Dynamic Diffusion Transformer, ICLR2025

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song,
Gao Huang, Fan Wang, Yang You

https://github.com/NUS-HPC-AI-Lab/Dynamic-Diffusion-Transformer

Introduction

- Diffusion Transformer (DiT) has demonstrate significant superioriry in visual generation

black forest labs.

Stable Diffusion 3 | Flux | Sora | WanX | Movie Gen

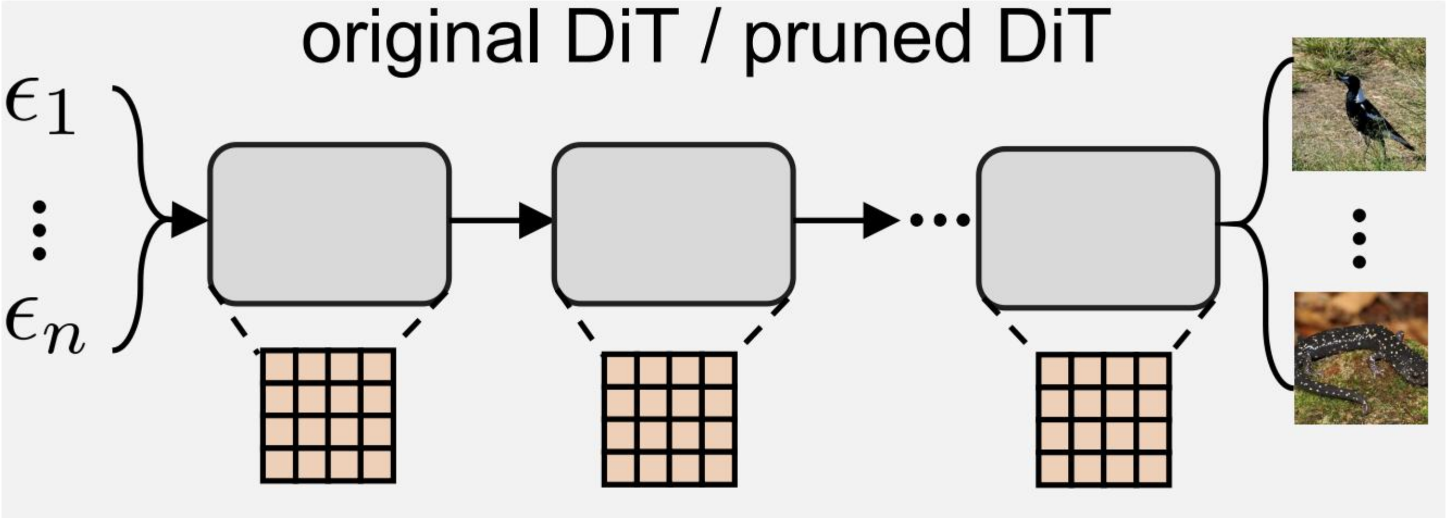
- DiT faces significant efficiency challenges during generation

William Peebles and Saining Xie. Scalable diffusion models with transformers. In ICCV, pp. 4195–4205, 2023.

Introduction

Previous solution

- Efficient diffusion samplers
- Global acceleration e.g. Cache
- Model compression e.g. pruning

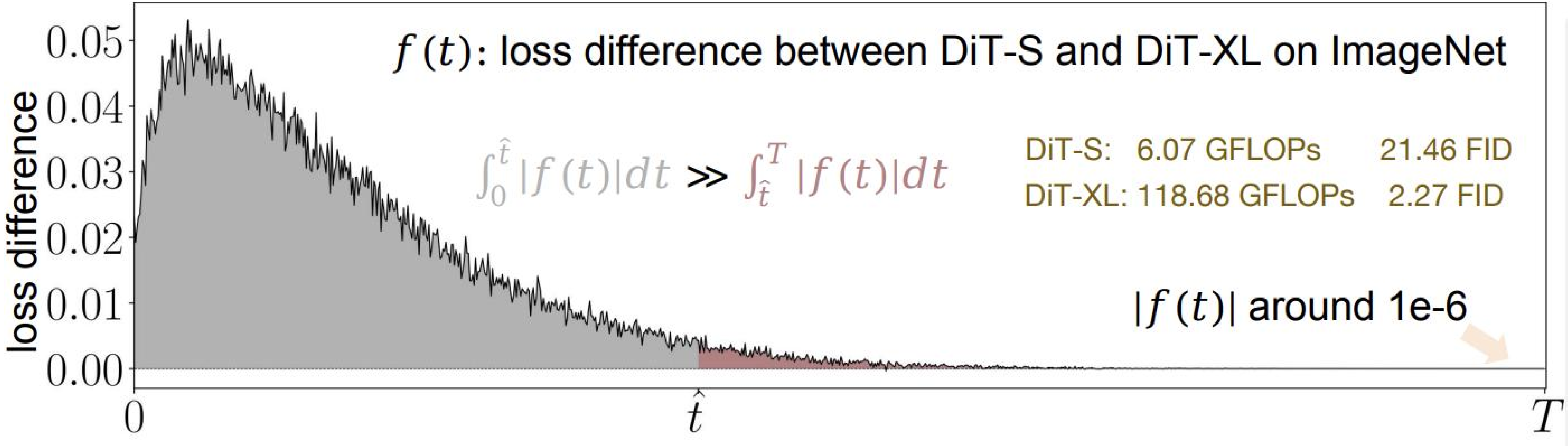


Problem

- A fixed model width across all diffusion timesteps
- Same computational cost to every image patch

Introduction

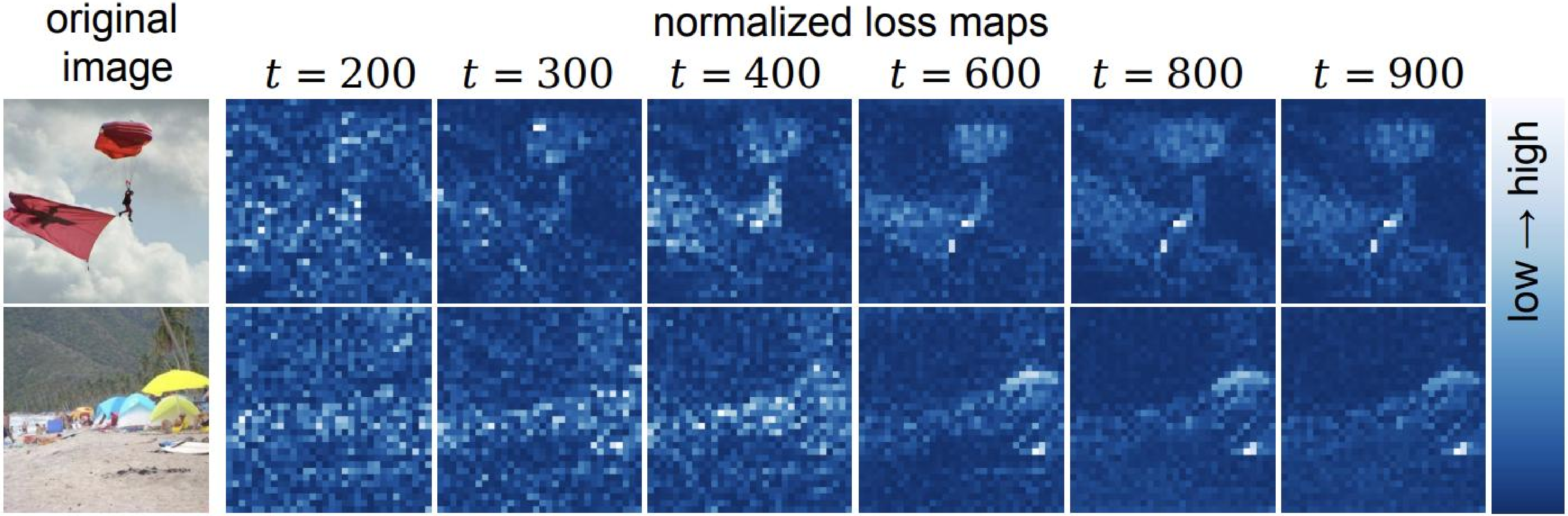
Redundancy from timestep perspective



- Loss differences diminish substantially for t > tˆ, and even approach negligible levels as t nears the prior distribution (t → T)
- The same architecture across all timesteps, leading to excessive computational costs at timesteps where the task complexity is low.

Introduction

Redundancy from spatial perspective

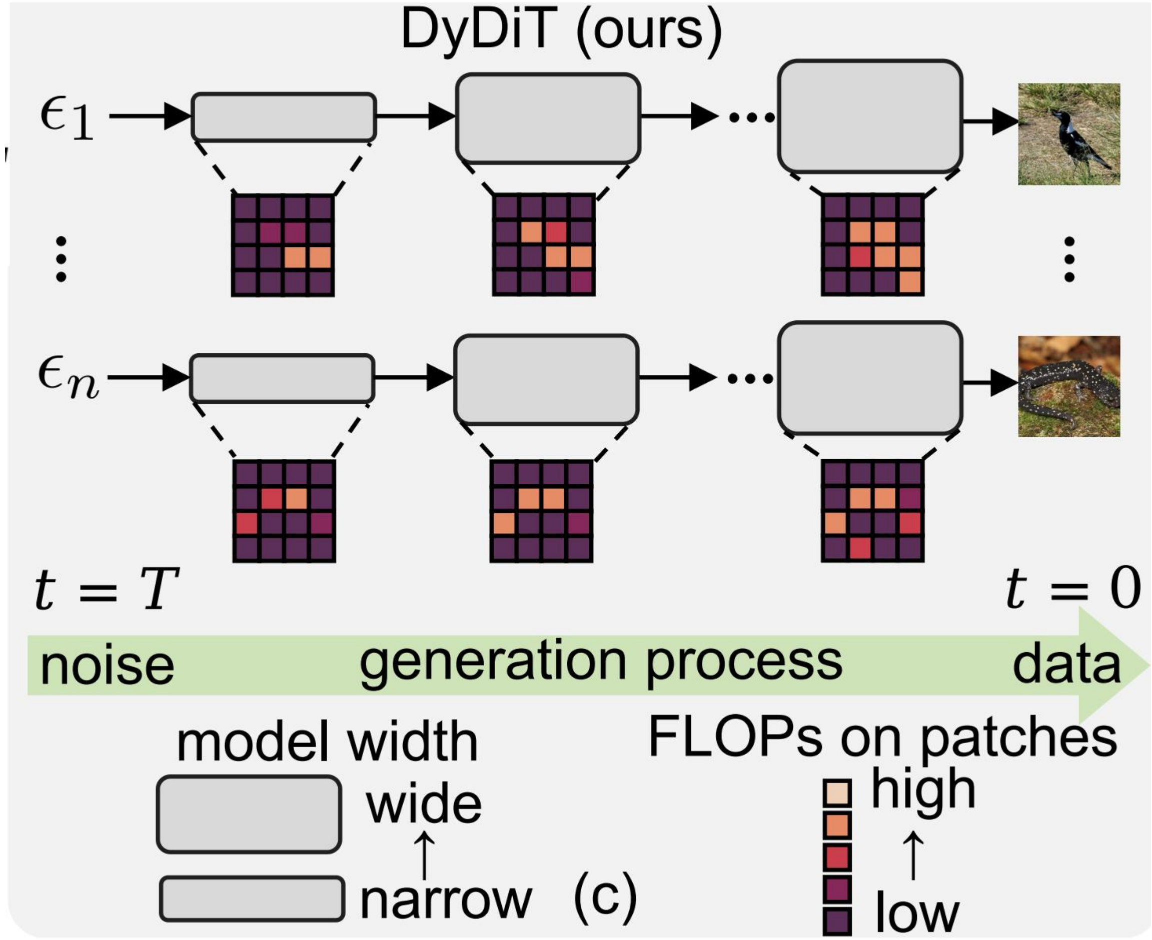


- The difficulty of noise prediction varies across spatial regions
- Uniform computational treatment of all patches introduces redundancy and is likely suboptimal.

Introduction

We propose Dynamic Diffusion Transformer (DyDiT):

- Timestep-wise Dynamic Width (TDW)
- Spatial-wise Dynamic Token (SDT)



Methodology

Timestep-wise dynamic width (TDW)

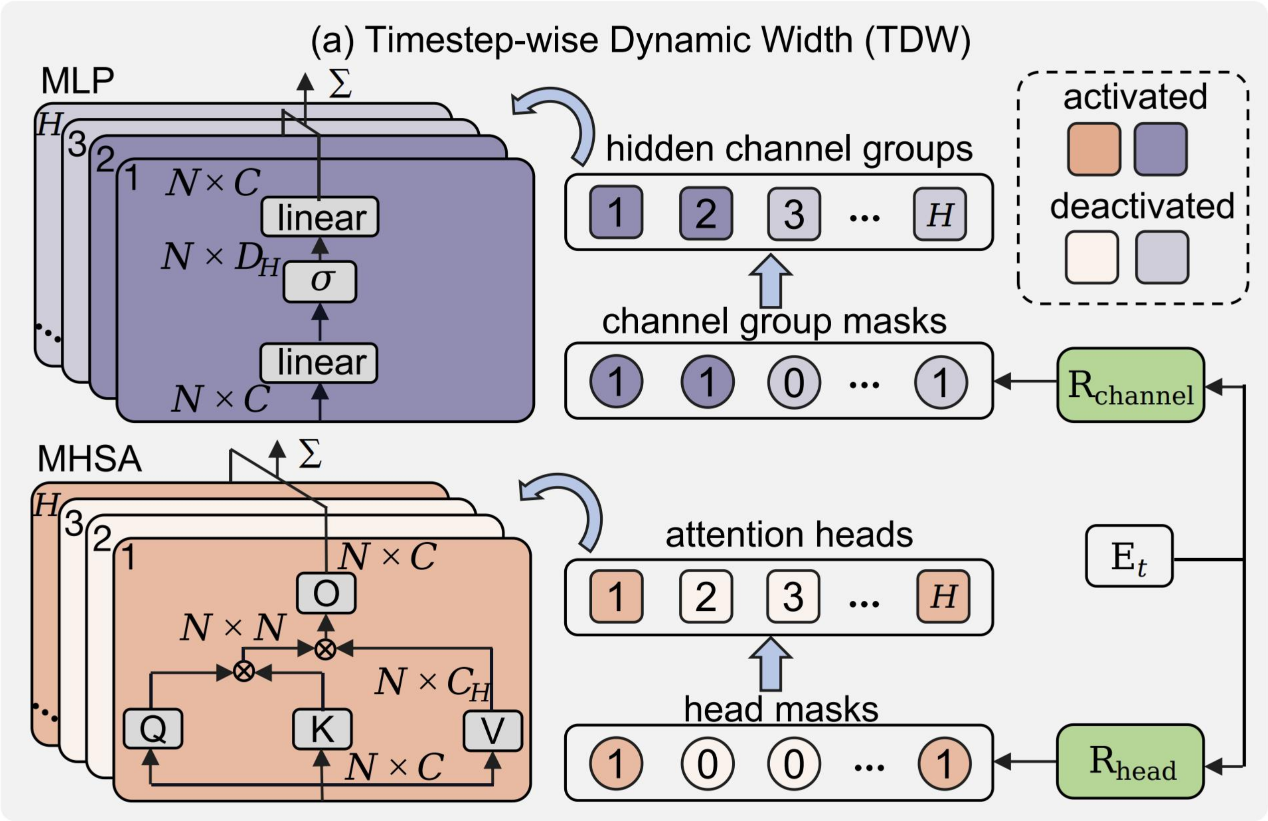
$$\mathbf{S}_{\text{head}} = \mathrm{R}_{\text{head}}(\mathbf{E}_t) \in [0,1]^H$$

$$\mathbf{S}_{\text{channel}} = \mathrm{R}_{\text{channel}}(\mathbf{E}_t) \in [0,1]^H$$

convered into binary masks $\mathbf{M}_{\text{head}}$ $\mathbf{M}_{\text{channel}}$

$$\mathrm{MHSA}(\mathbf{X}) = \sum_{h:\mathbf{M}^h_{\text{head}}=1} \mathbf{X}^h_{\text{attn}} \mathbf{W}^{h,:,:}_{\text{O}},$$

$$\mathrm{MLP}(\mathbf{X}) = \sum_{h:\mathbf{M}^h_{\text{channel}}=1} \sigma(\mathbf{X}^h_{\text{hidden}}) \mathbf{W}^{h,:,:}_{2}.$$

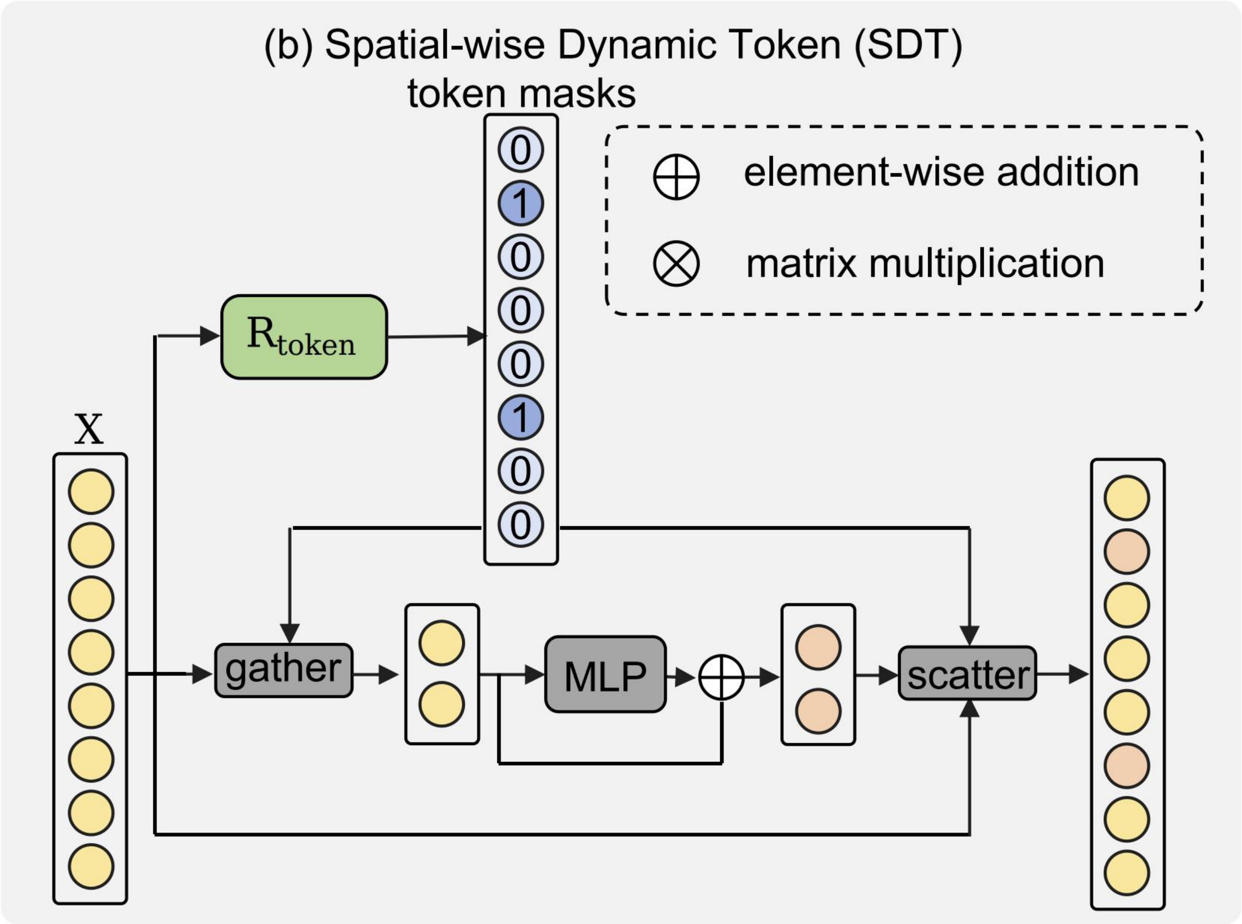


Methodology

Spatial-wise dynamic token (SDT)

$$\mathbf{S}_{\text{token}} = \mathrm{R}_{\text{token}}(\mathbf{X}) \in [0, 1]^N$$

convered into binary mask $\mathbf{M}_{\text{token}}$



Methodology

FLOPs-aware end-to-end training

- t ~ Uniform(0, T) during training， approximately covers the entire computation graph.
- FLOPs $F_{\text{dynamic}}^{t_b}$ using masks $\{\bar{\mathbf{M}}_{\text{head}}^{t_b}, \mathbf{M}_{\text{channel}}^{t_b}, \bar{\mathbf{M}}_{\text{token}}^{t_b}\}$

$$\mathcal{L}_{\text{FLOPs}} = (\frac{1}{B} \sum_{t_b : b \in [1, B]} \frac{F_{\text{dynamic}}^{t_b}}{F_{\text{static}}} - \lambda)^2$$

$$\mathcal{L} = \mathcal{L}_{\text{DiT}} + \mathcal{L}_{\text{FLOPs}}$$

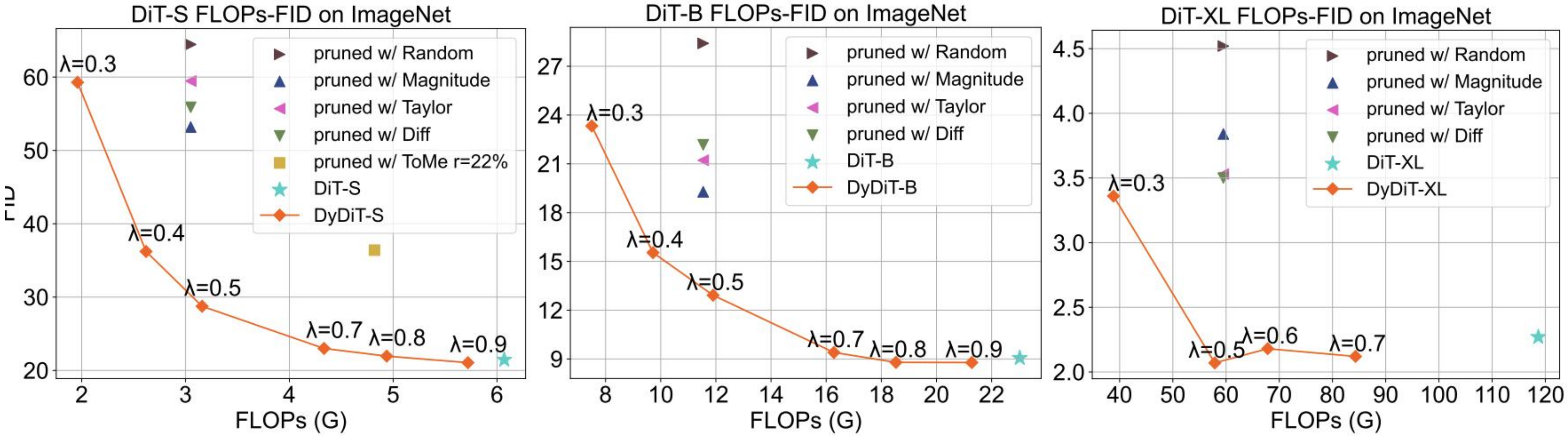
Experiments

Comparison with state-of-the-art diffusion models

Model	Params. (M) ↓	FLOPs (G) ↓	FID ↓	sFID ↓	IS ↑	Precision ↑	Recall ↑
			Static 256 × 256				
ADM	608	1120	4.59	5.25	186.87	0.82	0.52
LDM-4	400	104	3.95	-	178.22	0.81	0.55
U-ViT-L/2	**287**	<u>77</u>	3.52	-	-	-	-
U-ViT-H/2	501	113	2.29	-	247.67	**0.87**	0.48
DiffuSSM-XL	673	280	2.28	**4.49**	269.13	<u>0.86</u>	0.57
DiM-L	<u>380</u>	94	2.64	-	-	-	-
DiM-H	860	210	2.21	-	-	-	-
DiT-L	468	81	5.02	-	167.20	0.75	0.57
DiT-XL	675	118	2.27	4.60	277.00	0.83	0.57
DiffiT	561	114	<u>1.73</u>	-	276.49	0.80	<u>0.62</u>
SiT-XL	675	118	2.06	**4.49**	277.50	0.83	0.59
DiMR-XL	505	160	**1.70**	-	**289.00**	0.79	**0.63**
			Dynamic 256 × 256				
DyDiT-XL$_{\lambda=0.7}$	678	84.33	2.12	4.61	<u>284.31</u>	0.81	0.60
DyDiT-XL$_{\lambda=0.5}$	678	**57.88**	2.07	4.56	248.03	0.80	0.61

Experiments

Scaling up ability



Increased computation redundancy with larger models, allowing our method to reduce redundancy without compromising FID

Experiments

Visualization of dynamic architecture

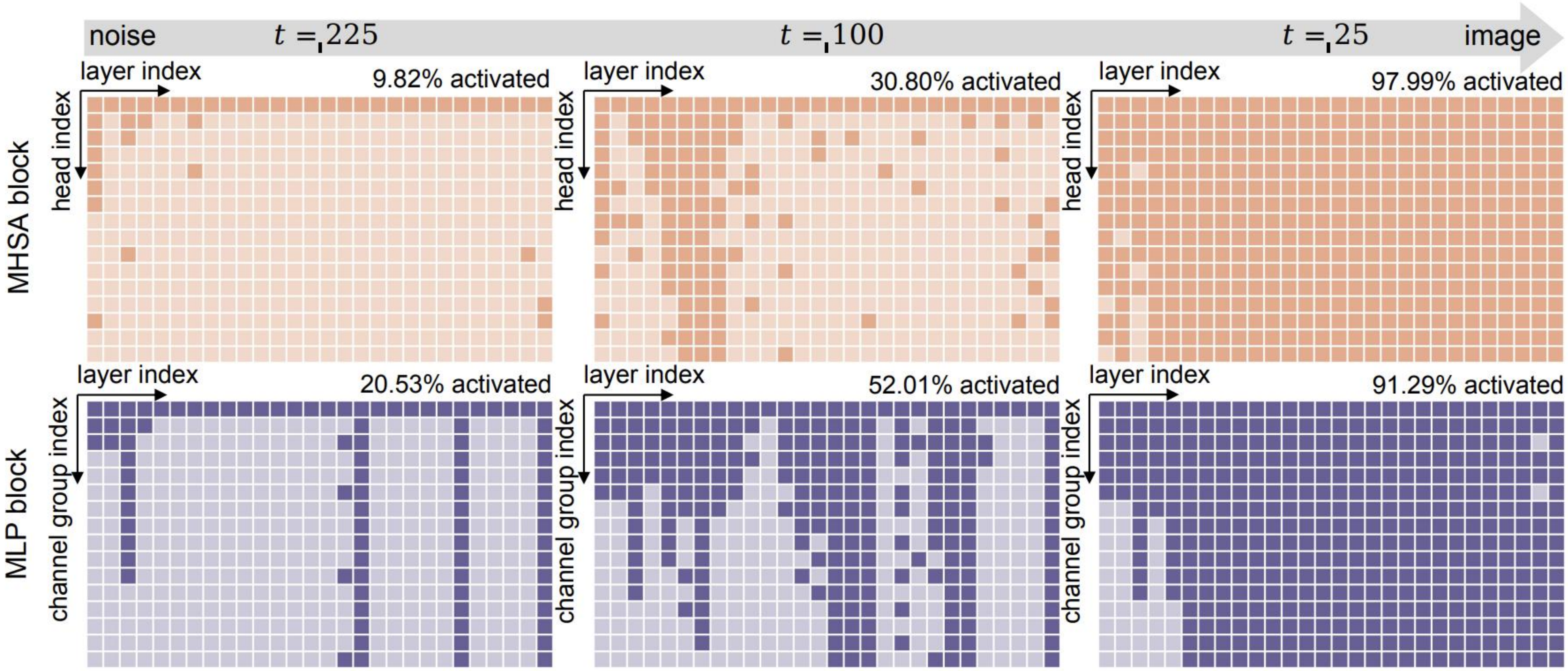


Figure 5: **Visualization of dynamic architecture.** and indicates the deactivated and activated heads in an MHSA block, while and denotes that the channel group is deactivated or activated in an MLP block, respectively. We conduct 250-step DDPM generation.

Experiments

Computational cost across different image patches

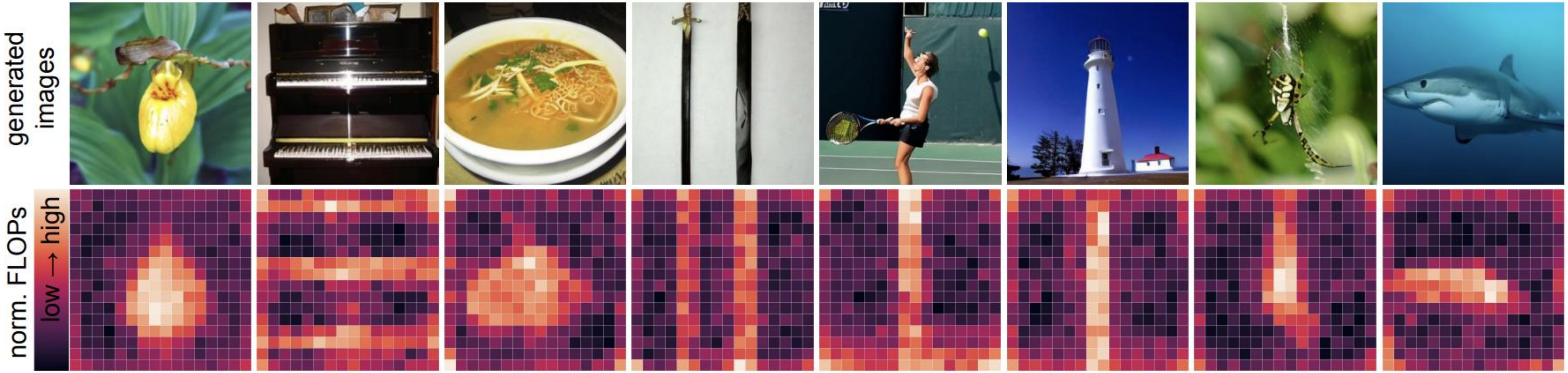


Figure 6: **Computational cost across different image patches.** We quantify the FLOPs cost on image patches over the generation process and normalize them into [0, 1] for better clarity.

Experiments

Compatibility

Table 4: **Combination with efficient samplers** (Song et al., 2020a; Lu et al., 2022).

Model	250-DDPM		50-DDIM		20-DPM-solver++		10-DPM-solver++	
	s/image ↓	FID ↓	s/image ↓	FID ↓	s/image ↓	FID ↓	s/image ↓	FID ↓
DiT-XL	10.22	2.27	2.00	2.26	0.84	4.62	0.42	11.66
DyDiT-XL$_{\lambda=0.7}$	7.76	2.12	1.56	2.16	0.62	4.28	0.31	11.10
DyDiT-XL$_{\lambda=0.5}$	5.91	2.07	1.17	2.36	0.46	4.22	0.23	11.31

Table 18: **Combined with DeepCache.** "interval" denotes the interval of cached timestep in DeepCache (Ma et al., 2023).

Model	interval	s/image ↓	FID ↓
DiT-XL	0	10.22	2.27
DiT-XL	2	5.02	2.47
DiT-XL	5	2.03	6.73
DyDiT-XL$_{\lambda=0.5}$	0	5.91	2.08
DyDiT-XL$_{\lambda=0.5}$	2	2.99	2.43
DyDiT-XL$_{\lambda=0.5}$	3	2.01	3.37

Thanks!

https://github.com/NUS-HPC-AI-Lab/Dynamic-Diffusion-Transformer