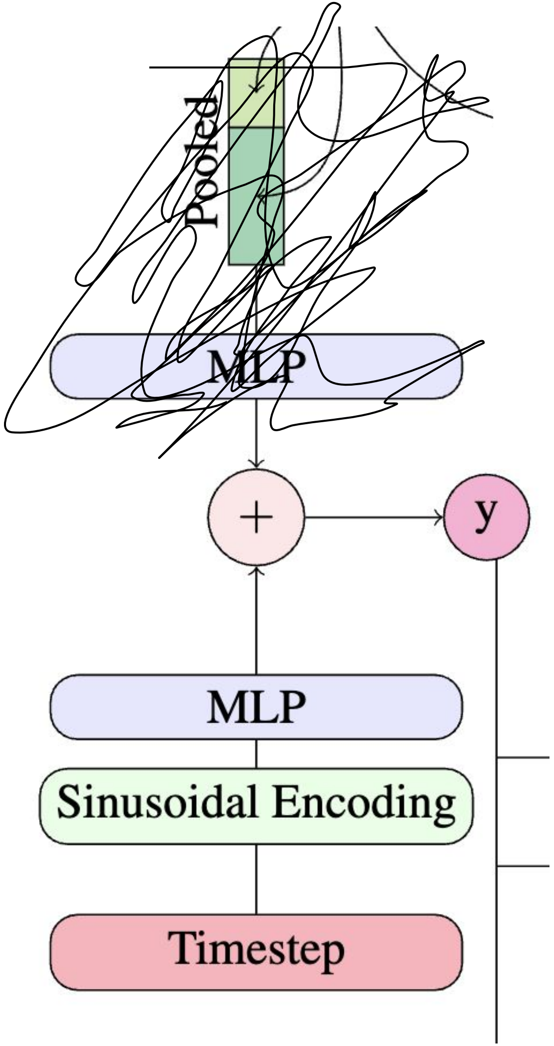
3. Personalization



3. Video generation

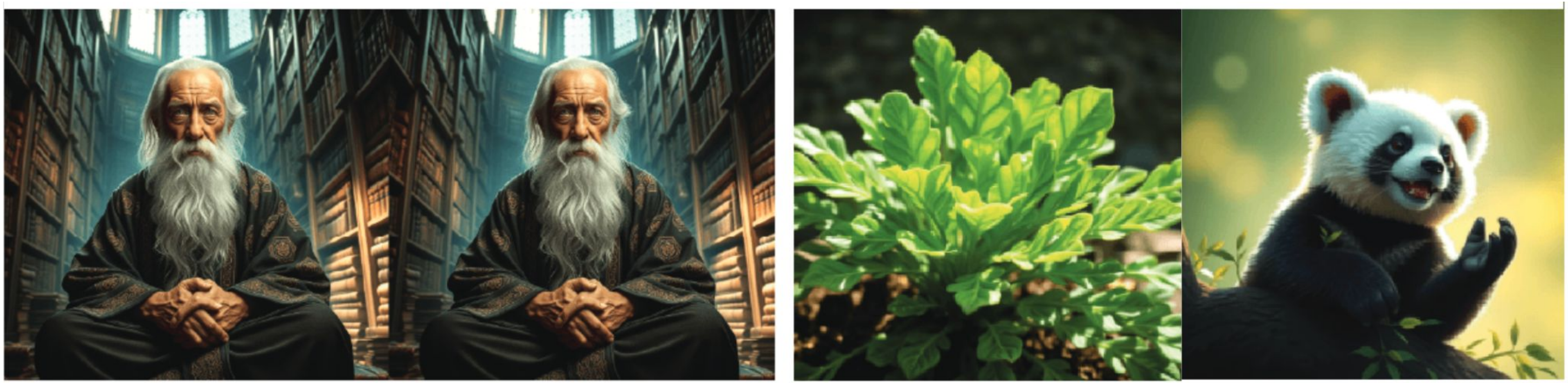
- Text-2-Video generation is quite similar to Text-2-Image.
- However, there are important differences:
  - Input data has now additional **temporal dimension**.
  - **Positional embeddings** have to account for this.
  - In addition to relevance and image quality, **motion consistency** and **smooth transition** between frames are crucial.
- Some approaches treat video clips as a single sequence, whereas make several predictions based on previous frames.

3. Video generation

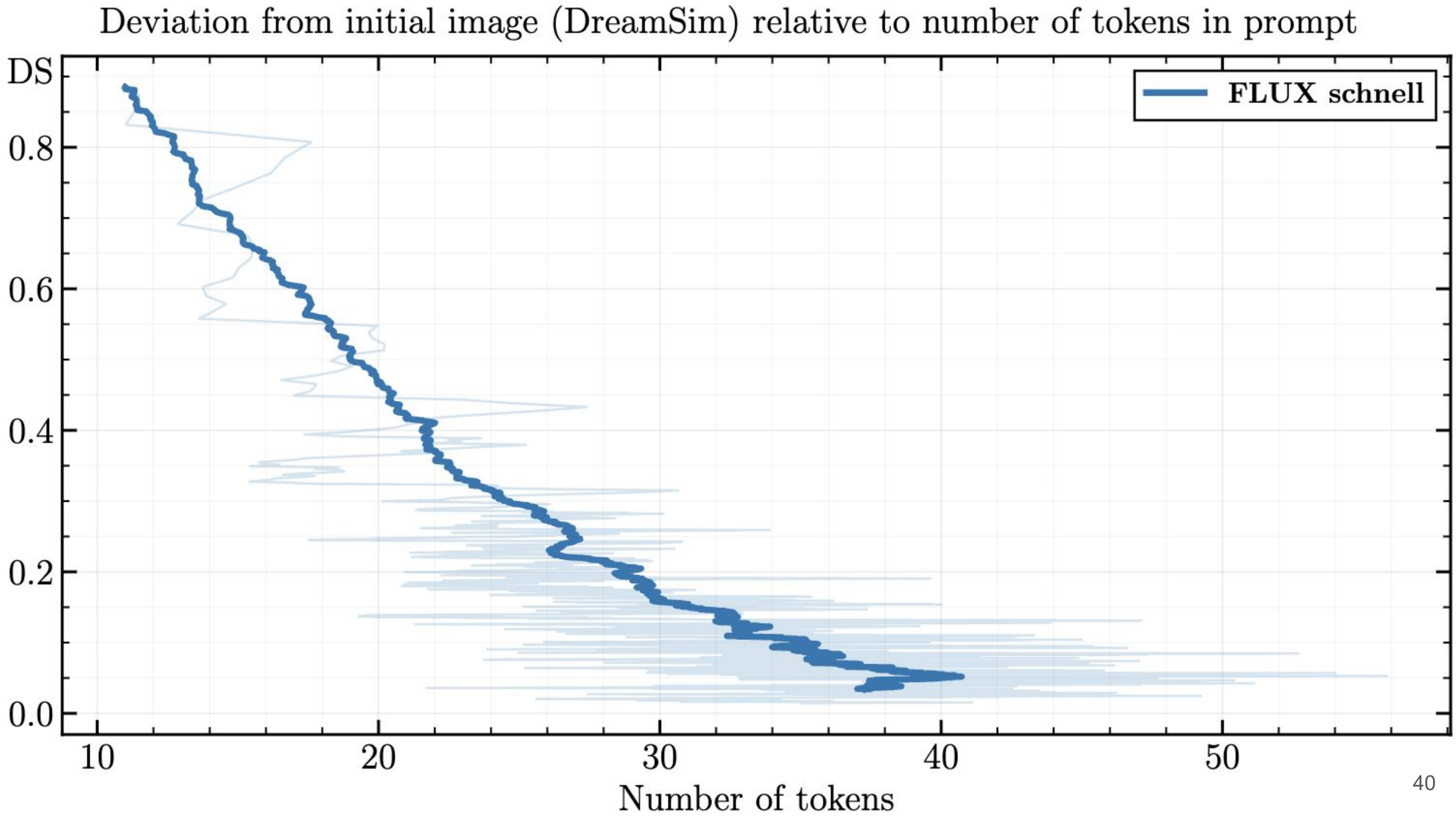



4. Example of possible research

$$y = e_t \cancel{+ h_y} \quad \in \mathbb{R}^{B \times d_y}$$

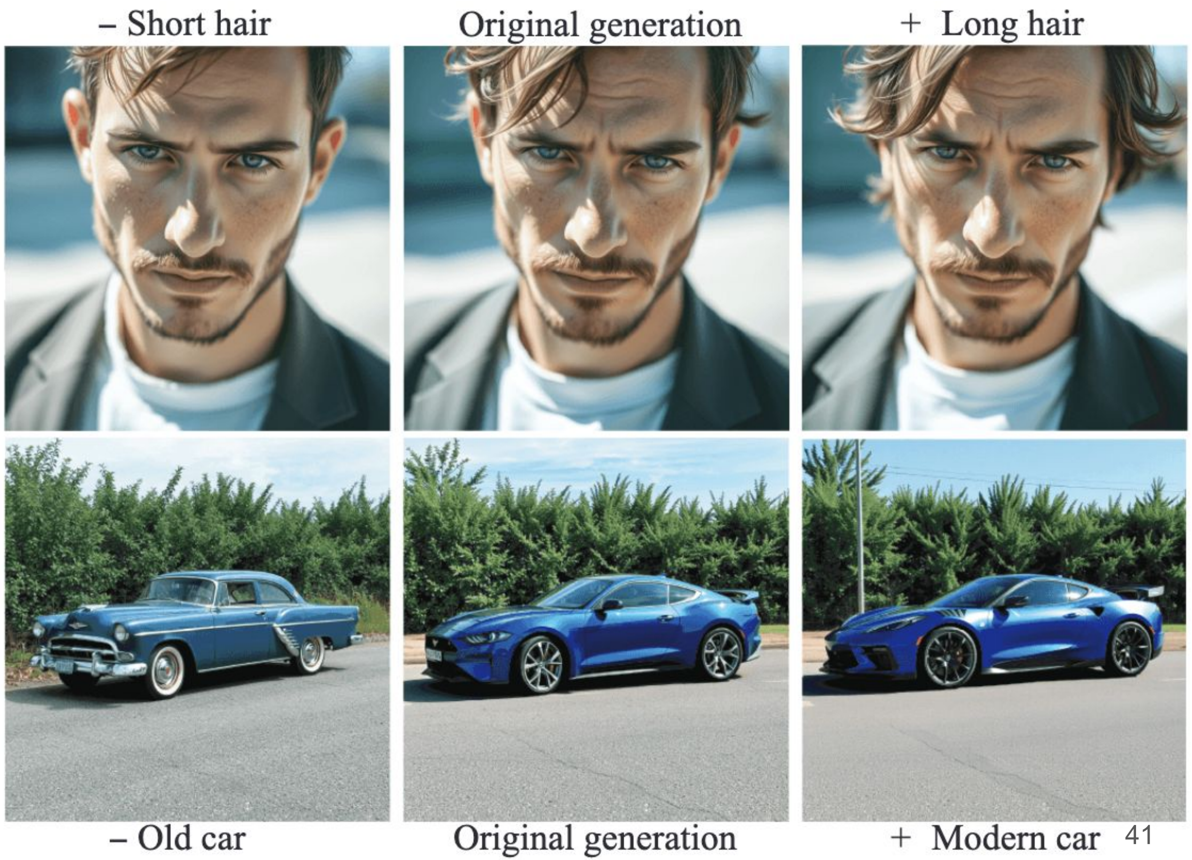


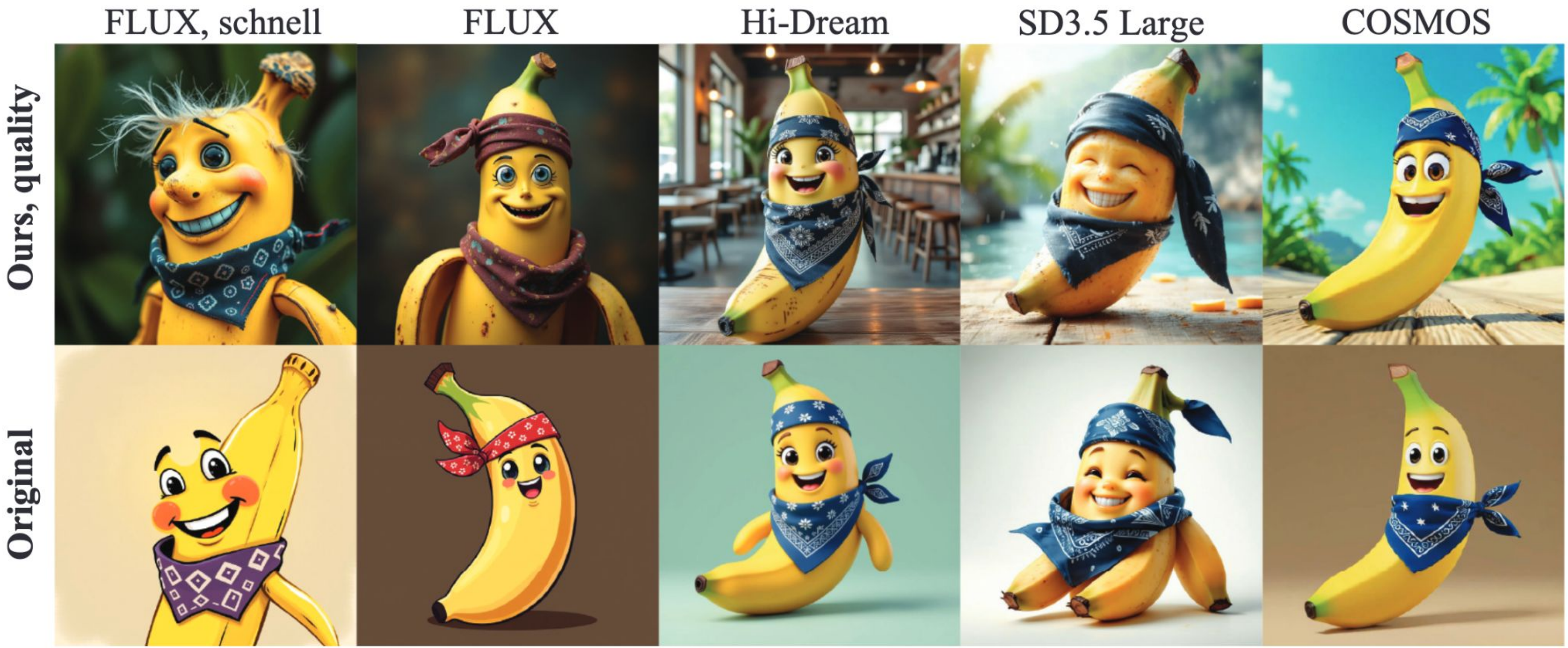
w/o CLIP, long | with CLIP, long | w/o CLIP, short | with CLIP, short


Deviation from initial image (DreamSim) relative to number of tokens in prompt
DS
0.8
0.6
0.4
0.2
0.0
10
20
30
40
50
Number of tokens
FLUX schnell

$$\mathbf{y}(\mathbf{p}, t) \rightarrow \hat{\mathbf{y}}(\mathbf{p}, \mathbf{p}_+, \mathbf{p}_-, t) = \mathbf{y}(\mathbf{p}, t) + w \cdot \left(\mathbf{y}(\mathbf{p}_+, t) - \mathbf{y}(\mathbf{p}_-, t)\right).$$

p+ = Long hair | Modern car
p− = Short hair | Old car





a smiling banana wearing a bandana