TGML Lab | SUT | Winter 1403

DiT: SCALABLE DIFFUSION MODELS WITH TRANSFORMERS

2023 IEEE / CVF International Conference on Computer Vision

William Peebles (UC Berkeley) **Saining Xie (NYU)**

Presentation by: Maryam Rezaee & Mahshid Dehghani

TABLE OF CONTENTS

01

Setting the Stage

1.1 Core Idea

- ML Renaissance
- DiT Proposal

1.2 Related Work

- Key Themes
- Vision Transformer

1.1 Core Idea

ML Renaissance

- Transformers have revolutionized ML but mostly remain in the autoregressive fields.
- Diffusion models (integral to image generation advances) mainly use U-Net (convolutional) despite attention addition.
- U-Net is effective, but the inductive bias is not needed; transformers could replace it for architecture unification.

1.1 Core Idea

DiT Proposal

- Use Vision Transformer (ViT) principles but for diffusion.
- Keep diffusion model quality and robustness while benefiting from transformer scalability and efficiency.
- Step closer to standardized architecture for more possibilities.
- Achieve state-of-the-art performance!
- How? **Take LDMs' VAE latent space & use Transformer inside!**

1.2 Related Work

Key Themes

- **Transformers:** Autoregressive and generative tasks, including ViTs, autoregressive pixel models, and CLIP image embeddings.
- **Denoising Diffusion Probabilistic Models:** State-of-the-art in image generation; improvements include sampling, classifier-free guidance, and multi-resolution pipelines.
- **Architecture Complexity:** Works in both FLOPs and parameter counts; UNet in DM has already been studied via FLOPs.

floating-point operations per second

Setting the Stage

- Core Idea
- Related Work

Inside DiT

- Preliminaries
- Architecture

Testing Grounds

- Setup
- Results
- Inference

Looking Ahead

- Applications
- Enhancement

1.2 Related Work

Vision Transformer (ViT)

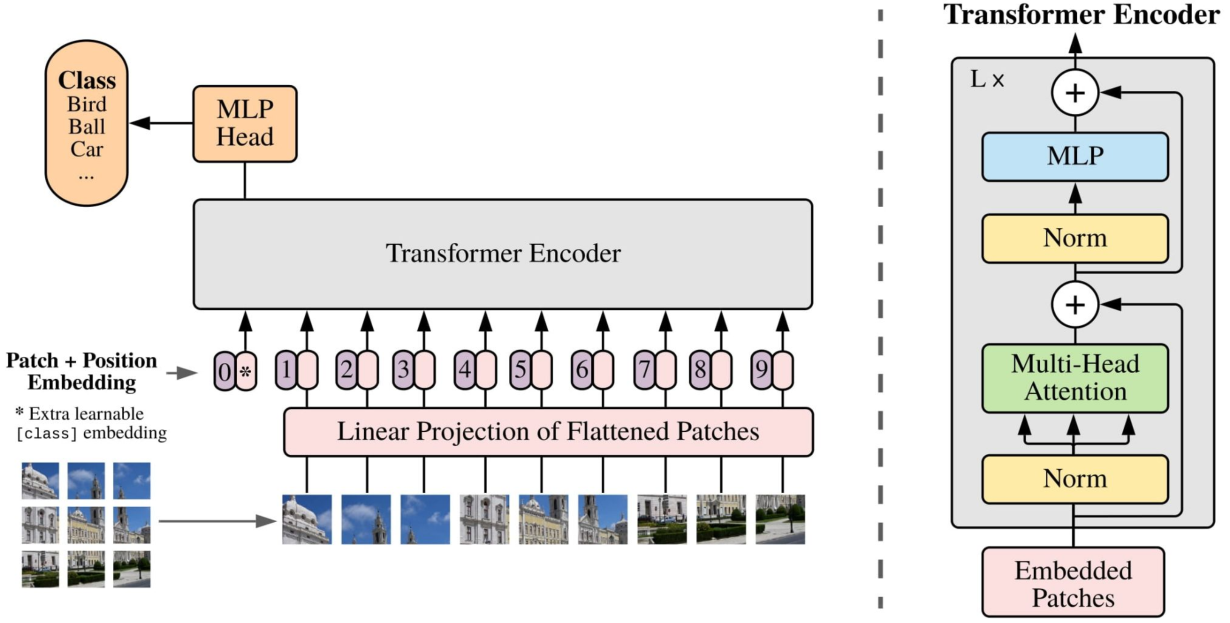


Figure 1. ViT Architecture

1.2 Related Work

Vision Transformer (ViT) ***(cont.)***

- Key features include:
  - Global attention capable of learning relationships between distant parts of the image (difficult for CNNs, needs many layers).
  - Scalability, less computational cost, less prone to overfitting than CNNs when scaled up and benefits more from large datasets.
  - No inductive biases, can learn any patterns in data without limits. However, this also makes ViT more reliant on large datasets to learn patterns effectively.

02

Inside DiT

2.1 Key Preliminaries

- Diffusion Formulation
- Classifier-Free Guidance
- Latent Diffusion Models

2.2 Final Architecture

- Design Overview
- Input Structure
- Block Details

2.1 Key Preliminaries

Diffusion Formulation ***(reminder)***

- Forward Process:
  - add noise to real data $$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$
  - sample (reparam. trick) $$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I)$$
- Reverse Process: $$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$ $$x_{t_{\max}} \sim \mathcal{N}(0, I) \qquad x_{t-1} \sim p_\theta(x_{t-1}|x_t)$$
  - full training loss (for Σ_θ) $$L(\theta) = -p(x_0|x_1) + \sum_t D_{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t))$$
  - simplified loss (for ϵ_θ) $$L_{\text{simple}}(\theta) = \|\epsilon_\theta(x_t) - \epsilon_t\|_2^2$$

2.1 Key Preliminaries

Classifier-Free Guidance

- Conditional Diffusion Models: $p_\theta(x_{t-1}|x_t, c)$
- Classifier-Free Guidance: need $p(c|x)$ so align with high $p(x|c)$
    - Why? Bayes' Rule! $\nabla_x \log p(c|x) \propto \nabla_x \log p(x|c) - \nabla_x \log p(x)$
    - Final formula? $\hat{\epsilon}_\theta(x_t, c) = \epsilon_\theta(x_t, \varnothing) + s \cdot (\epsilon_\theta(x_t, c) - \epsilon_\theta(x_t, \varnothing))$
    - Training?
        1. Randomly drop some c for null embedding to learn w/ and w/out c.
        2. If $s > 1$ then stronger focus on condition.
        3. if $s = 1$ then no guidance.

2.1 Key Preliminaries

Latent Diffusion Models

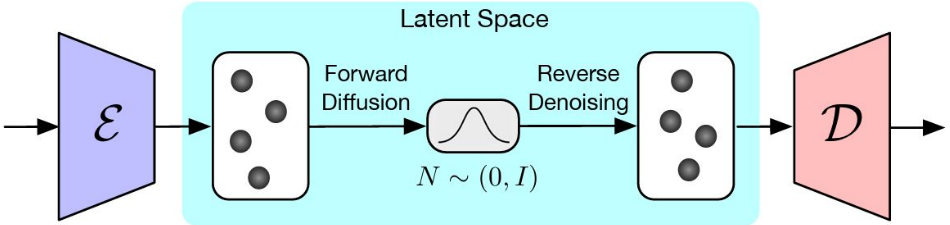


Figure 2. LDM

- **Motivation:** Pixel-space diffusion is expensive.
- **Solution:** LDMs!
  - Learn an autoencoder (VAE) for images x: $z = E(x)$
  - Train a diffusion model in the smaller latent space z.
  - Sample z from the diffusion model.
  - Decode z to an image with the decoder: $x = D(z)$
  - **Note:** E and D are pretrained and frozen!

Setting the Stage

- Core Idea
- Related Work

Inside DiT

- Preliminaries
- Architecture

Testing Grounds

- Setup
- Results
- Inference

Looking Ahead

- Applications
- Enhancement

2.2 Final Architecture

Design Overview

- Faithful to ViTs for its benefits!
- Process:
  - Take noised latent from VAE
  - Extract patches as tokens
  - Linearly embed tokens into d
  - Add sine-cosine pos embedding
  - Also process condition (time, label, etc.))
  - Pass through DiT block (more later)
  - Apply layer norm (can be adaptive)
  - Use linear decoder for ϵ & Σ

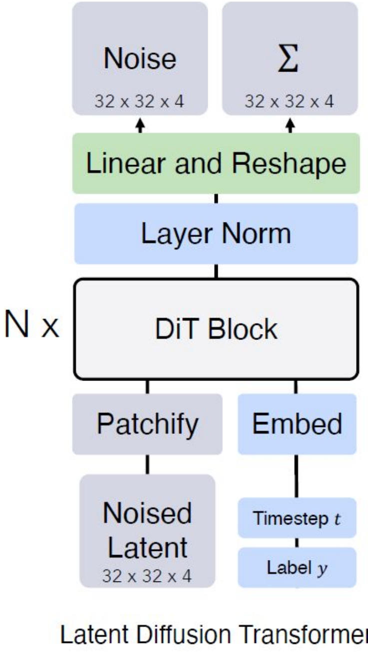


Figure 3. DiT Architecture

2.2 Final Architecture

Input Structure

- VAE's z shape: $I \times I \times C$
- Patch shape: $p \times p \times C$
- Patch count: $T = (I / p)^2$ (tokens)
- Input shape: $T \times d$

Note:

p does not affect parameter count,

but it affects transformer compute.

$\Rightarrow$ smaller p, increased compute.

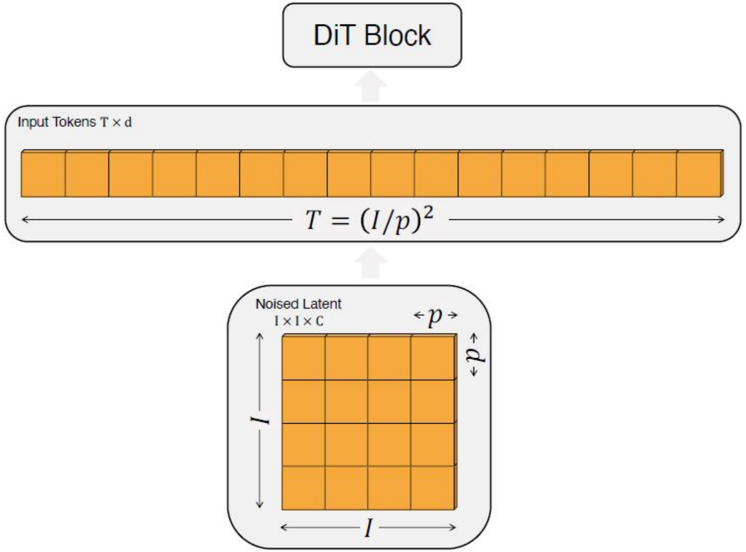


Figure 4. Input Specification for DiT

Setting the Stage

- Core Idea
- Related Work

Inside DiT

- Preliminaries
- Architecture

Testing Grounds

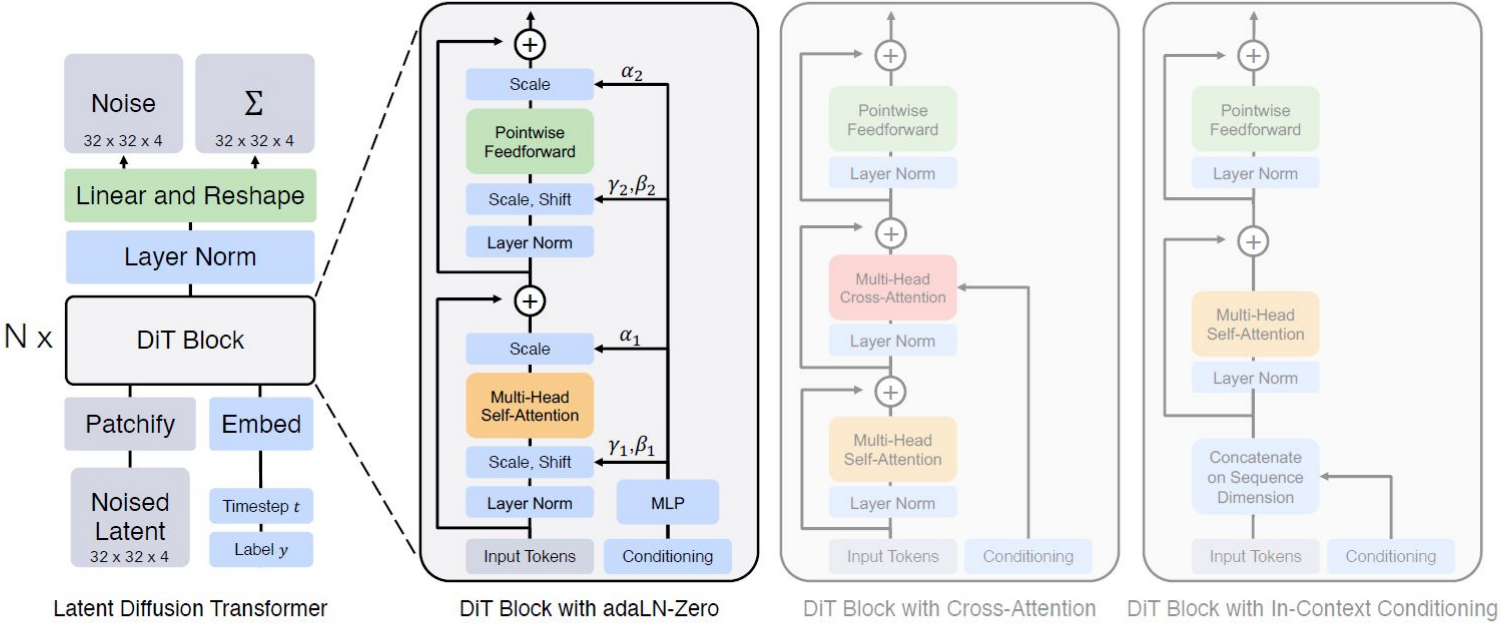
- Setup
- Results
- Inference

Looking Ahead

- Applications
- Enhancement

2.2 Final Architecture

Figure 5. ***Details of DiT Block Architecture***



Setting the Stage

- Core Idea
- Related Work

Inside DiT

- Preliminaries
- Architecture

Testing Grounds

- Setup
- Results
- Inference

Looking Ahead

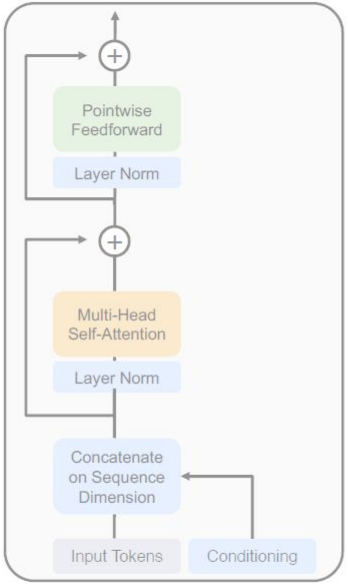
- Applications
- Enhancement

2.2 Final Architecture

Block Details ***(cont.)***

In-Context Conditioning Block | **Design 1**

- **Process:**
  - Append conditional info (timestep t or label c) to the input sequence as regular tokens.
  - Proceed with ViT as before.
  - Remove conditional tokens at the end of block.
- **Pros & Cons:**
  - Simple, low overhead, and compatible with ViT.
  - Little flexibility or sophistication in processing.



DiT Block with In-Context Conditioning

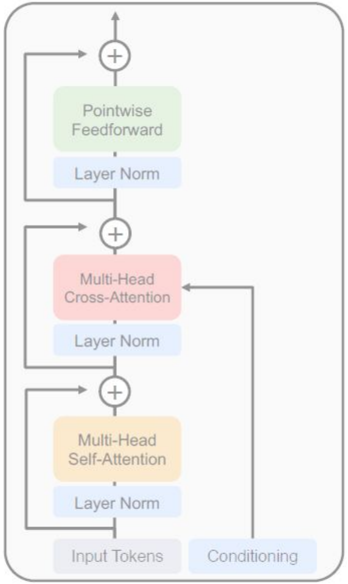
Figure 5.1

2.2 Final Architecture

Block Details ***(cont.)***

Cross-Attention Block | **Design 2**

- **Process:**
  - Create separate sequence for conditional info (timestep t or label c).
  - Use cross-attention to attend to every image token via the conditional tokens.
- **Pros & Cons:**
  - More sophisticated and interactive.
  - Large computational overhead.



DiT Block with Cross-Attention

Figure 5.2

2.2 Final Architecture

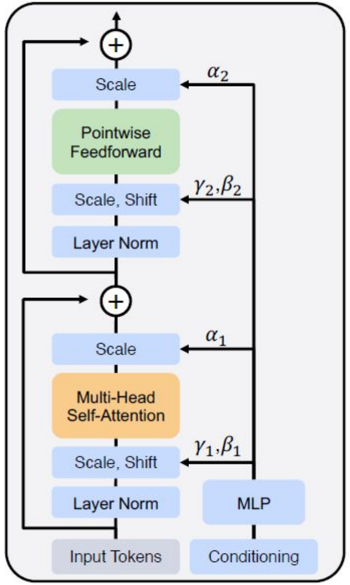
Block Details ***(cont.)***

adaLN-Zero Block | **Design 3**

- **Formula:**
    - Standard LayerNorm: $\hat{x} = \frac{x - \mu}{\sigma} \cdot \gamma + \beta$
    - adaLN: $\beta, \gamma = \mathrm{MLP}(t + c)$
    - adaLN-Zero: $\mathrm{Output} = x + \alpha \cdot f(\hat{x})$

$$\alpha, \beta, \gamma = MLP(t + c) \quad \text{all } 0 - \text{init}$$

- **Pros & Cons:**
    - Better adaptation, almost no overhead, faster.
    - More restricted (same norm on all tokens).



DiT Block with adaLN-Zero

Figure 5.3

03

Testing Grounds

3.1 Experimental Setup

Model Complexity Metrics

- Parameter Count
  - Total number of trainable parameters in a model.
  - Used as proxy for model complexity.
  - Does not account for image resolution!
- GFLOPS (Giga Floating-Point Operation Per Second)
  - Floating point calculation during one forward pass.
    - Matrix multiplication
    - Addition
    - Transformation of data
  - Accounts for both parameter utilization and image resolution!

3.1 Experimental Setup

Model Design Space

- Hyperparameters of test design space in Table 1.
- Additionally, patch sizes considered: p = 2, 4, 8

Model	Layers N	Hidden size d	Heads	Gflops (I=32, p=4)
DiT-S	12	384	6	1.4
DiT-B	12	768	12	5.6
DiT-L	24	1024	16	19.7
DiT-XL	28	1152	16	29.1

Table 1. Details of DiT Model Designs

3.1 Experimental Setup

Other Settings

- Data:
  - ImageNet Datasets: $256 \times 256 \quad \text{and} \quad 512 \times 512$
  - Only augmentation used: horizontal flip
- Optimization:
  - AdamW: $\mathrm{LR} = 10e - 4$
  - No weight decay!
- EMA (Exponential Moving Average) maintained like all gen. lit. and hyperparameters retained from ADM (Adversarial Diffusion Model)

3.1 Experimental Setup

Other Settings ***(cont.)***

- VAE from Stable Diffusion:

$$x_{shape} = 256\times256\times3 \xrightarrow{z=E(x)} z_{shape} = 32\times32\times3$$

- Evaluation Metrics:
  - FID (Fréchet Inception Distance)
  - IS (Inception Score)
  - Precision/Recall
- Compute:
  - Implemented in JAX
  - Trained at 5.7 itr/s on TPU v3-256

$$\frac{(5.7\,\text{itr/s}\times 800{,}000\,\text{itr})}{(60\,\text{min}\times 60\,\text{s})}\times 2\,\$ \approx 2{,}500\,\$$$

$$\text{TPU V3 Price} = 2\ \$/\text{hour}$$

3.2 Results & Analysis

Conditioning Strategies

- Based on Transformer Complexity $O(T^2 d)$, DiT blocks' compute equals:

DiT Block Type	GFLOPS Overhead	GFLOPS
In-Context Conditioning Block	Negligible	119.4
Cross-Attention Block	~15% increase	137.6
AdaLN Block	Minimal	118.6
AdaLN-Zero Block	Minimal	118.6

Table 2. Details of DiT Model Designs

3.2 Results & Analysis

Comparing different conditioning strategies *(DiT Block Types)*

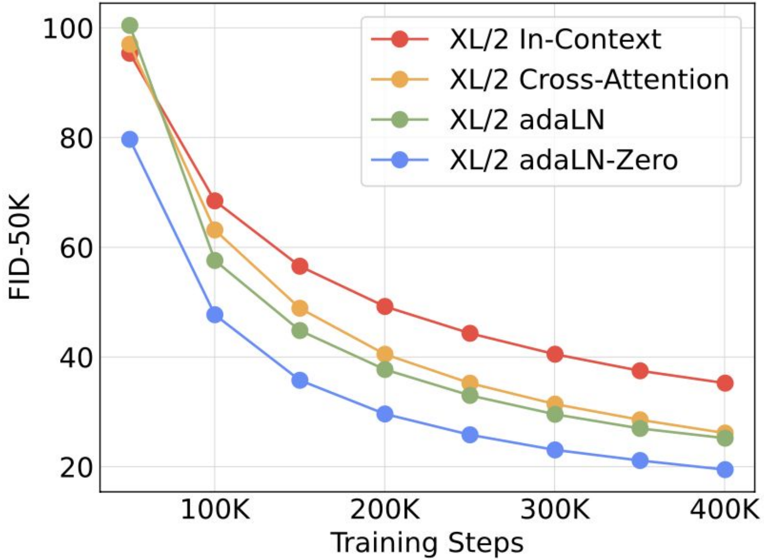


Figure 6. FIDs of DiT Blocks

Setting the Stage

- Core Idea
- Related Work

Inside DiT

- Preliminaries
- Architecture

Testing Grounds

- Setup
- Results
- Inference

Looking Ahead

- Applications
- Enhancement

3.2 Results & Analysis

ImageNet generation with Diffusion Transformers (DiTs)

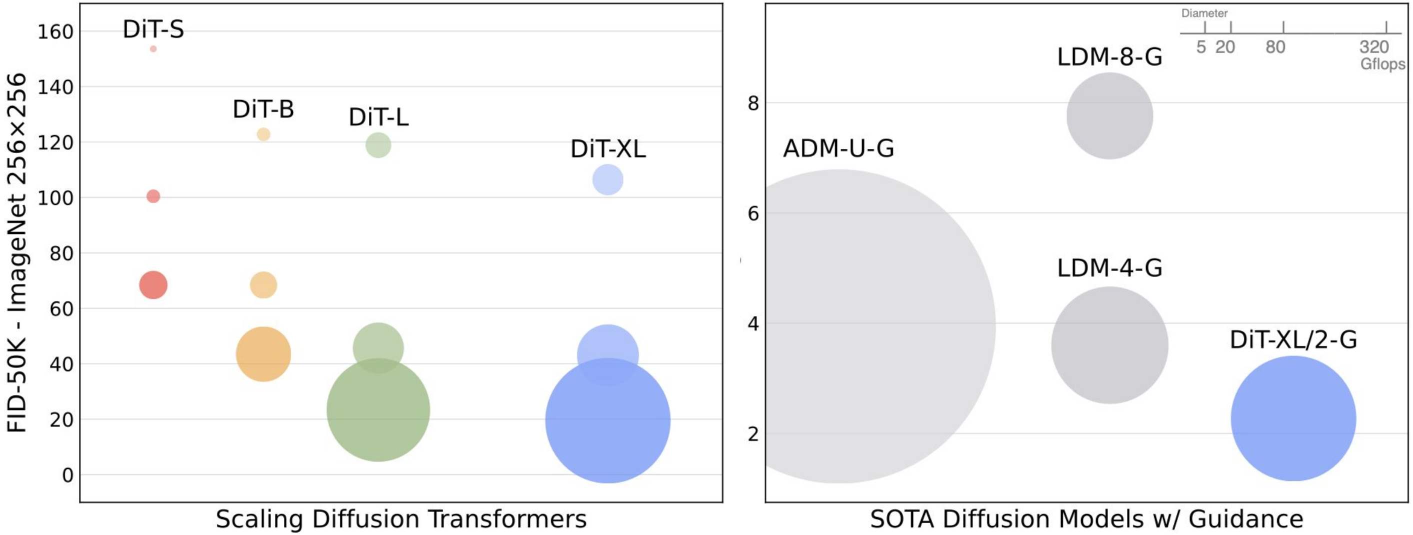


Figure 7. Overall FID on ImageNet

3.2 Results & Analysis

Scaling the DiT model improves FID at all stages of training

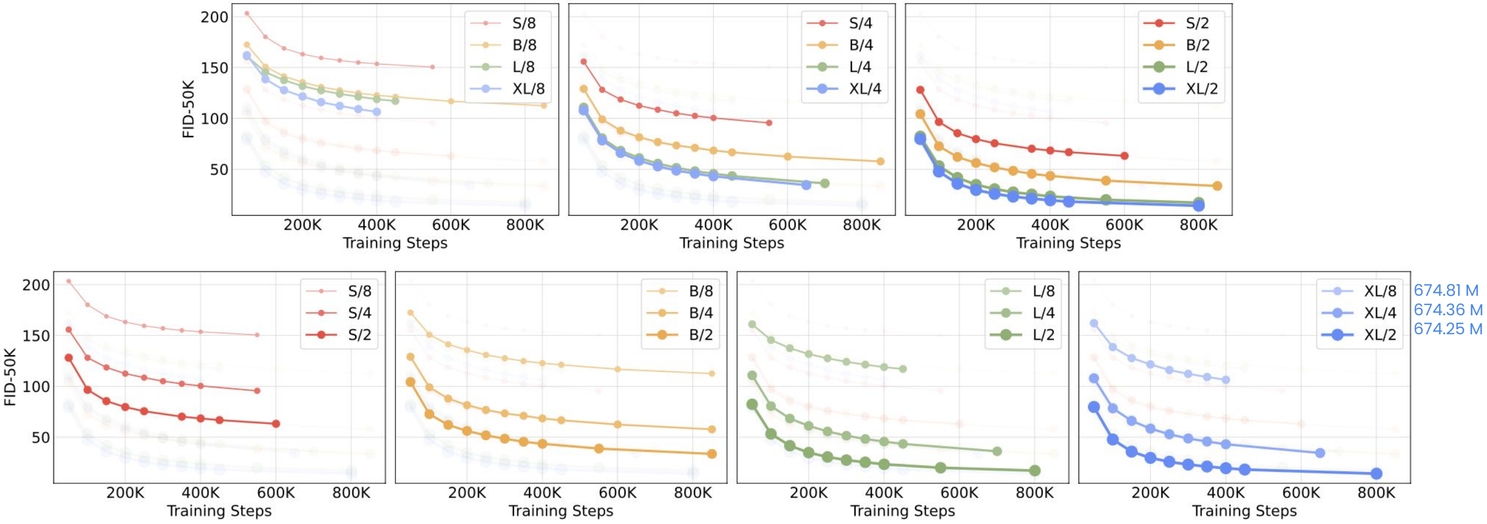


Figure 8. Model Scaling Effects

3.2 Results & Analysis

Increasing transformer forward pass Gflops increases sample quality

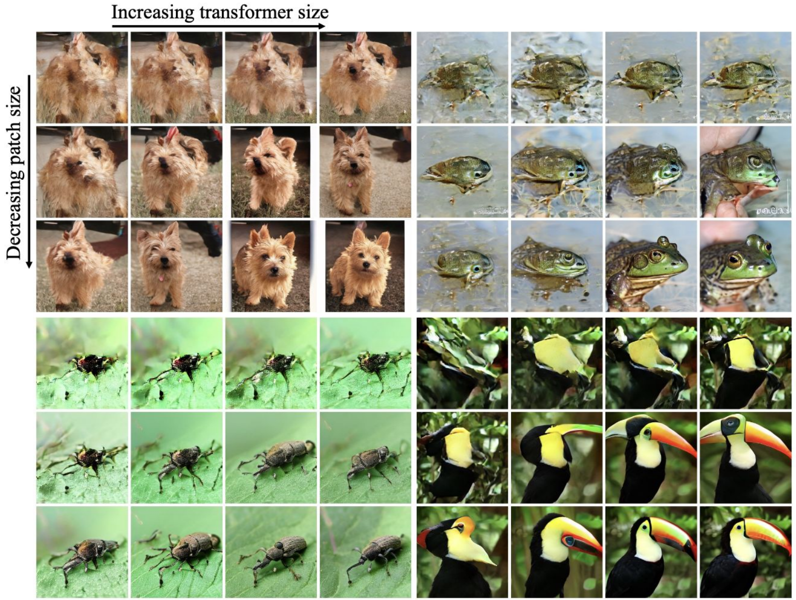


Figure 9. Model Scaling Effects 2

Setting the Stage

- Core Idea
- Related Work

Inside DiT

- Preliminaries
- Architecture

Testing Grounds

- Setup
- Results
- Inference

Looking Ahead

- Applications
- Enhancement

3.2 Results & Analysis

Transformer Gflops are strongly correlated with FID

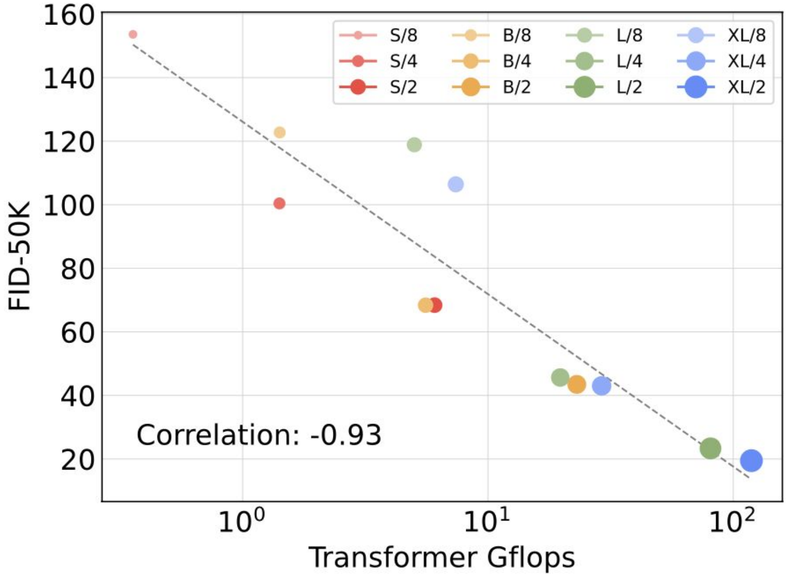


Figure 10. Model Scaling Effects 3

3.2 Results & Analysis

Benchmarking class-conditional generation on ImageNet

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

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

Table 3. Vs State-of-the-art Methods

Setting the Stage

- Core Idea
- Related Work

Inside DiT

- Preliminaries
- Architecture

Testing Grounds

- Setup
- Results
- Inference

Looking Ahead

- Applications
- Enhancement

3.2 Results & Analysis

Scaled-up sampling compute does not compensate for a lack of model compute

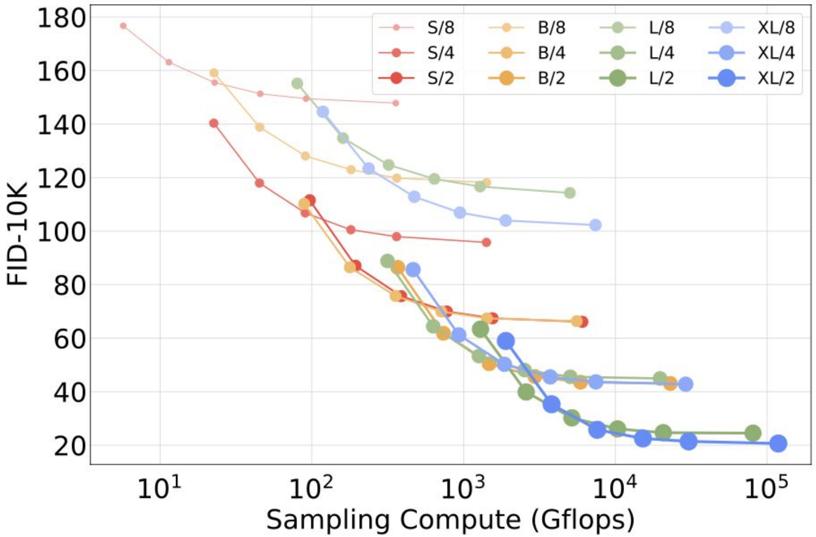


Figure 11. Model Scaling 4

Setting the Stage

- Core Idea
- Related Work

Inside DiT

- Preliminaries
- Architecture

Testing Grounds

- Setup
- Results
- Inference

Looking Ahead

- Applications
- Enhancement

3.3 Inference in Practice

Let's take a look at some code! *inference time < 1 min*

Figure 12. DiT Generations

04

Looking Ahead

4.1 Applications

- OpenAI Sora
- Other Models

4.2 Enhancements

- Limitations
- Future Work

4.1 Applications

Sora: A Diffusion Transformer

OpenAI   Research   Products   Safety   Company

Scaling transformers for video generation

Sora is a diffusion model[21, 22, 23, 24, 25]; given input noisy patches (and conditioning information like text prompts), it's trained to predict the original "clean" patches. Importantly, Sora is a diffusion *transformer*.[26] Transformers have demonstrated remarkable scaling properties across a variety of domains, including language modeling,[13, 14] computer vision,[15, 16, 17, 18] and image generation.[27, 28, 29]

Figure 13. Sora Page at OpenAI

4.1 Applications

Sora: A Diffusion Transformer *(cont.)*

- OpenAI's Sora has a DiT architecture:

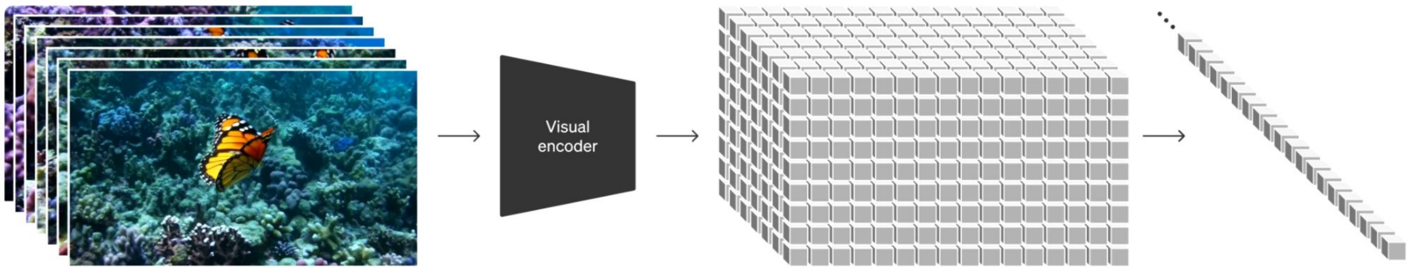


Figure 14. Sora Architecture

4.1 Applications

Other Models Include:

- DeepMind's Veo2 AI
- NVIDIA's Cosmos World Foundation Model For Physical AI

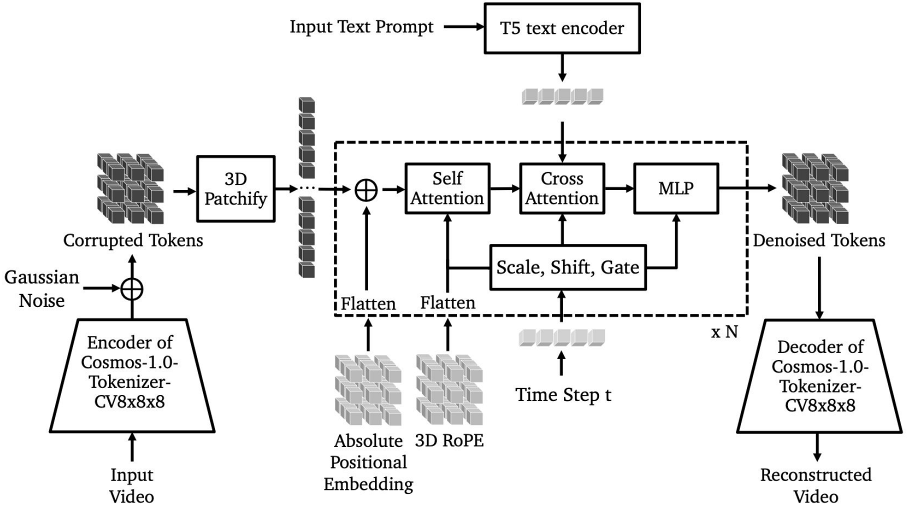


Figure 15. NVIDIA's Cosmos WFM

4.2 Enhancement

Limitations

- Computational Inefficiency
  - Training Cost
  - Inference Latency
- High Memory Usage
- Limited Adaptability
  - Task-Specific Fine-Tuning
  - Adaptation to Non-Image Data
- Sensitive to HyperParameter Settings
- Quality of Results

4.2 Enhancement

Future Works

- Enhancing Training Efficiency
  - *Optimization Algorithms*
  - *Regularization Methods*
  - *Loss Function*
  - *Sparse Attention*
- Accelerating Inference
  - *Sampling Algorithms*
  - *Cache Mechanisms*
  - *Dynamic Architecture*
  - *Token Pruning*
- Improving Scalability
  - *Hybrid Architectures*
  - *Multi-Scale Tokenization*
- Expanding Modality
  - *Cross-Modal Learning*
  - *3D Applications*
  - *Audio*
- Reduce Model Size
  - *Quantization*
  - *Pruning*

REFERENCES

1. [LINK] Peebles, W., & Xie, S. (2023). **Scalable Diffusion Models with Transformers.** *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*
2. [LINK] Dosovitskiy, A., et al. (2021). **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.** *International Conference on Learning Representations (ICLR)*
3. [LINK] Fu, C., et al. (2023). **A Latent Diffusion Model for Protein Structure Generation.** *Second Learning on Graphs Conference (LoG 2023)*
4. [LINK] Liu, Y., et al. (2024). **Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models.** *arXiv:2402.17177 [cs.CV]*
5. [LINK] **Video generation models as world simulator: Sora** (2024). *OpenAI*. openai.com
6. [LINK] **Veo 2.** (2024). *Google DeepMind.* deepmind.google
7. [LINK] NVIDIA: Agarwal, N., et al. (2025). **Cosmos World Foundation Model Platform for Physical AI.** *arXiv:2501.03575 [cs.CV]*

THANKS!

Any questions?

Maryam Rezaee & Mahshid Dehghani

TGML Lab | Sharif University of Technology

Under the supervision of

Dr. Fatemeh SeyyedSalehi

Template by Slidesgo and Maryam Rezaee