LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Generative Models in Computer Vision

Dr. Tao Hu, Ommer-Lab PostDoc

https://taohu.me

Agenda

- Brief Theory of Generative Models
- Our work introduction
  - ZigMa: A DiT-style Zigzag Mamba Diffusion Model

Generative AI: New Era

DALL·E

Here are two images showcasing the University of Calgary during winter, with snow-covered landscapes and gentle snowflakes adding to the serene atmosphere.

Generative AI: New Era

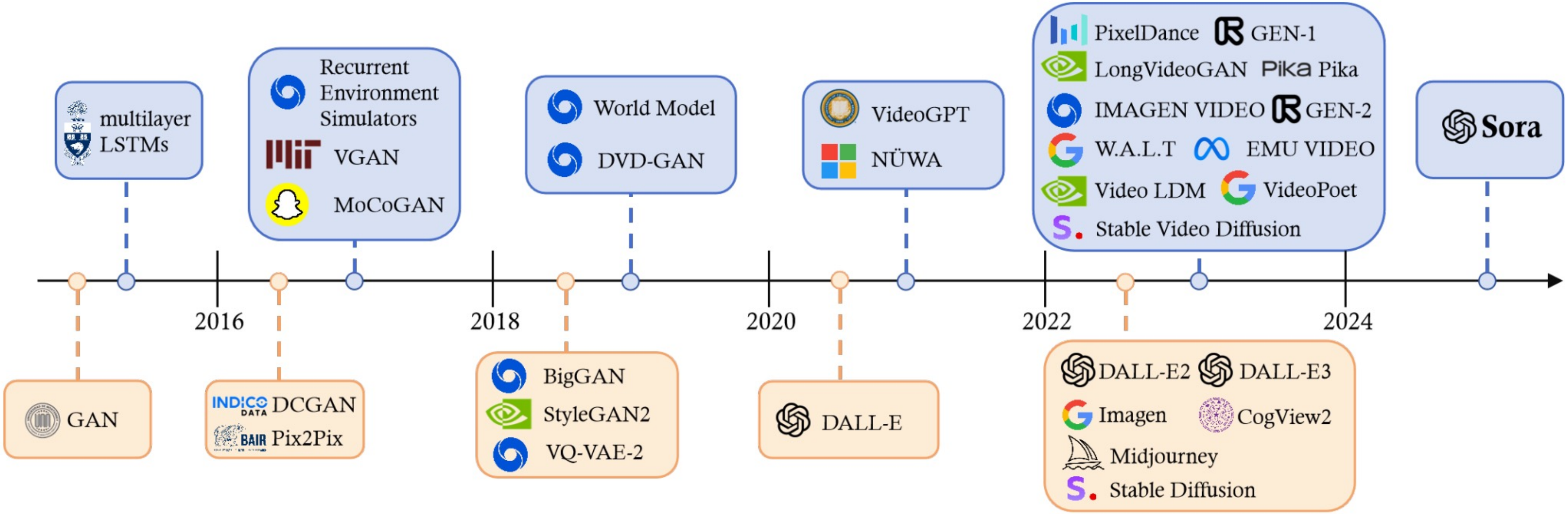


Figure 3: History of Generative AI in Vision Domain.

Source: https://arxiv.org/abs/2402.17177

Generative AI: New Era

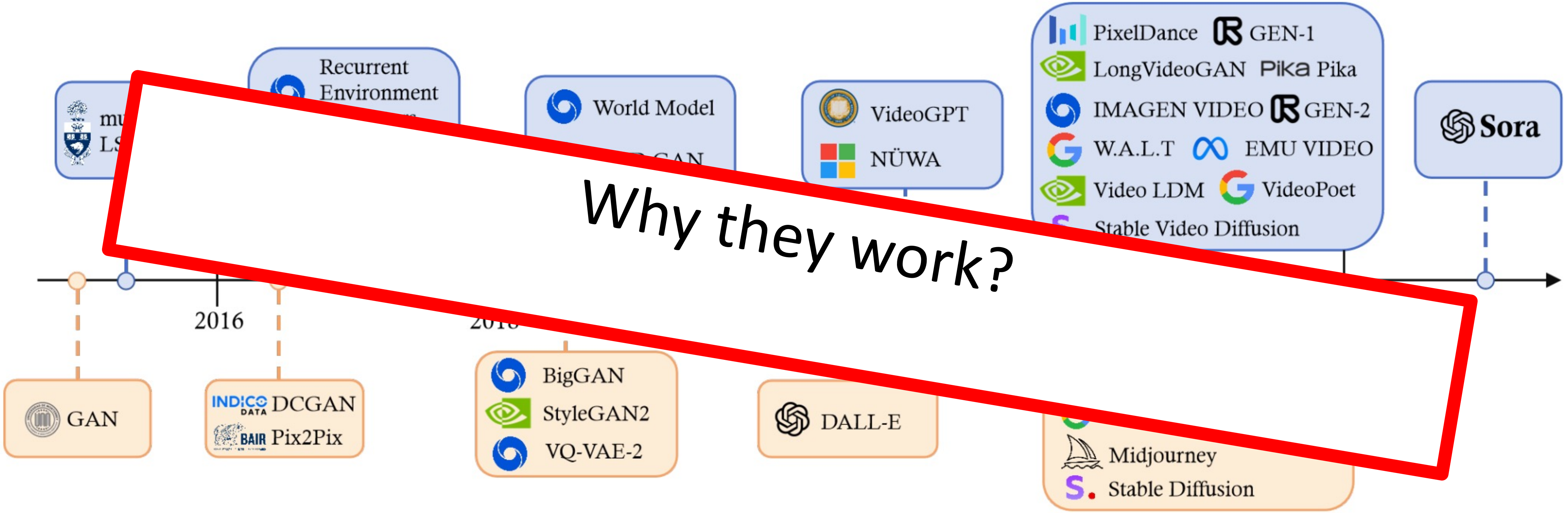
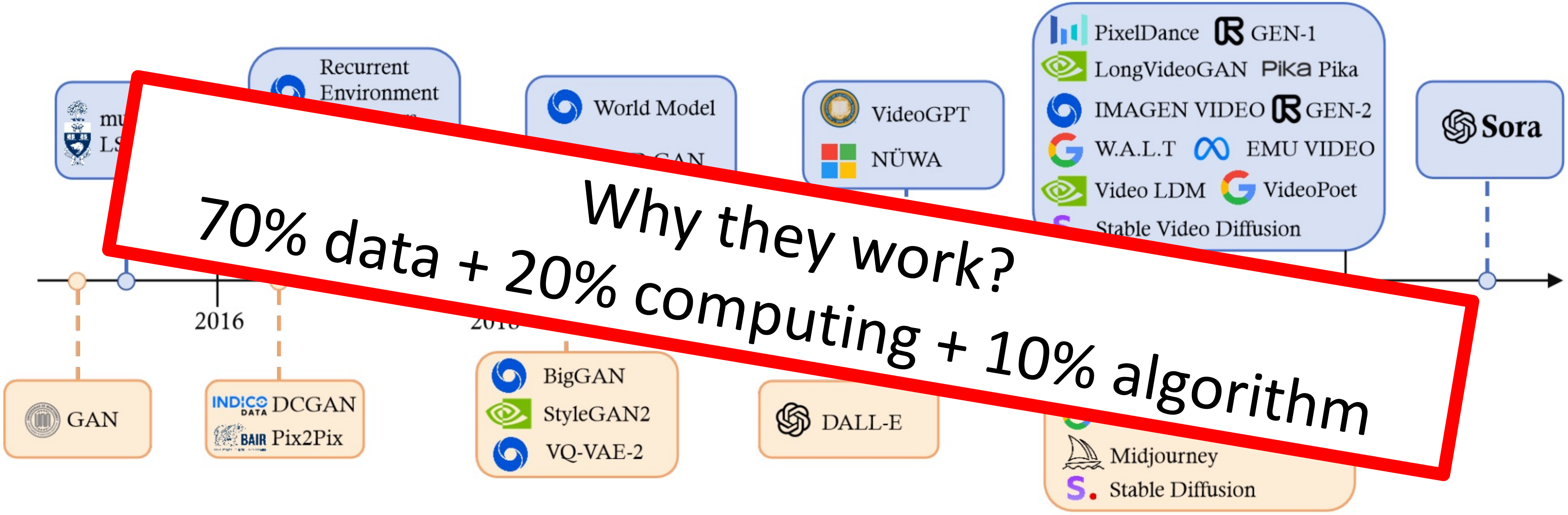


Figure 3: History of Generative AI in Vision Domain.

Source: https://arxiv.org/abs/2402.17177

Generative AI: New Era



Why they work?

70% data + 20% computing + 10% algorithm

Figure 3: History of Generative AI in Vision Domain.

Source: https://arxiv.org/abs/2402.17177

Generative AI: New Era

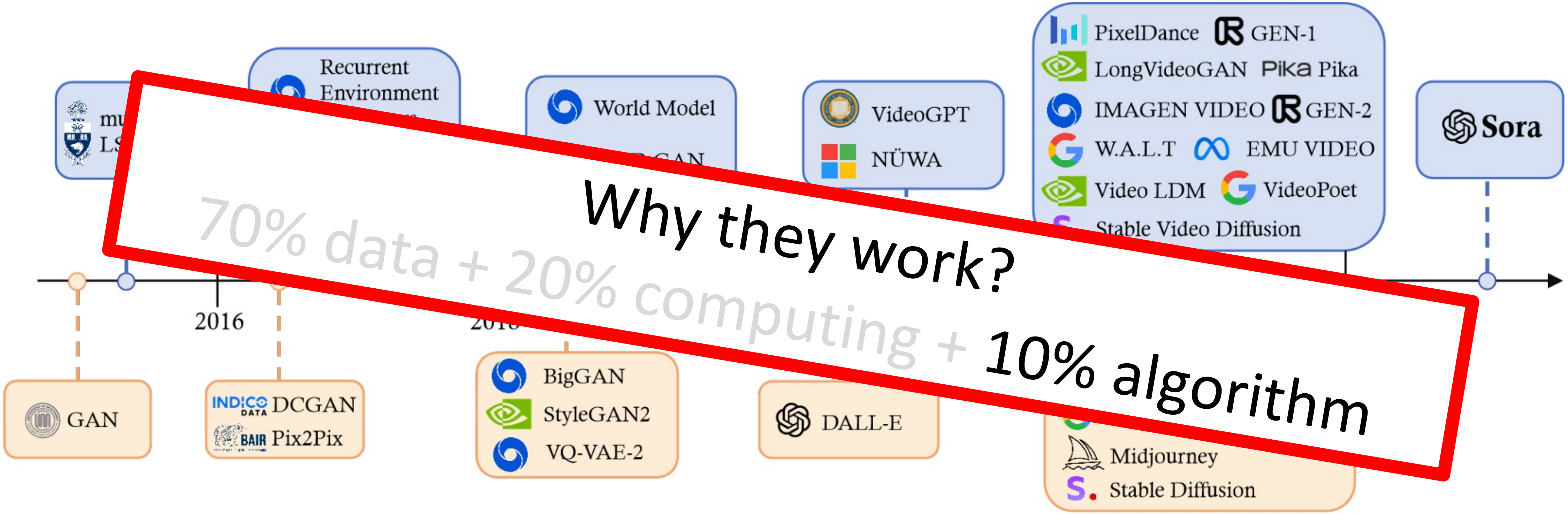


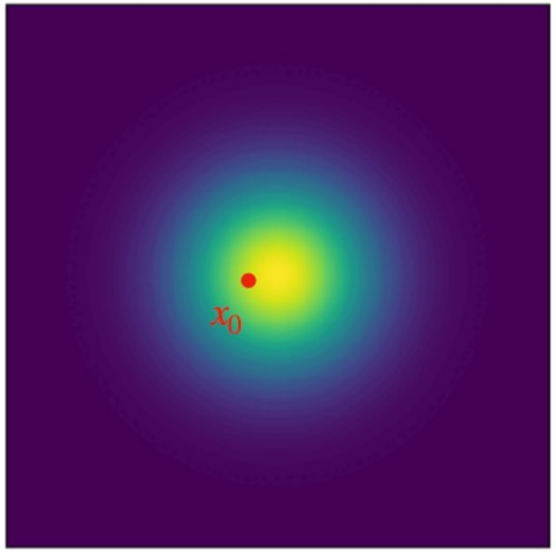
Figure 3: History of Generative AI in Vision Domain.

Source: https://arxiv.org/abs/2402.17177

Generative Models

$\mathbb{R}^d$

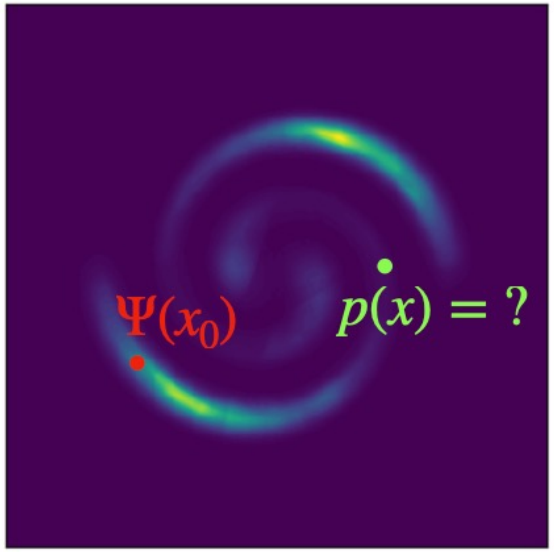
$p(x)$



$x_0 \sim p$

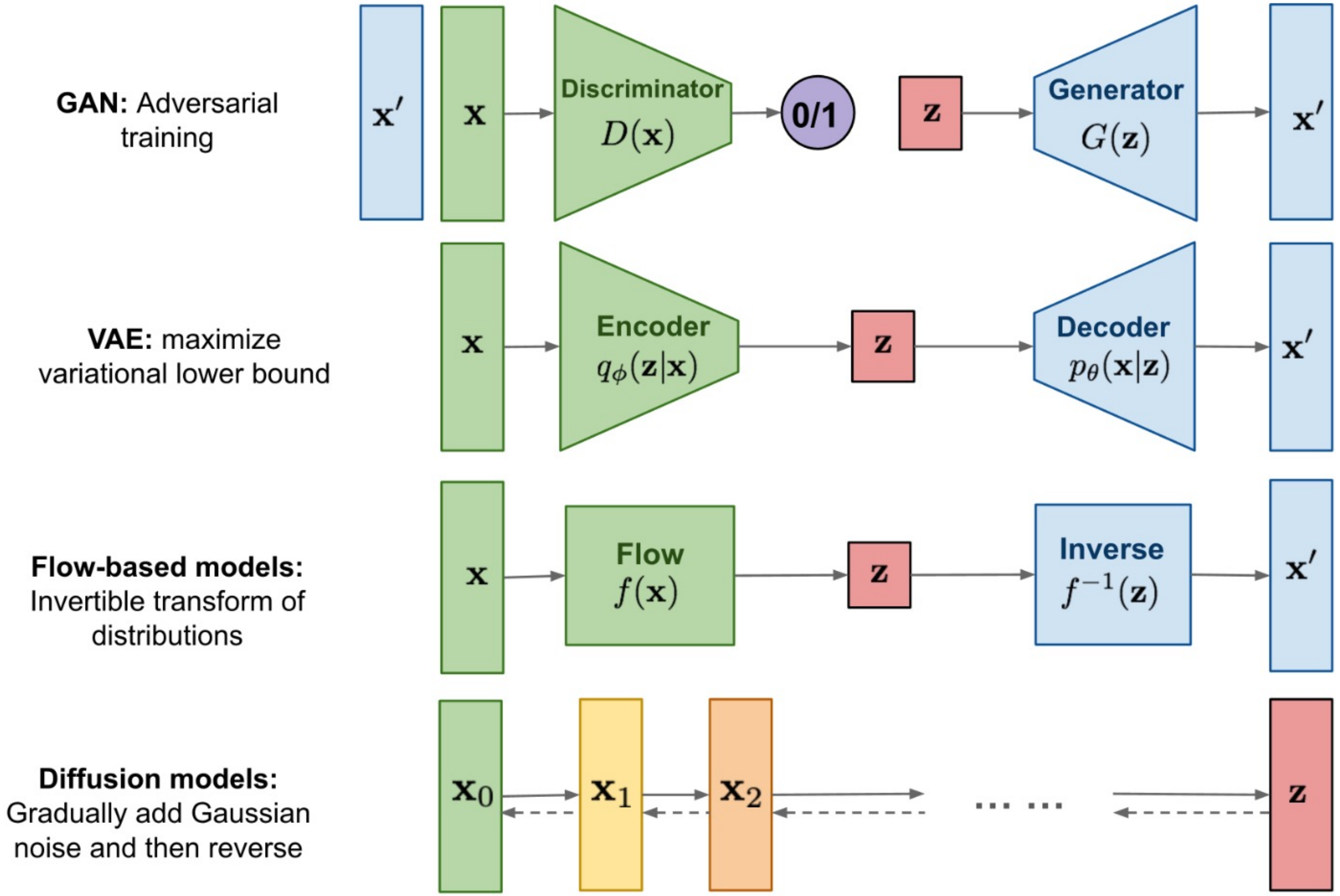
$\xrightarrow{\Psi}$

$q(x)$



$\Psi(x_0) \sim q$

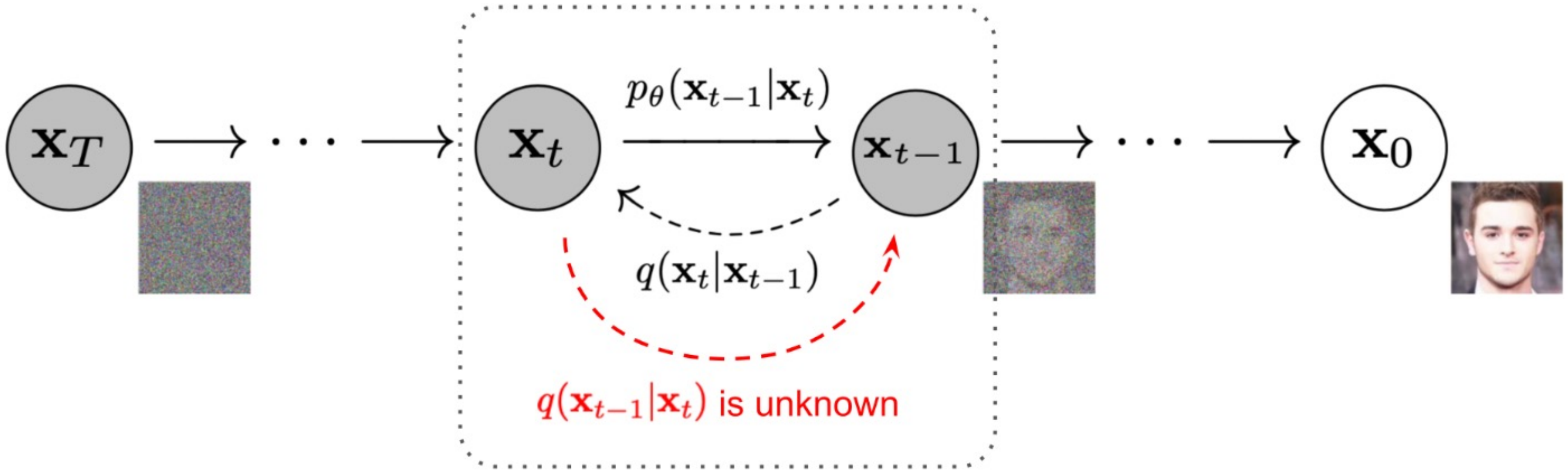
Generative Models



From Lilian Weng's blog

Diffusion Basics

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

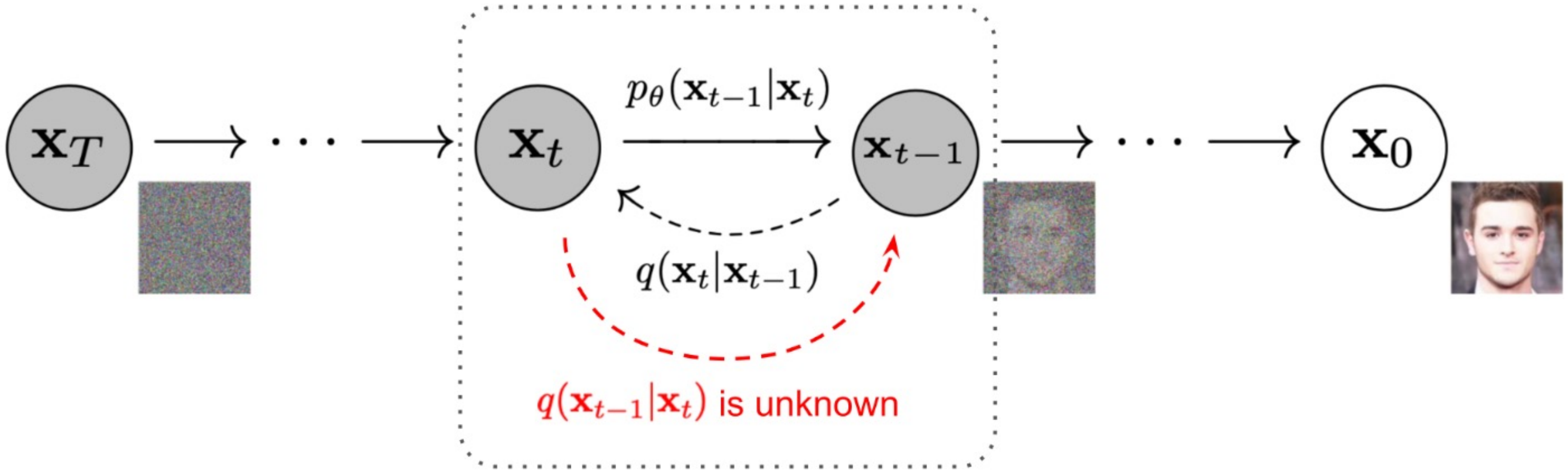


$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} & \text{;where } \boldsymbol{\epsilon}_{t-1}, \boldsymbol{\epsilon}_{t-2}, \dots \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&= \sqrt{\alpha_t \alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2} & \text{;where } \bar{\boldsymbol{\epsilon}}_{t-2} \text{ merges two Gaussians (*).} \\
&= \dots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}
\end{aligned}$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

Diffusion Basics

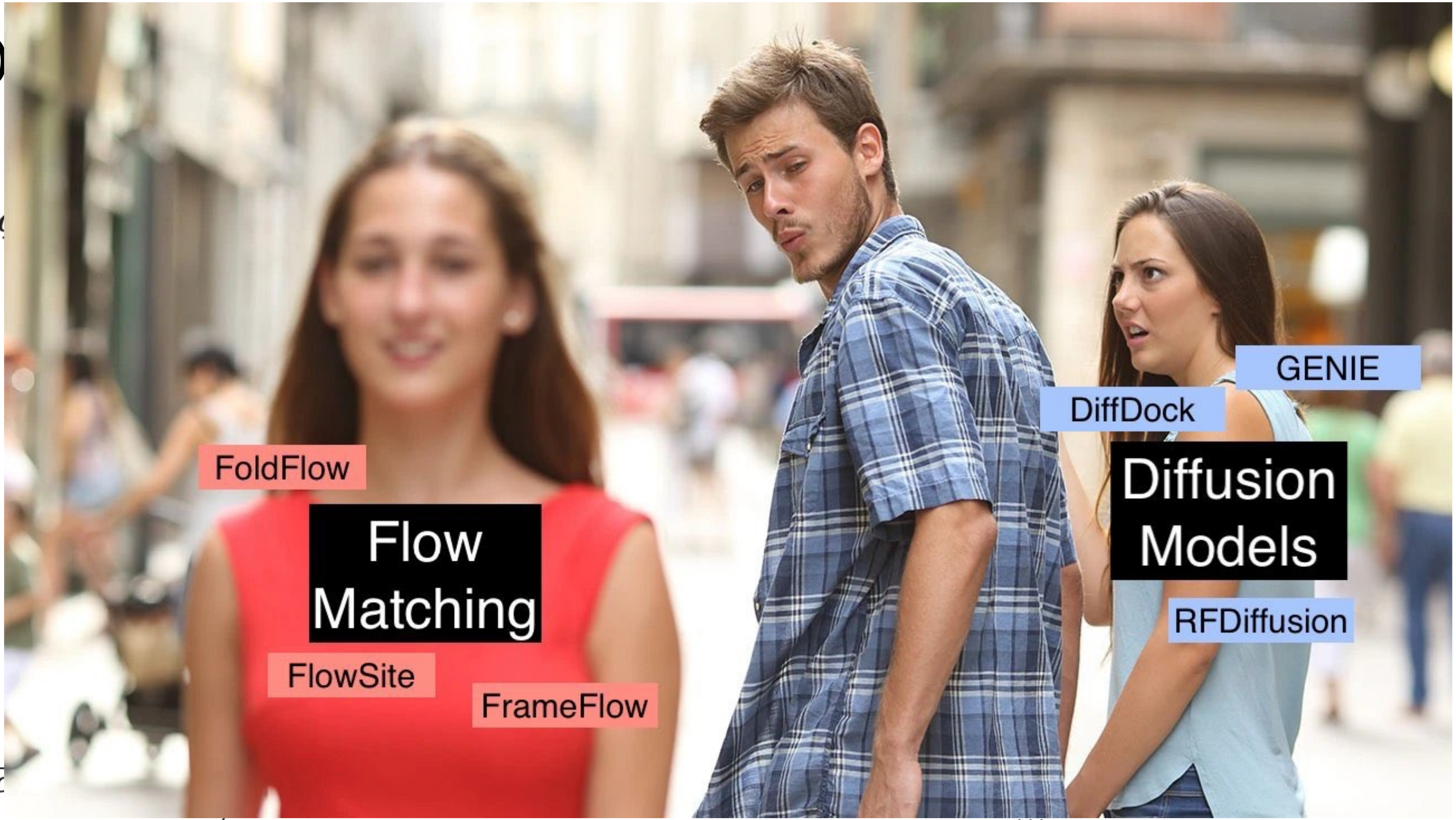
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$



$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} & \text{;where } \boldsymbol{\epsilon}_{t-1}, \boldsymbol{\epsilon}_{t-2}, \dots \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2} & \text{;where } \bar{\boldsymbol{\epsilon}}_{t-2} \text{ merges two Gaussians (*).} \\
&= \dots \\
&\boxed{= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}
\end{aligned}$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

D



$$\mathbf{x}_t = \sqrt{\alpha \ldots}$$

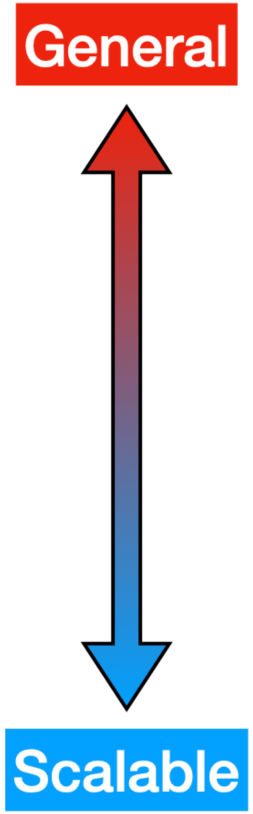
$$= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}} \bar{\boldsymbol{\epsilon}}_{t-2} \qquad \text{;where } \bar{\boldsymbol{\epsilon}}_{t-2} \text{ merges two Gaussians (*).}$$

$$= \ldots$$

$$\boxed{= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}}$$

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})$$

Flow Matching Basics



Generative models

$\cup$

Flows

$\cup$

Flow Matching

$\cup$

Diffusion Models

Flow Matching: Tractable Loss

The gradients of losses coincide:
$\nabla_{\theta} L_{\text{FM}} = \nabla_{\theta} L_{\text{CFM}}$

$$L_{\text{FM}}(q\|p_1) = \min \mathbb{E}_{t,\,p_t(x)} \| v_t(x) - u_t(x) \|^2$$

$$L_{\text{CFM}}(q\|p_1) = \min \mathbb{E}_{t,\,q(x_1),\,p_t(x|x_1)} \| v_t(x) - u_t(x \,|\, x_1) \|^2$$

$$= \min \mathbb{E}_{t,\,q(x_1),\,p(x_0)} \| v_t(x_t) - \dot{x}_t \|^2$$

$x_t \sim p_t(x \,|\, x_1)$
$\dot{x}_t = u_t(x_t \,|\, x_1)$

[L et al. 2022]

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN Dr. TAO HU

Comparison: Flow Matching vs. Diffusion

Algorithm 2: Diffusion training.

Input : dataset q, noise p

Initialize s^θ

while *not converged* **do**

$t \sim \mathcal{U}([0,1])$ ▷ sample time

$x_1 \sim q(x_1)$ ▷ sample data

$x_t = p_t(x_t|x_1)$ ▷ sample conditional prob

Gradient step with

$\nabla_\theta \| s_t^\theta(x_t) - \nabla_{x_t} \log p_t(x_t|x_1) \|^2$

Output: v^θ

$p_t(x_t|x_1)$ closed-form from of SDE $dx_t = f_t dt + g_t dw$

- *Variance Exploding*: $p_t(x|x_1) = \mathcal{N}(x|x_1, \sigma_{1-t}^2 I)$
- *Variance Preserving*: $p_t(x|x_1) = \mathcal{N}(x|\alpha_{1-t}x_1, (1-\alpha_{1-t}^2)I)$

$$\alpha_t = e^{-\frac{1}{2}T(t)}$$

$p(x_0)$ is Gaussian

$p_0(\cdot|x_1) \approx p$

Comparison: Flow Matching *v.s.* Diffusion

Algorithm 1: Flow Matching training.

Input : dataset q, noise p
Initialize v^θ
while *not converged* **do**
$t \sim \mathcal{U}([0,1])$ ▷ sample time
$x_1 \sim q(x_1)$ ▷ sample data
$x_0 \sim p(x_0)$ ▷ sample noise
$x_t = \Psi_t(x_0|x_1)$ ▷ conditional flow
Gradient step with $\nabla_\theta \|v_t^\theta(x_t) - \dot{x}_t\|^2$
Output: v^θ

Algorithm 2: Diffusion training.

Input : dataset q, noise p
Initialize s^θ
while *not converged* **do**
$t \sim \mathcal{U}([0,1])$ ▷ sample time
$x_1 \sim q(x_1)$ ▷ sample data
$x_t = p_t(x_t|x_1)$ ▷ sample conditional prob
Gradient step with
$\nabla_\theta \|s_t^\theta(x_t) - \nabla_{x_t} \log p_t(x_t|x_1)\|^2$
Output: v^θ

$p_t(x_t \mid x_1)$ general
$p(x_0)$ is general

$p_t(x_t \mid x_1)$ closed-form from of SDE $dx_t = f_t dt + g_t dw$

- *Variance Exploding*: $p_t(x \mid x_1) = \mathcal{N}(x \mid x_1, \sigma_{1-t}^2 I)$
- *Variance Preserving*: $p_t(x \mid x_1) = \mathcal{N}(x \mid \alpha_{1-t} x_1, (1-\alpha_{1-t}^2) I)$

$$\alpha_t = e^{-\frac{1}{2}T(t)}$$

$p(x_0)$ is Gaussian
$p_0(\cdot \mid x_1) \approx p$

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
Dr. TAO HU

Beyond Diffusion Models

- A more general framework that simplifies diffusion models. This generalized framework can provide more freedom for the model design.

- A special case in this framework can be used to facilitate sampling.

- Extension to Schrodinger Bridge between two distributions.

Beyond Diffusion Models

- A more general framework that simplifies diffusion models. This generalized framework can provide more freedom for the model design.

- A special case in this framework can be used to facilitate sampling.

- Extension to Schrodinger Bridge between two distributions.

Open questions:

- Ordinary Differential Equation or Stochastic Differential Equation?
  - Stochastic Interpolants: A Unifying Framework for Flows and Diffusions. Albergo et al.
  - Elucidating the Design Space of Diffusion-Based Generative Models, Karras et al, NeurIPS22.
  - Minimizing Trajectory Curvature of ODE-based Generative Models. Lee et al, ICML23
- Latent space or pixel space?
  - NeurIPS23 Tutorial:
    - Latent Diffusion Models: Is the Generative AI Revolution Happening in Latent Space?

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN Dr. TAO HU

ZigMa: A DiT-style Zigzag Mamba Diffusion Model

Vincent Tao Hu, Stefan Andreas Baumann, Ming Gui, Olga Grebenkova, Pingchuan Ma, Johannes Fischer, and Björn Ommer







Background

- Quadratic Complexity in Attention, it hinders the application on many downstream tasks that requires long token number, e.g., high-resolution image generation, long video generation, etc.

- Diffusion Models is not so generalizable, and flexible. Painful noise schedule and need to be a gaussian distribution.

Background

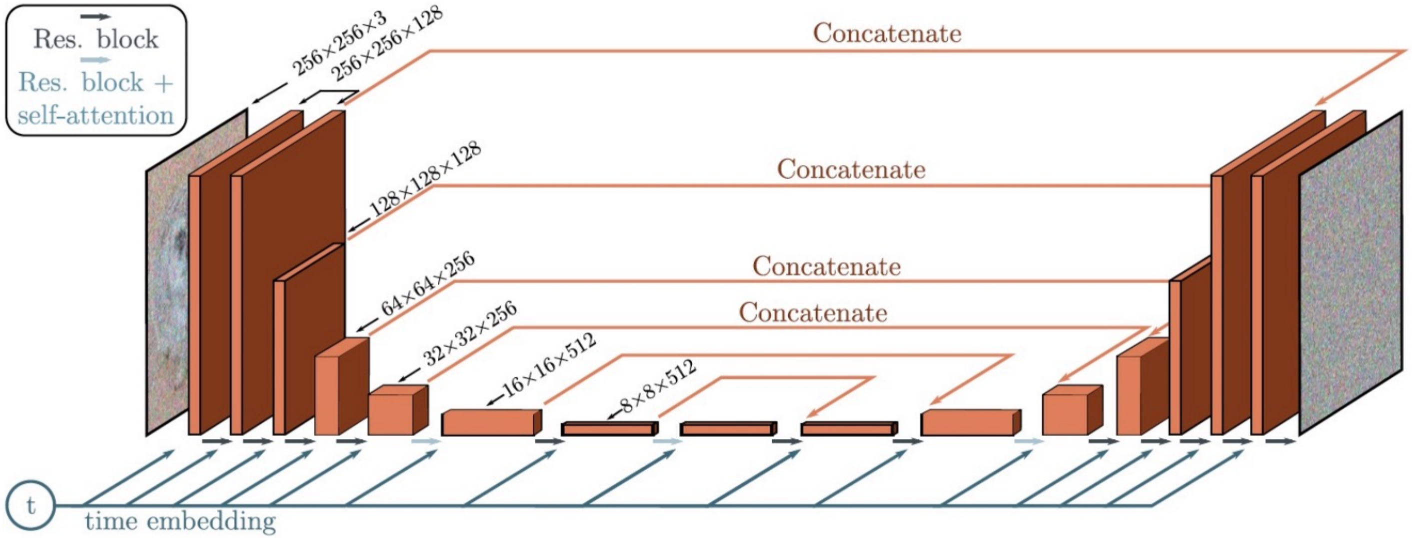
- Quadratic Com… hinders the application on many downstream tasks … umber, e.g., high-resolution image generatio… on, etc.

Mamba:
Linear Complexity

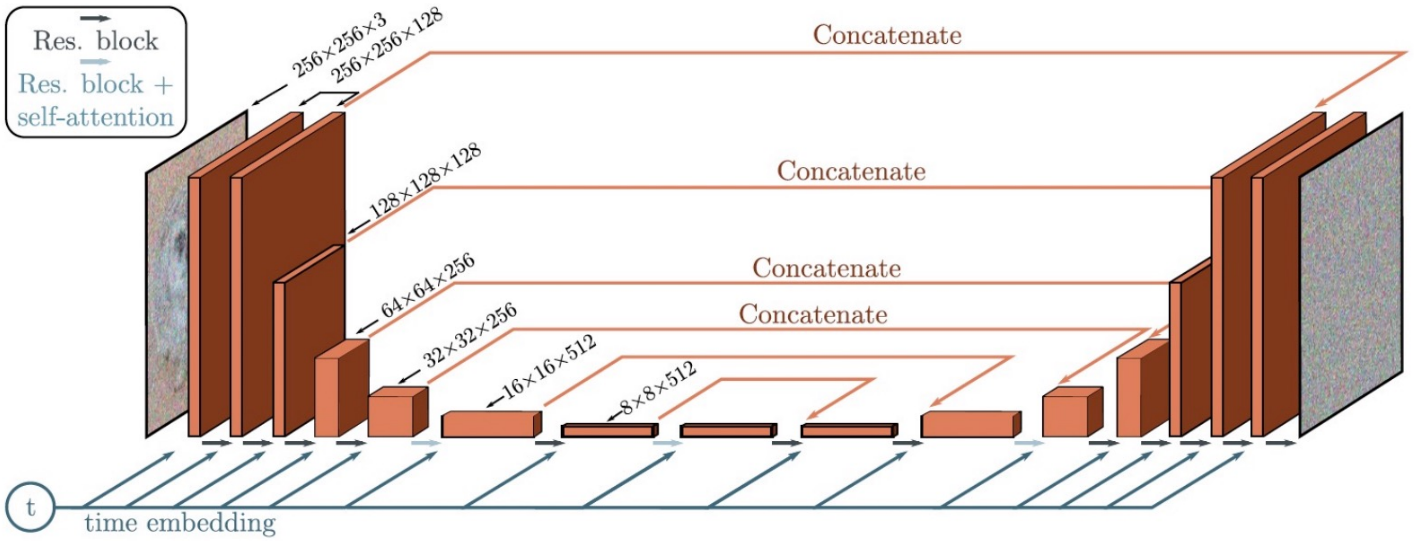
- Diffusion Model… nd flexible. Painful noise schedule and need to be…

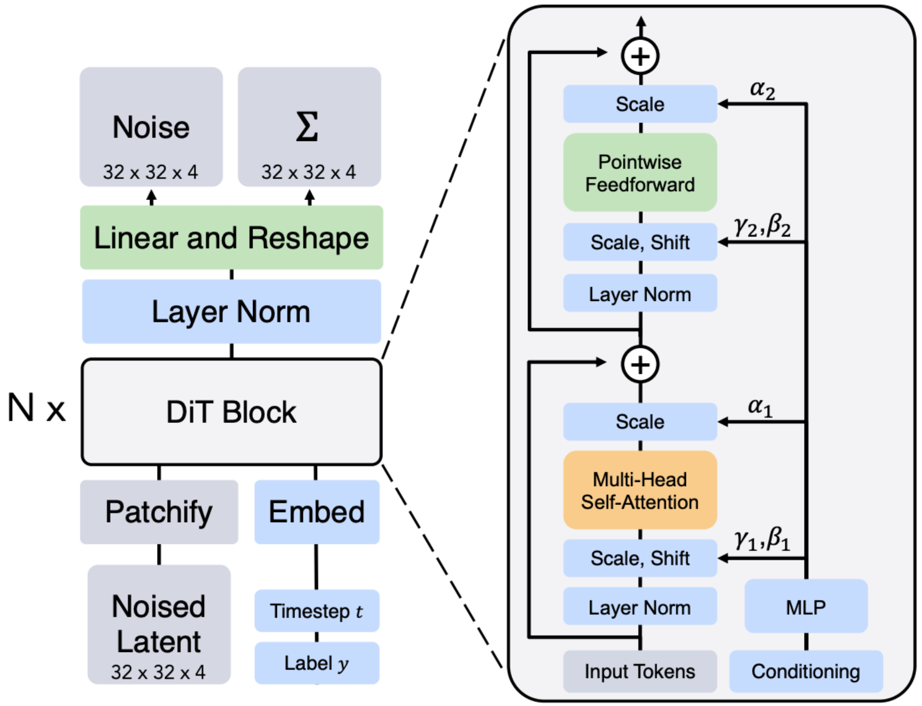
New framework:
Stochastic Interpolant

Background: Network Choice of (f(x(t),t))



Background: Network Choice of (f(x(t),t))

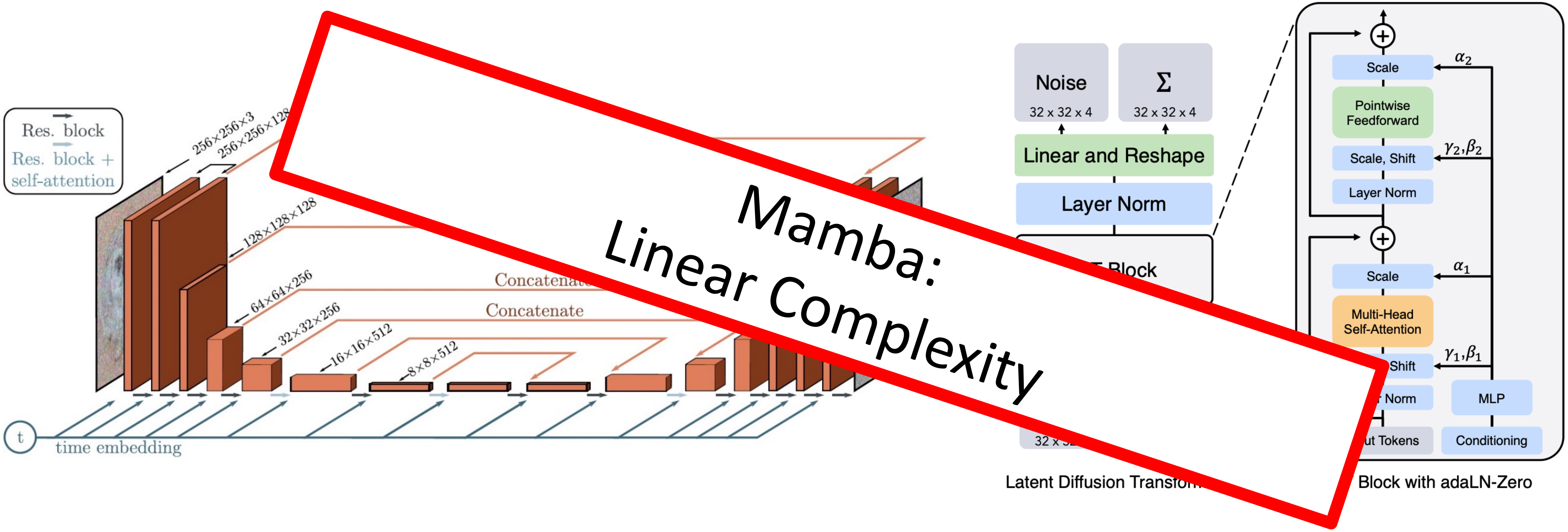




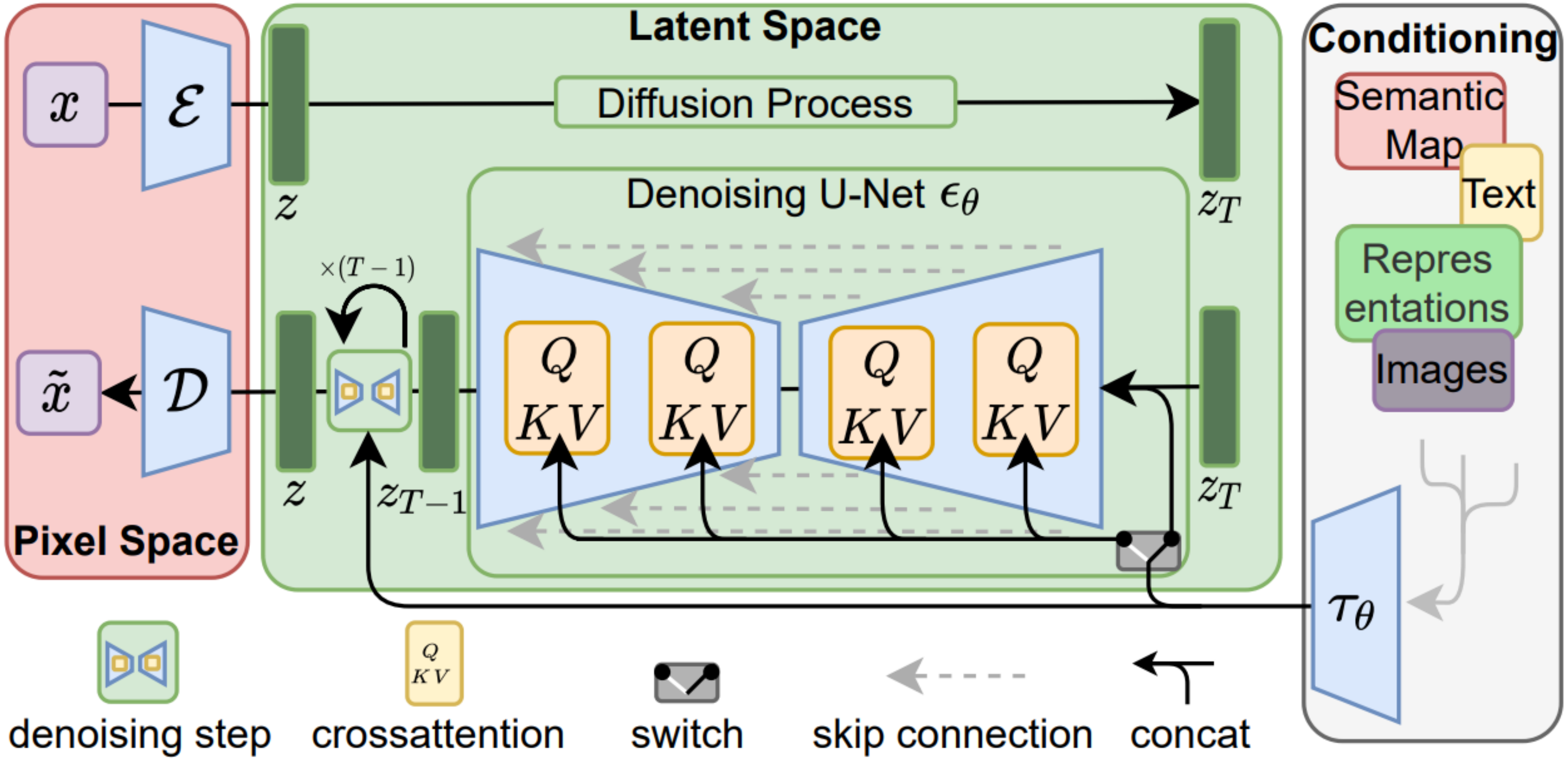
Latent Diffusion Transformer

DiT Block with adaLN-Zero

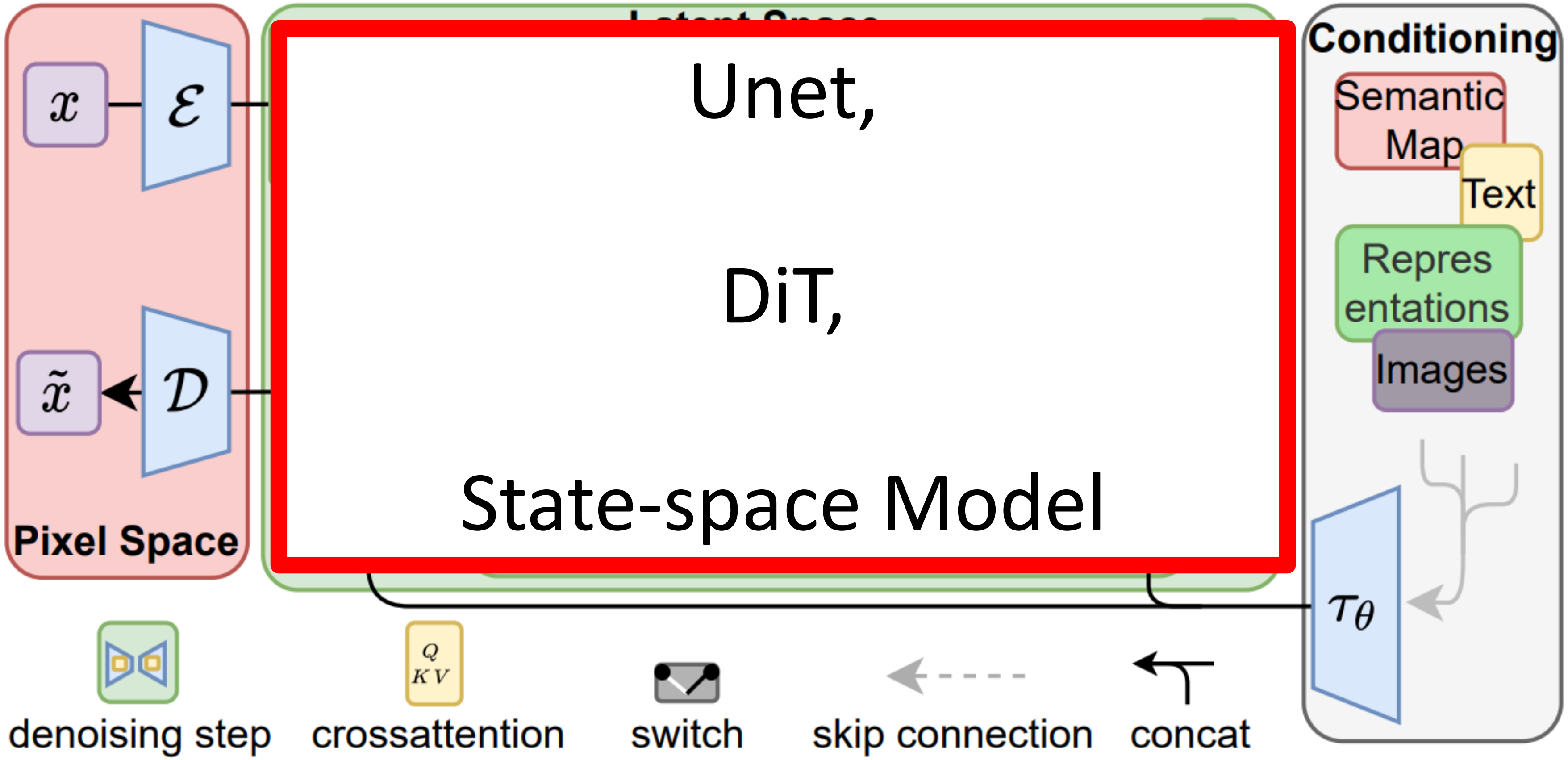
Background: Network Choice of (f(x(t),t))



Background: Network Choice of (f(x(t),t))



Background: Network Choice of (f(x(t),t))



Why State-Space Model?

- for image 256x256, latent space 4x32x32.
  - What if we need to generate 10k x 10k image?
  - What if we need to generate a 1k frames of videos?

Background: Mamba- a new State-Space Model

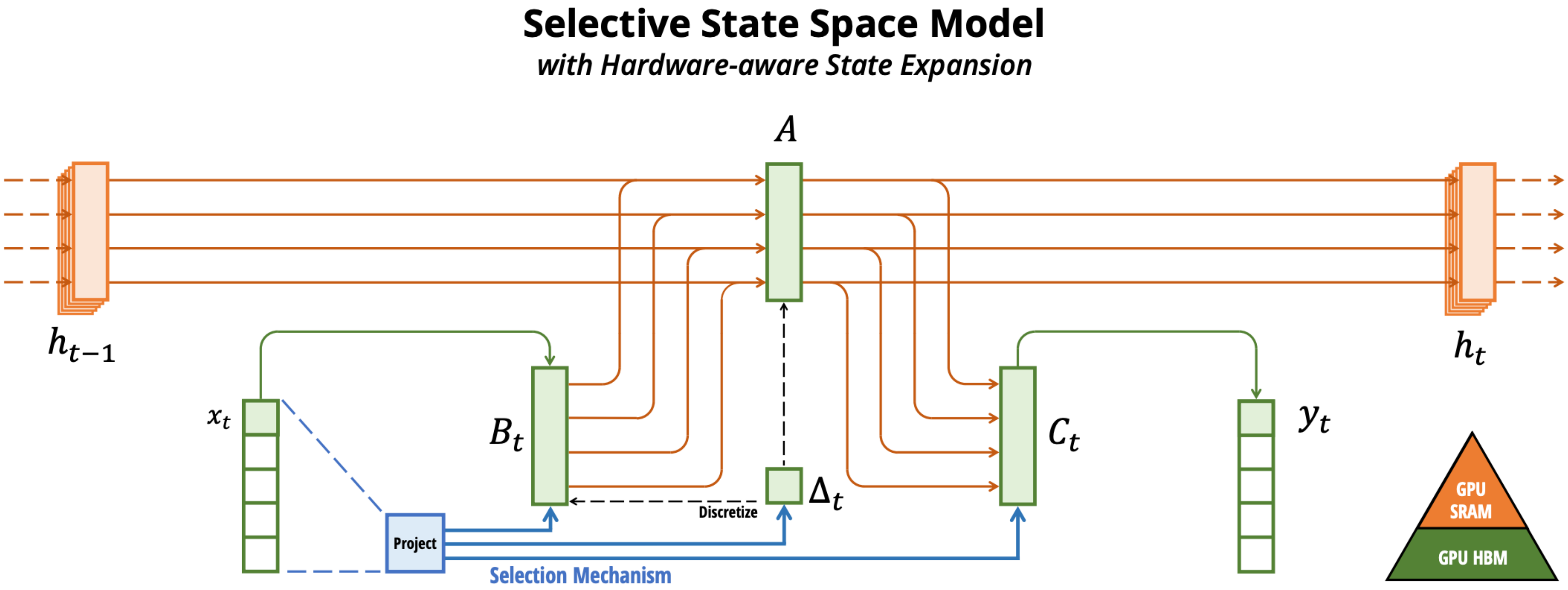
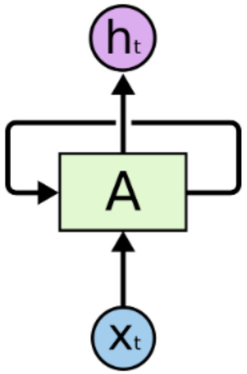
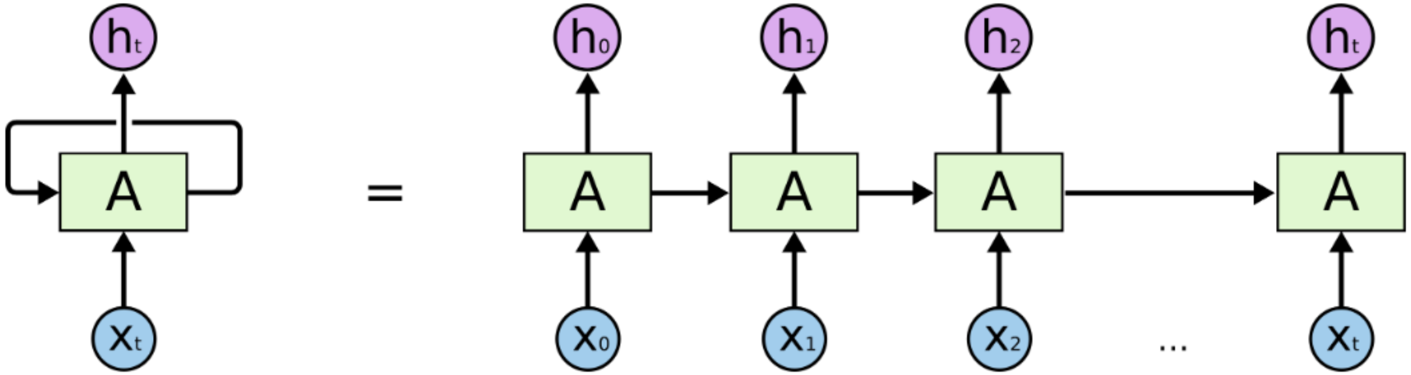


Figure 1: (**Overview**.) Structured SSMs independently map each channel (e.g. $D = 5$) of an input x to output y through a higher dimensional latent state h (e.g. $N = 4$). Prior SSMs avoid materializing this large effective state (DN, times batch size B and sequence length L) through clever alternate computation paths requiring time-invariance: the $(\Delta, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ parameters are constant across time. Our selection mechanism adds back input-dependent dynamics, which also requires a careful hardware-aware algorithm to only materialize the expanded states in more efficient levels of the GPU memory hierarchy.

RNN

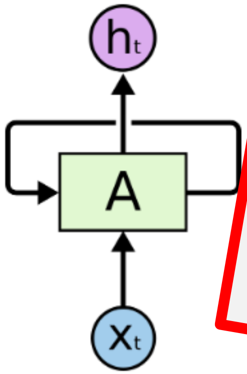


Recurrent Neural Networks have loop



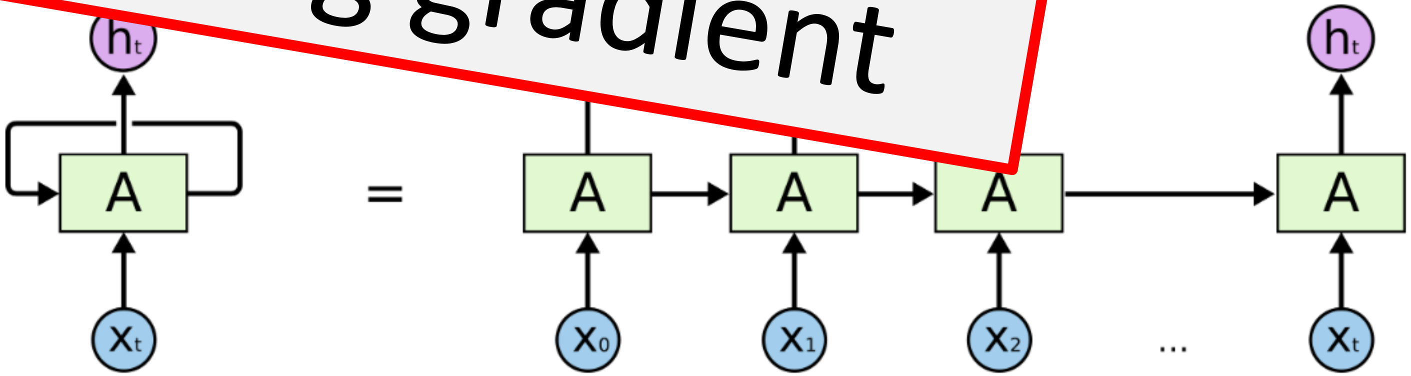
An unrolled recurrent neural network.

RNN



Recurrent Neural Networks have loop

unscalable training and diminishing gradient



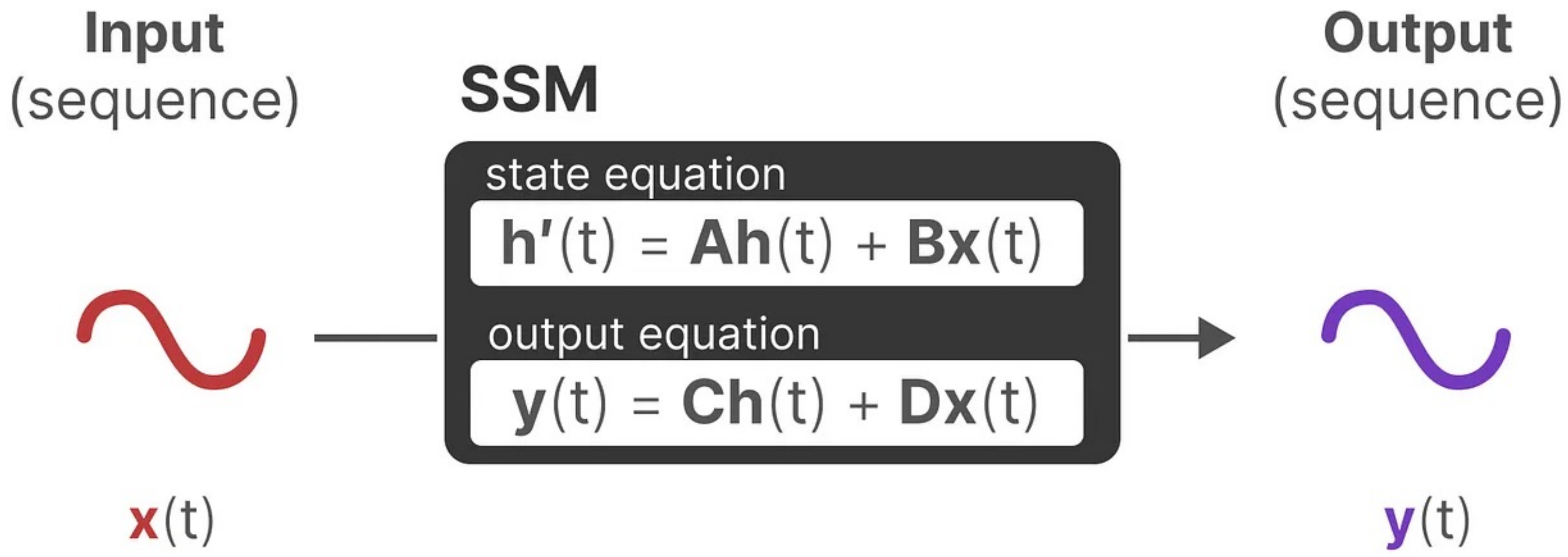
An unrolled recurrent neural network.

RNN

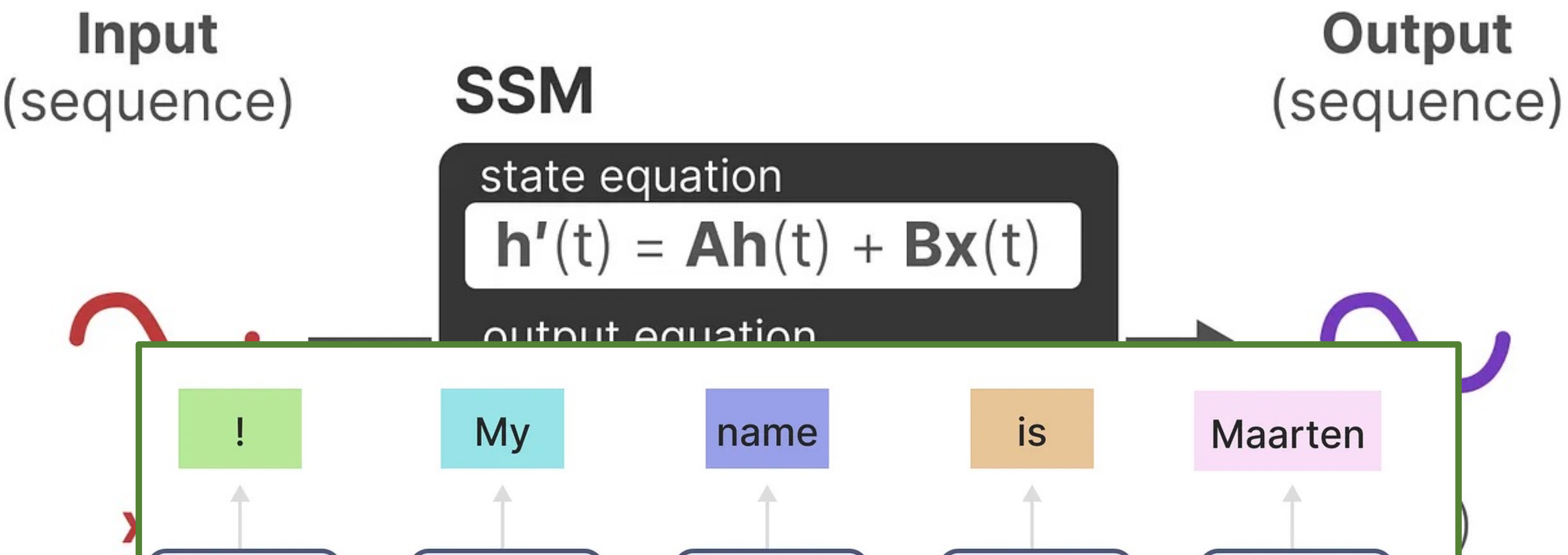
	Training	Inference
Transformers	**Fast!** (parallelizable)	**Slow...** (scales **quadratically** with sequence length)
RNNs	**Slow...** (not parallelizable)	**Fast!** (scales **linearly** with sequence length)

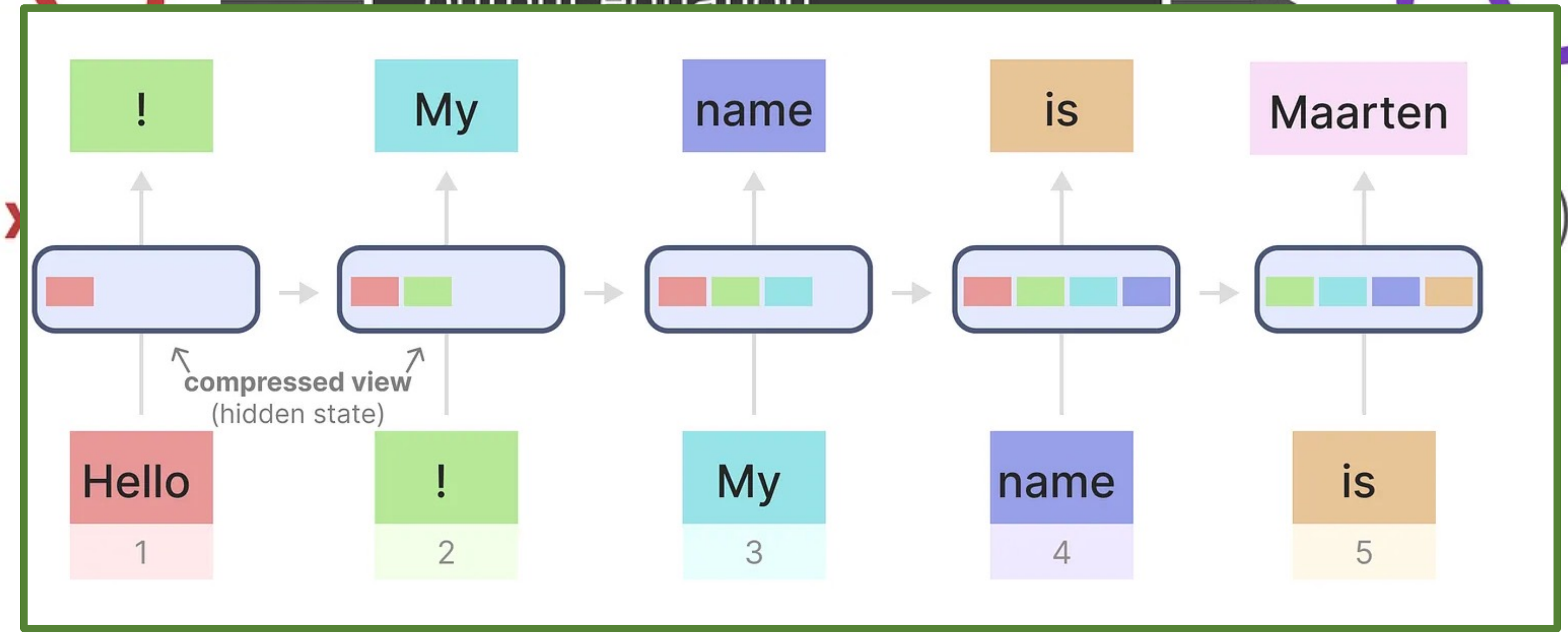
https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mamba-and-state

A basic equations of State-Space Model



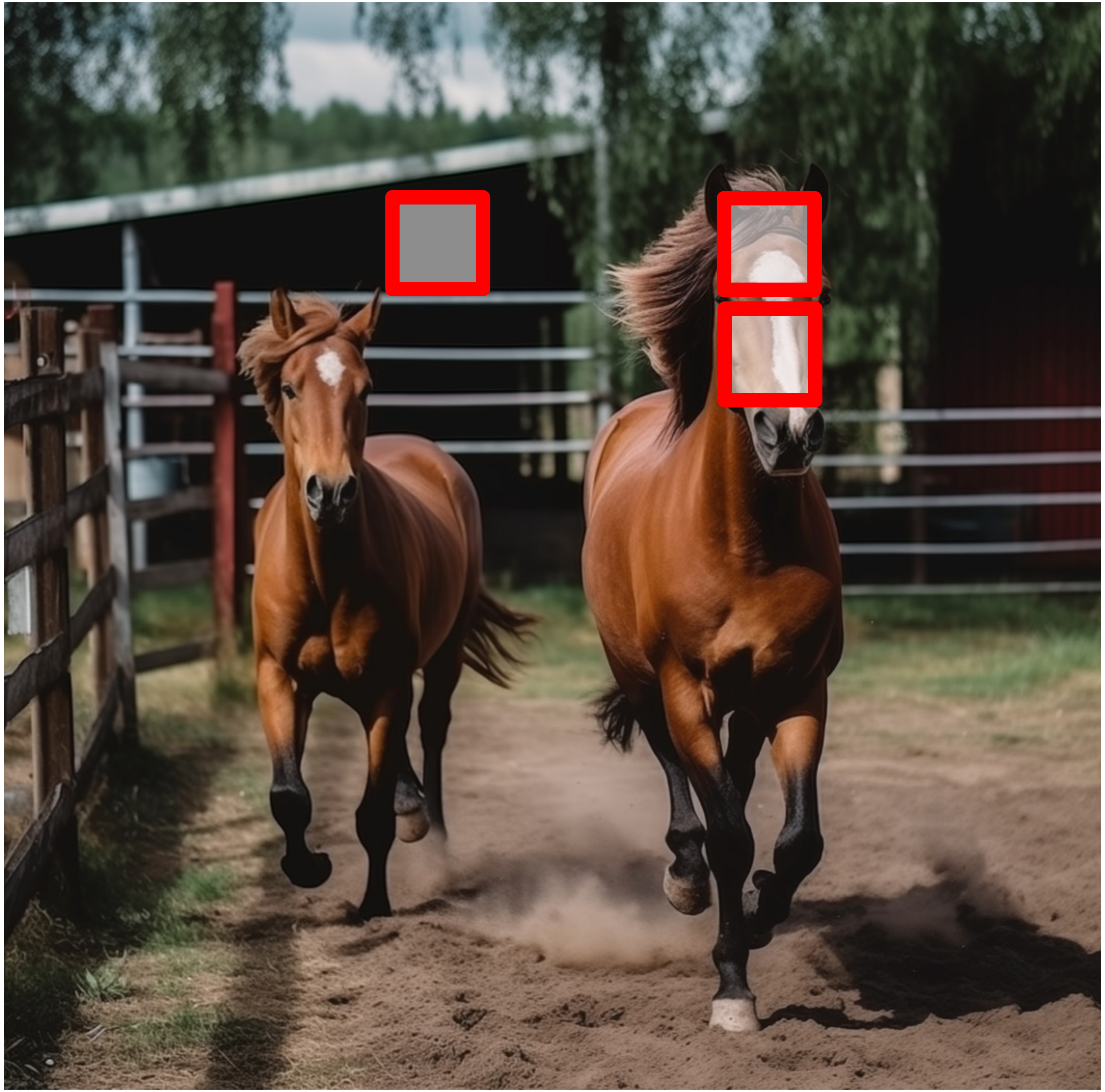
A basic equations of State-Space Model

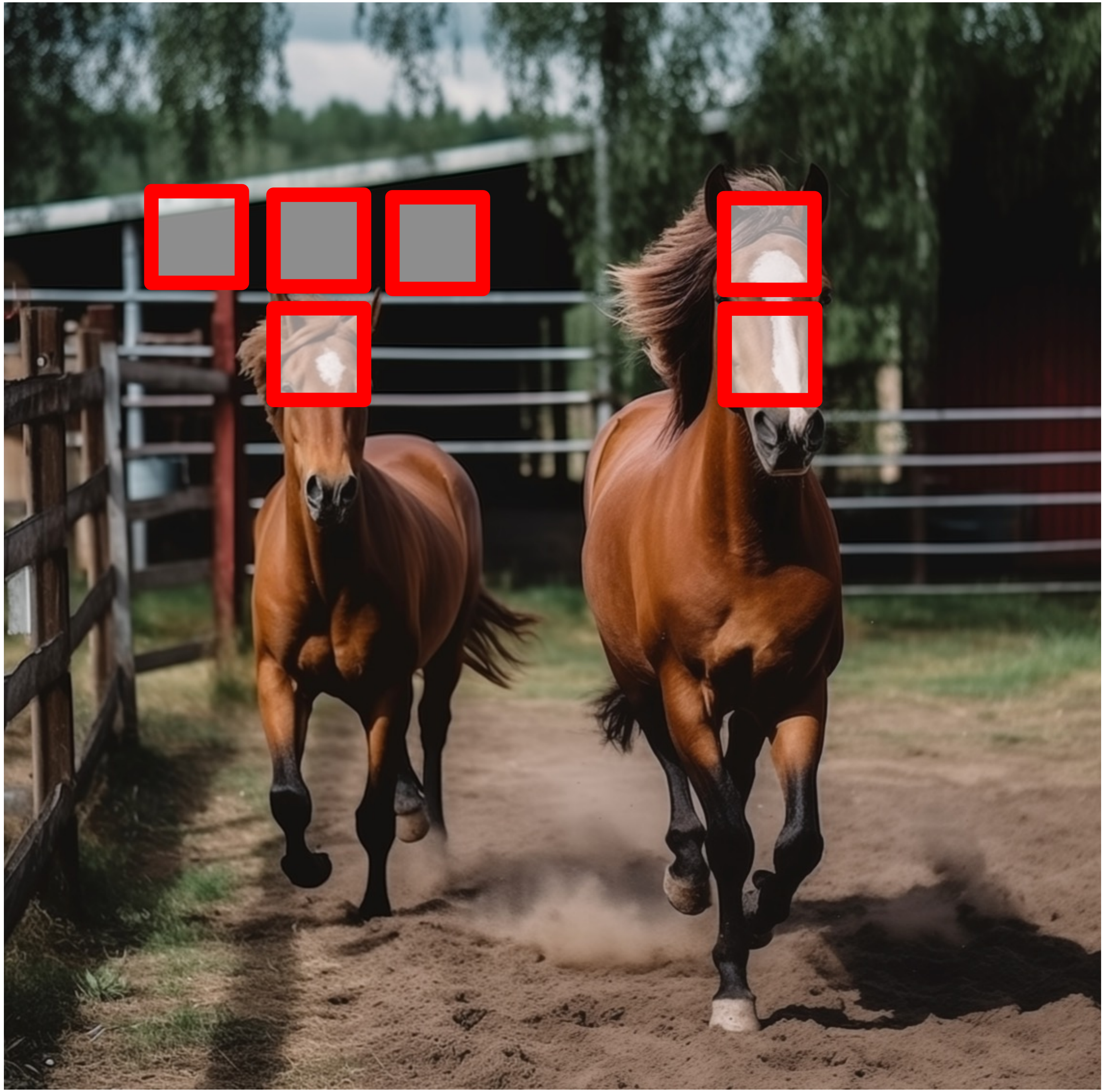


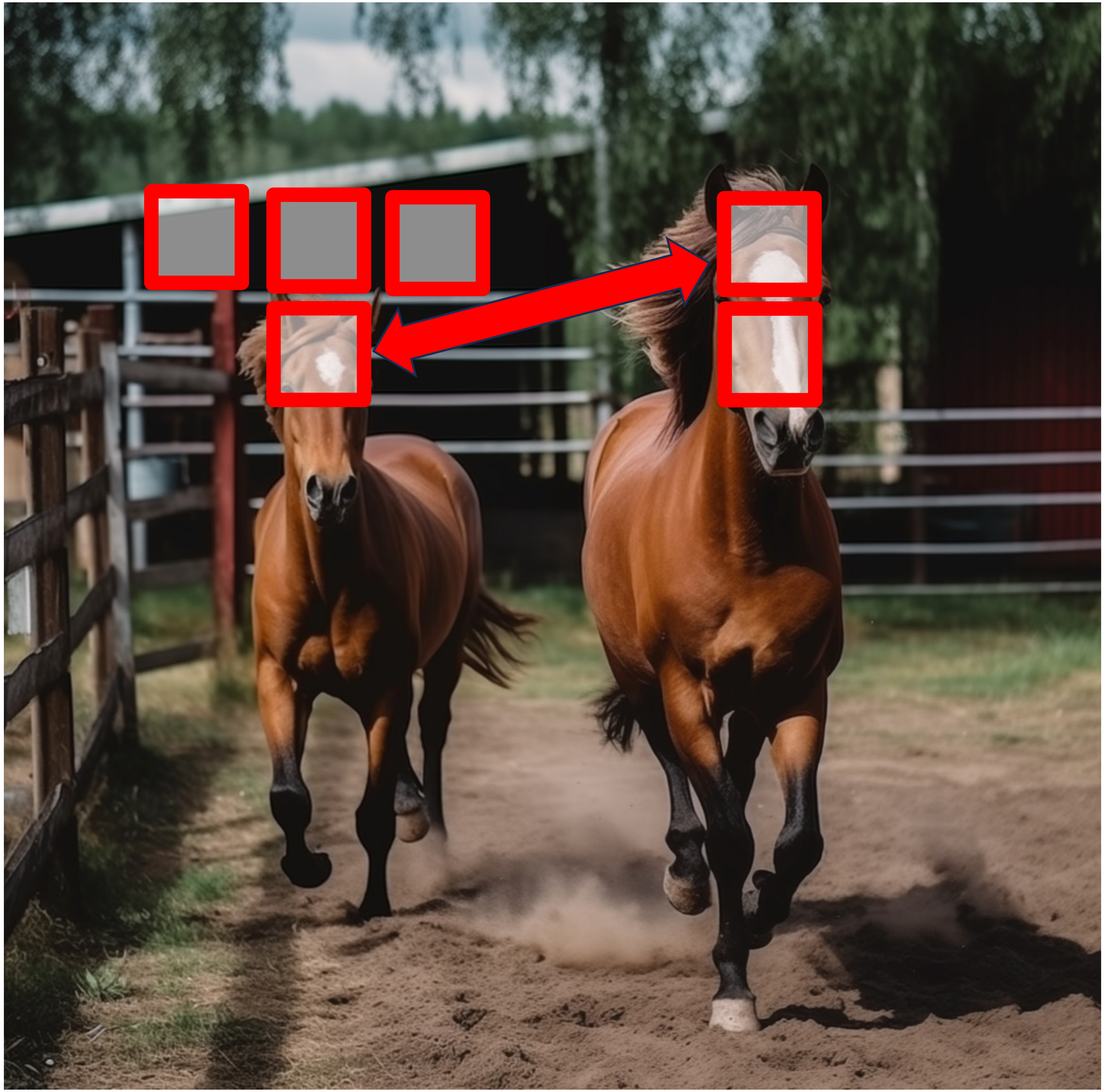


Challenges for State-Space Model

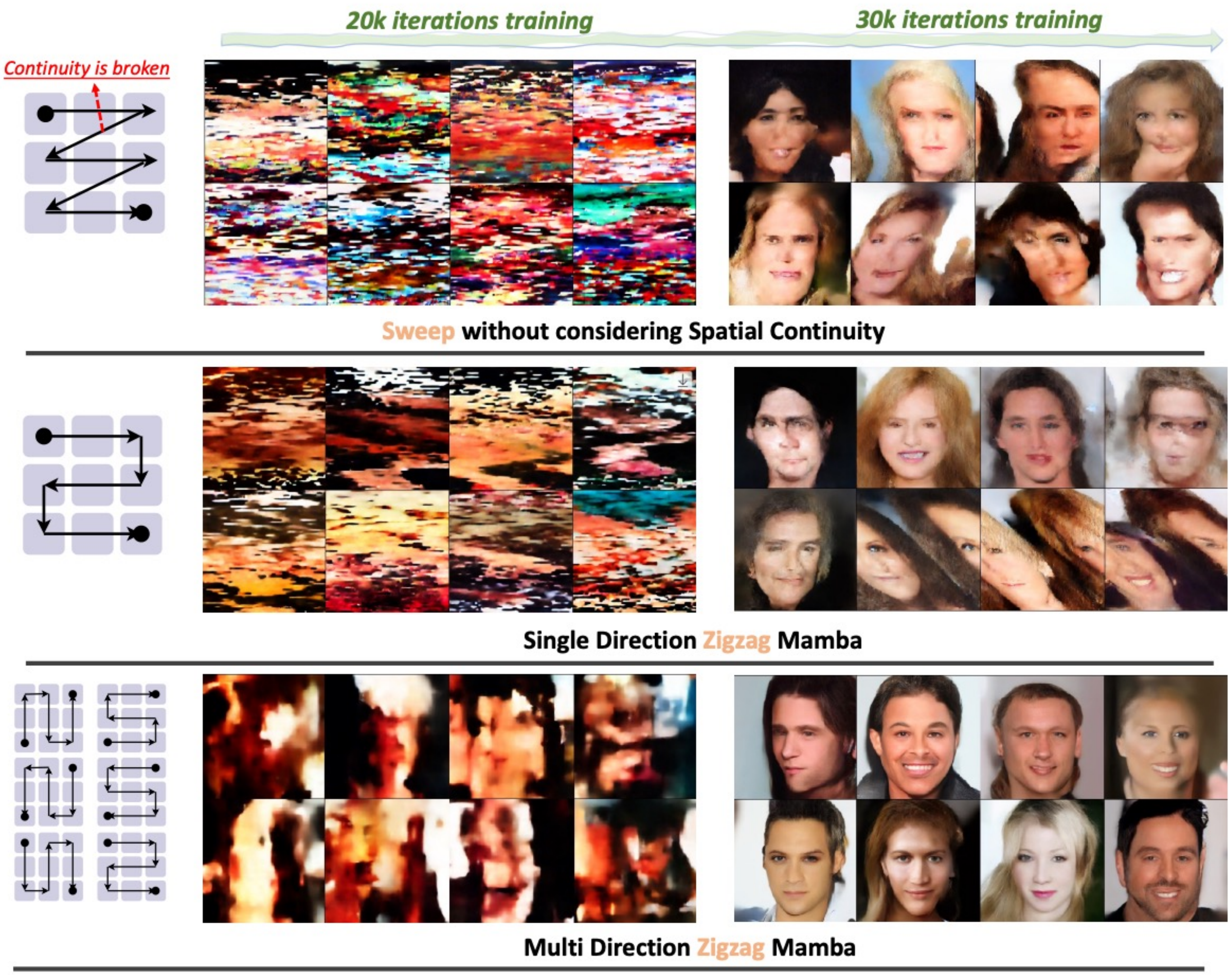
- State-space Model is similar to a Linear-RNN, so it’s sensitive to the order of the token
- Neighbourhood tokens need to be semantically similar.
- Challenging to fuse various modalities e.g., image and text.







Our Solution



Method

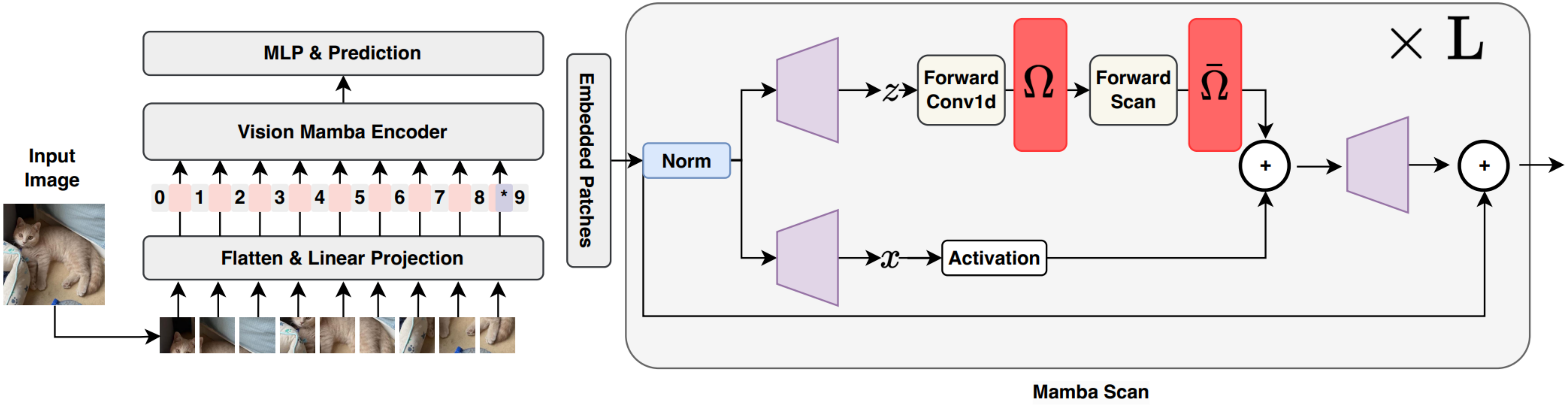


Figure 2: ZigMa. Our backbone is structured in L layers, mirroring the style of DiT [65]. We use the single-scan Mamba block as the primary reasoning module across different patches. To ensure the network is positionally aware, we've designed an arrange-rearrange scheme based on the single-scan Mamba. Different layers follow pairs of unique rearrange operation Ω and reverse rearrange $\bar{\Omega}$, optimizing the position-awareness of the method.

Method

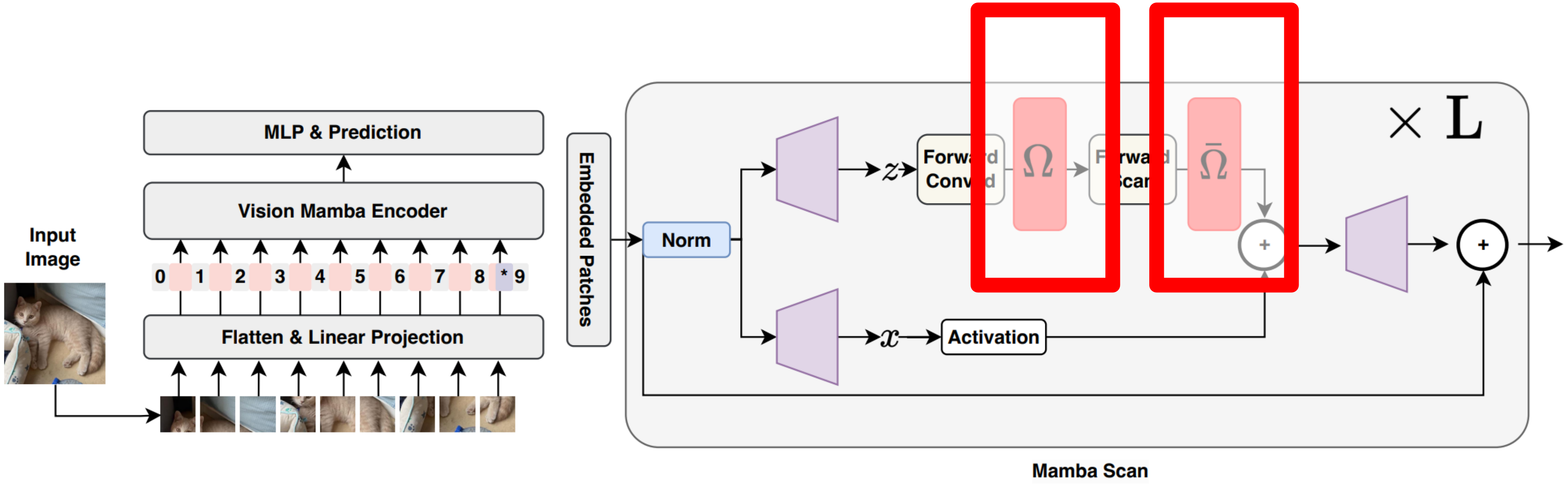
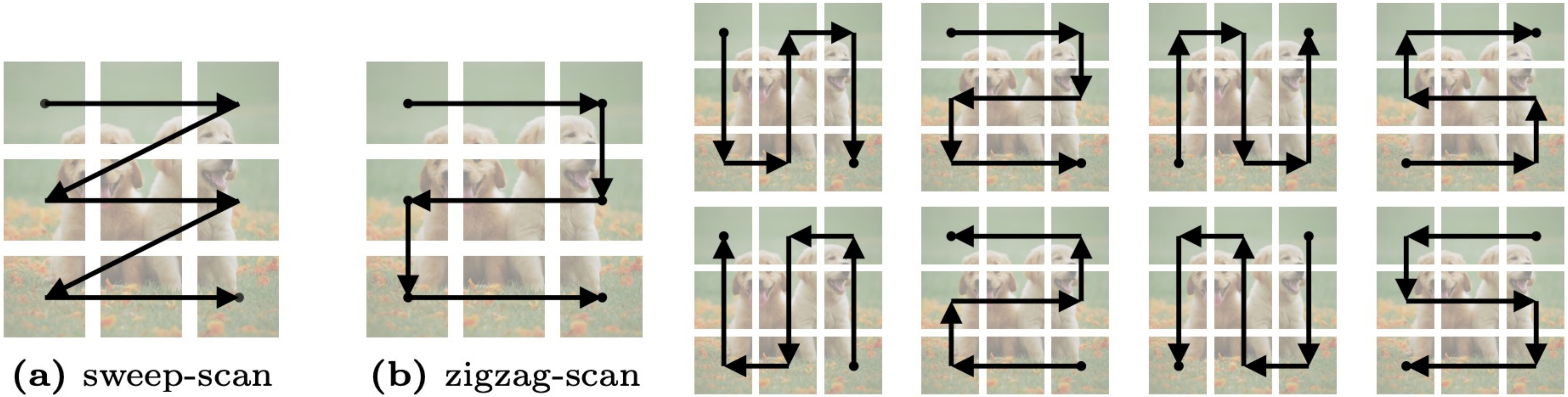


Figure 2: ZigMa. Our backbone is structured in L layers, mirroring the style of DiT [65]. We use the single-scan Mamba block as the primary reasoning module across different patches. To ensure the network is positionally aware, we've designed an arrange-rearrange scheme based on the single-scan Mamba. Different layers follow pairs of unique rearrange operation Ω and reverse rearrange $\bar{\Omega}$, optimizing the position-awareness of the method.

Method



(a) sweep-scan (b) zigzag-scan

(c) zigzag-scan with 8 schemes

Method

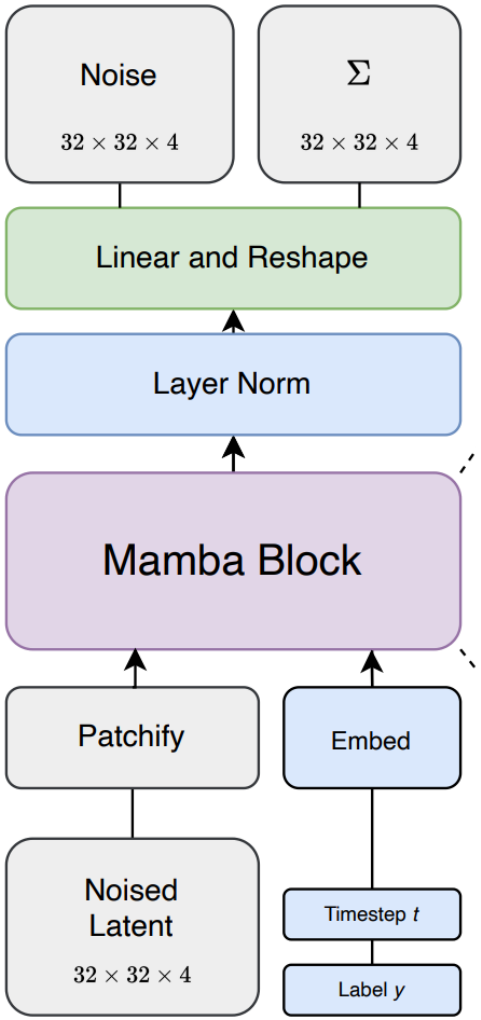


Figure 4: The Detail of our Zigzag Mamba block. The detail of Mamba Scan is shown in Figure 2. The condition can include a timestep and a text prompt. These are fed into an MLP, which separately modulates the Mamba scan for long sequence modeling and cross attention for multi-modal reasoning.

Method

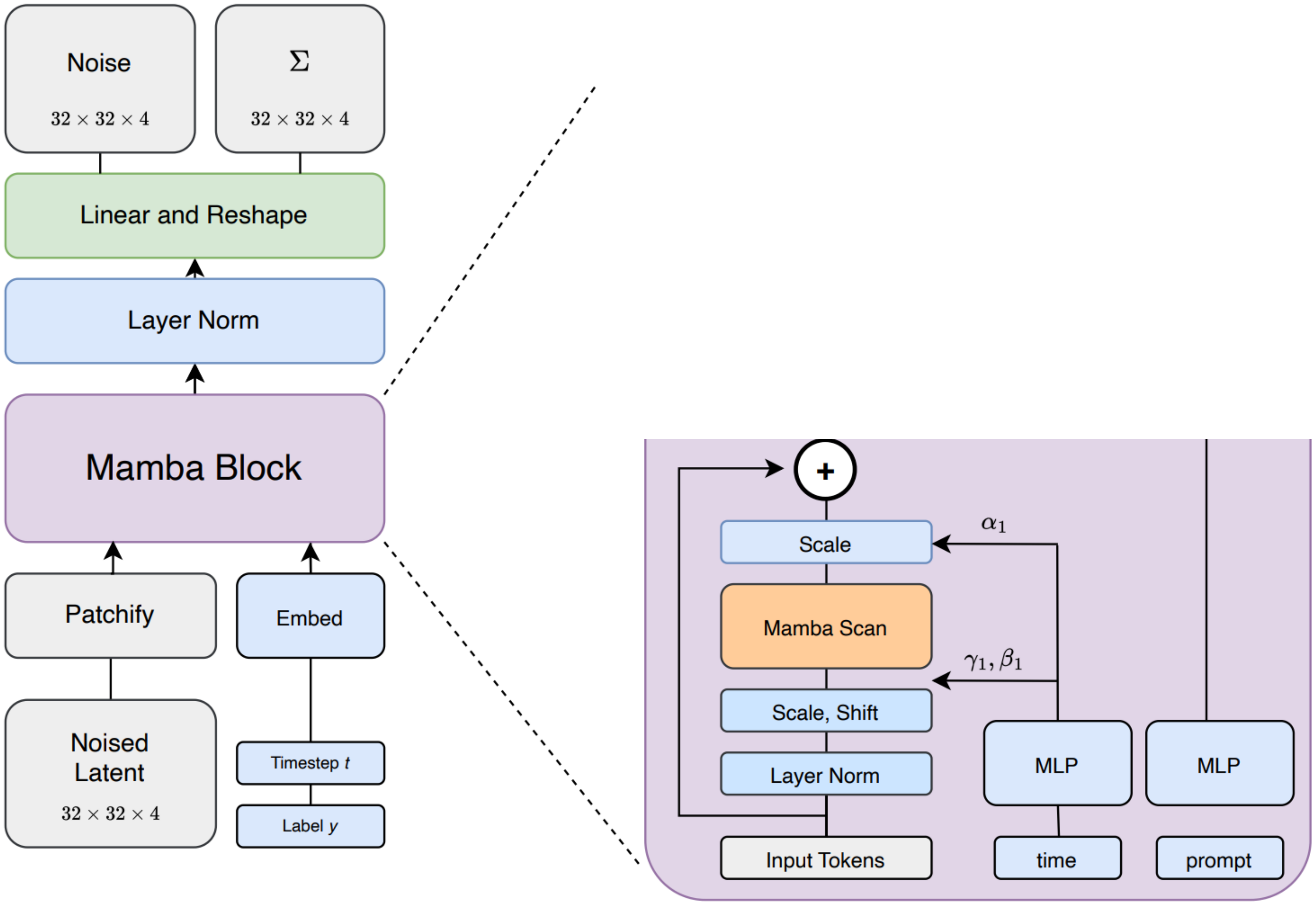


Figure 4: The Detail of our Zigzag Mamba block. The detail of Mamba Scan is shown in Figure 2. The condition can include a timestep and a text prompt. These are fed into an MLP, which separately modulates the Mamba scan for long sequence modeling and cross attention for multi-modal reasoning.

Method

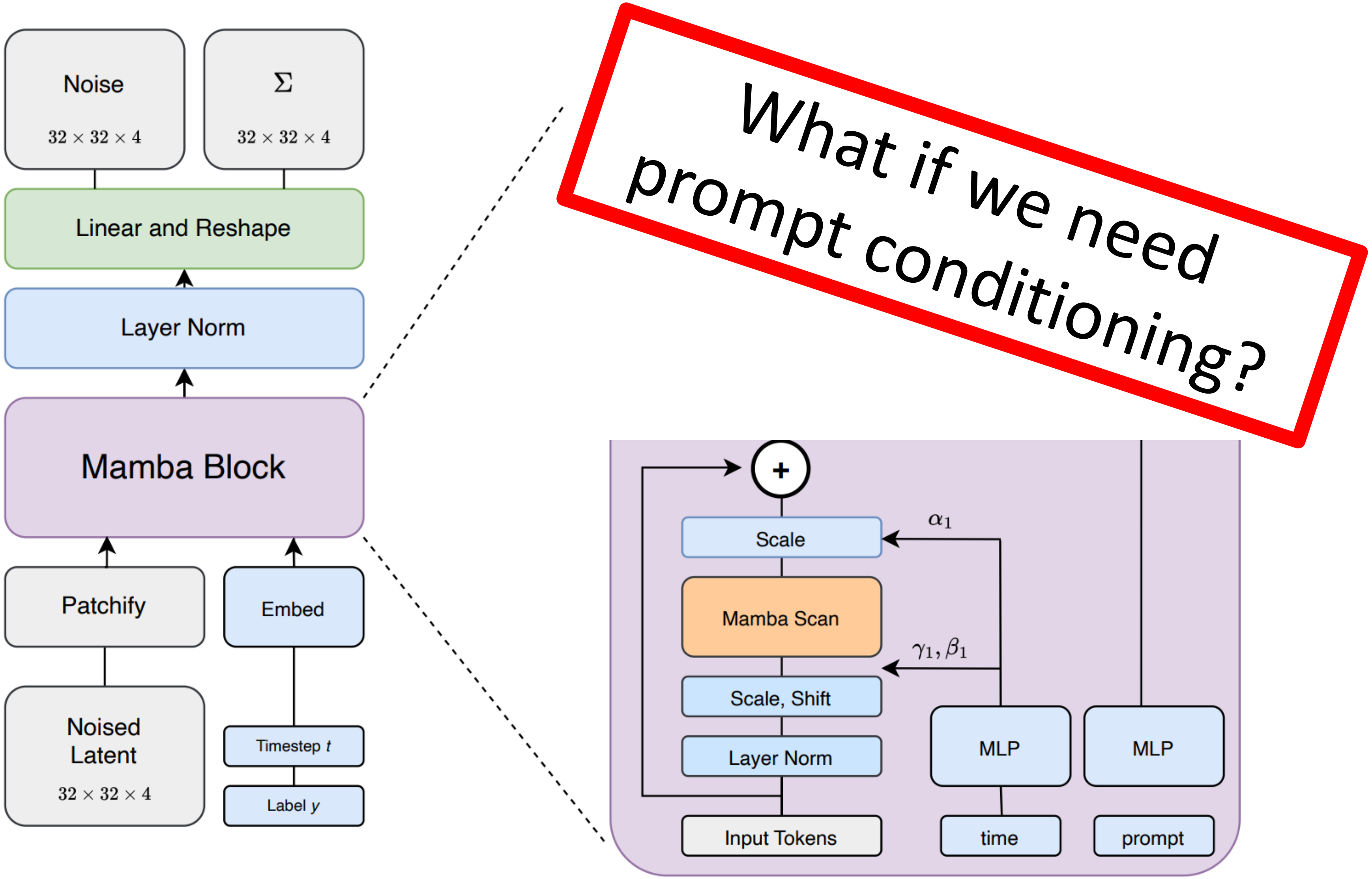


Figure 4: The Detail of our Zigzag Mamba block. The detail of Mamba Scan is shown in Figure 2. The condition can include a timestep and a text prompt. These are fed into an MLP, which separately modulates the Mamba scan for long sequence modeling and cross attention for multi-modal reasoning.

Method

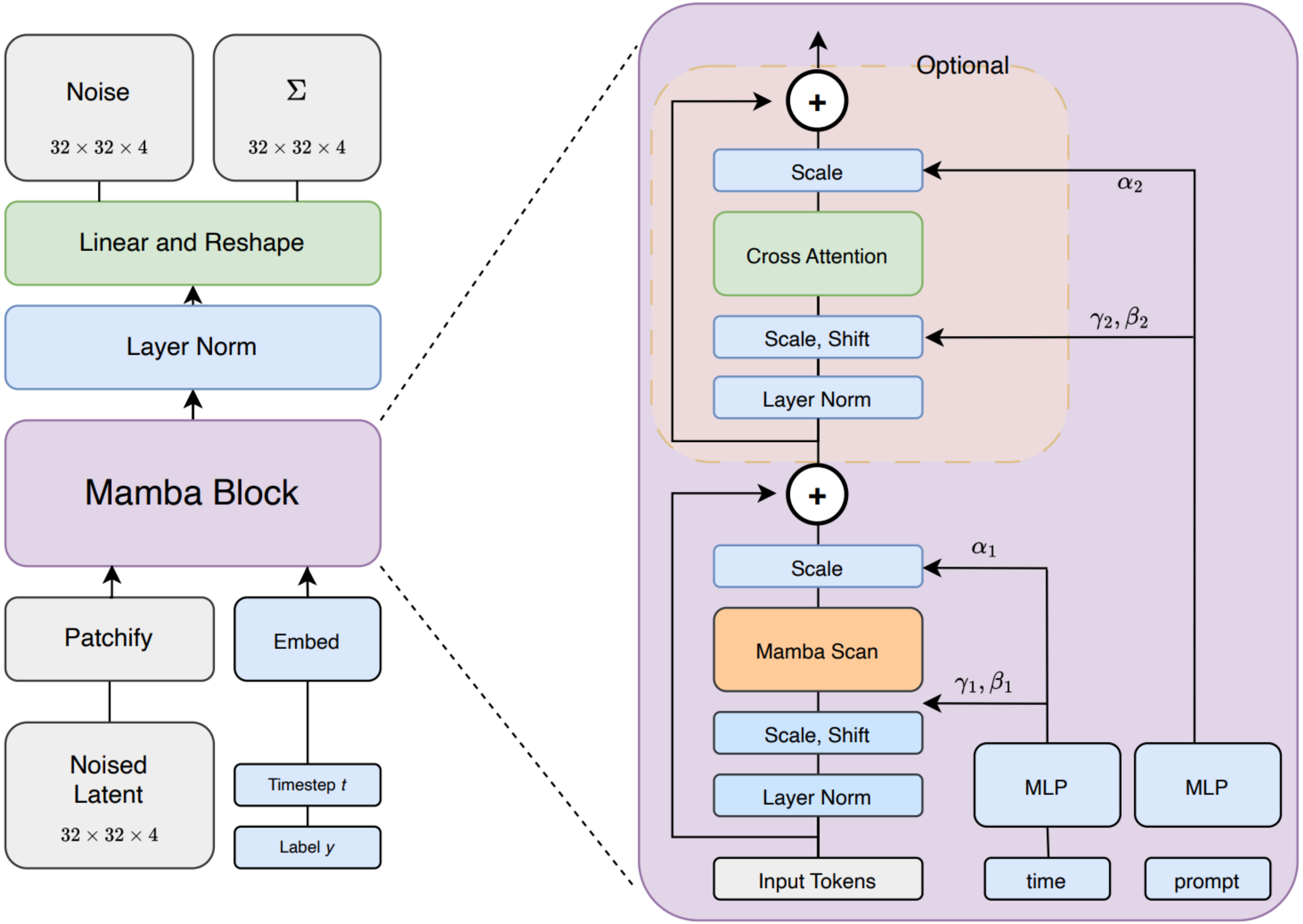
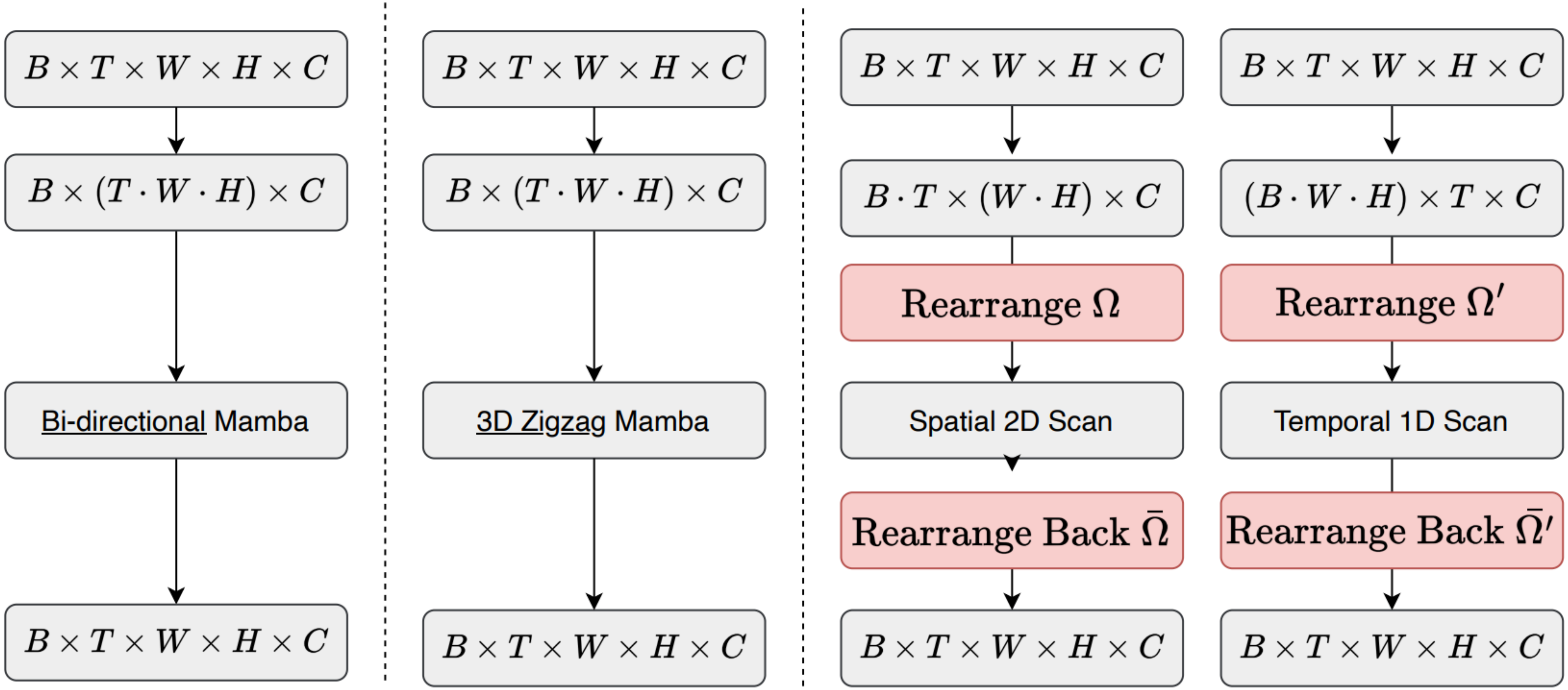


Figure 4: The Detail of our Zigzag Mamba block. The detail of Mamba Scan is shown in Figure 2. The condition can include a timestep and a text prompt. These are fed into an MLP, which separately modulates the Mamba scan for long sequence modeling and cross attention for multi-modal reasoning.

Generalizing to 3D video



(a). sweep-scan (b). 3D zigzag-scan (c). 2D zigzag scan + 1D normal scan

Diffusion Framework

$$\mathcal{L}_{\mathrm{s}}(\theta) = \int_0^T \mathbb{E}[\|\sigma_t \mathbf{s}_\theta(\mathbf{x}_t, t) + \boldsymbol{\varepsilon}\|^2]\mathrm{d}t. \tag{11}$$

$$\mathcal{L}_{\mathrm{v}}(\theta) = \int_0^T \mathbb{E}[\|\mathbf{v}_\theta(\mathbf{x}_t, t) - \dot{\alpha}_t \mathbf{x}_* - \dot{\sigma}_t \boldsymbol{\varepsilon}\|^2]\mathrm{d}t, \tag{12}$$

where θ represents the Zigzag Mamba network that we described in the previous section, we adopt the linear path for training, due to its simplicity and relatively straight trajectory:

$$\alpha_t = 1 - t, \qquad \sigma_t = t. \tag{13}$$

Diffusion Framework

Sampling based on vector v and score s. Following [3, 76], the time-dependent probability distribution $p_t(\mathbf{x})$ of $\mathbf{x}_t$ also coincides with the distribution of the reverse-time SDE [6]:

$$d\mathbf{X}_t = \mathbf{v}(\mathbf{X}_t, t)dt + \frac{1}{2} w_t \mathbf{s}(\mathbf{X}_t, t)dt + \sqrt{w_t} d\bar{\mathbf{W}}_t, \tag{7}$$

$$\|^2]dt. \tag{11}$$

where θ represents the Zigzag Mamba network that we d
section, we adopt the linear path for training, due to its simplicity and
straight trajectory:

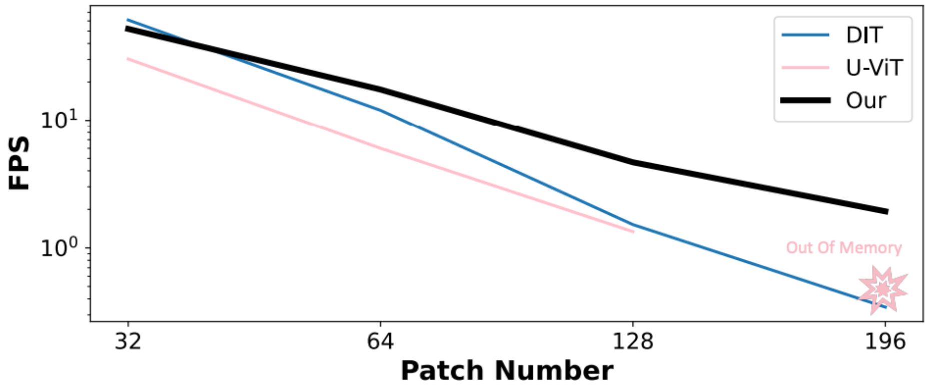
$$\alpha_t = 1 - t, \qquad \sigma_t = t. \tag{13}$$

Result

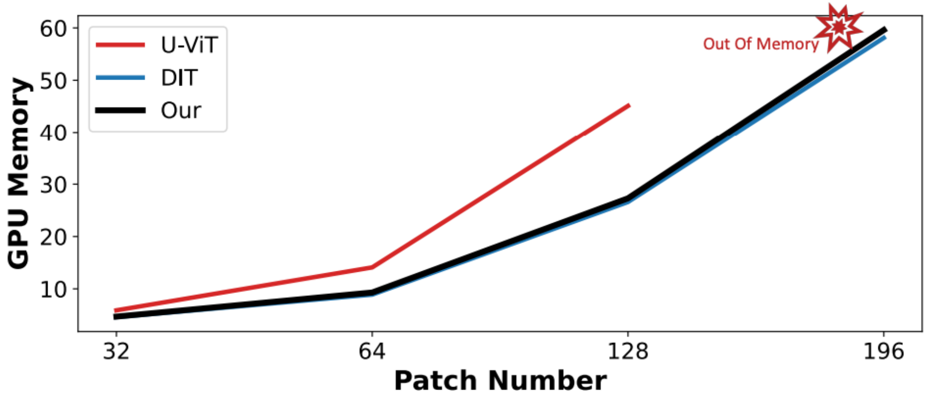
Table 1: Ablation of Scanning Scheme Number. We evaluate various zigzag scanning schemes. Starting from a simple "Sweep" baseline, we consistently observe improvements as more schemes are implemented.

	MultiModal-CelebA256			MultiModal-CelebA512		
	FID^{5k}	FDD^{5k}	KID^{5k}	FID^{5k}	FDD^{5k}	KID^{5k}
Sweep	158.1	75.9	0.169	162.3	103.2	0.203
Zigzag-1	65.7	47.8	0.051	121.0	78.0	0.113
Zigzag-2	54.7	45.5	0.041	96.0	59.5	0.079
Zigzag-8	45.5	26.4	0.011	34.9	29.5	0.023

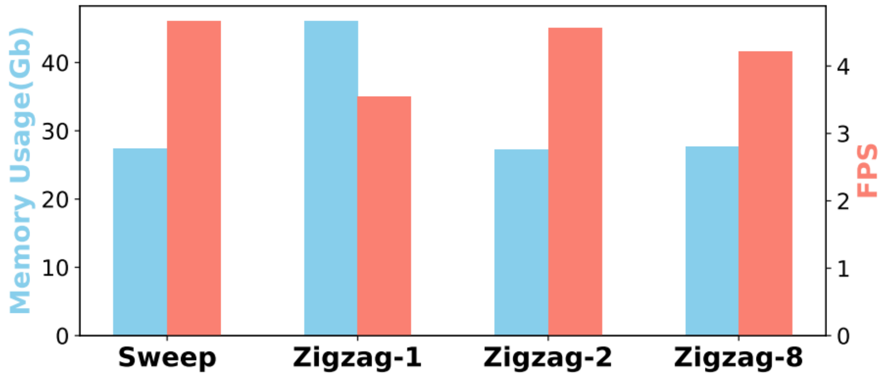
Result



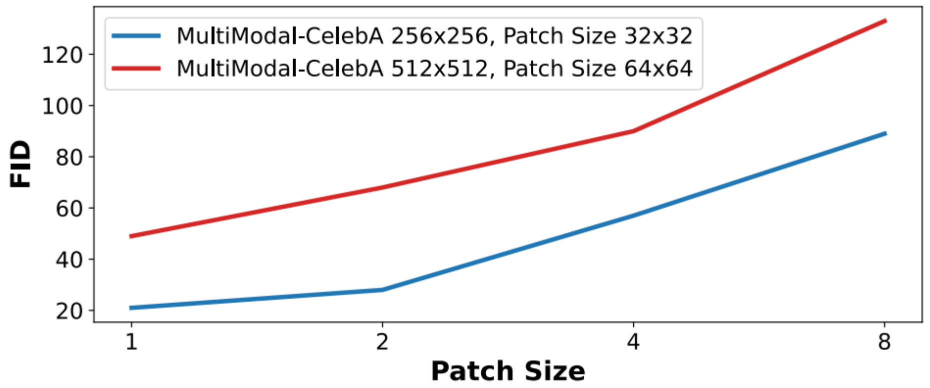
(a) FPS *v.s.* Patch Number.



(b) GPU Memory *v.s.* Patch Number.

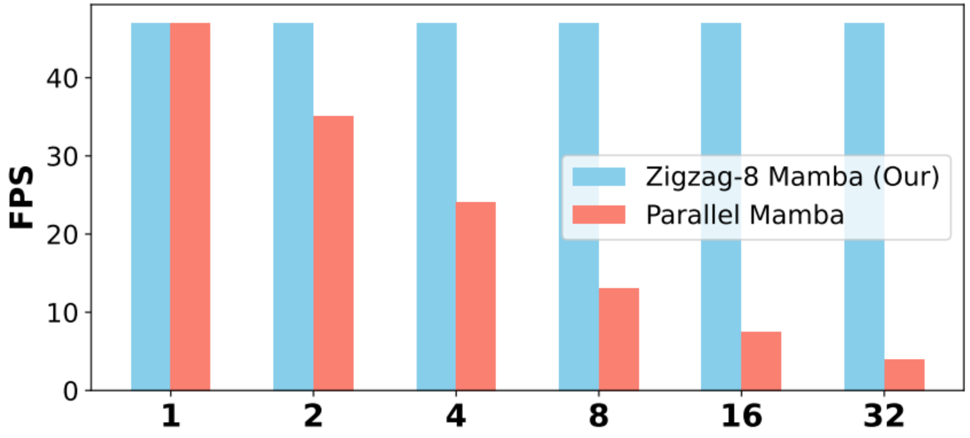


(c) GPU usage of variants of our method.

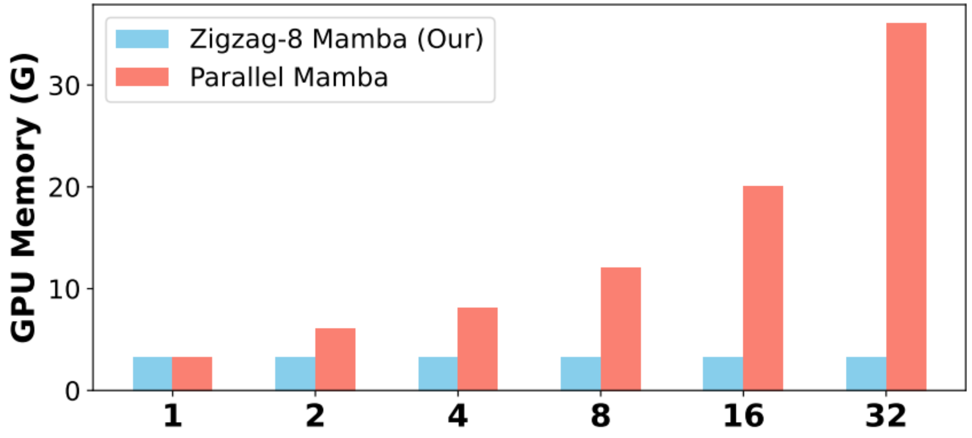


(d) FPS *v.s.* Patch Size.

Result



(a) Order Receptive Field *v.s.* FPS.



(b) Order Receptive Field *v.s.* GPU Memory.

Figure 8: The ablation study about Order Receptive Field, FPS, GPU Memory.

Main results

Table 2: Main Results on MS-COCO dataset. Our method consistently outperforms the baseline and can achieve even better results when the training scale is increased.

Variants	FID^{5k}
Sweep	195.1
Zigzag-1	73.1
Bidirection Mamba [96]	60.2
Zigzag-8	41.8
Zigzag-8 $\times$ 16GPU	33.8

Table 4: Video Scan Scheme on UCF101 dataset. Our method outperforms the baseline and can achieve even better results when the training scale is increased.

Method	Frame-FID^{5k}	FVD^{5k}
Bidirection Mamba [96] -4GPU	256.1	320.2
3D Zigzag Mamba -4GPU	238.1	282.3
Factorized 3D Zigzag Mamba -4GPU	216.1	210.2
Bidirection Mamba [96] -16GPU	146.2	201.1
Factorized 3D Zigzag Mamba -16GPU	121.2	140.1

Results

Position Embedding	Scan	FID^{5k}
–	sweep2	21.33
–	zigzag8	14.27
Sinusoidal	sweep2	18.47
Sinusoidal	zigzag8	14.03
Learnable	sweep2	16.38
Learnable	zigzag8	13.32

Learnable PE is best

Scan-ORF	FID^{5k}
hilbert-8	27.38
hilbert-2	61.67
zigzag-2	15.45
zigzag-8	13.32

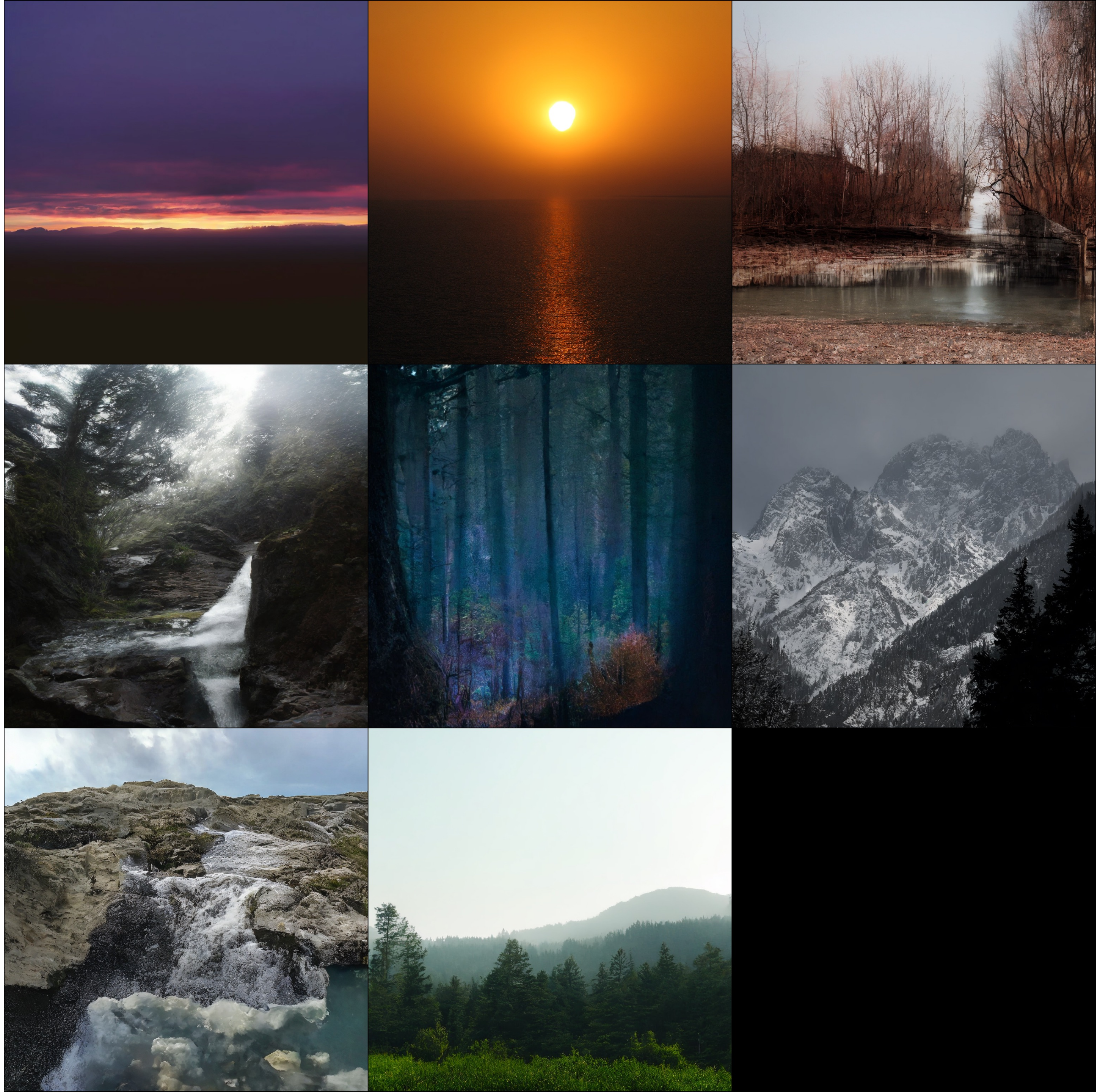
Hilbert path is too complex to learn

ORF of ZigZag	FID^{5k}
sweep1	130.00
sweep2	16.38
zigma-2	15.45
4	13.46
6	13.42
8	13.32

Scan patch will saturate around 4.
But we recommend to use 8, as there is no extra parameter and memory burden.

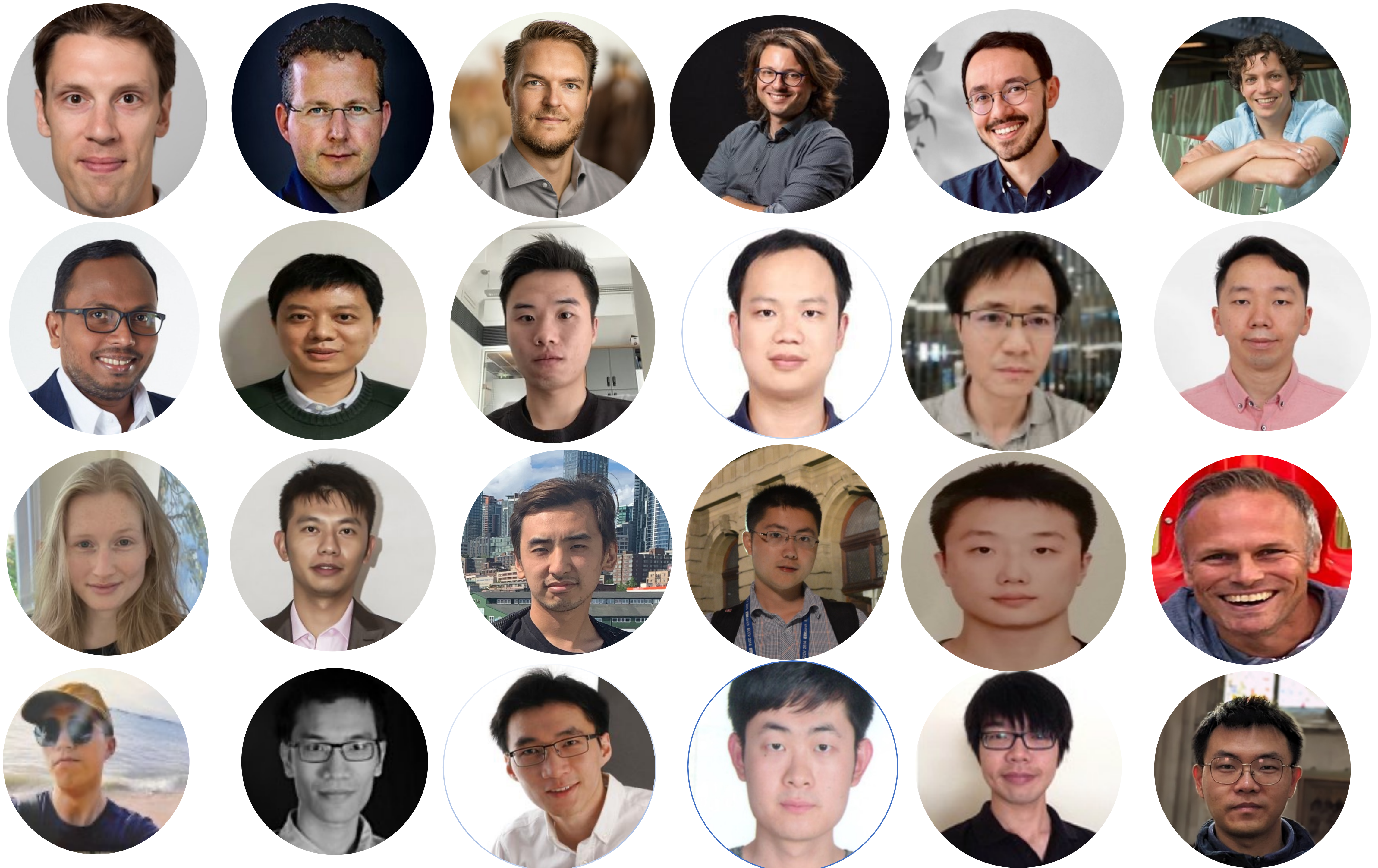
Visualization

Visualization



Conclusion

- A new ***scalable*** backbone: Mamba
- A ***generalizable*** framework: Stochastic Interpolant
- Generalized to 3D video



Thank you

Acknowledgements

- Some ppts are from Yaron Lipman
- https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mamba-and-state
- *Lilian Weng's blog*