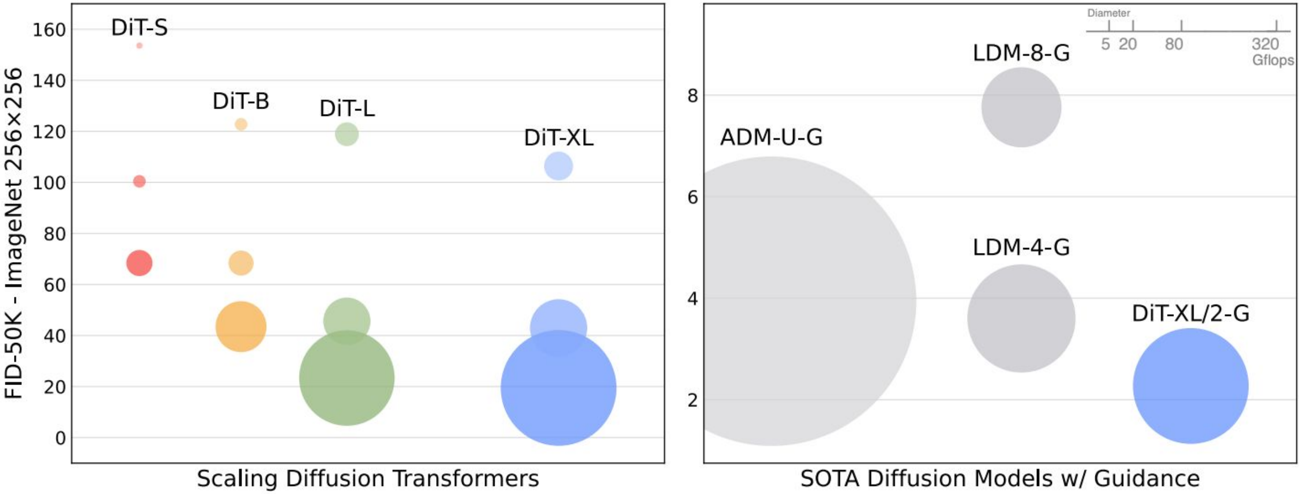
Scalable Diffusion Models with Transformers

William Peebles, Saining Xie
Presenter: Zhenrui Yue

Intro & Motivation

1. Existing diffusion models adopt convolutional U-Net as backbone
2. Explore alternative architecture choices for generative modeling research
3. Introduce diffusion transformers (DiTs) and study the scaling behavior of transformers with respect to network complexity vs. sample quality

There is a strong correlation between the network complexity (in Gflops) and sample quality (in FID). By scaling-up DiT and training with high capacity (118.6 Gflops), the authors achieve state-of-the-art results (2.27 FID) on ImageNet generation benchmark.



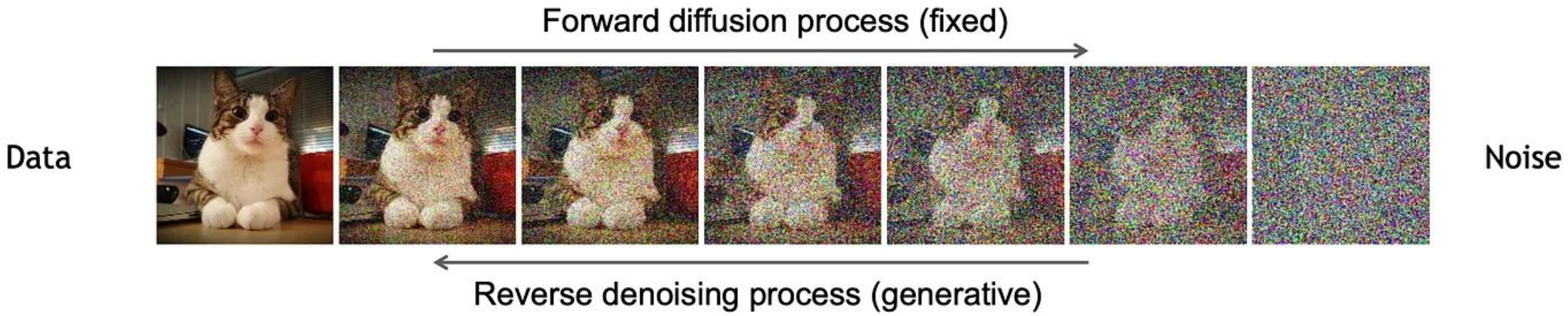
Diffusion Transformers - Preliminaries

Forward Noising Process:

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

Reverse Denoising Process:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$



Diffusion Transformers - Preliminaries

Learning via variational lower bound:

→ Reparameterize μ_θ as ϵ_θ and train with L_simple

→ Learn reverse process covariance Σ_θ with L

$$\mathcal{L}(\theta) = -p(x_0|x_1) + \sum_t \mathcal{D}_{KL}(q^*(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))$$
$$\mathcal{L}_{simple}(\theta) = ||\epsilon_\theta(x_t) - \epsilon_t||_2^2$$

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Diffusion Transformers - Preliminaries

Classifier-Free Guidance

→ Reverse process conditioned on class c: $p_\theta(x_{t-1}|x_t, c)$

→ $\hat{\epsilon}_\theta(x_t, c) \;=\; \epsilon_\theta(x_t, \emptyset) + s \cdot \nabla_x \log p(x|c) \propto \epsilon_\theta(x_t, \emptyset) + s \cdot (\epsilon_\theta(x_t, c) - \epsilon_\theta(x_t, \emptyset))$

Latent Diffusion Models

→ Learn an autoencoder with a learned encoder E

→ Train a diffusion model of representations z = E(x) instead of x (E is frozen)

→ Off-the-shelf convolutional VAEs & transformer-based DDPMs

Diffusion Transformer Design Space

DiT is largely based on Vision Transformers (ViTs)

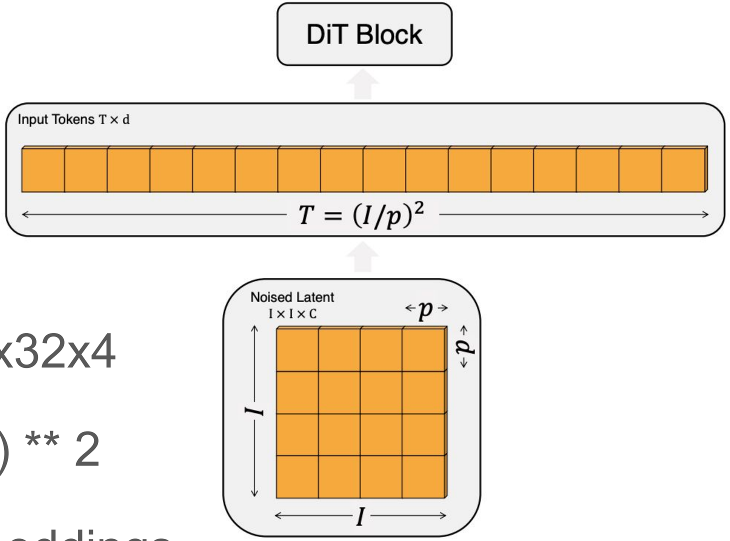
For Patching:

→ Input is the encoded image z

→ For 256x256x3 image, input z has shape 32x32x4

→ Patch size p = 2, 4, 8, Token length T = (I / p) ** 2

→ Additionally apply sine-cosine positional embeddings



Diffusion Transformer Design Space

Incorporating Timestamp & Class in DiT Blocks:

→ Timestampe t and class c are appended as special tokens

→ Additional cross attention layer for (t, c) above self-attention

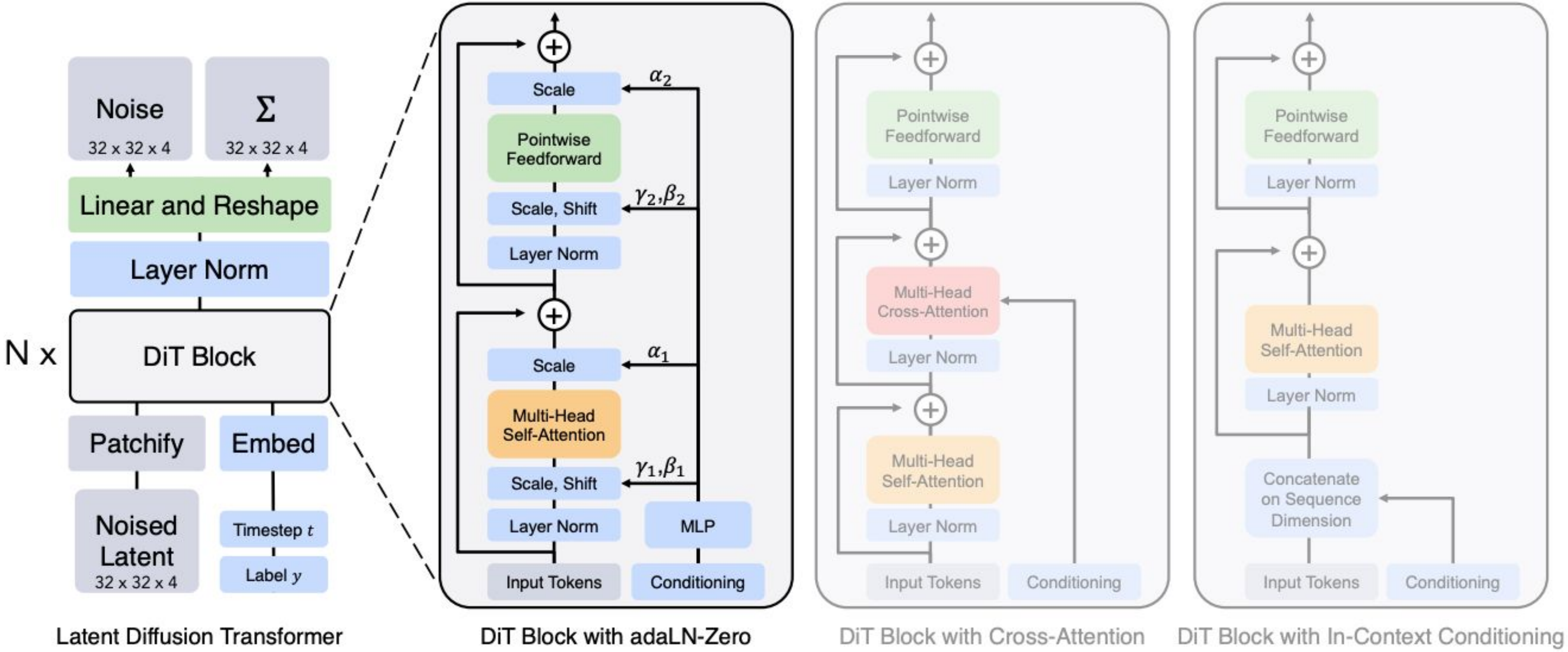
→ Adaptive layer norm (adaLN) by regressing gamma / beta upon (t, c)

→ Zero-initializing the final layer in adaLN prior to residual connections

Decoder Blocks:

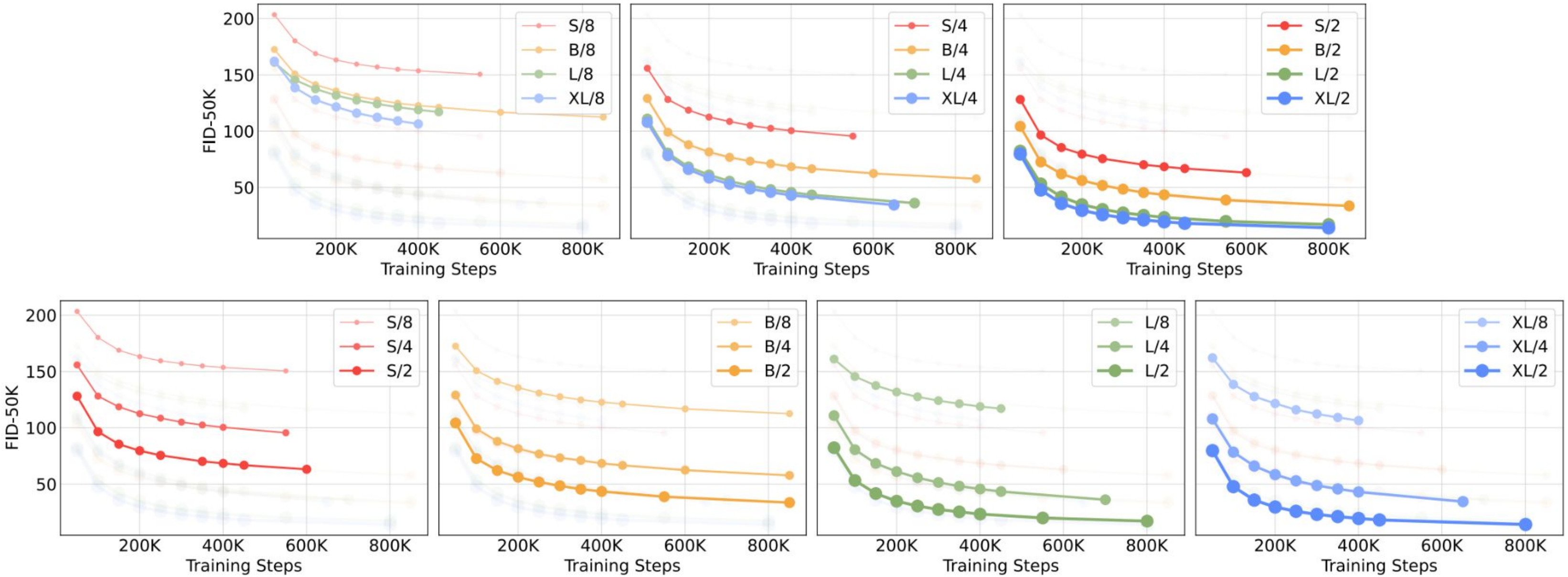
→ Linear decoder to predict noise and covariance (p x p x 2C shape)

Diffusion Transformer Design Space

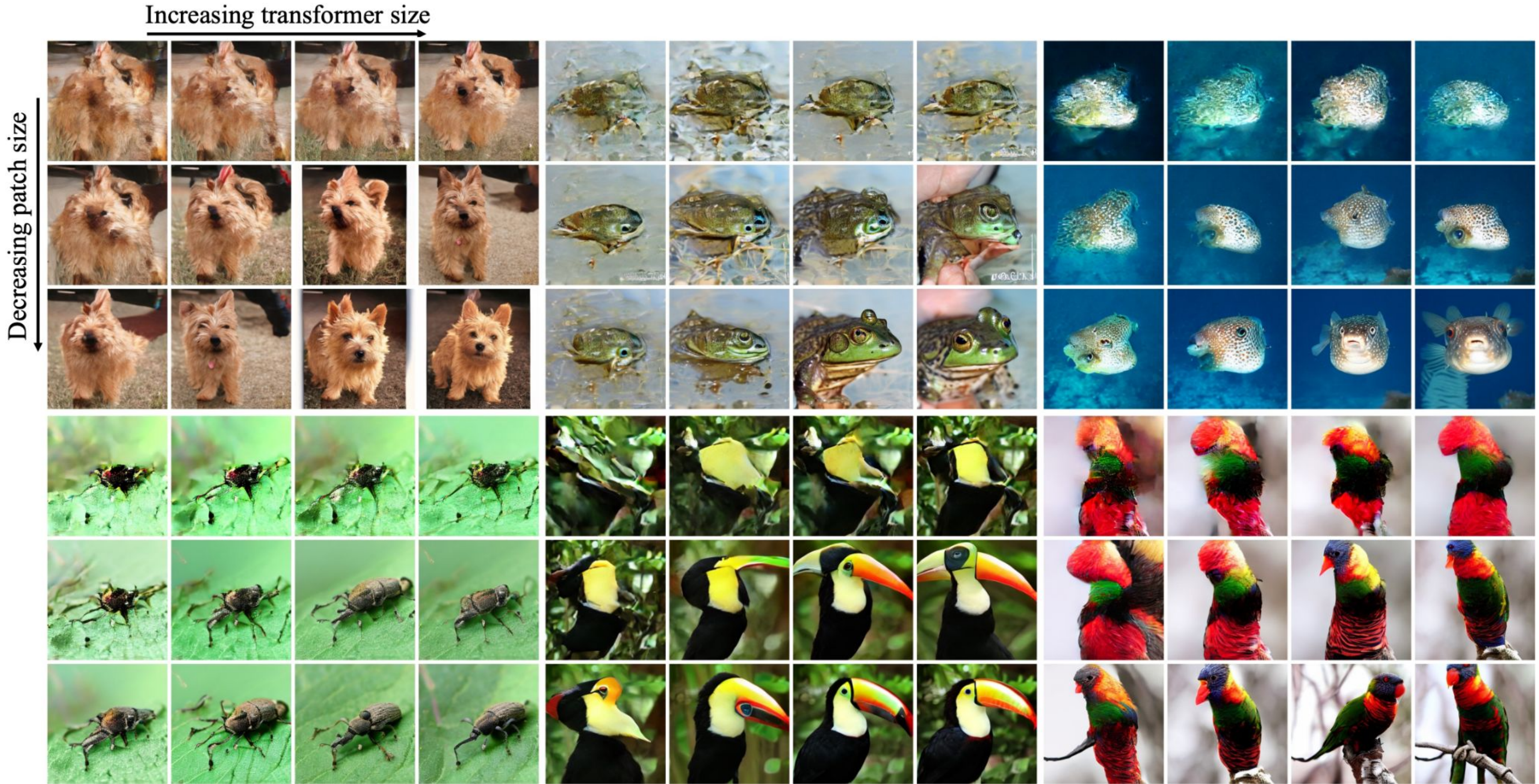


Training Diffusion Transformers

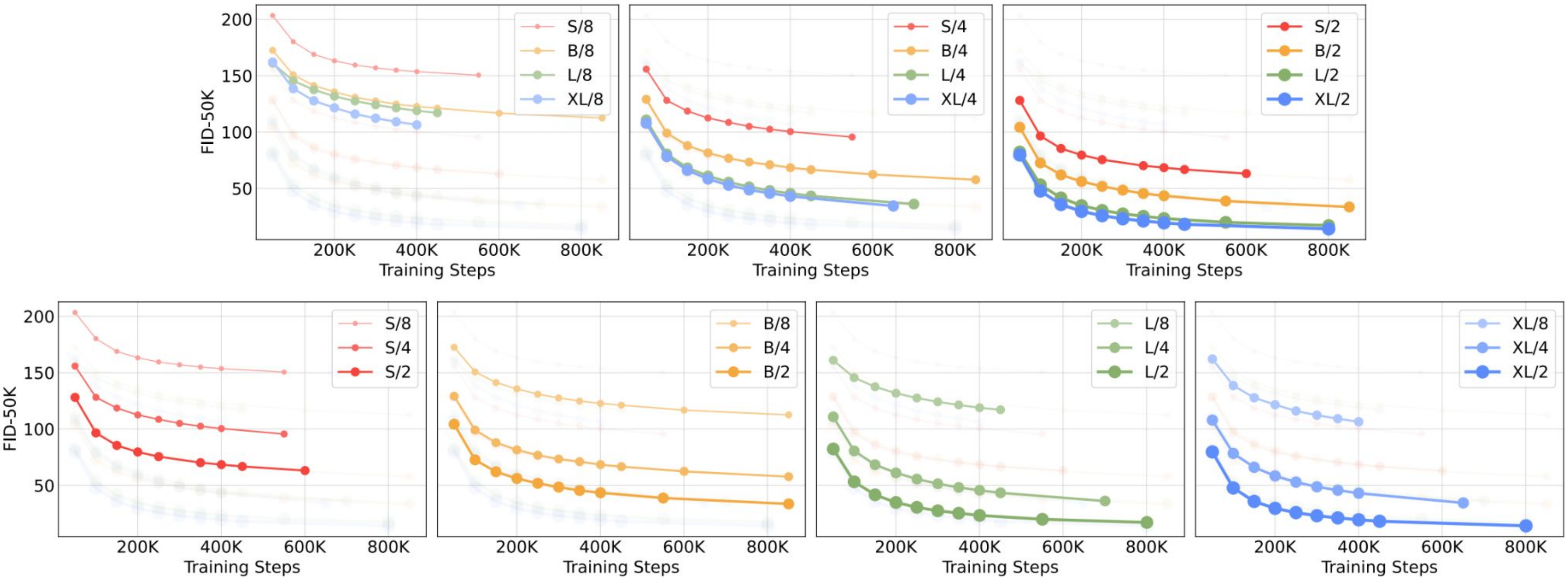
1. Leverage pretrained VAE from stable diffusion
2. After sampling, decode pixels using VAE decoder
3. Evaluate with FID scores and 250 DDPM sampling steps



Model and Patch Size



Model and Patch Size / DiT Gflops



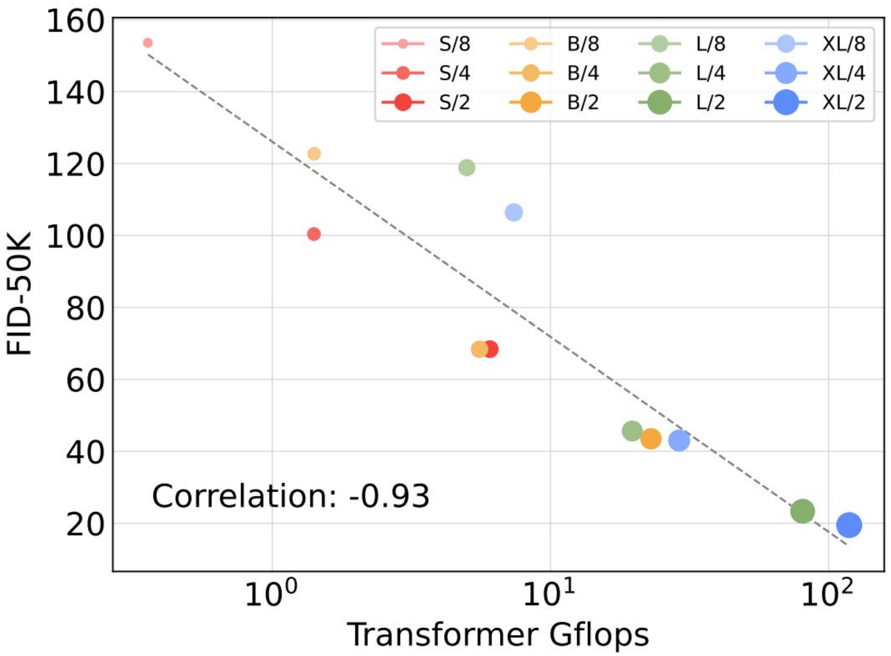
Model and Patch Size / DiT Gflops

Model / Patch Size:

1. Improvements in FID are obtained by making transformer models larger
2. Improvements in FID are obtained by reducing patch sizes (scaling tokens)

Gflops:

1. Scaling model Gflops is the key to improved FID performance
2. Given constant Gflops, different DiT configs obtain similar FID values

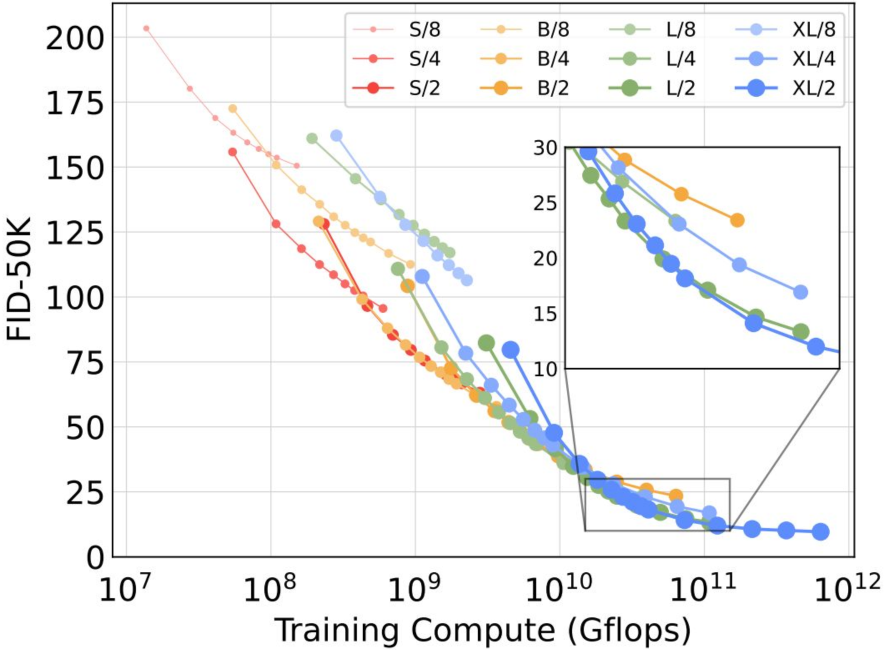


DiT Model Size / Efficiency

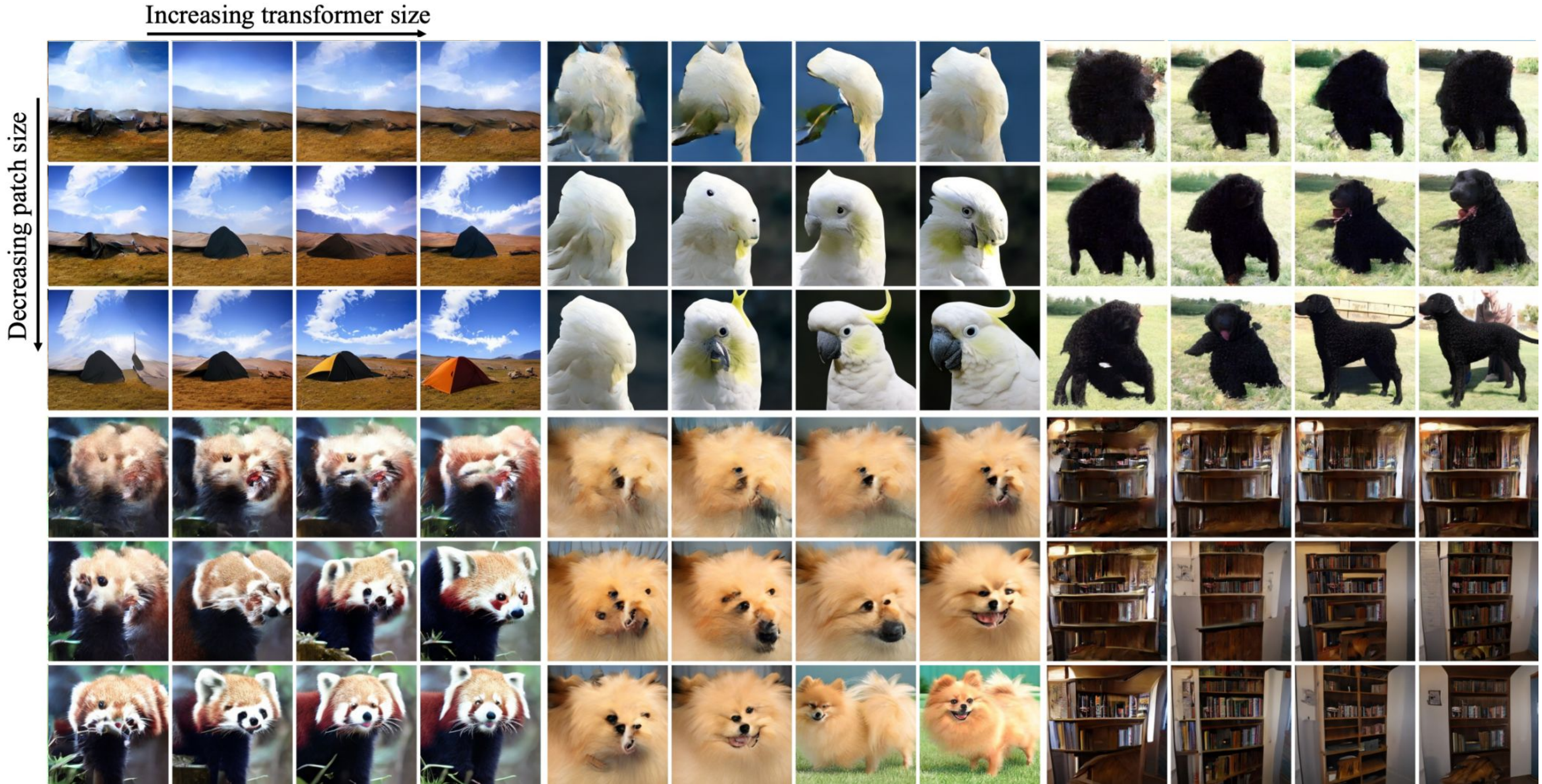
Training compute:

Model Gflops * batch * training steps * 3

1. Given constant training Gflops, larger DiT models are more efficient
2. Models that are identical except for patch size have different performance profiles even when controlling training Gflops
3. Similar observation on qualitative examples



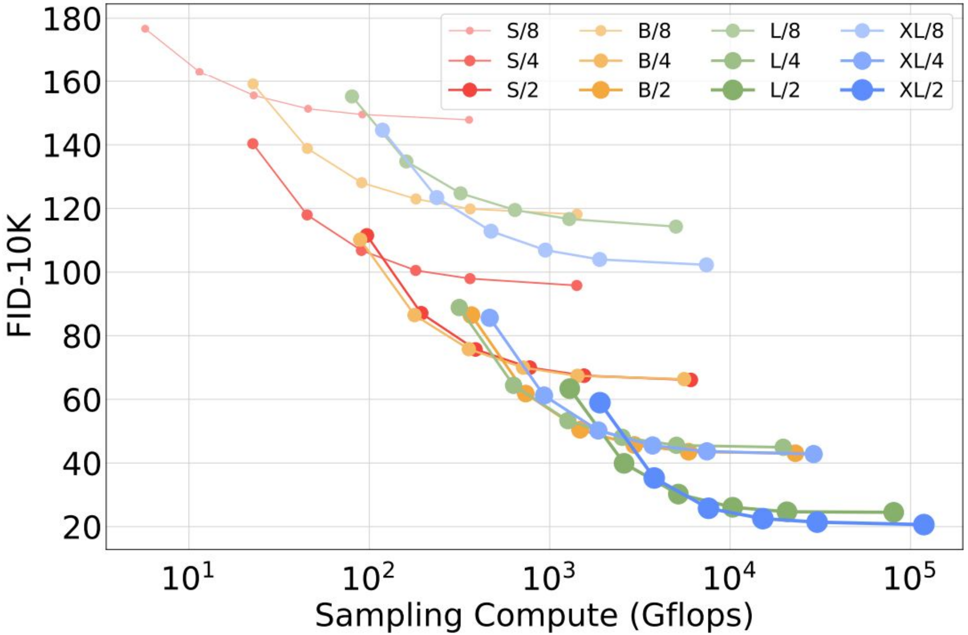
Further Qualitative Examples



Scaling & Sampling

Scaling Model vs. Sampling Compute

1. Consider constant sampling compute (DiT-XL/2 w/ 128 steps OR DiT-L/2 w/ 1000 steps)
2. In most cases, scaling-up sampling compute (steps) cannot compensate for the lack of model compute



Comparison to SOTA Models

For both 256 x 256 and 512 x 512 image resolution, the proposed DiT can outperform existing methods and remains compute efficient.

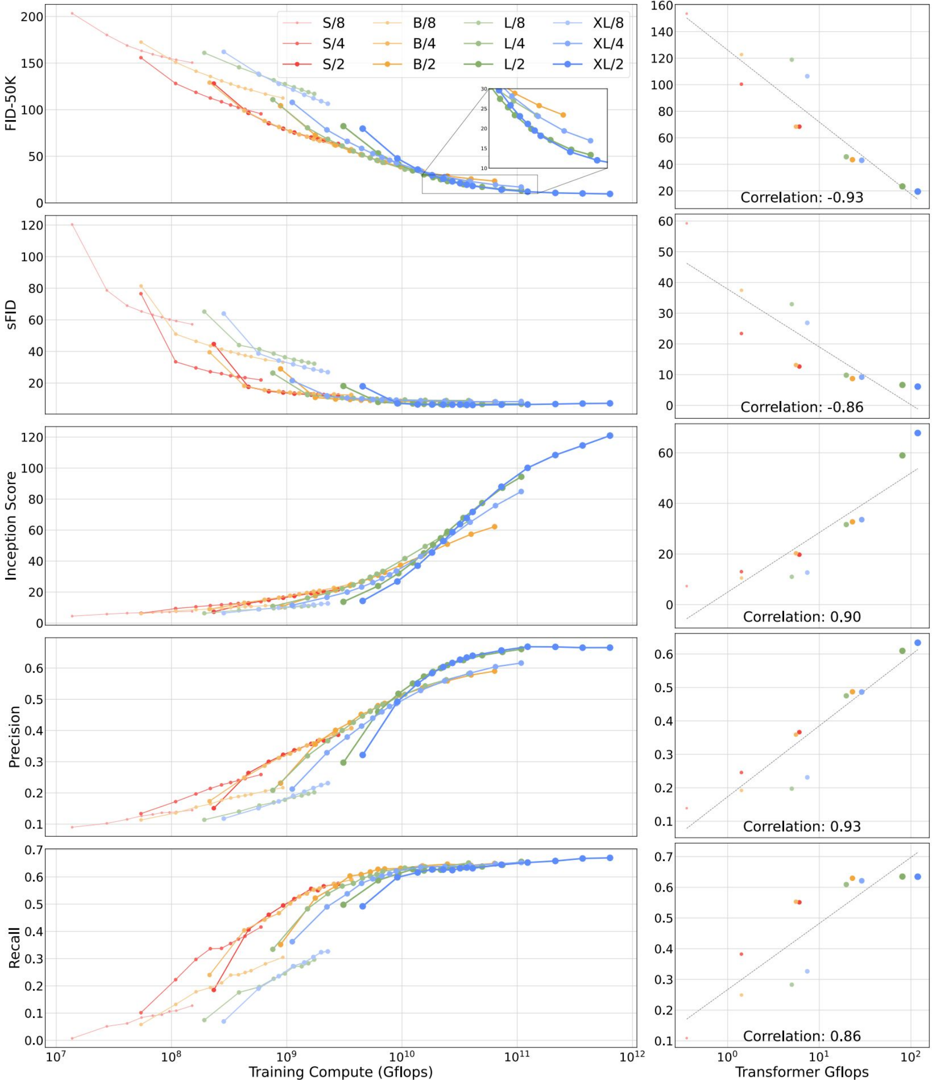
Class-Conditional ImageNet 512×512					
Model	FID↓	sFID↓	IS↑	Precision↑	Recall↑
BigGAN-deep [2]	8.43	8.13	177.90	0.88	0.29
StyleGAN-XL [53]	2.41	4.06	267.75	0.77	0.52
ADM [9]	23.24	10.19	58.06	0.73	0.60
ADM-U	9.96	5.62	121.78	0.75	**0.64**
ADM-G	7.72	6.57	172.71	**0.87**	0.42
ADM-G, ADM-U	3.85	5.86	221.72	0.84	0.53
DiT-XL/2	12.03	7.12	105.25	0.75	**0.64**
DiT-XL/2-G (cfg=1.25)	4.64	5.77	174.77	0.81	0.57
DiT-XL/2-G (cfg=1.50)	**3.04**	**5.02**	**240.82**	0.84	0.54

Class-Conditional ImageNet 256×256					
Model	FID↓	sFID↓	IS↑	Precision↑	Recall↑
BigGAN-deep [2]	6.95	7.36	171.4	0.87	0.28
StyleGAN-XL [53]	2.30	4.02	265.12	0.78	0.53
ADM [9]	10.94	6.02	100.98	0.69	0.63
ADM-U	7.49	5.13	127.49	0.72	0.63
ADM-G	4.59	5.25	186.70	0.82	0.52
ADM-G, ADM-U	3.94	6.14	215.84	0.83	0.53
CDM [20]	4.88	-	158.71	-	-
LDM-8 [48]	15.51	-	79.03	0.65	0.63
LDM-8-G	7.76	-	209.52	0.84	0.35
LDM-4	10.56	-	103.49	0.71	0.62
LDM-4-G (cfg=1.25)	3.95	-	178.22	0.81	0.55
LDM-4-G (cfg=1.50)	3.60	-	247.67	**0.87**	0.48
DiT-XL/2	9.62	6.85	121.50	0.67	**0.67**
DiT-XL/2-G (cfg=1.25)	3.22	5.28	201.77	0.76	0.62
DiT-XL/2-G (cfg=1.50)	**2.27**	**4.60**	**278.24**	0.83	0.57

Additional Results

Metrics

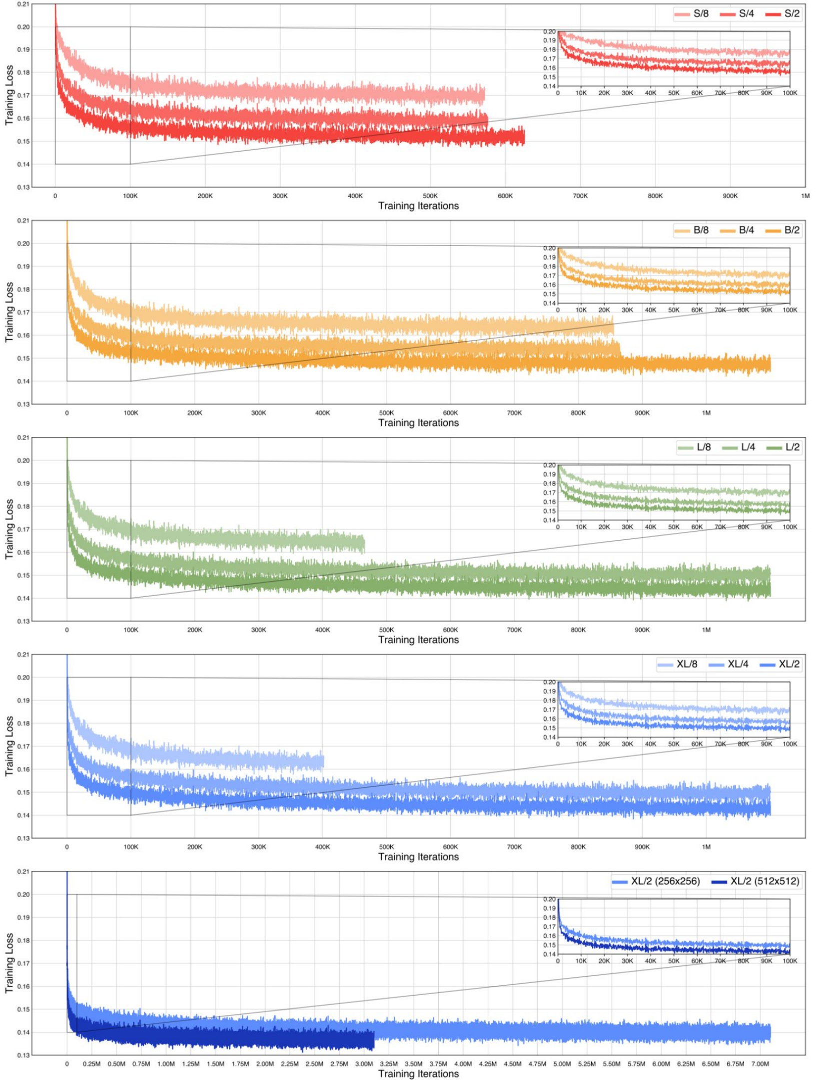
1. Additional metrics: sFID, Inception Score, Precision, Recall
2. FID-driven analysis in the paper generalizes to the other metrics - across every metric, scaled-up DiT models are more compute-efficient and model Gflops are highly-correlated with performance. In particular, Inception Score and Precision benefit heavily from increased model scale.



Additional Results

Training Loss

1. Training curves for different DiT model sizes
2. Increasing DiT model Gflops (via transformer size or number of input tokens) causes the training loss to decrease more rapidly and saturate at a lower value. This phenomenon is consistent with trends observed with language models, where scaled up transformers demonstrate both improved loss curves and downstream performance.



Additional Results

VAE Decoder Ablations

1. XL/2 continues to outperform all prior models when using the LDM decoder.

Class-Conditional ImageNet 256×256, DiT-XL/2-G (cfg=1.5)					
Decoder	FID↓	sFID↓	IS↑	Precision↑	Recall↑
original	2.46	5.18	271.56	0.82	0.57
ft-MSE	2.30	4.73	276.09	0.83	0.57
ft-EMA	2.27	4.60	278.24	0.83	0.57

Conclusion

1. Introduced Diffusion Transformers (DiTs), a simple transformer-based backbone for diffusion models that outperforms prior U-Net models.

2. Incorporated a series of improvements upon ViT such as cross attention, adaptive layer norm etc., leading to improved generation performance

3. Given the promising scaling results in this paper, future work should continue to scale DiTs to larger models and token counts. Alternatively, DiT could also be explored as a drop-in backbone for text-to-image models like DALLE.

Thoughts & Discussion

1. Conditioned on class

   → Text prior is not considered

2. Only experimented on ImageNet

   → Further experiments and evaluation may be beneficial

3. Focuses on the scaling of DiT rather than efficiency

   → Sampling is still somewhat expensive

   → Maybe MoE or fewer steps like InstaFlow

   → Model size are much smaller than LLMs (33M to 675M)