CLEAR: Conv-Like Linearization Revs Pre-Trained Diffusion Transformers Up

Songhua Liu[†,‡], **Zhenxiong Tan**[‡], **and Xinchao Wang**[‡*]
[†]School of Artificial Intelligence, Shanghai Jiao Tong University, [‡]National University of Singapore
liusonghua@sjtu.edu.cn, zhenxiong@u.nus.edu, xinchao@nus.edu.sg

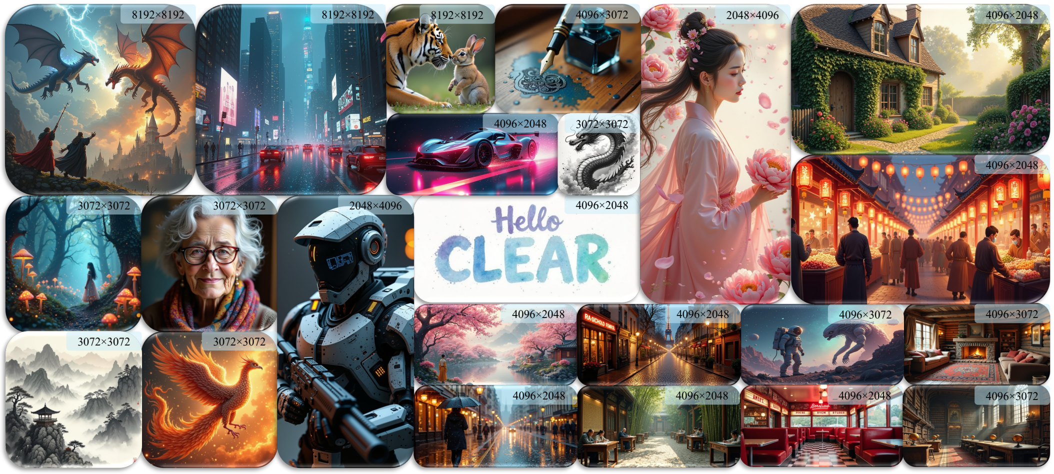


Figure 1: Ultra-resolution results generated by the linearized FLUX.1-dev model with our approach CLEAR. Resolution is marked on the top-right corner of each result in the format of `width`×`height`. Corresponding prompts can be found in the appendix.

Abstract

Diffusion Transformers (DiT) have become a leading architecture in image generation. However, the quadratic complexity of attention mechanisms, which are responsible for modeling token-wise relationships, results in significant latency when generating high-resolution images. To address this issue, we aim for a linear attention mechanism in this paper that reduces the complexity of pre-trained DiTs to linear. We begin our exploration with a comprehensive summary of existing efficient attention mechanisms and identify four key factors crucial for the successful linearization of pre-trained DiTs: locality, formulation consistency, high-rank attention maps, and feature integrity. Based on these insights, we introduce a convolution-like local attention strategy termed CLEAR, which limits feature interactions to a local window around each query token, and thus achieves linear complexity. Our experiments indicate that by fine-tuning the attention layer on merely 10K self-generated samples for 10K iterations, we can effectively transfer knowledge from a pre-trained DiT to a student model with linear complexity, yielding results comparable to those of the teacher model. Simultaneously, it reduces attention computations by 99.5% and accelerates generation by 6.3 times for generating 8K-resolution images. Furthermore, we investigate favorable properties in the distilled attention layers, such as zero-shot generalization across various models and plugins, as well as improved support for multi-GPU parallel inference. Models and codes are available here.

[*]Corresponding Author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Diffusion models [42, 14, 50, 26] have gained widespread attention in text-to-image generation, proving to be highly effective for producing high-quality and diverse images from textual prompts [10, 64]. Traditionally, architectures based on UNet [51, 50] have dominated this field due to their robust generative capabilities. In recent years, Diffusion Transformers (DiTs) [44, 1, 6, 17, 36, 18, 5] have emerged as a promising alternative, achieving leading performance in this field. Unlike the UNet-based architectures, DiTs leverage the attention mechanism [56] to model intricate token-wise relationships with remarkable flexibility, enabling them to capture nuanced dependencies across all tokens in images and texts, and thus produce visually rich and coherent outputs.

Despite their impressive performance, the attention layers—which model intricate pairwise token relationships with quadratic complexity—can introduce substantial latency in high-resolution image generation. As shown in Fig. 2, FLUX.1-dev [34], a state-of-the-art text-to-image DiT, requires over 30 minutes to generate 8K-resolution images with 20 denoising steps, even with hardware-aware optimizations like FlashAttention [12, 11].

Focusing on these drawbacks, we are curious about one question: *Is it possible to convert a pre-trained DiT to achieve linear complexity?* The answer is not straightforward, in fact, as it remains unclear whether existing efficient attention mechanisms—despite their recent widespread exploration [13, 31, 65, 49, 58, 52, 3, 68, 8, 67, 71, 28, 21]—can be effectively applied to pre-trained DiTs.

To answer this question, we initiate our exploration with a summary of previous methods dedicated to efficient attention, categorizing them into three main strategies: formulation variation, key-value compression, and key-value sampling. We then experiment with fine-tuning the model by replacing the original attention layers with these efficient alternatives. Results indicate that while formulation variation strategies have proven effective in attention-based UNets [39] and DiTs trained from scratch [62], they do not yield similar success with pre-trained DiTs. Key-value compression often leads to distorted details, and key-value sampling highlights the necessity of local tokens for each query to generate visually coherent results.

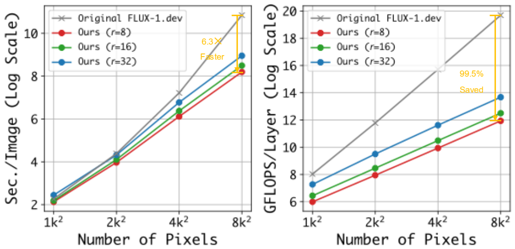


Figure 2: Comparison of speed and GFLOPS between the proposed linearized DiT and the original FLUX.1-dev. Speed is evaluated by performing 20 denoising steps on a single H100 GPU. FLOPS is calculated with the approximation: $4 \times \sum M \times c$, where c is the feature dimension and M denotes the attention masks. $\log_2$ is applied on both vertical axes for better visualization. The raw data are supplemented in the appendix.

Building on these observations, we figure out four elements crucial for for linearizing pre-trained DiTs, including locality, formulation consistency, high-rank attention maps, and feature integrity. Satisfying all these criteria, we present a convolution-like linearization strategy termed *CLEAR*, where each query interacts only with tokens within a predefined distance r. Since the number of key-value tokens interacting with each query is fixed, the resulting DiT achieves linear complexity with respect to image resolution.

To our surprise, such a concise design yields results comparable to original FLUX.1-dev after a knowledge distillation process [25] with merely 10K fine-tuning iterations on 10K self-generated samples. As shown in Fig. 1, CLEAR exhibits satisfactory cross-resolution generalizability, a property also reflected in UNet-based diffusion models [2, 15, 23, 29]. For ultra-high-resolution generation like 8K, it reduces attention computations by 99.5% and accelerates the original DiT by 6.3 times, as shown in Fig. 2. The distilled local attention layers are also compatible with different variants of the teacher model, *e.g.*, FLUX.1-dev and FLUX.1-schnell, and various pre-trained plugins like ControlNet [69] without requiring any adaptation.

As the token interactions are performed locally, it is convenient for CLEAR to support multi-GPU parallel inference. We further develop a patch-parallel paradigm that minimizes communication overhead. Our contribution can be summarized as follows:

- We provide a taxonomic overview of recent efficient attention mechanisms and identify four elements essential for linearizing pre-trained DiTs.

- Based on them, we propose a convolution-like local attention mechanism termed CLEAR as an alternative to default attention, which is the first linearization strategy tailored for pre-trained DiT to the best of our knowledge.
- We delve into multiple satisfactory properties of CLEAR through experiments, including its comparable performance with the original DiT, linear complexity, cross-resolution generalizability, cross-model/plugin generalizability, support for parallel inference, *etc.*

2 Efficient Attention: A Taxonomic Overview

The attention mechanism [56] is known for its flexibility in modeling token-wise relationships. It takes a query matrix $Q \in \mathbb{R}^{n\times c}$, a key matrix $K \in \mathbb{R}^{m\times c}$, and a value matrix $V \in \mathbb{R}^{m\times c'}$ as input and produces an output matrix $O \in \mathbb{R}^{n\times c'}$ via:

$$O = \text{softmax}\left(\frac{QK^\top}{\sqrt{c}}\right)V, \tag{1}$$

where n and m are the numbers of query and key-values tokens respectively, and c and c' are the feature dimensions for query-key and value tokens. In line with standard design conventions, we assume $c = c'$ throughout this paper, and in the case of self-attention, Q, K, and V come from the same feature maps with $m = n$.

As shown in Eq. 1, self-attention involves constructing $n \times n$ attention maps to model pair-wise token-to-token relationships, which results in both time and memory complexity. To address this issue, numerous studies focus on developing efficient attention mechanisms. In this section, we summarize recent work in this area and assess its applicability to DiT linearization. Specifically, we categorize existing approaches into three main categories: formulation variation, key-value compression, and key-value sampling.

2.1 Formulation Variation

Revisiting Eq. 1, if the `softmax` operation is omitted, we can first compute $K^\top V$, yielding a $c \times c$ matrix with linear time in relation to n. In this way, a series of linear attention mechanisms apply kernel functions $f(\cdot)$ and $g(\cdot)$ to Q and K respectively to mimic the effect of `softmax`:

$$O = f(Q)g(K)^\top V, \tag{2}$$

such as Mamba2 [13], Gated Linear Attention [65], and Generalized Linear Attention [39]. Another mainstream of methods try to replace the `softmax` operation with efficient alternatives, like `sigmoid` [49], relu^2 [28, 67], and Nystrom-based approximation [63].

2.2 Key-Value Compression

In the default setting of self-attention, the numbers of query and key-value tokens are consistent, *i.e.*, $m = n$, and the shape of the attention map would be $n \times n$. It is thus promising to compress key-value tokens so that m can be smaller than n to reduce the complexity. Following this routine, PixArt-Sigma [5] compress KV tokens locally with a downsampling `Conv2d` operator. Agent Attention [21] first conducts attention with downsampled Q and full-sized K and V to select agent KV tokens for compression. Then, original Q would interact with these compressed tokens. Similarly, Slot Attention [71] adopts learnable slots to obtain agent KV. Linformer [58] introduces learnable maps to obtain compressed tokens from the original ones.

2.3 Key-Value Sampling

Efficient attention based on key-value sampling is based on the assumption that not all key-value tokens are important for a query and the attention matrix is highly sparse. Comparing with key-value compression, it prunes key-value tokens for each query instead of producing new tokens. For instance, Strided Attention [7] samples one key-value token at a regular interval. Routing Attention [52] samples key-value tokens based on grouping. Swin Transformer [40] divides feature maps into non-overlapping local windows and performs attention independently for each window. Neighborhood

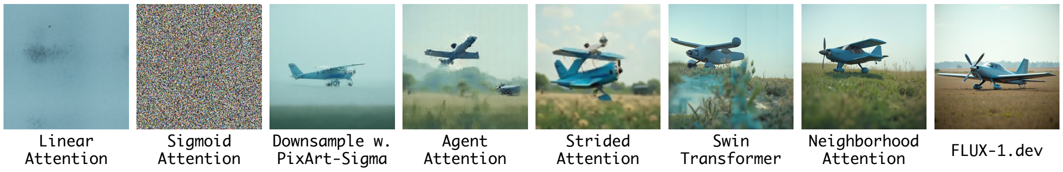


Figure 3: Preliminary results of various efficient attention methods on FLUX-1.dev. The prompt is “`A small blue plane sitting on top of a field`".

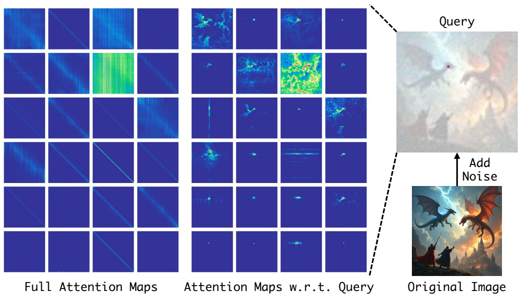


Figure 4: Visualization of attention maps by various heads. Attention in pre-trained DiTs is largely conducted in a local fashion.

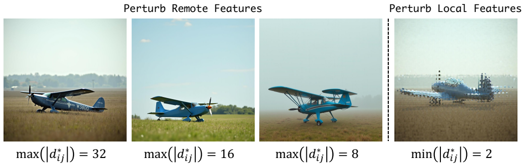


Figure 5: We try perturbing remote and local features respectively through clipping the relative distances required for rotary position embedding. Perturbing remote features has no obvious impact on image quality, whereas altering local ones results in significant distortion. The text prompt and the original result are consistent with Fig. 3.

Attention [22] selects key-value tokens within a local window around each query. BigBird [68] uses a selection strategy combining neighborhood attention and random attention, and LongFormer [3] combines neighborhood attention with some global tokens that are visible to all tokens.

3 Methods

3.1 What are Crucial for Linearizing DiTs?

Building on the overview of recent efficient attention mechanisms in Sec. 2, we explore a key question here: *What specific features are essential for successfully linearizing pre-trained DiTs*? We thus try substituting all the attention layers in FLUX.1-dev with various efficient alternatives and fine-tuning parameters in these layers. The preliminary text-to-image results are shown in Fig. 3, through which we figure out four key elements: locality, formulation consistency, high-rank attention maps, and feature integrity. According to these perspectives, we summarize some previous efficient attention methods in Tab. 1.

Locality indicates that key-value tokens fallen in the neighborhood of a query are included for attention. From Fig. 3, we observe that many methods equipped with this feature yield at least plausible results, like PixArt-Sigma, Swin Transformer, and Neighborhood Attention. Particularly, comparing the results of Neighborhood Attention and Strided Attention, we find that incorporating local key-value tokens diminishes a lot of distorted patterns.

Method	Locality	Formulation Consistency	High-Rank Attention Maps	Feature Integrity
Linear Attention [13, 39, 65, 31]	Yes	No	No	Yes
Sigmoid Attention [49]	Yes	No	Yes	Yes
PixArt-Sigma [5]	Yes	Yes	Yes	No
Agent Attention [21]	Maybe	Yes	Yes	No
Strided Attention [7]	No	Yes	Yes	Yes
Swin Transformer [40]	Yes	Yes	No	Yes
Neighborhood Attention [22]	Yes	Yes	Yes	Yes

Table 1: Summary of existing efficient attention mechanisms based on the four factors crucial for linearizing DiTs.

The reason for these phenomena is that pre-trained DiTs, such as FLUX, rely heavily on local features to manage token relationships. To validate this, we visualize attention maps in Fig. 4 and observe that the most significant attention scores fall in the local area around each query.

In Fig. 5, we provide further evidence to illustrate the importance of local features, that perturbing remote features would not damage the quality of FLUX.1-dev much. Specifically, FLUX.1-dev relies on rotary position embedding [55] to perceive spatial relationships and is sensitive to the relative distance $(d_{ij}^{(x)}, d_{ij}^{(y)})$ on the two axes of a 2D feature map, where indices i and j denote query and key token indices respectively. We perturb remote features by clipping the relative distances for rotary position embedding to a maximum value r when they exceed this threshold, *i.e.*, $d_{ij}^{(*)'} = d_{ij}^{(*)}.\text{clip}(-r, r)$. As shown in Fig. 5(left), the results are reasonable for a 64×64 feature map when r is as small as 8. Conversely, if we perturb local features by setting their minimum absolute distances to r, even with r as small as 2, the result still collapses as shown in Fig. 5(right)—emphasizing the importance of locality.

Formulation Consistency denotes that the efficient attention still applies the `softmax`-based formulation of the scaled dot-product attention. LinFusion [39] has shown that linear attention approaches like linear attention achieve promising results in attention-based UNets. However, we find that it is not the case for pre-trained DiTs, as shown in Fig. 3. We speculate that it is due to attention layers being the only modules for token interactions in DiTs, unlike the case in U-Nets. Substituting all of them would have a substantial impact on the final outputs. Other formulations like Sigmoid Attention fail to converge within a limited number of iterations, unable to mitigate the divergence between the original and modified formulations. It is thus beneficial to maintain consistency with the original attention function.

High-Rank Attention Maps means that attention maps calculated by efficient attention alternatives should be sufficient to capture the intricate token-wise relationships. As visualized in Fig. 4, extensive attention scores are concentrated along the diagonal, indicating that the attention maps do not exhibit the low-rank property assumed by many prior works. That is why methods like linear attention and Swin Transformer largely produce blocky patterns.

Feature Integrity implies that raw query, key, and value features are more favorable than the compressed ones. Although PixArt-Sigma has demonstrated that applying KV compression on deep layers would not hurt the performance much, this approach is not suitable for completely linearizing pre-trained DiTs. As shown in Fig. 3, methods based on KV compression, such as PixArt-Sigma and Agent Attention, tend to produce distorted textures compared to the results from Swin Transformer and Neighborhood Attention, which highlights the necessity to preserve the integrity of the raw query, key, and value tokens.

3.2 Conv-Like Linearization

Given the above analysis of the crucial factors for linearizing DiTs, Neighborhood Attention is the only scheme satisfying all 4 constraints. Motivated by this, we propose CLEAR, a conv-like linearization strategy tailored for pre-trained DiTs. Specifically, given that state-of-the-art text-to-image DiTs, like FLUX and StableDiffusion 3 series, typically adopt text-image joint self-attention, in CLEAR, for each text query, it still gathers features from all text and image key-value tokens; while for each image query, it interacts with all text tokens and local key-value tokens fallen in a local neighborhood around it. Since the number of text tokens and the local neighborhood size remains constant as resolution increases, the overall complexity scales linearly with the number of image tokens.

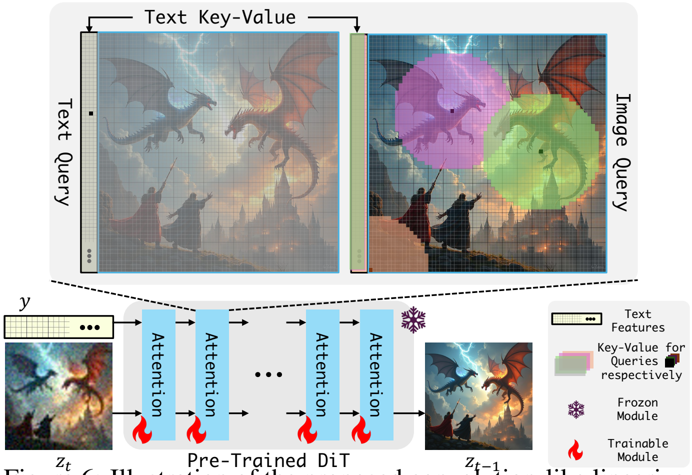


Figure 6: Illustration of the proposed convolution-like linearization strategy for pre-trained DiTs. In each text-image joint attention module, text queries aggregate information from all text and image tokens, while each image token gathers information only from tokens within a local circular window.

Unlike Neighborhood Attention and standard 2D convolution, which use a square sliding window, CLEAR employs circular windows, selecting key-value tokens within a predefined Euclidean radius r for each query. Compared with corresponding square windows, the computation overhead introduced by this design is $\sim \frac{\pi}{4}$ times. Formally, the attention mask M is constructed as follows:

$$M_{ij} = \begin{cases} 1, & \text{if } i \leq n_{text} \text{ or } j \leq n_{text} \text{ or } d_{ij}^{(x)2} + d_{ij}^{(y)2} < r^2; \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

Method/Setting	Against Original FID (↓)	 LPIPS (↓)	 CLIP-I (↑)	 DINO (↑)	Against Real FID (↓)	 LPIPS (↓)	CLIP-T (↑)	IS (↑)	GFLOPS (↓)
Original FLUX-1.dev	-	-	-	-	34.93	0.81	31.06	38.25	260.9
Sigmoid Attention [49]	447.80	0.91	41.34	0.25	457.69	0.84	17.53	1.15	260.9
Linear Attention [13, 39, 65, 31]	324.54	0.85	51.37	2.17	325.58	0.87	19.16	2.91	174.0
PixArt-Simga [5]	30.64	0.56	86.43	71.45	33.38	0.88	31.12	32.14	67.7
Agent Attention [21]	69.85	0.65	78.18	56.09	54.31	0.87	30.38	21.03	80.5
Strided Attention [7]	24.88	0.61	85.50	70.72	35.27	0.89	30.62	32.05	67.7
Swin Transformer [40]	18.90	0.65	85.72	73.43	32.20	0.87	30.64	34.68	67.7
CLEAR ($r = 8$)	15.53	0.64	86.47	74.36	32.06	0.83	30.69	34.47	63.5
w. distill	13.07	0.62	88.56	77.66	33.06	0.82	30.82	35.92	63.5
CLEAR ($r = 16$)	14.27	0.60	88.51	78.35	32.36	0.89	30.90	37.13	80.6
w. distill	13.72	0.58	88.53	77.30	33.63	0.88	30.65	37.84	80.6
CLEAR ($r = 32$)	11.07	0.52	89.92	81.20	33.47	0.82	30.96	37.80	154.1
w. distill	8.85	0.46	92.18	85.44	34.88	0.81	31.00	39.12	154.1

Table 2: Quantitative results of the original FLUX-1.dev, previous efficient attention methods, and CLEAR proposed in this paper with various r on 5,000 images from the COCO2014 validation dataset at a resolution of 1024×1024.

where n_{text} denotes the number of text tokens. Fig. 6 illustrates this paradigm.

3.3 Training and Optimization

Although each query only has access to tokens within a local window, stacking multiple Transformer blocks enables each token to gradually capture holistic information—similar to the way convolutional neural networks operate. To promote functional consistency between models before and after fine-tuning, we employ a knowledge distillation objective during the fine-tuning process. Specifically, the conventional flow matching loss [17, 38] is included:

$$\mathcal{L}_{fm} = \|(\epsilon - z_0) - \epsilon_\theta(z_t, t, y)\|_2^2, \tag{4}$$

where z_0 is denotes the feature of an image x encoded with a pre-trained VAE encoder $\mathcal{E}(\cdot)$ while z_t is its noisy version at the t-th timestep, y is the text condition, and $\epsilon_\theta(\cdot)$ is the DiT backbone for denoising with parameters θ. Beyond that, we encourage consistency between the linearized student model and the original teacher model, in terms of predictions and attention outputs:

$$\begin{aligned} \mathcal{L}_{pred} &= \|\epsilon_\theta(z_t, t, y) - \epsilon_{\theta_{org}}(z_t, t, y)\|_2^2, \\ \mathcal{L}_{attn} &= \frac{1}{L}\sum_{l=1}^{L} \|\epsilon_\theta^{(l)}(z_t, t, y) - \epsilon_{\theta_{org}}^{(l)}(z_t, t, y)\|_2^2, \end{aligned} \tag{5}$$

where θ_{org} denotes parameters of the original teacher DiT, L is the number of attention layers applying the loss term, and the superscript $^{(l)}$ indicates the layer index. The training objectives can be written as:

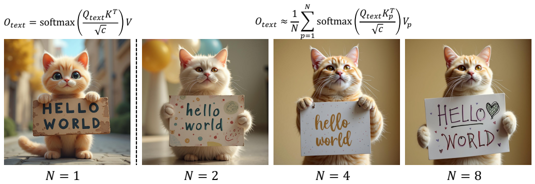
$$\min_\theta \mathbb{E}_{z \sim \mathcal{E}(x), y, \epsilon \sim \mathcal{N}(0,1), t}[\mathcal{L}_{fm} + \alpha \mathcal{L}_{pred} + \beta \mathcal{L}_{attn}], \tag{6}$$

where α and β are hyper-parameters controlling the weights of the corresponding loss terms. Only parameters in the attention layers are trainable.

For the training data, we find that training on samples generated by the original DiT model yields significantly better results than training on a real image dataset, even when the real dataset contains much more higher-quality data. Please refer to Sec. 4.3 for more discussions.

3.4 Multi-GPU Parallel Inference

Since attention is confined to a local window around each query, CLEAR offers greater efficiency for multi-GPU patch-wise parallel inference compared to the full attention in the original DiTs, which is particularly valuable for generating ultra-high-resolution images. Specifically, each GPU is responsible for processing an image patch, and the GPU communication is only required in the boundary areas. In other



$O_{text} = \text{softmax}\left(\frac{Q_{text}K^T}{\sqrt{c}}\right)V$
$O_{text} \approx \frac{1}{N}\sum_{p=1}^{N} \text{softmax}\left(\frac{Q_{text}K_p^T}{\sqrt{c}}\right)V_p$
HELLO WORLD
hello world
hello world
HELLO WORLD
$N = 1$
$N = 2$
$N = 4$
$N = 8$

Figure 7: To enhance multi-GPU parallel inference, each text query aggregates only the key-value tokens from the patch managed by its assigned GPU, then averages the attention results across all GPUs, which also generates high-quality images.

Setting	PSNR (↑)	SSIM (↑)	FID (↓)	LPIPS (↓)	CLIP-I (↑)	DINO (↑)	CLIP-T (↑)	IS (↑)	GFLOPS (↓)
–$1024 \times 1024 \to 2048 \times 2048$–									
FLUX-1.dev	-	-	-	-	-	-	31.11	24.53	3507.9
CLEAR ($r = 8$)	27.57	0.91	13.55	0.12	98.97	98.37	31.09	25.05	246.2
CLEAR ($r = 16$)	27.60	0.92	13.43	0.12	98.97	98.34	31.08	25.46	352.6
CLEAR ($r = 32$)	28.95	0.94	10.87	0.10	99.23	98.82	31.09	25.48	724.3
–$2048 \times 2048 \to 4096 \times 4096$–									
FLUX-1.dev	-	-	-	-	-	-	31.29	24.36	53604.4
CLEAR ($r = 8$)	26.19	0.87	20.87	0.22	98.02	96.56	31.16	25.87	979.3
CLEAR ($r = 16$)	26.98	0.88	16.20	0.19	98.48	97.64	31.25	25.13	1433.2
CLEAR ($r = 32$)	27.70	0.90	13.56	0.17	98.72	98.21	31.20	24.81	3141.7

Table 3: Quantitative results of the original FLUX-1.dev and our CLEAR with various r on 1,000 images from the COCO2014 validation dataset at resolutions of 2048×2048 and 4096×4096.

words, if we divide a $H \times W$ feature map into N patches along the vertical dimension, with each GPU handling a $\frac{H}{N} \times W$ patch, the communication cost for image tokens between each adjacent GPUs is $\mathcal{O}(r \times W)$ in CLEAR comparing with $\mathcal{O}(H \times W)$ in the original DiT.

Nevertheless, since each text token requires information from all image tokens, the exact attention computation in CLEAR still necessitates synchronization of all key-value tokens specially for text tokens, which compromises its potential in this regard. Fortunately, as shown in Fig. 7, we empirically find that without any training or adaptation, the original attention computation for text tokens can be effectively approximated by a patch-wise average while not hurting the performance too much, *i.e.*,

$$O_{text} \approx \frac{1}{N} \sum_{p=1}^{N} \text{softmax}\left(\frac{Q_{text} K_p^\top}{\sqrt{c}}\right) V_p, \tag{7}$$

where p is the patch/GPU index. Consequently, we only need to aggregate attention outputs for text tokens, resulting in a constant communication cost and eliminating the need to transmit all tokens.

Moreover, our pipeline is orthogonal to existing strategies for patch parallelism such as Distrifusion [35], which applies asynchronous computation and communication by using staled feature maps. Building CLEAR on top of these optimizations would lead to even greater acceleration.

4 Experiments

4.1 Settings and Implementation Details

In this paper, we primarily conduct experiments with the FLUX model series due to its state-of-the-art performance in text-to-image generation. Studies on more DiTs can be found in the appendix. We replace all the attention layers in FLUX-1.dev with the proposed CLEAR and experiment with three various window sizes with $r = 8$, $r = 16$, and $r = 32$. Leveraging FlexAttention in PyTorch [43], CLEAR, as a sparse attention mechanism, can be efficiently implemented on GPUs with low-level optimizations. We fine-tune parameters in attention layers on 10K samples with 1024×1024 resolution generated by FLUX-1.dev itself for 10K iterations under a total batch size 32 using the loss function defined in Eq. 6. $\mathcal{L}_{attn}$ is applied on `single_transformer_blocks` of FLUX, whose layer indices are $20 \sim 57$. Following previous works on architectural distillation for diffusion models [32, 39], both hyper-parameters α and β are set as 0.5. Other hyper-parameters, including schedulers, optimizers, *etc*, follow the default settings provided by Diffusers [57]. The training is conducted on 4 H100 GPUs supported by DeepSpeed ZeRO-2 [48], which takes ~ 1 day to finish. Unless otherwise specified, all inference is conducted on a single H100 GPU.

Following previous works [35, 39], we quantitatively study the proposed method on the validation set of COCO2014 [37] and randomly sample 5,000 images along with their prompts for evaluation. Since CLEAR aims to linearize a pre-trained DiT, we also benchmark our method against the results by the original DiT using consistent random seeds. Following conventions [53, 33, 60, 66], we consider FID [24], LPIPS [70], CLIP image similarity [47], and DINO image similarity [4] in this setting as metrics. For settings requiring pixel-wise alignment like image upsampling and ControlNet [69], we additionally incorporate PSNR [27] and multi-scale SSIM [59] for reference. While comparing with real images in COCO, we only include FID and LPIPS for distributional distances. Furthermore,

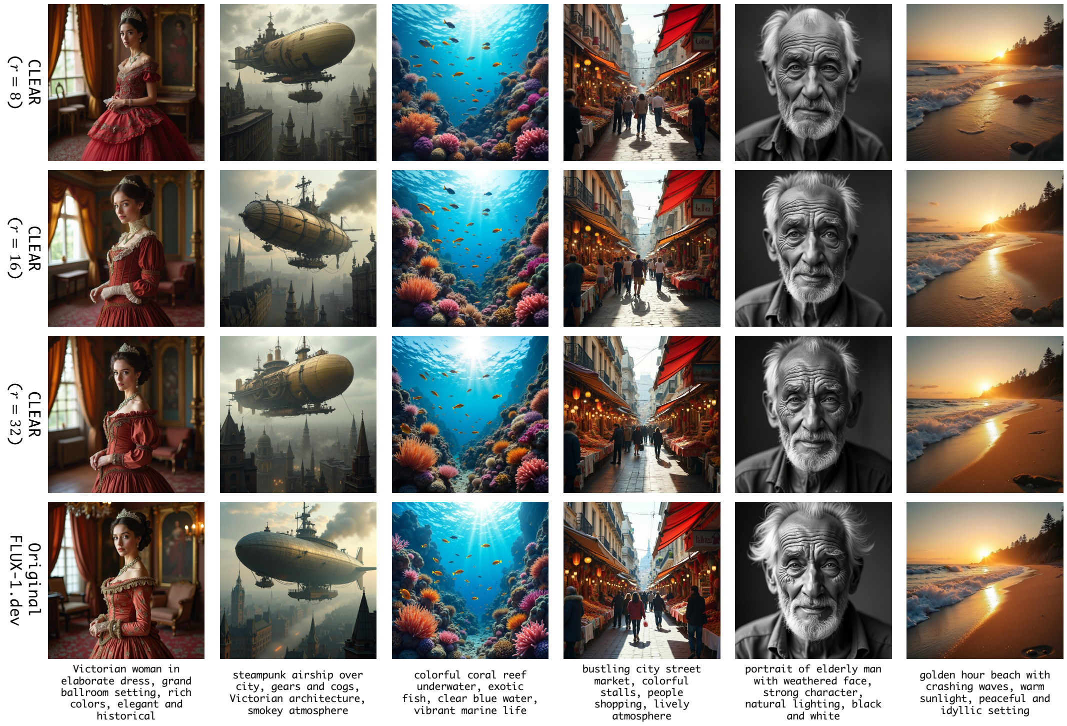

CLEAR ($r = 8$)
CLEAR ($r = 16$)
CLEAR ($r = 32$)
Original FLUX-1.dev
Victorian woman in elaborate dress, grand ballroom setting, rich colors, elegant and historical
steampunk airship over city, gears and cogs, Victorian architecture, smokey atmosphere
colorful coral reef underwater, exotic fish, clear blue water, vibrant marine life
bustling city street market, colorful stalls, people shopping, lively atmosphere
portrait of elderly man with weathered face, strong character, natural lighting, black and white
golden hour beach with crashing waves, warm sunlight, peaceful and idyllic setting

Figure 8: Qualitative results by the linearized FLUX-1.dev with CLEAR and the original model.

CLIP text similarity [47], Inception Score (IS) [54], and the number of floating point operations (FLOPS) are adopted to reflect textual alignment, general image quality, and computational burden, respectively. Text prompts for qualitative examples are generated by GPT-4o.

4.2 Main Comparisons

We aim to linearize a pre-trained DiT in this paper, and the linearized model is expected to perform comparably to the original one. As illustrated in Sec. 3.1, most efficient attention algorithms result in suboptimal performance for the target problem, as quantitatively supported by the evaluation in Tab. 2. In contrast, the proposed convolution-like linearization strategy achieves comparable or even superior performance to the original FLUX-1.dev while requiring less computation, underscoring its potential for effectively linearizing pre-trained DiTs. Please refer to the appendix for an analysis of training dynamics and convergence.

With the knowledge distillation loss terms defined in Eq. 5, the differences between the outputs of the linearized models and the original model are further minimized. More evaluations with the GenEval benchmark [19] and GPT scores can be found in the appendix. Qualitatively, as shown in Fig. 8, the linearized models produced by CLEAR yield comparable results.

For efficiency, CLEAR significantly reduces FLOPS compared to the original FLUX. However, due to the complexity of kernel implementation, as shown in Fig. 2, its practical speedup does not fully match the theoretical gains. Nevertheless, at 1K resolution, CLEAR with $r = 8$ and $r = 16$ still outperforms the original FLUX, with acceleration gains increasing as the resolution scales up.

4.3 Empirical Studies

In this part, we examine several noteworthy properties of CLEAR, including resolution extrapolation, zero-shot generalization across different models and plugins, multi-GPU parallel inference, and the effects of various training data.

Resolution Extrapolation. One of the key advantages of a linearized diffusion model is its ability to efficiently generate ultra-high-resolution images [39]. However, many previous studies have revealed that it is challenging for diffusion models to generate images beyond their native resolution during training. [15, 23, 2, 61, 16]. They thus apply a practical solution for generating high-resolution

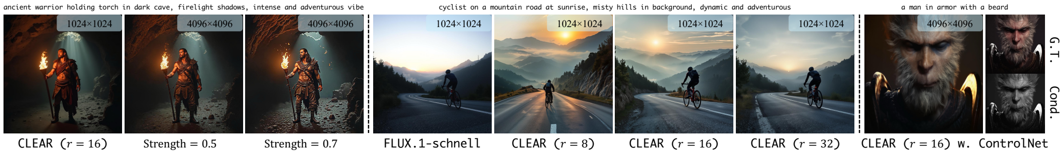

ancient warrior holding torch in dark cave, firelight shadows, intense and adventurous vibe
1024×1024
4096×4096
4096×4096
CLEAR (r = 16)
Strength = 0.5
Strength = 0.7
cyclist on a mountain road at sunrise, misty hills in background, dynamic and adventurous
1024×1024
1024×1024
1024×1024
1024×1024
FLUX.1-schnell
CLEAR (r = 8)
CLEAR (r = 16)
CLEAR (r = 32)
a man in armor with a beard
4096×4096
G.T.
Cond.
CLEAR (r = 16) w. ControlNet

Figure 9: Qualitative examples of using CLEAR with SDEdit [41] for high-resolution generation (left), FLUX-1.schnell in a zero-shot manner (middle), and ControlNet [69] (right). `G.T.` and `Cond.` denote ground-truth and condition images, separately.

Condition	Setting	Against Original						Against GT		CLIP-T (↑)	IS (↑)	RMSE (↓)
		PSNR (↑)	SSIM (↑)	FID (↓)	LPIPS (↓)	CLIP-I (↑)	DINO (↑)	FID (↓)	LPIPS (↓)			
Gray	FLUX-1.dev	-	-	-	-	-	-	40.25	0.32	30.16	22.22	0.039
	CLEAR ($r = 8$)	25.95	0.93	26.14	0.19	93.39	94.24	43.82	0.31	29.90	21.29	0.036
	CLEAR ($r = 16$)	28.24	0.95	16.86	0.13	96.00	96.73	40.45	0.31	30.19	22.34	0.040
	CLEAR ($r = 32$)	30.59	0.97	11.57	0.09	97.33	98.12	40.21	0.31	30.21	21.94	0.042
Tile	FLUX-1.dev	-	-	-	-	-	-	38.20	0.31	30.16	21.54	0.019
	CLEAR ($r = 16$)	30.12	0.97	9.1	0.13	99.25	99.04	39.73	0.34	30.11	21.77	0.021
Blur	FLUX-1.dev	-	-	-	-	-	-	38.72	0.31	30.20	21.42	0.028
	CLEAR ($r = 16$)	28.92	0.96	10.56	0.13	99.02	98.67	39.66	0.33	30.14	21.67	0.033

Table 4: Quantitative zero-shot generalization results of the proposed CLEAR to a pre-trained ControlNet with gray, tiled, and blur image conditions on 1,000 images from the COCO2014 validation dataset. `RMSE` here denotes the Root Mean Squared Error between the extracted conditions and the input conditions.

Setting	Against Original				Against Real		CLIP-T (↑)	IS (↑)	GFLOPS (↓)
	FID (↓)	LPIPS (↓)	CLIP-I (↑)	DINO (↑)	FID (↓)	LPIPS (↓)			
CLEAR ($r = 16$)	**13.72**	**0.58**	**88.53**	**77.30**	33.63	**0.88**	30.65	37.84	80.6
Square Neighborhood	13.77	**0.58**	88.47	76.53	**33.16**	**0.88**	**30.69**	**37.96**	92.1
Convolution	210.2	0.73	57.86	21.20	190.04	0.89	23.26	6.63	**78.9**

Table 5: Comparisons of circular neighborhood, square neighborhood, and standard attention.

images in a coarse-to-fine manner and devise adaptive strategies for components such as position embeddings and attention scales. The proposed CLEAR, on the other hand, makes architectural modifications to a pre-trained diffusion backbone, making it seamlessly applicable to them.

In this paper, we adopt SDEdit [41], a simple yet effective baseline adapting an image to a larger scale, for generating high-resolution images. In addition, we also enlarge the NTK factor of rotary position embeddings from 1 to 10 following [45], balance the entropy shift of attention using a log-scale attention factor [30], and disable the resolution-aware dynamic shifting [17] in the denoising scheduler. By adjusting the editing strength in SDEdit, as shown in Fig. 9(left), we can effectively control the trade-off between fine details and content preservation. In the appendix, we also try building CLEAR on top of various methods for resolution extrapolation.

Quantitatively, we measure the dependency between results by CLEAR with those by the original FLUX-1.dev. As shown in Tab. 3, we achieve MS-SSIM scores as high as 0.9, showcasing the effectiveness of the linearized model with CLEAR as an efficient alternative to the original FLUX.

Circular Neighborhood vs Square Neighborhood vs Standard Convolution. Compared with Neighborhood Attention [22], a noticeable difference in CLEAR is its circular neighborhood instead of a square. Computationally, the FLOPS of a circular neighborhood with radius r is $\sim \frac{\pi}{4}\times$ that of a square neighborhood with a side length $2r - 1$. We empirically find that they achieve comparable performance, as shown in Tab. 5. We also experiment with the standard convolution and find that it fails to generate visually plausible images, which highlights the importance of formulation consistency mentioned in Sec. 3.1.

Compatibility with DiT Plugins. It is favorable that substituting the original attention layers with linearized ones would not impact the functionality of plugins trained for the original DiT. As shown in Fig. 9(right), CLEAR exhibits this property by supporting the pre-trained ControlNet [69] using grayscale images as a condition for FLUX-1.dev. Quantitative results can be found in Tab. 4, including grayscale images, tiled images, and blur images as conditions, respectively.

Multi-GPU Parallel Inference. A linear complexity DiT can inherently support patch-wise multi-GPU parallel inference. In practice, for text-image joint attention used in modern DiT architectures [17, 34], we have to figure out a communication-efficient solution to handle text tokens, which requires gathering information from all key-value tokens. In Eq. 7, we propose an approximation

Setting	Against Original				Against Real		CLIP-T (↑)	IS (↑)
	FID (↓)	LPIPS (↓)	CLIP-I (↑)	DINO (↑)	FID (↓)	LPIPS (↓)		
CLEAR ($r = 16$)	-	-	-	-	33.63	0.88	30.65	37.84
$N = 2$	11.55	0.51	90.46	80.89	33.74	0.81	31.21	39.26
$N = 4$	12.78	0.54	89.74	79.99	33.07	0.81	31.27	40.01
$N = 8$	14.21	0.57	88.92	78.65	32.26	0.80	31.22	39.34

Table 6: Results of patch-wise multi-GPU parallel inference with various numbers of patches using the approximation in Eq. 7.

# of GPUs	Synchronous			Asynchronous		
	FLUX-1.dev	CLEAR ($r = 16$)	CLEAR ($r = 8$)	FLUX-1.dev	CLEAR ($r = 16$)	CLEAR ($r = 8$)
–1024 × 1024–						
1	11.13	11.40	11.00	-	-	-
2	$7.98_{\times 1.39}$	$8.52_{\times 1.34}$	$7.85_{\times 1.40}$	$7.64_{\times 1.46}$	$8.10_{\times 1.41}$	$7.50_{\times 1.47}$
4	$5.93_{\times 1.88}$	$6.01_{\times 1.90}$	$5.38_{\times 2.04}$	$5.64_{\times 1.97}$	$5.67_{\times 2.01}$	$5.11_{\times 2.15}$
8	$4.84_{\times 2.30}$	NA	$4.37_{\times 2.52}$	$4.49_{\times 2.48}$	NA	$3.90_{\times 2.82}$
–2048 × 2048–						
1	52.25	42.98	39.18	-	-	-
2	$30.96_{\times 1.69}$	$26.26_{\times 1.64}$	$23.96_{\times 1.64}$	$30.17_{\times 1.73}$	$25.41_{\times 1.69}$	$23.01_{\times 1.70}$
4	$18.94_{\times 2.76}$	$15.64_{\times 2.75}$	$13.86_{\times 2.83}$	$18.58_{\times 2.81}$	$15.12_{\times 2.84}$	$13.4_{\times 2.92}$
8	$12.97_{\times 4.03}$	$9.72_{\times 4.42}$	$8.40_{\times 4.66}$	$12.57_{\times 4.16}$	$9.30_{\times 4.62}$	$8.04_{\times 4.87}$
–4096 × 4096–						
1	372.43	207.83	173.53	-	-	-
2	$200.16_{\times 1.86}$	$115.02_{\times 1.81}$	$96.65_{\times 1.80}$	OOM	$112.34_{\times 1.85}$	$91.84_{\times 1.89}$
4	$105.59_{\times 3.53}$	$59.65_{\times 3.48}$	$49.70_{\times 3.49}$	OOM	$57.42_{\times 3.62}$	$48.57_{\times 3.57}$
8	$59.18_{\times 6.29}$	$32.33_{\times 6.43}$	$26.88_{\times 6.46}$	OOM	$31.23_{\times 6.65}$	$26.26_{\times 6.61}$

Table 7: Efficiency of multi-GPU parallel inference measured by sec./50 denoising steps on a HGX H100 8-GPU server. We adapt Distrifusion [35] to FLUX-1.dev here for asynchronous communication. The ratios of acceleration are highlighted with red. Results of CLEAR with $r = 16$ at the 1024×1024 resolution are not available (NA) because the patch size processed by each GPU is smaller than the boundary size. OOM denotes encountering out-of-memory error.

based on patch-wise averaging and validate its effectiveness quantitatively in Tab. 6. Results indicate a high correlation in semantics before and after this approximation, demonstrating its practical effectiveness.

For CLEAR, since there are only token interactions in the boundary areas of the patch handled by each GPU and the approximation of feature aggregation for text tokens defined in Eq. 7, we achieve satisfactory efficiency on multi-GPU parallel inference. Furthermore, we can apply the asynchronous communication strategy in Distrifusion [35] to achieve even more significant acceleration. As shown in Tab. 7, the acceleration becomes more significant with the increase of image resolution, while the original DiT encounters an out-of-memory (OOM) error due to the necessity of caching all key-value tokens.

5 Conclusions

In this paper, we present CLEAR, a convolution-like local attention strategy that effectively linearizes the attention mechanism in pre-trained Diffusion Transformers (DiTs), making them significantly more efficient for high-resolution image generation. Specifically, we identified four key elements—locality, formulation consistency, high-rank attention maps, and feature integrity—that are essential for successful linearization in the context of pre-trained DiTs. CLEAR leverages these principles by restricting attention to a circular local field around each query, achieving linear complexity while retaining high-quality results comparable to the original model. Our experiments demonstrate that fine-tuning on merely 10K self-generated samples allows for efficient knowledge transfer to a student model, leading to a 99.5% reduction in attention computations and a 6.3× acceleration in 8K-resolution image generation. Moreover, CLEAR, once trained, supports zero-shot generalization across different models and plugins and improves multi-GPU parallel inference capabilities, offering broader applicability and scalability.

Acknowledgment

This project is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Award Number: MOE-T2EP20122-0006).

References

[1] 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] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. In *International Conference on Machine Learning*, pages 1737–1752. PMLR, 2023.

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

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

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

[7] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[8] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[9] Djork-Arné Clevert. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[10] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[11] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[12] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[13] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

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

[15] Ruoyi Du, Dongliang Chang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Demofusion: Democratising high-resolution image generation with no $$$. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6159–6168, 2024.

[16] Ruoyi Du, Dongyang Liu, Le Zhuo, Qin Qi, Hongsheng Li, Zhanyu Ma, and Peng Gao. I-max: Maximize the resolution potential of pre-trained rectified flow transformers with projected flow. *arXiv preprint arXiv:2410.07536*, 2024.

[17] 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. In *Forty-first International Conference on Machine Learning*, 2024.

[18] Peng Gao, Le Zhuo, Ziyi Lin, Chris Liu, Junsong Chen, Ruoyi Du, Enze Xie, Xu Luo, Longtian Qiu, Yuhang Zhang, et al. Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers. *arXiv preprint arXiv:2405.05945*, 2024.

[19] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

[20] Dongchen Han, Ziyi Wang, Zhuofan Xia, Yizeng Han, Yifan Pu, Chunjiang Ge, Jun Song, Shiji Song, Bo Zheng, and Gao Huang. Demystify mamba in vision: A linear attention perspective. *arXiv preprint arXiv:2405.16605*, 2024.

[21] Dongchen Han, Tianzhu Ye, Yizeng Han, Zhuofan Xia, Siyuan Pan, Pengfei Wan, Shiji Song, and Gao Huang. Agent attention: On the integration of softmax and linear attention. In *European Conference on Computer Vision*, pages 124–140. Springer, 2025.

[22] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[23] Yingqing He, Shaoshu Yang, Haoxin Chen, Xiaodong Cun, Menghan Xia, Yong Zhang, Xintao Wang, Ran He, Qifeng Chen, and Ying Shan. Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

[24] 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.

[25] Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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

[27] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010.

[28] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. Transformer quality in linear time. In *International conference on machine learning*, pages 9099–9117. PMLR, 2022.

[29] Linjiang Huang, Rongyao Fang, Aiping Zhang, Guanglu Song, Si Liu, Yu Liu, and Hongsheng Li. Fouriscale: A frequency perspective on training-free high-resolution image synthesis. *arXiv preprint arXiv:2403.12963*, 2024.

[30] Zhiyu Jin, Xuli Shen, Bin Li, and Xiangyang Xue. Training-free diffusion model adaptation for variable-sized text-to-image synthesis. *Advances in Neural Information Processing Systems*, 36:70847–70860, 2023.

[31] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[32] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: Architecturally compressed stable diffusion for efficient text-to-image generation. In *Workshop on Efficient Systems for Foundation Models@ ICML2023*, 2023.

[33] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023.

[34] Black Forest Labs. Flux: Official inference repository for flux.1 models, 2024. Accessed: 2024-11-12.

[35] Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Kai Li, and Song Han. Distrifusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7183–7193, 2024.

[36] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. *arXiv preprint arXiv:2405.08748*, 2024.

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

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

[39] Songhua Liu, Weihao Yu, Zhenxiong Tan, and Xinchao Wang. Linfusion: 1 gpu, 1 minute, 16k image. *arXiv preprint arXiv:2409.02097*, 2024.

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[41] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[42] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[44] 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.

[45] Bowen Peng and Jeffrey Quesnelle. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation, 2023.

[46] Hao Peng, Jungo Kasai, Nikolaos Pappas, Dani Yogatama, Zhaofeng Wu, Lingpeng Kong, Roy Schwartz, and Noah A Smith. Abc: Attention with bounded-memory control. *arXiv preprint arXiv:2110.02488*, 2021.

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[48] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

[49] Jason Ramapuram, Federico Danieli, Eeshan Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, et al. Theory, analysis, and best practices for sigmoid self-attention. *arXiv preprint arXiv:2409.04431*, 2024.

[50] 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.

[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[52] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

[53] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[54] 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.

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

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[57] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

[58] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[59] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[60] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. *arXiv preprint arXiv:2302.13848*, 2023.

[61] Haoning Wu, Shaocheng Shen, Qiang Hu, Xiaoyun Zhang, Ya Zhang, and Yanfeng Wang. Megafusion: Extend diffusion models towards higher-resolution image generation without further tuning. *arXiv preprint arXiv:2408.11001*, 2024.

[62] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Yujun Lin, Zhekai Zhang, Muyang Li, Yao Lu, and Song Han. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

[63] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14138–14148, 2021.

[64] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

[65] Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

[66] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. 2023.

[67] Haoran You, Yichao Fu, Zheng Wang, Amir Yazdanbakhsh, et al. When linear attention meets autoregressive decoding: Towards more effective and efficient linearized large language models. *arXiv preprint arXiv:2406.07368*, 2024.

[68] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

[69] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[70] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[71] Yu Zhang, Songlin Yang, Ruijie Zhu, Yue Zhang, Leyang Cui, Yiqiao Wang, Bolun Wang, Freda Shi, Bailin Wang, Wei Bi, et al. Gated slot attention for efficient linear-time sequence modeling. *arXiv preprint arXiv:2409.07146*, 2024.

A Details of Efficient Attention Alternatives

The vanilla scaled dot-product attention, although effective and flexible, introduces quadratic computational complexity. Many works have focused on its efficient alternatives. In Sec. 2, we provide a taxonomic overview of recent works and will supplement more details regarding the specific formulations and implementations here.

Linear Attention avoids the `softmax` operation in the vanilla attention, supporting computing $K^\top V$ first with the associative property of matrix-wise multiplication, and thus achieves linear complexity. Before that, non-negative kernel functions $f(\cdot)$ and $g(\cdot)$ are applied on Q and K respectively such that the similarity between each query-key pair is non-negative. Furthermore, the similarity score between each query-key pair is normalized by the sum of similarity scores of between this query and all key tokens separately, to mimic the functionalities of `softmax`. Following [31, 20, 39], we implement $f(\cdot)$ and $g(\cdot)$ by the elu function [9]. Formally, the operation for the i-th query can be written as[2]:

$$O_i = \frac{(\mathrm{elu}(Q_i)+1)(\mathrm{elu}(K)+1)^\top}{(\mathrm{elu}(Q_i)+1)\sum_{j=1}^{m}(\mathrm{elu}(K_j)+1)^\top}V. \tag{8}$$

Sigmoid Attention replaces the `softmax` with the formulation of `sigmoid`, which removes the need to compute the `softmax` normalization, and thus achieves acceleration:

$$O = \mathrm{sigmoid}(\frac{QK^\top}{\sqrt{c}} + b)V, \tag{9}$$

where b is a hyper-parameter. In this paper, we follow the official implementation of `FlashSigmoid` with hardware-aware optimization[3] when applying Sigmoid Attention to DiTs.

PixArt-Sigma achieves acceleration by spatially down-sampling the key-value token maps [5]. Following the official implementation[4], we use learnable group-wise $\mathrm{Conv}4\times4$ kernels with $\mathrm{stride} = 4$ and initialize the weights to $\frac{1}{16}$ so that it is equivalent to an average pooling operation at the beginning. Formally, it can be written as:

$$O = \mathrm{softmax}(\frac{Q\mathrm{Conv}_k(K)^\top}{\sqrt{c}})\mathrm{Conv}_v(V). \tag{10}$$

Although it has been demonstrated that such a strategy can work well at relatively deep layers of DiTs, the results are still unsatisfactory for a completely linearized DiT.

Agent Attention performs efficient attention operations via agent tokens A from a down-sampled query token map [21]:

$$A = \mathrm{softmax}(\frac{\mathrm{Down}(Q)K^\top}{\sqrt{c}})V. \tag{11}$$

The derived agent tokens A are then used as value tokens:

$$O = \mathrm{softmax}(\frac{Q\mathrm{Down}(Q)^\top}{\sqrt{c}})A. \tag{12}$$

Such operations can be viewed as an adaptive token down-sampling strategy.

Slot Attention implemented in this paper is adapted from [71, 46], which contain s key-value memory slots derived by adaptively aggregating key-value tokens:

$$\tilde{K} = \mathrm{softmax}(\frac{PX^\top}{\sqrt{c}})K, \quad \tilde{V} = \mathrm{softmax}(\frac{PX^\top}{\sqrt{c}})V, \tag{13}$$

where $P \in \mathbb{R}^{s\times c}$ is learnable and introduced for modeling the writing intensity of each input token to each memory slot. These slots are then used as alternatives to original key-value tokens for attention computation:

$$O = \mathrm{softmax}(\frac{Q\tilde{K}^\top}{\sqrt{c}})\tilde{V}. \tag{14}$$

[2] https://github.com/LeapLabTHU/MLLA

[3] https://github.com/apple/ml-sigmoid-attention

[4] https://github.com/PixArt-alpha/PixArt-sigma

Setting	**Running Time (Sec. / 20 Steps)**				**TFLOPS / Layer**			
	1024×1024	2048×2048	4096×4096	8192×8192	1024×1024	2048×2048	4096×4096	8192×8192
FLUX-1.dev	4.81	20.90	148.97	1842.48	0.26	3.51	53.60	847.73
CLEAR ($r = 8$)	4.38	15.67	69.41	293.50	0.06	0.25	0.98	3.92
CLEAR ($r = 16$)	4.56	17.19	83.13	360.83	0.09	0.35	1.43	5.79
CLEAR ($r = 32$)	5.45	19.95	109.57	496.22	0.15	0.72	3.14	13.09

Table 8: Raw data for Fig. 2 on efficiency comparisons.

This strategy presents a different fashion for adaptive key-value compression.

Strided Attention samples tokens at a regular interval [7]. As a sparse attention strategy, the attention mask of the l-th layer with a down-sampling ratio of $r \times r$ can be constructed in the following way:

$$M_{ij}^{(l)} = \begin{cases} 1, & \text{if } i \leq n_{text} \text{ or } j \leq n_{text} \text{ or} \\ & (d_{ij}^{(x)} \% r = r_x \text{ and } d_{ij}^{(y)} \% r = r_y); \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

where $r_x = l \% r$ and $r_y = l // r$ ensure that each token has a chance to be sampled as key-value tokens.

Swin Transformer adopts a sliding window partition strategy [40], where attention interactions are independently conducted for each window. Formally, the attention map can be constructed via:

$$M_{ij} = \begin{cases} 1, & \text{if } i \leq n_{text} \text{ or } j \leq n_{text} \text{ or} \\ & \text{ tokens } i \text{ and } j \text{ in the same window;} \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

We set the window size to 16 and apply a shift of 8 for windows in layers with odd indices, following the approach described in the original manuscript. The rank of the image-to-image attention mask corresponds to the number of windows, which poses challenges in achieving the high-rank requirement introduced in the main manuscript needed for linearizing DiTs. In other words, all tokens within the same window share the same set of key and value tokens. This results in many duplicate rows in M, significantly reducing its rank.

Neighborhood Attention performs like convolution: for each query, it only samples key-value tokens within a neighborhood sliding window:

$$M_{ij} = \begin{cases} 1, & \text{if } i \leq n_{text} \text{ or } j \leq n_{text} \\ & \text{or } (d_{ij}^{(x)} < r \text{ and } d_{ij}^{(y)} < r); \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

Comparing with Swin Transformer, each query token in neighborhood attention has a distinct key and value token set, and each row in M is linearly independent of each other, which ensures the high-rank property. The formulation used in CLEAR is developed based on such Neighborhood Attention. The difference is in the functions of checking whether a key-value token is in the neighborhood of a given query token, as shown in Eq. 3.

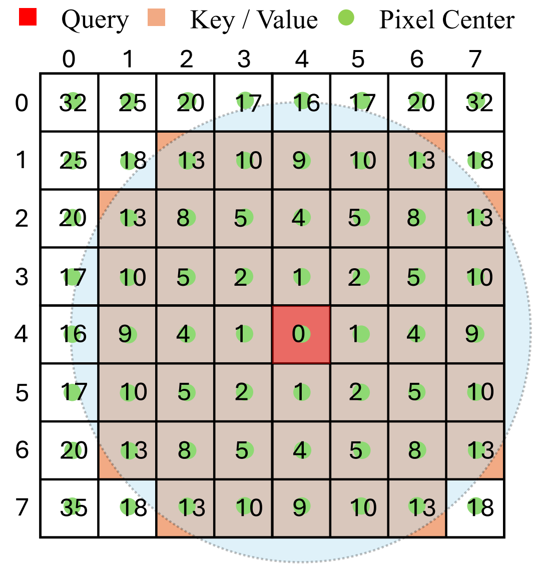
For CLEAR, we provide a further illustration of the $r = 4$ case in Fig. 10 for better clarity.

B Training Dynamics

We supplement the curves of training losses of various efficient attention alternatives in Fig. 11. The conclusion is consistent with the main manuscript, that strategies fulfilling the requirements of locality, formulation consistency, high-rank attention map, and feature integrity yield the most satisfactory training convergence.

C Raw Data for Efficiency Comparisons

We supplement raw data for Fig. 2 on efficiency comparisons in Tab. 8 for better clarity.



	0	1	2	3	4	5	6	7
0	32	25	20	17	16	17	20	32
1	25	18	13	10	9	10	13	18
2	20	13	8	5	4	5	8	13
3	17	10	5	2	1	2	5	10
4	16	9	4	1	0	1	4	9
5	17	10	5	2	1	2	5	10
6	20	13	8	5	4	5	8	13
7	35	18	13	10	9	10	13	18

Numbers: $(x_q - x_k)^2 + (y_q - y_k)^2$,
$(x_q, y_q) = (4,4)$, $r = 4$, $threshold = r^2 = 16$

Figure 10: The neighborhood region for an exemplar query in CLEAR when $r = 4$.

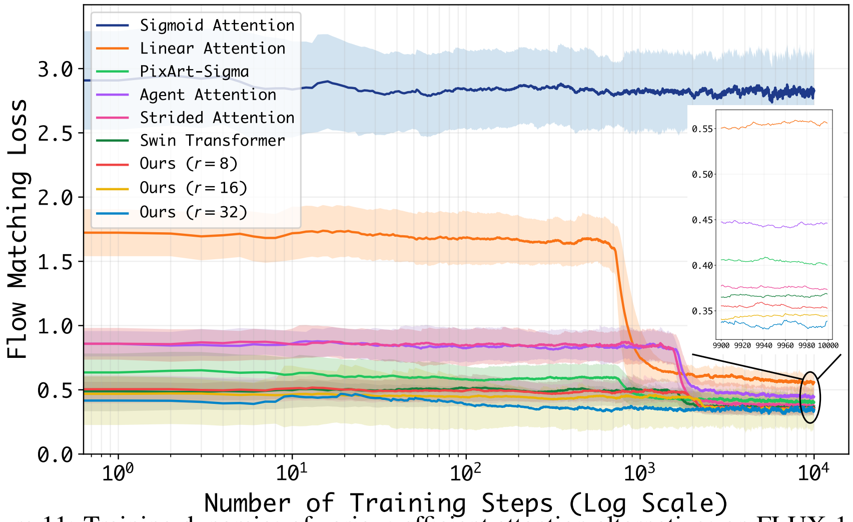


Figure 11: Training dynamics of various efficient attention alternatives on FLUX-1.dev.

Setting	Against Original				Against Real		CLIP-T (↑)	IS (↑)
	FID (↓)	LPIPS (↓)	CLIP-I (↑)	DINO (↑)	FID (↓)	LPIPS (↓)		
FLUX-1.dev	-	-	-	-	29.19	0.83	31.53	36.41
CLEAR ($r = 8$)	13.62	0.62	88.91	78.36	33.51	0.81	31.35	38.42
CLEAR ($r = 16$)	12.51	0.58	90.43	81.32	34.43	0.82	31.38	39.66
CLEAR ($r = 32$)	12.43	0.57	90.70	82.61	33.57	0.83	31.48	39.68

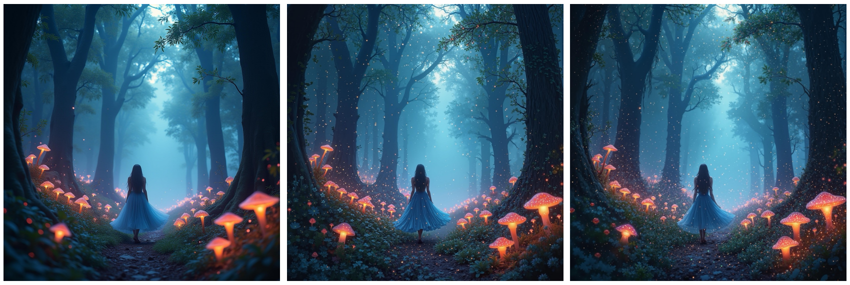
Table 9: Quantitative zero-shot generalization results to FLUX-1.schnell using CLEAR layers trained on FLUX-1.dev.

D Circular Neighborhood or Square Neighborhood

In fact, circular or square windows do not fundamentally affect the performance and the linearization properties. Our main conclusion of the manuscript is the four key factors for successful linearization of pre-trained DiTs. The standard neighborhood attention, which follows these four principles, is definitely effective, and so is the introduced method based on circular windows. They yield

Setting	Aesthetic			Prompt Alignment			Overall			Win Rate vs Other		
	1K	2K	4K	1K	2K	4K	1K	2K	4K	1K	2K	4K
FLUX-1.dev	**89.92**	87.70	89.88	91.88	90.90	91.52	**88.53**	87.46	88.38	0.57	0.54	0.60
CLEAR ($r = 16$)	89.62	**90.22**	**90.09**	**92.13**	**91.94**	**92.29**	88.52	**88.71**	**88.81**	-	-	-
SANA	82.42	86.86	87.63	84.68	90.68	90.57	84.32	86.96	86.71	0.83	0.57	0.62

Table 10: GPT scores compared with the original FLUX-1.dev model and SANA, a recent linear-complexity DiT trained from scratch.


1024×1024
by FLUX-1.dev

4096×4096
w. SDEdit

4096×4096
w. I-Max

Figure 12: The linearized DiTs by CLEAR are compatible with various pipelines dedicated to high-resolution inference. The prompt is shown in Fig. 17.

Setting	One Obj.	Two Obj.	Count	Color	Pos.	Color Attr.	Img Acc.	Text Acc.	Overall
FLUX-1.dev	98.44	81.82	**72.50**	77.93	21.75	41.75	64.10	79.57	0.657
CLEAR ($r = 16$)	99.06	83.84	69.69	**79.79**	**23.25**	48.75	**66.00**	81.19	**0.674**
CLEAR ($r = 8$)	**99.38**	78.03	48.75	76.60	16.50	41.00	58.52	76.13	0.600
+ Coarse Tokens	98.75	**84.85**	67.81	79.52	21.00	**52.00**	**66.00**	**81.74**	0.673

Table 12: Studies on the GenEval benchmark.

comparable performance as mentioned in Tab. 5. Given that the FLOPS can be reduced largely while preserving the performance, which is a free-lunch benefit, we apply the latter form. As shown in Tab. 11, it is indeed more efficient in terms of wall-clock time, especially at higher resolutions.

E Results on More DiTs

We additionally deploy our method on DiT models other than FLUX used in the main manuscript to demonstrate the universality of the proposed CLEAR. Here, we consider StableDiffusion3.5-Large[5] [17] (SD3.5-L), another state-of-the-art text-to-image generation DiT. We use the default setting of $r = 16$, which yields the best trade-off between quality and efficiency according to our experiments. Results on the COCO2014 validation dataset are shown in Tab. 16.

Time (s) / 20 Steps	$1K \times 1K$	$2K \times 2K$	$4K \times 4K$	$8K \times 8K$
Square ($r = 8$)	4.63	16.40	72.70	310.50
Circular ($r = 8$)	4.38	15.67	69.41	293.50
Square ($r = 16$)	4.84	18.77	87.09	382.07
Circular ($r = 16$)	4.56	17.19	83.13	360.83

Table 11: Comparisons on the running time required by circular and square attention windows at various resolution scales.

We also supplement more qualitative comparisons with results by the original FLUX-1.dev and SD3.5-L in Fig. 15. Results indicate an overall comparable performance. Due to the absence of explicit long-distance token interactions, our method may underperform in capturing overall structural properties, such as potential symmetry. Additionally involving more or less global tokens, such as down-sampled tokens as employed in PixArt-Sigma [5], could potentially mitigate this issue. However, as the primary objective of this paper is to highlight the significance of locality as a simple yet effective baseline, we leave detailed design explorations to future work.

[5] https://huggingface.co/stabilityai/stable-diffusion-3.5-large

	2K×2K				4K×4K			
Method	**Aesthetic**	**Prompt Align.**	**Overall**	**Win Rate**	**Aesthetic**	**Prompt Align.**	**Overall**	**Win Rate**
w/o SDEdit	86.32	89.08	85.37	0.87	84.15	87.22	82.36	0.92
w/o NTK	87.73	91.32	87.10	0.75	86.54	91.35	85.93	0.78
w Dynamic Shifting	88.44	91.68	86.59	0.70	84.02	91.40	84.29	0.73
Ours	**90.22**	**91.94**	**88.71**	–	**90.09**	**92.29**	**88.81**	–

Table 13: Quantitative comparison at 2K and 4K resolutions.

F GPT Evaluation

Following prior work [5, 16], we use GPT as a human-like evaluator to assess the quality of generated images across three aspects: aesthetics, prompt alignment, and overall quality. Here, we consider three candidates: the original FLUX-1.dev, its linearized version with CLEAR ($r = 16$), and SANA [62], a linear-complexity text-to-image DiT trained from scratch. We further instruct GPT-4o to compare CLEAR separately against the original FLUX-1.dev and SANA, reporting the win rates. Results across various resolutions are presented in Tab. 10, further demonstrating the effectiveness and superior performance of the proposed convolution-like linearization approach.

G More High-Resolution Results

In the main manuscript, we build our CLEAR on top of SDEdit [41], a simple yet effective strategy for image generation given a conditional image, for coarse-to-fine high-resolution generation. We demonstrate here that our method is also compatible with a variety of pipelines dedicated to resolution extrapolation. As shown in Fig. 12, we deploy CLEAR on I-Max [16], a concurrent work for training-free high-resolution generation with pre-trained DiTs, and observe that it may yield a more optimal balance between preserving low-resolution content and capturing high-resolution details. For instance, as shown in Fig. 12, I-Max preserves the textures of the dresses from the low-resolution result with minimal variation while effectively enhancing clear high-resolution details.

In fact, it requires two steps to achieve efficient high-resolution generation. The first is to adapt the original DiT to make it effective at higher scales. The second is to make it more efficient. **CLEAR mainly addresses the second step.** For the first step, the techniques are mainly adapted from previous works, *e.g.*, SDEdit and NTK rotary embedding. That is why we do not include so many details about them.

Nevertheless, we fully agree with the reviewer that it would be helpful to supplement the related details. We include some key information here and will further elaborate on it in the revision.

- SDEdit is a simple yet effective training-free method for image editing based solely on a pre-trained text-to-image diffusion model, which can be used for high-resolution generation in a coarse-to-fine manner. Specifically, we first generate an image at the native resolution scale of the diffusion model. Then, we resize it to a larger size and add a certain noise to it. A noise scale of 0.7 is adopted empirically in our experiments. Starting from this point, the model conducts the remaining denoising steps. In this way, the original image structures are preserved and low-level details are refined.
- NTK rotary embedding applies a scaling factor s to the rotary base b used for rotary positional embedding, *i.e.*, $b' = bs$, where $s = \sqrt{\frac{L_{cur}}{L_{native}}}$ and L_{cur} is the current sequence length while L_{native} is the native length seen in the training time.
- Resolution-aware dynamic shifting applies a factor s to the projection function of the current denoising time step t for the flow matching scheduler: $t' = \frac{st}{1+(s-1)t}$, where s is positively related to the image resolution. When this is enabled, at a high resolution, the projection function would appear too "skew", *i.e.*, the number of denoising steps allocated to the stage of high noise level is insufficient.

Regarding the three factors, we conduct the GPT evaluation following the same protocol in Tab. 10, and the results are shown in Tab. 13, highlighting their effectiveness.

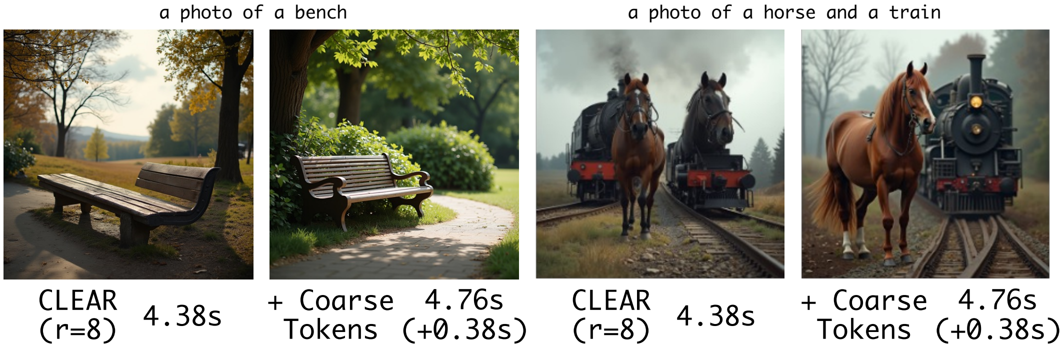

a photo of a bench
a photo of a horse and a train
CLEAR (r=8) 4.38s
+ Coarse Tokens 4.76s (+0.38s)
CLEAR (r=8) 4.38s
+ Coarse Tokens 4.76s (+0.38s)

Figure 13: Potential issues of small r. Adding down-sampled coarse key-value tokens can effectively address the problems.

FID (Against Original) / CLIP-T	$\alpha = 0$	$\alpha = 0.05$	$\alpha = 0.5$	$\alpha = 5$
$\beta = 0$	14.27 / 30.90	13.98 / 30.77	13.78 / 30.66	13.70 / 30.56
$\beta = 0.05$	13.90 / 30.81	13.88 / 30.81	13.81 / 30.72	13.68 / 30.64
$\beta = 0.5$	13.83 / 30.68	13.82 / 30.68	13.72 / 30.65	13.47 / 30.59
$\beta = 5$	13.77 / 30.65	13.69 / 30.66	13.45 / 30.62	13.44 / 30.58

Table 14: Grid search on α and β for FID and CLIP-T metrics.

H Cross-DiT Generalization

We empirically find that the trained CLEAR layers for one DiT are also applicable for others within the same series without any further adaptation efforts. For example, as shown in Fig. 9(middle), the CLEAR layers trained on FLUX-1.dev can be directly applied to inference on FLUX-1.schnell, a timestep-distilled DiT supporting 4-step inference, yielding results similar to those of the original FLUX-1.schnell. Such zero-shot generalization is quantitatively evaluated in Tab. 9.

I Hybrid Attention Extension

In practice, we observe that a small r, *e.g.*, $r = 8$, tends to produce results with inconsistent layouts and multi-objects, as shown in Fig. 13. As an extension, we propose a hybrid structure that incorporates coarse tokens as additional key-value tokens to enhance holistic interactions. Similar to [5], we conduct $8\times$ downsampling to the key-value maps to derive these global tokens. As shown in Fig. 13 and Tab. 12, we find that these issues can be effectively resolved with minimal additional computational overhead, underscoring the practical benefits. However, as this paper primarily emphasizes the importance of local tokens, we do not adopt this strategy by default.

J Studies on Loss Weights

In this part, we study the individual contributions of the two additional loss terms in Eq. 6 by varying the weights α and β. In general, both loss terms contribute to the consistency between the distilled and original models. Nevertheless, we find that they are complementary:

- α controls the weight of $\mathcal{L}_{pred}$, regulating the consistency between the final output results, which, according to our experiments shown in the grid search in Tab. 14, appears to be relatively more effective than $\mathcal{L}_{attn}$. For example, the cases of $\alpha > 0, \beta = 0$ generally yield better FiD scores than those of $\alpha = 0, \beta > 0$.
- β controls the weight of $\mathcal{L}_{attn}$, regulating the consistency between the intermediate attention results, which, according to our experiments, serves as an auxiliary measure to facilitate the training convergence. To verify this, we present the values of the flow matching loss $\mathcal{L}_{fm}$ after moving average at various steps in the training progress in Fig. 14. Results indicate that $\mathcal{L}_{attn}$ can facilitate training by offering informative intermediate supervision signals beyond the relatively late supervision provided by $\mathcal{L}_{pred}$ at the final outputs.

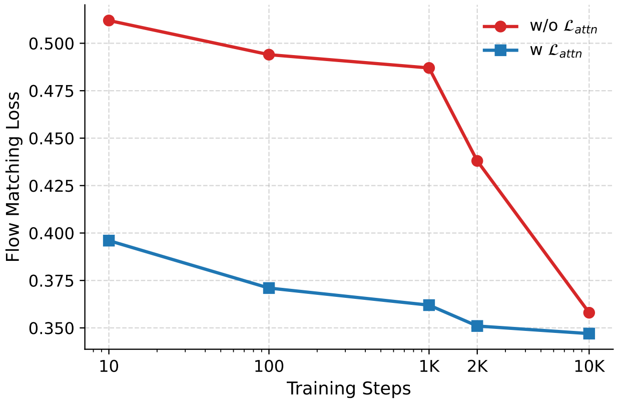


Figure 14: Effectiveness of $\mathcal{L}_{attn}$ on accelerating the training convergence.

	Aesthetic	Prompt Alignment	Overall
Train $r = 8$, Eval $r = 8$	89.19	88.38	87.96
Train $r = 8$, Eval $r = 16$	89.49	88.83	88.31
Train $r = 16$, Eval $r = 16$	89.62	92.13	88.52
Train $r = 16$, Eval $r = 8$	86.88	79.08	82.95

Table 15: Evaluation under different train and evaluation window radius (r).

Method/Setting	**Against Original**				**Against Real**		**CLIP-T** (↑)	**IS** (↑)	**GFLOPS** (↓)
	FID (↓)	**LPIPS** (↓)	**CLIP-I** (↑)	**DINO** (↑)	**FID** (↓)	**LPIPS** (↓)			
SD3.5-L	-	-	-	-	34.10	0.83	31.40	36.06	206.5
w. CLEAR ($r = 16$)	11.21	0.57	90.9	81.47	36.98	0.83	31.23	36.28	63.8

Table 16: Quantitative results of the original SD3-Large and its linearized version by CLEAR proposed in this paper on 5,000 images from the COCO2014 validation dataset at a resolution of 1024×1024.

Overall, applying both terms achieves the best performance, and the performance is insensitive to the specific values of the hyper-parameters according to the grid search. There can be some subtle tradeoffs between the FID and CLIP-T metrics, where the default values achieve a good balance.

K Generalizability across Different r

Qualitatively, we find that the model trained with a smaller window radius r can be used for a larger one. However, the opposite case does not work. We speculate that it is because the information provided for the model is complete for inference with additional clues when r is larger, while the information would be insufficient if r becomes smaller. We supplement the quantitative results of GPT evaluation in Tab. 15.

L Limitation

One limitation of our approach is that the practical acceleration achieved by CLEAR does not fully meet the theoretical expectations indicated by FLOPS. It becomes less significant at relatively low resolutions and can even be slower than the original DiT when the resolution is below 1024×1024. This drawback arises partially because hardware optimization for sparse attention is inherently more challenging than the optimizations achieved by FlashAttention for full attention computation. Addressing this limitation may require developing fused CUDA operators specifically optimized for the specific sparse pattern of CLEAR, which is a valuable direction for future works.

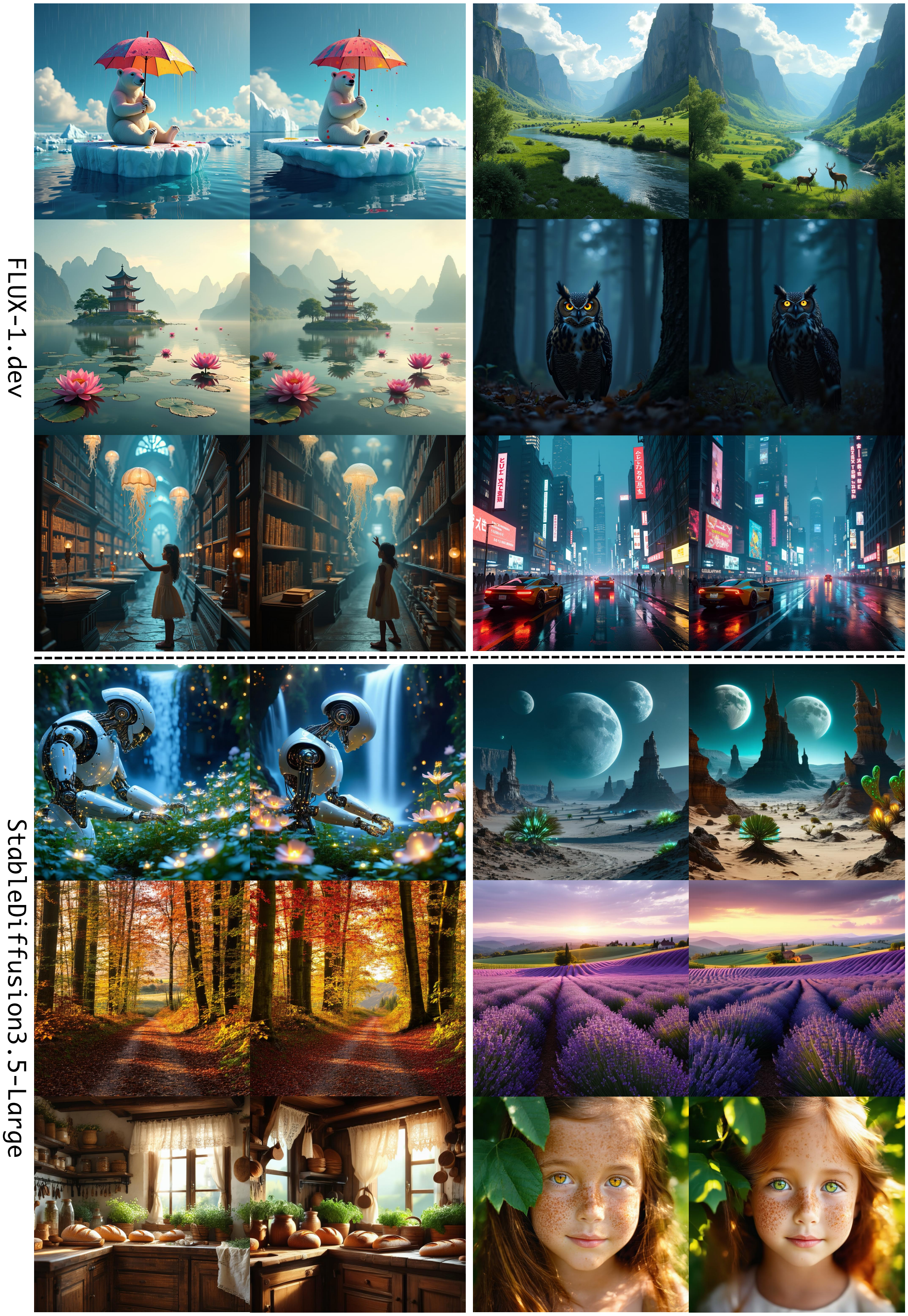


Figure 15: Qualitative comparisons on FLUX-1.dev (top) and SD3.5-Large (bottom). The left subplots are results by the original models while the right ones are by the CLEAR linearized models. Prompts are listed in Fig. 19.

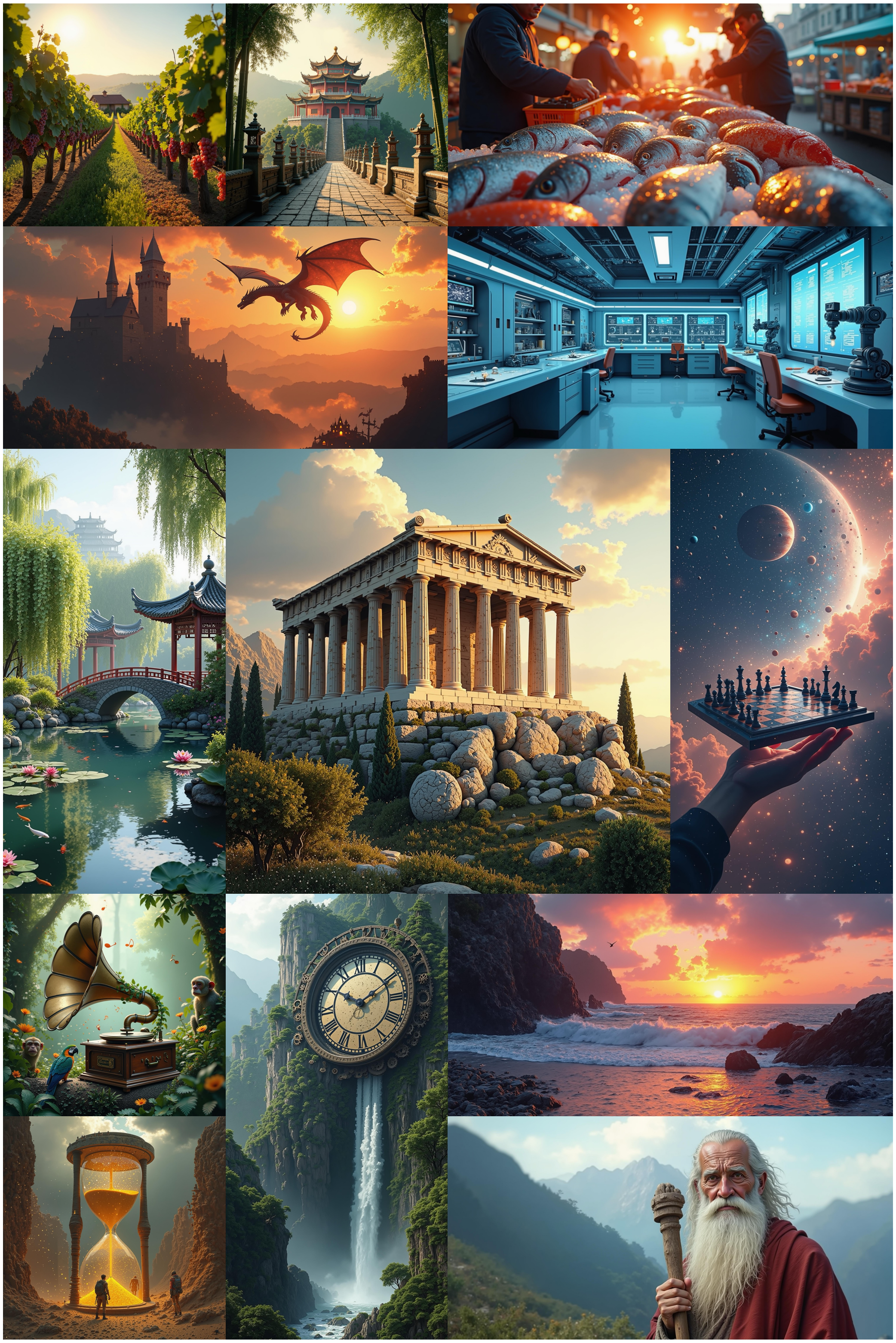


Figure 16: More 4K examples by the CLEAR linearized FLUX-1.dev. Prompts are listed in Fig. 19.

```
// Fig. 1, according to the top-left corner, from top to bottom, from left to right

// 1, also used in Fig. 4 and Fig. 6
"A high fantasy scene where a fierce battle is taking place in the sky between dragons and powerful
  wizards. One side of the scene shows wizards casting spells, their staffs glowing with magical
 energy, while on the other, dragons with scales of fire and lightning breathe torrents of flame.
 The sky is torn with storms of magic, and below, a medieval kingdom watches in awe as the skies
 blaze with the fury of battle.",
// 2
"futuristic cityscape, towering skyscrapers, neon lights, speeding cars, holographic advertisements
 , cyberpunk, ultra-realistic, high resolution, cinematic lighting, highly detailed, ultra HD, 8K,
 nighttime, rain-soaked streets, reflections on glass, vibrant colors, misty atmosphere",
// 3
"A tiger is kissing a rabbit",
// 4
"classic fountain pen with detailed engravings, glass ink bottle with reflections, subtle ink
 stains, warm lighting, rich wood desk, soft shadows, high detail on pen and bottle, ultra-
 realistic textures, vintage and refined, calm and artistic feel, close-up, high resolution, deep
 blue and golden accents",
// 5
"beautiful Chinese woman in hanfu, surrounded by blooming peonies, flowing silk robes, elegant and
 ethereal, soft lighting, pastel colors, highly detailed fabric textures, delicate hair ornaments,
 peony petals in the air, graceful pose, traditional hairpin",
// 6
"charming countryside cottage, early morning sunlight, mist in the air, lush garden, rustic and
 cozy, ivy-covered walls, wooden fence, high detail, ultra-realistic, peaceful atmosphere, blooming
  flowers, warm light, quiet and serene",
// 7
"futuristic racing car, sleek design, neon underglow, high-speed action, dust trail, dynamic motion
  blur, cinematic lighting, high resolution, ultra-realistic, ultra HD, 8K, dark background, neon
 lights, sparks flying, intense colors, reflections on car surface",
// 8
"majestic Chinese dragon, swirling clouds, water and ink effect, powerful presence, dynamic and
 dramatic, monochromatic ink wash, swirling motion, high detail on dragon scales, whirlwind of
 clouds, dragon's fierce eyes, ink splashes, ancient mystical aura",
// 9
"traditional Chinese night market, red lanterns, crowded stalls, vibrant atmosphere, warm and
 lively, golden lighting, realistic and bustling, intricate market details, traditional snacks,
 merchants in robes, lanterns casting glow, animated crowd in background",
// 10, also used in Fig. 2 (Appendix)
"enchanted forest, glowing plants, towering ancient trees, a mystical girl, magical aura, fantasy
 style, vibrant colors, ethereal lighting, bokeh effect, ultra-detailed, painterly, ultra HD, 8K,
 soft glowing lights, mist and fog, otherworldly ambiance, glowing mushrooms, sparkling particles",
```

Figure 17: GPT-generated prompts used in the main manuscript.

```
// 11
"portrait of an elderly female artist with silver hair, gentle smile, wearing glasses and colorful
 scarf, soft studio lighting, high detail wrinkles, ultra-realistic, warm lighting, creative and
 thoughtful, calm and wise, subtle background, rich textures, peaceful and inviting, close-up",
// 12
"futuristic soldier, robotic armor, high-tech weapon, visor with digital HUD, dark sci-fi, highly
 detailed, cinematic lighting, dynamic pose, ultra-realistic, ultra HD, 8K, neon accents, dark
 background, glowing HUD, intense expression, battle scars on armor",
// 13
"A watercolor-style sign reading 'Hello CLEAR' with soft gradients of blue, green, and purple,
 textured lettering, and subtle paint splashes",
// 14
"hidden paradise with peach blossoms, flowing river, distant mountains, quaint cottages, dreamlike
 and serene, vibrant colors, soft and warm lighting, idyllic landscape, blossoming peach trees,
 mist over river, villagers in traditional attire, sunlight filtering through petals",
// 15
"Parisian street at night, iconic street lights, cobblestone path, view of Eiffel Tower, vibrant
 city atmosphere, warm tones, rain reflections on street, historic architecture, romantic ambiance,
  ultra-realistic details, cinematic lighting, urban scene, high resolution",
// 16
"astronaut meeting alien creatures, cosmic background, colorful nebula, stars in background, high
 detail spacesuit, atmospheric lighting, sci-fi setting, calm and peaceful, otherworldly creatures,
  ultra-realistic, adventure in space, detailed environment",
// 17
"rustic wooden cabin interior, cozy and warm, fireplace glowing, wooden beams, vintage furniture,
 soft light from a window, warm and earthy tones, ultra-realistic details, rich textures, cozy
 blankets and cushions, peaceful ambiance, high resolution, natural wood grain visible",
// 18
"Chinese ink landscape painting, misty mountains, winding rivers, ancient pine trees, traditional
 ink wash painting, soft brushstrokes, monochromatic, ethereal and timeless, light mist among
 mountains, small thatched pavilion, subtle gradation of ink, natural flow",
// 19
"phoenix rising from flames, vibrant feathers, traditional Chinese mythological style, vivid and
 majestic, dynamic colors, dramatic lighting, intricate feather details, golden flames, radiant
 plumage, traditional patterns on wings, sense of rebirth",
// 20
"city street on a rainy day, wet pavement with reflections, people under umbrellas, soft city
 lights reflecting in water puddles, detailed raindrops, warm and cozy tones, misty atmosphere,
 ultra-realistic details, vibrant and deep colors, high contrast, peaceful rain ambiance, soft
 shadows, street lights glowing",
// 21
"ancient Chinese academy, surrounded by bamboo forest, stone paths, wooden study desks, calm and
 serene, warm lighting, natural greens, intricate woodwork, rustic textures, bamboo shadows on
 ground, calligraphy brushes, traditional scrolls, scholars in robes",
// 22
"1950s American diner, red leather booths, checkerboard floor, neon signs, nostalgic atmosphere,
 warm lighting, retro decor, vintage menu, chrome accents, classic style, cozy and inviting, high
 detail, ultra-realistic",
// 23
"ancient library, high shelves filled with old books, detailed wood carvings, dusty and dim
 lighting, massive wooden tables, vintage globes, warm light filtering through tall windows, ultra-
 realistic, intricate details on book spines, nostalgic atmosphere, high resolution, serene and
 historical feel",

// Fig. 7
"A cat holding a sign that says hello world"
```

Figure 18: GPT-generated prompts used in the main manuscript. (Cont.)

```
// Fig. 15, from top to bottom, from left to right

// 1
"a polar bear sitting on a floating iceberg, holding an umbrella while it rains colorful paint, the
  surrounding ocean reflecting the vibrant colors, ultra-detailed, photorealistic, ultra HD, 8K,
 surreal and artistic composition, bold contrasts, intricate reflections",
// 2
"lush green valley surrounded by towering cliffs, a winding river reflecting the blue sky, fluffy
 white clouds casting shadows, grazing deer in the distance, ultra-detailed, photorealistic, ultra
 HD, 8K, natural vibrancy, peaceful wilderness atmosphere, intricate water and vegetation textures"
 ,
// 3
"peaceful Chinese lake scene, a traditional pagoda on a small island, still water reflecting the
 structure, distant misty mountains, pink lotus flowers floating, warm morning light, ultra-
 detailed, photorealistic, ultra HD, 8K, serene atmosphere, traditional aesthetics, vibrant yet
 soft colors",
// 4
"owl in a dense forest at night, glowing yellow eyes, dark and mysterious atmosphere",
// 5
"a library where the books are glowing jellyfish floating mid-air, a young girl reaching out to
 touch one, shelves filled with ancient tomes, soft ambient lighting, ultra-detailed,
 photorealistic, ultra HD, 8K, whimsical and magical atmosphere, intricate textures",
// 6
"cyberpunk cityscape, glowing neon lights, futuristic skyscrapers, bustling streets, flying cars,
 nighttime setting, holographic advertisements, rain-soaked roads, ultra-detailed, cinematic
 lighting, ultra HD, 8K, vivid colors, dramatic atmosphere, intricate reflections, dystopian vibe",
// 7
"a futuristic robot tending a garden of glowing bioluminescent flowers, its metallic hands
 delicately handling the plants, a waterfall of stars in the background, ultra-detailed,
 photorealistic, ultra HD, 8K, ethereal lighting, blending nature and technology",
// 8
"alien desert landscape, multiple moons in the sky, strange rock formations, glowing plants,
 mysterious alien figures, science fiction style, ultra-detailed, cinematic lighting, ultra HD, 8K,
  vibrant colors, surreal ambiance, dramatic shadows, expansive vistas",
// 9
"autumn forest in golden hour, trees with vibrant red, orange, and yellow leaves, a narrow path
 covered in fallen foliage, sunlight casting warm hues, distant hills, ultra-detailed,
 photorealistic, ultra HD, 8K, rich colors, peaceful atmosphere, intricate details of leaves and
 bark",
// 10
"endless lavender fields at sunset, soft purple hues blending with golden sky, a small rustic
 farmhouse in the distance, rolling hills on the horizon, ultra-detailed, photorealistic, ultra HD,
  8K, delicate lavender flowers, serene ambiance, atmospheric depth",
// 11
"cozy rural kitchen, wooden cabinets, fresh bread on the counter, sunlight streaming through lace
 curtains, ceramic jars and fresh herbs, rustic charm, warm tones, ultra-detailed, photorealistic,
 ultra HD, 8K, soft ambient light, intricate wood grain textures, peaceful atmosphere",
// 12
"portrait of a young girl with freckles, natural outdoor setting, sunlight filtering through leaves
 , soft focus background, vibrant hair and vivid eye color, ultra-detailed, photorealistic, ultra
 HD, 8K, delicate facial textures, bright and innocent atmosphere, warm golden tones"
```

Figure 19: GPT-generated prompts used in the appendix.

```
// Fig. 16, according to the top-left corner, from top to bottom, from left to right

// 1
"sunlit vineyard in late summer, rows of grapevines heavy with ripe fruit, rustic farmhouse in the
 distance, soft hills and clear sky, warm golden light, ultra-detailed, photorealistic, ultra HD, 8
 K, intricate grape and leaf textures, serene countryside atmosphere",
// 2
"ancient Chinese temple on a hill, red walls and golden roofs, surrounded by lush green bamboo
 forest, stone lanterns lining the path, soft golden hour light, ultra-detailed, photorealistic,
 ultra HD, 8K, traditional Chinese architecture, peaceful ambiance, intricate carvings and ornate
 designs",
// 3
"bustling fish market at sunrise, vibrant colors of fresh seafood, fishermen unloading crates,
 intricate details of fish scales and ice, ambient light, bustling atmosphere, ultra-detailed,
 photorealistic, ultra HD, 8K, atmospheric realism, sharp textures, lively dynamics",
// 4
"majestic dragon flying over a medieval castle, fiery sunset, rolling hills, dramatic clouds,
 fantasy style, ultra-detailed, painterly aesthetic, ultra HD, 8K, warm hues, glowing embers,
 intricate textures, golden hour lighting",
// 5
"futuristic laboratory interior, glowing screens, robotic arms, holographic displays, sleek design,
  science fiction style, ultra-detailed, ultra HD, 8K, cold lighting, metallic textures, high-tech
 ambiance, detailed equipment",
// 6
"a serene Chinese garden, a curved stone bridge over a lotus-filled pond, elegant pavilions with
 ornate designs, weeping willow trees, koi fish swimming, gentle sunlight, ultra-detailed,
 photorealistic, ultra HD, 8K, traditional landscape design, tranquil atmosphere, vibrant yet
 harmonious colors",
// 7
"ancient Greek temple on a hilltop, surrounded by lush gardens, golden hour, marble columns,
 intricate carvings, mythological figures, painterly style, ultra-detailed, ultra HD, 8K, warm
 lighting, serene atmosphere, historical accuracy",
// 8
"a chessboard floating in a cosmic void, pieces made of planets and stars, a human hand reaching
 out to make a move, ultra-detailed, photorealistic, ultra HD, 8K, cosmic and abstract design,
 vivid lighting, surreal and thought-provoking atmosphere",
// 9
"a vintage gramophone in the middle of a lush rainforest, vines wrapping around the horn, music
 notes visibly floating in the air, animals like parrots and monkeys curiously gathered, ultra-
 detailed, photorealistic, ultra HD, 8K, vibrant colors, magical and whimsical atmosphere, rich
 textures",
// 10
"a giant clock embedded in a mountain cliff, waterfalls flowing through the clock's gears, lush
 greenery surrounding the scene, ultra-detailed, photorealistic, ultra HD, 8K, timeless and surreal
  atmosphere, intricate mechanical details, dramatic lighting",
// 11
"sunset over a rocky coastline, waves crashing against jagged cliffs, vivid orange and purple hues
 in the sky, seabirds flying above, tide pools with reflections, ultra-detailed, photorealistic,
 ultra HD, 8K, dynamic motion, tranquil yet dramatic atmosphere, intricate rock and water textures"
 ,
// 12
"a gigantic hourglass buried in a desert, golden sand slowly flowing between the chambers, a group
 of explorers climbing the hourglass, a storm brewing in the background, ultra-detailed,
 photorealistic, ultra HD, 8K, dramatic lighting, surreal and adventurous ambiance",
// 13
"portrait of a wise old man with a long white beard, wearing traditional robes, holding a wooden
 staff, mountain landscape in the background, soft diffused light, ultra-detailed, photorealistic,
 ultra HD, 8K, deep wrinkles, serene expression, mystical and timeless atmosphere"
```

Figure 20: GPT-generated prompts used in the appendix. (Cont.)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the abstract and introduction, we have clarified the focused problem and our contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in the appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We mention our experiment settings in Sec. 4 and include the detailed configurations in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We attach the source codes to reproduce the main experimental results as supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We mention our experiment settings in Sec. 4 and include the detailed configurations in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following the evaluation convention for text-to-image generation, only average results are reported.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We mention the related information in Sec. 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the guidelines and confirm that the conducted research follows them.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the impacts in the appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The applied safeguards are discussed in the appendix.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The license and terms of use of the used assets are respected and mentioned in the appendix.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.