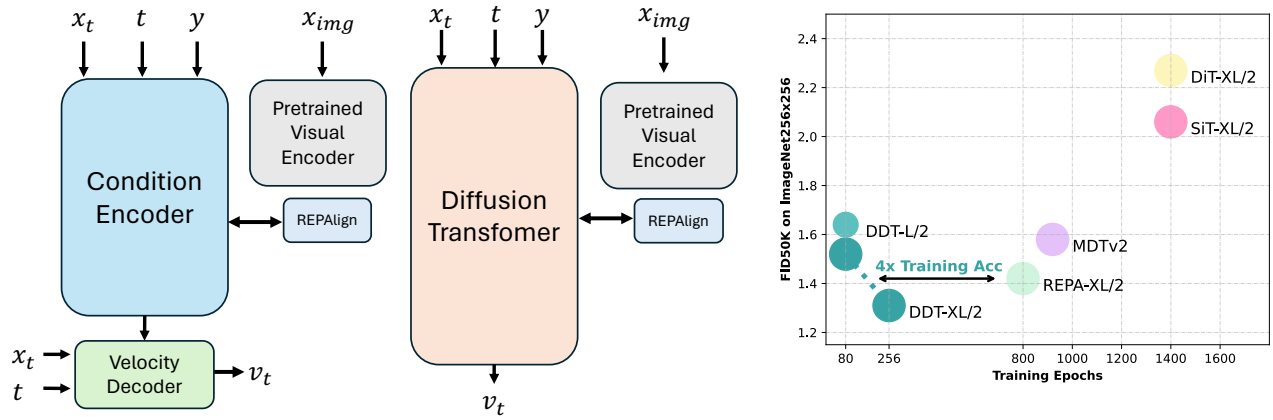
This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

DDT: Decoupled Diffusion Transformer

Shuai Wang[1] Zhi Tian[2] Weilin Huang[2] Limin Wang [1,✉]
[1]Nanjing University [2]ByteDance Seed

https://github.com/MCG-NJU/DDT



(a) Our Decoupled Diffusion Transformer (b) Conventional Diffusion Transformer (c) FID compared with Other Diffusion Models

Figure 1. **Our deoupled diffusion transformer (DDT-XL/2) achieves a SoTA 1.31 FID under 256 epochs.** Our decoupled diffusion transformer models incorporate a condition encoder to extract semantic self-conditions and a velocity decoder to decode velocity.

Abstract

Diffusion transformers have demonstrated remarkable generation quality, albeit requiring longer training iterations and numerous inference steps. In each denoising step, diffusion transformers encode the noisy inputs to extract the lower-frequency semantic component and then decode the higher frequency with identical modules. This scheme creates an inherent optimization dilemma: encoding low-frequency semantics necessitates reducing high-frequency components, creating tension between semantic encoding and high-frequency decoding. To resolve this challenge, we propose a new ***D****ecoupled* ***D****iffusion* ***T****ransformer (****DDT****), with a decoupled design of a dedicated condition encoder for semantic extraction alongside a specialized velocity decoder. Our experiments reveal that a more substantial encoder yields performance improvements as model size increases. For ImageNet* 256×256*, Our DDT-XL/2 achieves a new state-of-the-art performance of 1.31 FID (nearly* $4\times$ *faster training convergence compared to previous diffusion transformers). For ImageNet* 512×512*, Our DDT-XL/2 achieves a new state-of-the-art FID of 1.28. Additionally, as a beneficial by-product, our decoupled architecture enhances inference speed by enabling the sharing self-condition between adjacent denoising steps. To minimize performance degradation, we propose a novel statistical dynamic programming approach to identify optimal sharing strategies.*

1. Introduction

Image generation is a fundamental task in computer vision research, which aims at capturing the inherent data distribution of original image datasets and generating high-quality synthetic images through distribution sampling. Diffusion models [19, 21, 28, 29, 40] have recently emerged as highly promising solutions to learn the underlying data distribution in image generation, outperforming the GAN-based models [3, 39] and Auto-Regressive models [5, 42, 50].

The diffusion forward process gradually adds Gaussian noise to the pristine data following an SDE forward schedule [19, 21, 40]. The denoising process learns the score estimation from this corruption process. Once the score function is accurately learned, data samples can be synthesized by numerically solving the reverse SDE [21, 28, 29, 40].

✉ : Corresponding author (lmwang@nju.edu.cn).

Diffusion Transformers [31, 35] introduce the transformer architecture into diffusion models to replace the traditionally dominant UNet-based model [2, 10]. Empirical evidence suggests that, given sufficient training iterations, diffusion transformers outperform conventional approaches even without relying on long residual connections [35]. Nevertheless, their slow convergence rate still poses great challenge for developing new models due to the high cost.

In this paper, we want to tackle the aforementioned major disadvantages from a model design perspective. Classic computer vision algorithms [4, 17, 22] strategically employ encoder-decoder architectures, prioritizing large encoders for rich feature extraction and lightweight decoders for efficient inference, while contemporary diffusion models predominantly rely on conventional decoder-only structures. We systematically investigate the underexplored potential of decoupled encoder-decoder designs in diffusion transformers, by answering the question of *can decoupled encoder-decoder transformer unlock the capability of accelerated convergence and enhanced sample quality?*

Through investigation experiments, we conclude that the plain diffusion transformer has an optimization dilemma between abstract structure information extraction and detailed appearance information recovery. Further, the diffusion transformer is limited in extracting semantic representation due to the raw pixel supervision [27, 51, 52]. To address this issue, we propose a new architecture to explicitly decouple low-frequency semantic encoding and high-frequency detailed decoding through a customized encoder-decoder design. We call this encoder-decoder diffusion transformer model as **DDT** (**D**ecoupled **D**iffusion **T**ransformer). DDT incorporates a *condition encoder* to extract semantic self-condition features. The extracted self-condition is fed into a *velocity decoder* along with the noisy latent to regress the velocity field. To maintain the local consistency of self-condition features of adjacent steps, we employ direct supervision of representation alignment and indirect supervision from the velocity regression loss of the decoder.

In the ImageNet256×256 dataset, using the traditional off-shelf VAE [37], our decoupled diffusion transformer (DDT-XL/2) model achieves the state-of-the-art performance of 1.31 FID with interval guidance under only 256 epochs, approximately $4\times$ training acceleration compared to REPA [51]. In the ImageNet512×512 dataset, our DDT-XL/2 model achieves 1.28 FID within 500K finetuning steps.

Furthermore, our DDT achieves strong local consistency on its self-condition feature from the encoder. This property can significantly boost the inference speed by sharing the self-condition between adjacent steps. We formulate the optimal encoder sharing strategy solving as a classic minimal sum path problem by minimizing the performance drop of sharing self-condition among adjacent steps. We propose a statistic dynamic programming approach to find the optimal encoder sharing strategy with negligible second-level time cost. Compared with the naive uniform sharing, our dynamic programming delivers a minimal FID drop. Our contributions are summarized as follows.

- We propose a new decoupled diffusion transformer model, which consists of a condition encoder and a velocity decoder.
- We propose statistic dynamic programming to find the optimal self-condition sharing strategy to boost inference speed while keeping minimal performance downgradation.
- In the ImageNet256×256 dataset, using tradition SDf8d4 VAE, our decoupled diffusion transformer (DDT-XL/2) model achieves the SoTA 1.31 FID with interval guidance under only 256 epochs, approximately $4\times$ training acceleration compared to REPA [51].
- In the ImageNet512×512 dataset, our DDT-XL/2 model achieves the SoTA 1.28 FID, outperforming all previous methods with a significant margin.

2. Related Work

Diffusion Transformers. The pioneering work of DiT [35] introduced transformers into diffusion models to replace the traditionally dominant UNet architecture [2, 10]. Empirical evidence demonstrates that given sufficient training iterations, diffusion transformers outperform conventional approaches even without relying on long residual connections. SiT [31] further validated the transformer architecture with linear flow diffusion. Following the simplicity and scalability of the diffusion transformer [31, 35], SD3 [12], Lumina [53], and PixArt [6, 7] introduced the diffusion transformer to more advanced text-to-image areas. Moreover, recently, diffusion transformers have dominated the text-to-video area with substantiated visual and motion quality [1, 20, 23]. Our decoupled diffusion transformer (DDT) presents a new variant within the diffusion transformer family. It achieves faster convergence by decoupling the low-frequency encoding and the high-frequency decoding.

Fast Diffusion Training. To accelerate the training efficiency of diffusion transformers, recent advances have pursued multi-faceted optimizations. Operator-centric approaches [13, 44, 47, 48] leverage efficient attention mechanisms: linear-attention variants [13, 44, 48] reduced quadratic complexity to speed up training, while sparse-attention architectures [47] prioritized sparsely relevant token interactions. Resampling approaches [12, 16] proposed lognorm sampling [12] or loss reweighting [16] techniques to stabilize training dynamics. Representation learning enhancement approaches integrate external inductive biases:

Figure 2. **Selected 256×256 and 512×512 resolution samples.** Generated from DDT-XL/2 trained on ImageNet 256×256 resolution and ImageNet 512×512 resolution with CFG = 4.0.

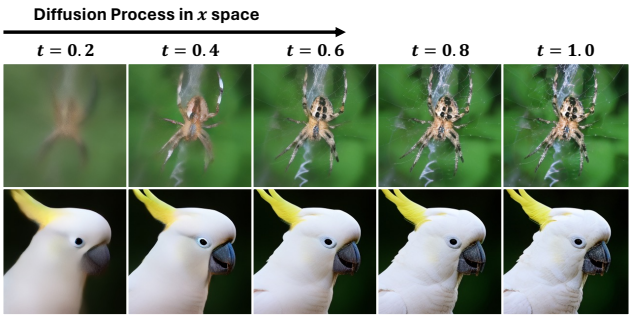


Figure 3. **The reverse-SDE process (generation) of SiT-XL/2 in x space.** There is a clear generation process from low frequency to high frequency. Most of the time is spent on generating high-frequency details (from $t = 0.4$ to $t = 1.0$).

REPA [51], RCG [26] and DoD [52] borrowed vision-specific priors into diffusion training, while masked modeling techniques [14, 15] strengthened spatial reasoning by enforcing structured feature completion during denoising. Collectively, these strategies address computational, sampling, and representational bottlenecks.

3. Preliminary Analysis

Linear-based flow matching [28, 29, 31] represents a specialized family of diffusion models that we focus on as our primary analytical subject due to its simplicity and efficiency. For the convenience, in certain situations, diffusion and flow-matching will be used interchangeably. In this

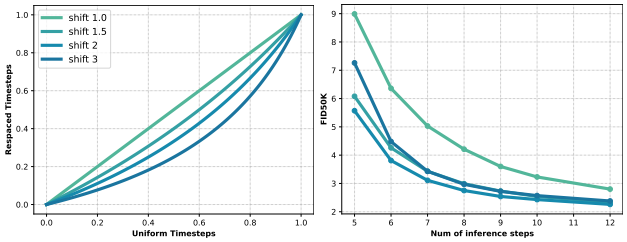


Figure 4. **The FID50K metric of SiT-XL/2 for different timeshift values.** We employ a 2-nd order Adams-like solver to collect the performance. Allocating more computation at noisy steps significantly improves the performance.

framework, $t = 0$ corresponds to the pure noise timestep.

As illustrated in Fig. 3, diffusion models perform autoregressive refinement on spectral components [11, 36]. The diffusion transformer encodes the noisy latent to capture lower-frequency semantics before decoding higher-frequency details. However, this semantics encoding process inevitably attenuates high-frequency information, creating an optimization dilemma. This observation motivates our proposal to decouple the conventional decode-only diffusion transformer into an explicit encoder-decoder arch.

When t increases to less noisy timesteps, semantic encoding becomes easier (due to noise reduction) while decoding complexity increases(See Appendix). Consider $f_{max}(t)$ as the maximum remaining frequency of the clean data at the denoising step t, the diffusion transformer encodes frequencies up to $f_{max}(t)$ and to progress to step s, it must decode a residual frequency of at least $f_{max}(s) -$

$f_{max}(t)$. Failure to decode these residual frequencies at step t creates a critical bottleneck for progression to subsequent steps. From this perspective, if allocating more of the calculations to more noisy timesteps can lead to an improvement, it means that diffusion transformers struggle with encoding lower frequencies to provide semantics. Otherwise, if allocating more of the calculations to less noisy timesteps can lead to an improvement, it means that flow-matching transformers struggle with decoding frequency residuals.

To figure out the bottom-necks of current diffusion models, we conducted a targeted experiment using SiT-XL/2 with a second-order Adams-like linear multistep solver. As shown in Fig. 4, by varying the time-shift values, we demonstrate that allocating more computation to early timesteps improves final performance compared to uniform scheduling. This reveals that diffusion models face challenges in more noisy steps. This leads to a key conclusion: **Current diffusion transformers are fundamentally constrained by their low-frequency semantic encoding capacity**. This insight motivates the exploration of encoder-decoder architectures with strategic encoder parameter allocation and validation with vigorous experiments.

Prior researches further support this perspective. While lightweight diffusion MLP heads demonstrate limited decoding capacity, MAR [27] overcomes this limitation through semantic features produced by its masked backbones, enabling high-quality image generation. Similarly, REPA [51] enhances low-frequency encoding through alignment with pre-trained vision foundations [34].

4. Method

Our decoupled diffusion transformer architecture comprises a condition encoder and a velocity decoder. The condition encoder extracted the low-frequency component from noisy input, class label, and timestep to serve as a self-condition for the velocity decoder; the velocity decoder processed the noisy latent with the self-condition to regress the high-frequency velocity. We train this model using the established linear flow diffusion framework. For brevity, we designate our model as **DDT** (**D**ecoupled **D**iffusion **T**ransformer).

4.1. Condition Encoder

The condition encoder mirrors the architectural design and input structure of DiT/SiT with improved micro-design. It is built with interleaved Attention and FFN blocks, without long residual connections. The encoder processes three inputs, the noisy latent $\boldsymbol{x}_t$, timestep t, and class label y, to extract the self-condition feature $\boldsymbol{z}_t$ through a series of stacked Attention and FFN blocks:

$$\boldsymbol{z}_t = \textbf{Encoder}\left(\boldsymbol{x}_t, t, y\right). \tag{1}$$

Specifically, the noisy latent $\boldsymbol{x}_t$ are patchfied into continuous tokens and then fed to extract the self-condition $\boldsymbol{z}_t$ with aforementioned encoder blocks. The timestep t and class label y serve as external-conditioning information projected into embedding. These external-condition embeddings are progressively injected into the encoded features of $\boldsymbol{x}_t$ using AdaLN-Zero[35] within each encoder block.

To maintain local consistency of $\boldsymbol{z}_t$ across adjacent timesteps, we adopt the representation alignment technique from REPA [51]. Shown in Eq. (2), this method aligns the intermediate feature $\mathbf{h}_i$ from the i-th layer in the condition encoder with the DINOv2 representation r_*. Consistent to REPA [51], the h_ϕ is the learnable projection MLP:

$$\mathcal{L}_{enc} = 1 - \cos(r_*, h_\phi(\mathbf{h_i})). \tag{2}$$

This simple regularization accelerates training convergence, as shown in REPA [51], and facilitates local consistency of $\boldsymbol{z}_t$ between adjacent steps. It allows sharing the self-condition $\boldsymbol{z}_t$ produced by the encoder between adjacent steps. Our experiments demonstrate that this encoder-sharing strategy significantly enhances inference efficiency with only negligible performance degradation.

Additionally, the encoder also receives indirect supervision from the decoder, which we elaborate on later.

4.2. Velocity Decoder

The velocity decoder adopts the same architectural design as the condition encoder and consists of several stacked interleaved Attention and FFN blocks, akin to DiT/SiT. It takes the noisy latent $\boldsymbol{x}_t$, timestep t, and self-conditioning $\boldsymbol{z}_t$ as inputs to estimate the velocity $\boldsymbol{v}_t$. Unlike the encoder, we assume that class label information is already embedded within $\boldsymbol{z}_t$. Thus, only the external-condition timestep t and self-condition feature $\boldsymbol{z}_t$ are used as condition inputs for the decoder blocks:

$$\boldsymbol{v}_t = \textbf{Decoder}\left(\boldsymbol{x}_t, t, \boldsymbol{z}_t\right). \tag{3}$$

As demonstrated previously, to further improve consistency of self-condition $\boldsymbol{z}_t$ between adjacent steps, we employ AdaLN-Zero [35] to inject $\boldsymbol{z}_t$ into the decoder feature. The decoder is trained with the flow matching loss as shown in Eq. (4):

$$\mathcal{L}_{dec} = \mathbb{E}[\int_0^1 ||(\boldsymbol{x}_{data} - \epsilon) - \boldsymbol{v}_t||^2 \mathrm{d}t]. \tag{4}$$

4.3. Sampling acceleration

By incorporating explicit representation alignment into the encoder and implicit self-conditioning injection into the decoder, we achieve local consistency of $\boldsymbol{z}_t$ across adjacent steps during training (shown in Fig. 5). This enables us to share $\boldsymbol{z}_t$ within a suitable local range, reducing the computational burden on the self-mapping encoder.

Formally, given total inference steps N and encoder computation bugets K, thus the sharing ratio is $1 - \frac{K}{N}$, we define Φ with $|\Phi| = K$ as the set of timesteps where the self-condition is recalculated, as shown in Equation 5. If the current timestep t is not in Φ, we reuse the previously computed $\boldsymbol{z}_{t-\Delta t}$ as $\boldsymbol{z}_t$. Otherwise, we recompute $\boldsymbol{z}_t$ using the encoder and the current noisy latent $\boldsymbol{x}_t$:

$$\boldsymbol{z}_t = \begin{cases} \boldsymbol{z}_{t-\Delta t}, & \text{if } t \notin \Phi \\ \textbf{Encoder}\left(\boldsymbol{x}_t, t, y\right), & \text{if } t \in \Phi \end{cases} \tag{5}$$

Uniform Encoder Sharing. This naive approach recaluculate self-condition $\boldsymbol{z}_t$ every $\frac{N}{K}$ steps. Previous work, such as DeepCache [32], uses this naive handcrafted uniform Φ set to accelerate UNet models. However, UNet models, trained solely with a denoising loss and lacking robust representation alignment, exhibit less regularized local consistency in deeper features across adjacent steps compared to our DDT model. Also, we will propose a simple and elegant statistic dynamic programming algorithm to construct Φ. Our statistic dynamic programming can exploit the optimal Φ set optimally compared to the naive approaches [32]. We will elaborate on our novel approach below.

Statistic Dynamic Programming. We construct the statistic similarity matrix of z_t among different steps $\mathbf{S} \in R^{N\times N}$ using cosine distance. The optimal Φ set would guarantee the total similarity cost $-\sum_k^K \sum_{i=\Phi_k}^{\Phi_{k+1}} S[\Phi_k, i]$ achieves global minimal. This question is a well-formed classic minimal sum path problem, it can be solved by dynamic programming. As shown in Eq. (7), we donate $\mathbf{C}_i^k$ as cost and $\mathbf{P}_i^k$ as traced path when $\Phi_k = i$. the state transition function from $\mathbf{C}_j^{k-1}$ to $\mathbf{C}_i^k$ follows:

$$\mathbf{C}_i^k = \min_{j=0}^{i}\{\mathbf{C}_j^{k-1} - \Sigma_{l=j}^{i}\mathbf{S}[j,l]\}. \tag{6}$$

$$\mathbf{P}_i^k = \operatorname{argmin}_{j=0}^{i}\{\mathbf{C}_i^{k-1} - \Sigma_{l=j}^{i}\mathbf{S}[j,l]\}. \tag{7}$$

After obtaining the cost matrix $\mathbf{C}$ and tracked path $\mathbf{P}$, the optimal Φ can be solved by backtracking $\mathbf{P}$ from $\mathbf{P}_N^K$.

5. Experiment

We conduct experiments on 256x256 ImageNet datasets. The total training batch size is set to 256. Consistent with methodological approaches such as SiT [31], DiT [35], and REPA [51], we employed the Adam optimizer with a constant learning rate of 0.0001 throughout the entire training process. To ensure a fair comparative analysis, we did not use gradient clipping and learning rate warm-up techniques. Our default training infrastructure consisted of $16\times$ or $8\times$ A100 GPUs. For sampling, we take the Euler solver with 250 steps as the default choice. As for the VAE, we take the off-shelf VAE-ft-EMA with a downsample factor of 8 from Huggingface[1]. We report FID [18], sFID [33], IS [38], Precision and Recall [24].

5.1. Improved baselines

Recent architectural improvements such as SwiGLU [45, 46], RoPE [41], and RMSNorm [45, 46] have been extensively validated in the research community [8, 30, 49]. Additionally, lognorm sampling [12] has demonstrated significant benefits for training convergence. Consequently, we developed improved baseline models by incorporating these advanced techniques, drawing inspiration from recent works in the field. The performance of these improved baselines is comprehensively provided in Tab. 2. To validate the reliability of our implementation, we also reproduced the results for REPA-B/2, achieving metrics that marginally exceed those originally reported in the REPA[51]. These reproduction results provide additional confidence in the robustness of our approach.

The improved baselines in our Tab. 2 consistently outperform their predecessors without REPA. However, upon implementing REPA, performance rapidly approaches a saturation point. This is particularly evident in the XL model size, where incremental technique improvements yield diminishingly small gains.

5.2. Metric comparison with baselines

We present the performances of different-size models at 400K training steps in Tab. 2. Our diffusion encoder-decoder transformer(DDT) family demonstrates consistent and significant improvements across various model sizes. Our DDT-B/2(8En4De) model exceeds Improved-REPA-B/2 by 2.8 FID gains. Our DDT-XL/2(22En6De) exceeds REPA-XL/2 by 1.3 FID gains. While the decoder-only diffusion transformers approach performance saturation with REPA[51], our DDT models continue to deliver superior results. The incremental technique improvements show diminishing gains, particularly in larger model sizes. However, our DDT models maintain a significant performance advantage, underscoring the effectiveness of our approach.

Training and Inference comparison Our experiments utilized 8×A100 GPUs (with 32 batches per GPU for both training and inference), employing **torch.compile** to optimize memory allocation and reduce redundant computations for both the baseline and our DDT approach. The $4\times$ **step-wise** acceleration in DDT is hardware-agnostic, as the total batch size(256) remains consistent with baselines.

5.3. System level comparision

ImageNet 256×256**.** We report the final metrics of DDT-XL/2 (22En6De) and DDT-L/2 (20En4De) at Tab. 1. Our

[1] https://huggingface.co/stabilityai/sd-vae-ft-ema

	Params	Epochs	256×256, w/o CFG				256×256, w/ CFG			
			FID↓	IS↑	Pre.↑	Rec.↑	FID↓	IS↑	Pre.↑	Rec.↑
MAR-B [27]	208M	800	3.48	192.4	0.78	0.58	2.31	281.7	0.82	0.57
CausalFusion [9]	368M	800	5.12	166.1	0.73	0.66	1.94	264.4	0.82	0.59
LDM-4 [37]	400M	170	10.56	103.5	0.71	0.62	3.6	247.7	0.87	0.48
DDT-L (Ours)	458M	80	7.98	128.1	0.68	**0.67**	**1.64**	**310.5**	0.81	**0.61**
MAR-L [27]	479M	800	2.6	221.4	0.79	0.60	1.78	296.0	0.81	0.60
VAVAE [49]	675M	800	2.17	205.6	0.77	0.65	1.35	295.3	0.79	0.65
CausalFusion [9]	676M	800	3.61	180.9	0.75	0.66	1.77	282.3	0.82	0.61
ADM [10]	554M	400	10.94	-	0.69	0.63	4.59	186.7	0.82	0.52
DiT-XL [35]	675M	1400	9.62	121.5	0.67	0.67	2.27	278.2	0.83	0.57
SiT-XL [31]	675M	1400	8.3	-	-	-	2.06	270.3	0.82	0.59
ViT-XL [16]	451M	400	8.10	-	-	-	2.06	-	-	-
U-ViT-H/2 [2]	501M	400	6.58	-	-	-	2.29	263.9	0.82	0.57
MaskDiT [14]	675M	1600	5.69	178.0	0.74	0.60	2.28	276.6	0.80	0.61
FlowDCN [47]	618M	400	8.36	122.5	0.69	0.65	2.00	263.1	0.82	0.58
RDM [43]	553M	/	5.27	153.4	0.75	0.62	1.99	260.4	0.81	0.58
REPA [51]	675M	800	5.9	157.8	0.70	0.69	1.42	305.7	0.80	0.64
DDT-XL (Ours)	675M	80	6.62	135.2	0.69	0.67	1.52	263.7	0.78	0.63
DDT-XL (Ours)	675M	256	6.30	146.7	0.68	0.68	**1.31**	**308.1**	0.78	0.62
DDT-XL (Ours)	675M	400	6.27	154.7	0.68	0.69	**1.26**	**310.6**	0.79	**0.65**

Table 1. **System performance comparison** on ImageNet 256 × 256 class-conditioned generation. Gray blocks mean the algorithm uses VAE trained or fine-tuned on ImageNet instead of the off-shelf SD-VAE-f8d4-ft-ema.

Model	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-B/2 [31]	33.0	6.46	43.7	0.53	0.63
REPA-B/2 [51]	24.4	6.40	59.9	0.59	0.65
REPA-B/2(Reproduced)	22.2	7.50	69.1	0.59	0.65
DDT-B/2† (8En4De)	**21.1**	7.81	**73.0**	**0.60**	**0.65**
Improved-SiT-B/2	25.1	6.54	58.8	0.57	0.64
Improved-REPA-B/2	19.1	6.88	76.49	0.60	0.66
DDT-B/2 (8En4De)	**16.32**	**6.63**	**86.0**	**0.62**	**0.66**
SiT-L/2 [31]	18.8	5.29	72.0	0.64	0.64
REPA-L/2 [51]	10.0	5.20	109.2	0.69	0.65
Improved-SiT-L/2	12.7	5.48	95.7	0.65	0.65
Improved-REPA-L/2	9.3	5.44	116.6	0.67	0.66
DDT-L/2 (20En4De)	**7.98**	5.50	**128.1**	**0.68**	**0.67**
SiT-XL/2 [31]	17.2	5.07	76.52	0.65	0.63
REPA-XL/2 [51]	7.9	5.06	122.6	0.70	0.65
Improved-SiT-XL/2	10.9	5.3	103.4	0.66	0.65
Improved-REPA-XL/2	8.14	5.34	124.9	0.68	0.67
DDT-XL/2 (22En6De)	**6.62**	**4.86**	**135.1**	**0.69**	**0.67**

Table 2. **Metrics of $400K$ training steps with different model sizes.** All results are reported without classifier-free guidance. gray means metrics are copied from the original paper, otherwise it is produced by our codebase. By default, our DDT models are built on improved baselines. DDT† means model built on naive baseline without architecture improvement and lognorm sampling, consistent to REPA. Our DDT models consistently outperformed their counterparts.

	TrainMem	TrainSpeed	InferMem	InferSpeed
Baseline-XL/2	29.3G	0.25s/it	13G	0.78s/image
DDT-XL/2	30.5G	0.26s/it	13G	0.84s/image

DDT models demonstrate exceptional efficiency, achieving convergence in approximately $\frac{1}{4}$ of the total epochs compared to REPA [51] and other diffusion transformer models. In order to maintain methodological consistency with REPA, we employed the classifier-free guidance with 2.0 in the interval $[0.3, 1]$, Our models delivered impressive results: DDT-L/2 achieved 1.64 FID, and DDT-XL/2 got 1.52 FID within just 80 epochs. By extending training to 256 epochs—still significantly more efficient than traditional 800-epoch approaches—our DDT-XL/2 established a new state-of-the-art benchmark of 1.31 FID on ImageNet 256×256, decisively outperforming previous diffusion transformer methodologies. To extend training to 400 epochs, our DDT-XL/2(22En6De) achieves 1.26 FID, nearly reaching the upper limit of SD-VAE-ft-EMA-f8d4, which has a 1.20 rFID on ImageNet256.

ImageNet 512×512**.** We provide the final metrics of DDT-XL/2 at Tab. 3. To validate the superiority of our DDT model, we take our DDT-XL/2 trained on ImageNet 256×256 under 256 epochs as the initialization, fine-tune out DDT-XL/2 on ImageNet 512×512 for $100K$ steps. We adopt the aforementioned interval guidance [25] and we achieved a remarkable state-of-the-art performance of 1.90 FID, decisively outperforming REPA by a significant 0.28

	ImageNet 512×512				
Model	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
BigGAN-deep [3]	8.43	8.13	177.90	0.88	0.29
StyleGAN-XL [39]	2.41	4.06	267.75	0.77	0.52
ADM-G [10]	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 [35]	3.04	5.02	240.82	0.84	0.54
SiT-XL/2 [31]	2.62	4.18	252.21	0.84	0.57
REPA-XL/2 [51]	2.08	4.19	274.6	0.83	0.58
FlowDCN-XL/2 [47]	2.44	4.53	252.8	0.84	0.54
DDT-XL/2 (500K)	**1.28**	4.22	**305.1**	0.80	**0.63**

Table 3. **Benchmarking class-conditional image generation on ImageNet 512×512.** Our DDT-XL/2(512×512) is fine-tuned from the same model trained on 256×256 resolution setting of 1.28M steps. We adopt the interval guidance with interval $[0.3, 1]$ and CFG of 3.0

performance margin. In Tab. 3, some metrics exhibit subtle degradation, we attribute this to potentially insufficient fine-tuning. When allocating more training iterations to DDT-XL/2, it achieves 1.28 FID at 500K steps with CFG3.0 within the time interval $[0.3, 1.0]$.

5.4. Acceleration by Encoder sharing

As illustrated in Fig. 5, there is a strong local consistency of the self-condition in our condition encoder. Even $\boldsymbol{z}_{t=0}$ has a strong similarity above 0.8 with $\boldsymbol{z}_{t=1}$. This consistency provides an opportunity to speed up inference by sharing the encoder between adjacent steps.

We employed the simple uniform encoder sharing strategy and the new novel statistics dynamic programming strategy. Specifically, for the uniform strategy, we only recalculate the self-condition $\boldsymbol{z}_t$ every K steps. For statistics dynamic programming, we solve the aforementioned minimal sum path on the similarity matrix by dynamic programming and recalculate $\boldsymbol{z}_t$ according to the solved strategy. As shown in Fig. 6, there is a significant inference speedup nearly without visual quality loss when K is smaller than 6. As shown in Tab. 4, the metrics loss is still marginal, while the inference speedup is significant. The novel statistics dynamic programming slightly outperformed the naive uniform strategy with less FID drop.

5.5. Ablations

We conduct ablation studies on ImageNet 256×256 with DDT-B/2 and DDT-L/2. For sampling, we take the Euler solver with 250 steps as the default choice without classifier-free guidance. For training, we train each model with 80 epochs(400k steps), and the batch size is set to 256.

Encoder-Decoder Ratio we systematically explored ratios ranging from $2:1$ to $5:1$ across different model sizes.

SharRatio	Acc	Φ	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
0.00	1.0×	Uniform	1.31	4.62	308.1	0.78	0.66
0.50	1.6×	Uniform	1.31	4.48	300.5	0.78	0.65
0.66	1.9×	Uniform	1.32	4.46	301.2	0.78	0.65
0.75	2.3×	Uniform	1.34	4.43	302.7	0.78	0.65
0.80	2.6×	Uniform	1.36	4.40	303.3	0.78	0.64
		StatisticDP	1.33	4.37	301.7	0.78	0.64
0.83	2.7×	Uniform	1.37	4.41	302.8	0.78	0.64
		StatisticDP	1.36	4.35	300.3	0.78	0.64
0.87	3.0×	Uniform	1.42	4.43	302.8	0.78	0.64
		StatisticDP	1.40	4.35	302.4	0.78	0.64

Table 4. **Statistic Dynamic Programming Inference Acceleration.** All results are reported with classifier-free guidance. Our StatisticDP outperforms uniform baseline with a moderate margin.

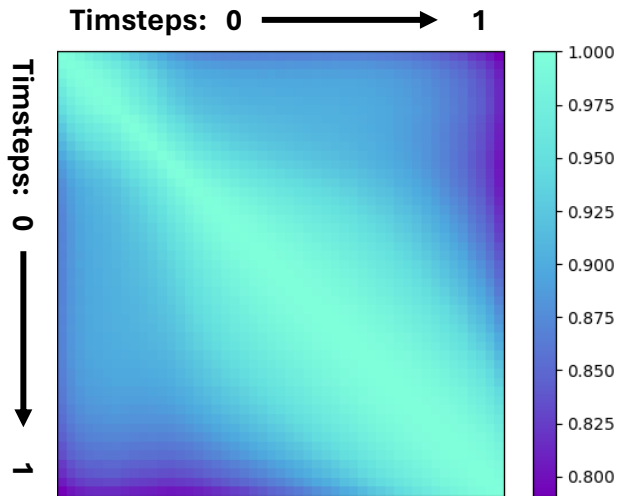


Figure 5. **The cosine similarity of self-condition feature $\boldsymbol{z}_t$ from encoder between different timesteps.** There is a strong correlation between adjacent steps, indicating the redundancy.

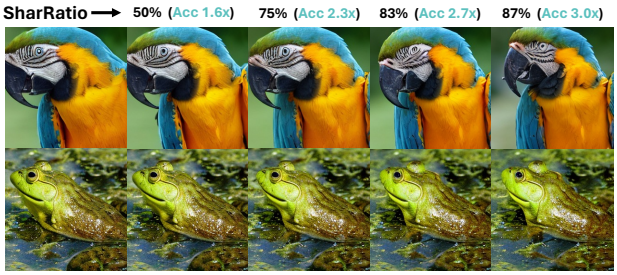


Figure 6. **Sharing the self-condition z_t in adjacent steps significant speedup the inference.**We tried various sharing frequency configurations. There is marginal visual quality down-gradation when the sharing frequency is reasonable.

in Fig. 7 and Fig. 8. Our notation mEnnDe represents models with m encoder layers and n decoder layers. The investigation experiments in Fig. 7 and Fig. 8 revealed critical

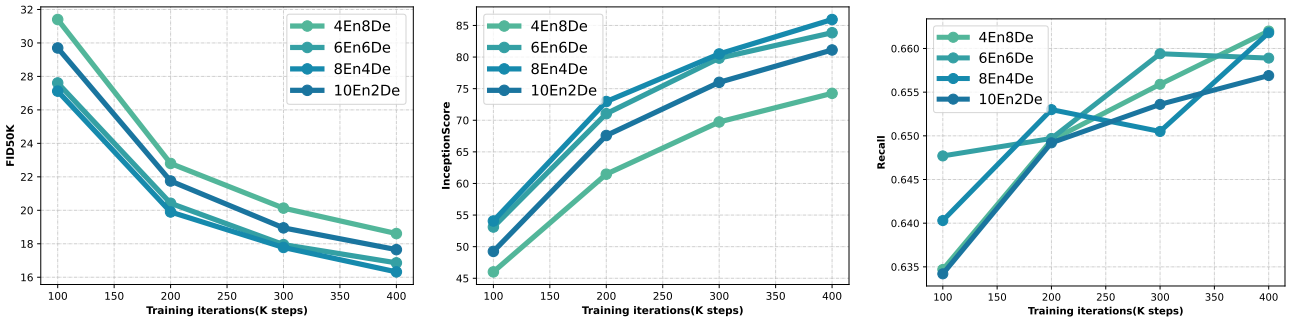


Figure 7. **The DDT-B/2 built upon Improved-baselines under various Encoder and Decoder layer ratio.** DDT-B/2(8En4De) achieves much faster convergence speed and better performance.

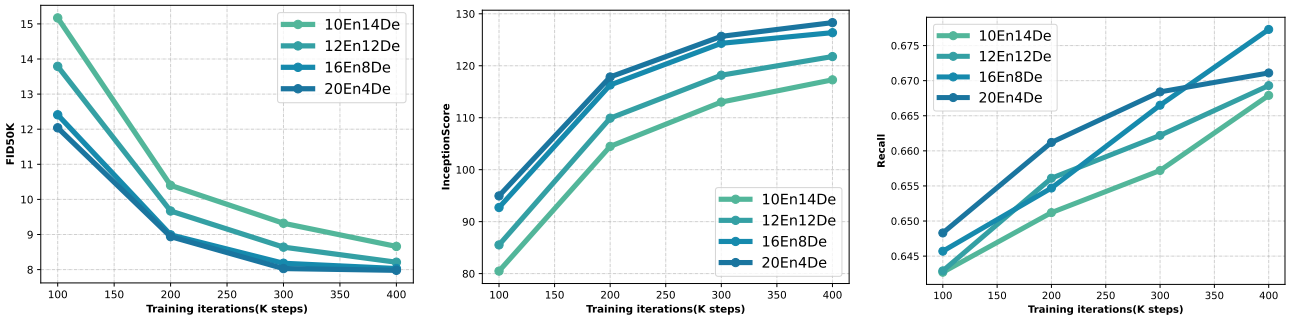


Figure 8. **The DDT-L/2 built upon Improved-baselines under various Encoder and Decoder layer ratio.** DDT-L/2 prefers an unexpected aggressive encoder-deocder ratio DDT-L/2(20En4De) achieves much faster convergence speed and better performance.

insights into architectural optimization. We observed that **a larger encoder is beneficial for further improving the performance as the model size increases**. For the Base model in Fig. 7, the optimal configuration emerged as 8 encoder layers and 4 decoder layers, delivering superior performance and convergence speed. Notably, the Large model in Fig. 8 exhibited a distinct preference, achieving peak performance with 20 encoder layers and 4 decoder layers, an unexpectedly aggressive encoder-decoder ratio. This unexpected discovery motivates us to scale the layer ratio in DDT-XL/2 to 22 encoder layers and 6 decoders to explore the performance upper limits of diffusion transformers.

Decoder Block types. In our investigation of decoder block types and their impact on high-frequency decoding performance, we systematically evaluated multiple architectural configurations. Our comprehensive assessment included alternative approaches such as simple 3×3 convolution blocks and naive MLP blocks. As shown in Tab. 5, the default (Attention with the MLP) setting achieves better results. Thanks to the encoder-decoder design, naive Conv blocks even achieve comparable results.

DecoderBlock	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Conv+MLP	16.96	7.33	85.1	0.62	0.65
MLP+MLP	24.13	7.89	65.0	0.57	0.65
Attn+MLP	16.32	6.63	86.0	0.62	0.66

Table 5. **Metrics of $400K$ training steps on DDT-B/2(8En4De) with different decoder blocks.** All results are reported without classifier-free guidance. The Default Attention + MLP configuration achieves best performance.

6. Conclusion

In this paper, we have introduced a novel Decoupled Diffusion Transformer, which rethinks the optimization dilemma of the traditional diffusion transformer by decoupling the low-frequency encoding and high-frequency decoding into dedicated components. Furthermore, we discovered that increasing the encoder capacity relative to the decoder yields increasingly beneficial results as the overall model scale grows. This insight provides valuable guidance for future model scaling efforts. Our experiments demonstrate that our DDT-XL/2 with an unexpected aggressive encoder-decoder layer ratio achieves great performance while requiring only 256 training epochs. This significant improvement in efficiency addresses one of the primary limitations of diffusion models: their lengthy training requirements. The decoupled architecture also presents opportunities for inference optimization through our proposed encoder result sharing mechanism. Our statistical dynamic programming approach for determining optimal sharing strategies enables faster inference while minimizing quality degradation, demonstrating that architectural innovations can yield benefits beyond their primary design objectives.

Acknowledgment. This work is supported by the National Key R&D Program of China (No. 2022ZD0160900), the Basic Research Program of Jiangsu (No. BK20250009), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 2

[2] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22669–22679, 2023. 2, 6

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1, 7

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[5] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022. 1

[6] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-\alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 2

[7] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-\sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024. 2

[8] Xiangxiang Chu, Jianlin Su, Bo Zhang, and Chunhua Shen. Visionllama: A unified llama interface for vision tasks. *arXiv preprint arXiv:2403.00522*, 2024. 5

[9] Chaorui Deng, Deyao Zh, Kunchang Li, Shi Guan, and Haoqi Fan. Causal diffusion transformers for generative modeling. *arXiv preprint arXiv:2412.12095*, 2024. 6

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2, 6, 7

[11] Sander Dieleman. Diffusion is spectral autoregression, 2024. 3

[12] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024. 2, 5

[13] Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, and Junshi Huang. Diffusion-rwkv: Scaling rwkv-like architectures for diffusion models. *arXiv preprint arXiv:2404.04478*, 2024. 2

[14] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 23164–23173, 2023. 3, 6

[15] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23164–23173, 2023. 3

[16] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7441–7451, 2023. 2, 6

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[20] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 2

[21] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022. 1

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 2

[23] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 2

[24] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019. 5

[25] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance

in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*, 2024. 6

[26] Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. *Advances in Neural Information Processing Systems*, 37:125441–125468, 2024. 3

[27] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2025. 2, 4, 6

[28] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 1, 3

[29] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 1, 3

[30] Zeyu Lu, Zidong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and Lei Bai. Fit: Flexible vision transformer for diffusion model. *arXiv preprint arXiv:2402.12376*, 2024. 5

[31] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024. 2, 3, 5, 6, 7

[32] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15762–15772, 2024. 5

[33] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021. 5

[34] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4

[35] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2, 4, 5, 6, 7

[36] Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. *arXiv preprint arXiv:2206.13397*, 2022. 3

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 6

[38] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 5

[39] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 1, 7

[40] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1

[41] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 5

[42] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 1

[43] Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. *arXiv preprint arXiv:2309.03350*, 2023. 6

[44] Yao Teng, Yue Wu, Han Shi, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Dim: Diffusion mamba for efficient high-resolution image synthesis. *arXiv preprint arXiv:2405.14224*, 2024. 2

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 5

[46] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 5

[47] Shuai Wang, Zexian Li, Tianhui Song, Xubin Li, Tiezheng Ge, Bo Zheng, and Limin Wang. Flowdcn: Exploring dcn-like architectures for fast image generation with arbitrary resolution. *arXiv preprint arXiv:2410.22655*, 2024. 2, 6, 7

[48] Jing Nathan Yan, Jiatao Gu, and Alexander M Rush. Diffusion models without attention. *arXiv preprint arXiv:2311.18257*, 2023. 2

[49] Jingfeng Yao and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025. 5, 6

[50] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. *arXiv preprint arXiv:2411.00776*, 2024. 1

[51] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024. 2, 3, 4, 5, 6, 7

[52] Xiaoyu Yue, Zidong Wang, Zeyu Lu, Shuyang Sun, Meng Wei, Wanli Ouyang, Lei Bai, and Luping Zhou. Diffusion models need visual priors for image generation. *arXiv preprint arXiv:2410.08531*, 2024. 2, 3

[53] Le Zhuo, Ruoyi Du, Han Xiao, Yangguang Li, Dongyang Liu, Rongjie Huang, Wenze Liu, Lirui Zhao, Fu-Yun Wang, Zhanyu Ma, et al. Lumina-next: Making lumina-t2x stronger and faster with next-dit. *arXiv preprint arXiv:2406.18583*, 2024. 2