ENAT: Rethinking Spatial-temporal Interactions in Token-based Image Synthesis

Zanlin Ni[1*] **Yulin Wang**[1*] **Renping Zhou**[1] **Yizeng Han**[1]
Jiayi Guo[1] **Zhiyuan Liu**[1] **Yuan Yao**[2†] **Gao Huang**[1†]
[1]Tsinghua University [2]National University of Singapore

Abstract

Recently, token-based generation approaches have demonstrated their effectiveness in synthesizing visual content. As a representative example, non-autoregressive Transformers (NATs) can generate decent-quality images in just a few steps. NATs perform generation in a progressive manner, where the latent tokens of a resulting image are incrementally revealed step-by-step. At each step, the unrevealed image regions are padded with `[MASK]` tokens and inferred by NAT, with the most reliable predictions preserved as newly revealed, visible tokens. In this paper, we delve into understanding the mechanisms behind the effectiveness of NATs and uncover two important interaction patterns that naturally emerge from NAT's paradigm: *Spatially* (within a step), although `[MASK]` and visible tokens are processed uniformly by NATs, the interactions between them are highly asymmetric. In specific, `[MASK]` tokens mainly gather information for decoding. On the contrary, visible tokens tend to primarily provide information, and their deep representations can be built only upon themselves. *Temporally* (across steps), the interactions between adjacent generation steps mostly concentrate on updating the representations of a few critical tokens, while the computation for the majority of tokens is generally repetitive. Driven by these findings, we propose EfficientNAT (ENAT), a NAT model that explicitly encourages these critical interactions inherent in NATs. At the spatial level, we *disentangle* the computations of visible and `[MASK]` tokens by encoding visible tokens independently, while decoding `[MASK]` tokens conditioned on the fully encoded visible tokens. At the temporal level, we prioritize the computation of the critical tokens at each step, while maximally *reusing* previously computed token representations to supplement necessary information. ENAT improves the performance of NATs notably with significantly reduced computational cost. Experiments on ImageNet-256^2 & 512^2 and MS-COCO validate the effectiveness of ENAT. Code and pre-trained models will be released at `https://github.com/LeapLabTHU/ENAT`.

1 Introduction

Recent years have witnessed an unprecedented growth in the field of AI-generated content (AIGC). In computer vision, diffusion models [10, 59, 61] have emerged as an effective approach. On the contrary, within the context of natural language processing, content is typically synthesized via the generation of discrete tokens using Transformers [72, 19, 5, 55]. Such discrepancy has excited a growing interest in exploring token-based generation paradigms for visual synthesis [7, 85, 33, 87, 6, 35]. Different from diffusion models, these approaches utilize a discrete data format akin to language models. This makes them straightforward to harness well-established language model optimizations such as the

[*] Equal contribution.
[†] Corresponding authors.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

refined scaling strategies [5, 54, 31, 73] and the progress in model infrastructure [65, 12, 8, 34, 96]. Moreover, explorations in this field may facilitate the development of more advanced, scalable multimodal models with a unified token space [17, 68, 18, 44, 90] as well as general-purpose vision foundation models that integrate visual understanding and generation capabilities [35, 69].

The recent advances in token-based visual generation have seen the rise of non-autoregressive Transformers (NATs) [7, 33, 6, 53], which are distinguished by their abilities to fulfill efficient and high-quality visual synthesis. As shown[3] in Figure 1, NATs follow a progressive generation paradigm: at each generation step, a certain number of latent tokens of the resulting image are decoded in parallel, and the model carries out this process iteratively to produce the final complete token maps. More specifically, at each step, the unknown latent tokens of the image are represented with `[MASK]` tokens and concatenated with the tokens that have been decoded (*i.e.*, visible tokens). Then, the full set of `[MASK]` and visible tokens is fed into a Transformer-based model, predicting the proper values of the unknown tokens, with the most reliable predictions preserved as the increments of visible tokens for the next generation step.

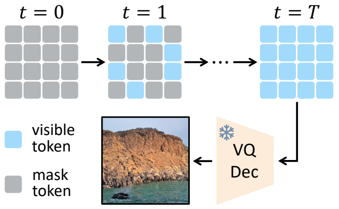


Figure 1: **The generation process of NATs** starts from a masked canvas, decode multiple tokens per step, and are then mapped to the pixel space using a pre-trained VQ-decoder [13].

In this paper, we seek to advance the understanding of the mechanisms behind the effectiveness of NATs' progressive generation procedures. Our investigation uncovers two important findings regarding the *spatial* and *temporal* interactions within NATs: *Spatially*, at each generation step, even though both `[MASK]` and visible tokens are treated equivalently within the computational graphs of NATs, the visible tokens naturally learn to mainly provide information for `[MASK]` tokens to infer the unknown image content, and their corresponding deep representations can be built in the absence of `[MASK]` tokens. *Temporally*, the interactions between adjacent generation steps mainly concentrate on updating the representations of a small number of "critical tokens" on top of the previous steps. In fact, the computation for the remaining majority of tokens is generally repetitive.

Inspired by these findings, we propose to develop novel NAT models to explicitly encourage these critical interaction mechanisms emerged naturally when trained for visual generation, yielding EfficientNAT (ENAT). Specifically, at the *spatial* level, we *disentangle* the computations of visible and `[MASK]` tokens by encoding visible tokens independently of `[MASK]` tokens. `[MASK]` tokens are then processed by attending to the fully contextualized features of visible tokens, as shown in Figure 3b. As an interesting observation derived from disentanglement, we find that prioritizing the computation for visible tokens, particularly when the computation is maximized for visible tokens and minimized for `[MASK]` tokens (even with only a single network layer), further improves the performance of NATs by a large margin. At the *temporal* level, we concentrate computation on the "critical tokens" while maximally *reusing* the representation of previously computed tokens to supplement the necessary information, as illustrated in Figure 4b.

Empirically, the effectiveness of ENAT is validated on ImageNet 256×256 [60], ImageNet 512×512 [60] and MS-COCO [36]. ENAT is able to achieve significantly reduced computational cost compared to conventional NATs while outperforming them notably (*e.g.*, 24% relative improvement with 1.8× lower cost, see Table 6a).

2 Related Work

Image tokenizer and token-based image generation models. Language models use algorithms like Byte Pair Encoding or WordPiece to convert text into tokens. Similarly, an image tokenizer transforms images into visual tokens for token-based image generation. Key works in this field include Discrete VAE [58], VQVAE [71], and VQGAN [13], with VQGAN-based tokenizers being most popular for their superior image reconstruction abilities. These tokenizers have enabled the advent of high-performance, scalable token-based generative models [85, 56, 87, 6]. Early token-based models were mainly autoregressive, generating images one token at a time [48, 13, 11, 85]. In contrast, non-autoregressive transformers (NATs)[7, 33, 6, 53] generate multiple tokens simultaneously, speeding up the process while maintaining high image quality. Recently, visual autoregressive models[70] introduced a next-scale prediction strategy, also demonstrating their promise in image synthesis.

[3] We illustrate with 4×4 tokens for simplicity; the actual token map size may be 16×16 or larger.

Efficient image synthesis has witnessed significant progress recently. Though the efficiency issue is relatively less explored in token-based image synthesis, it has been extensively studied in diffusion-based models. This includes advanced samplers [40, 41, 37], distillation methods [62, 83], quantization and compression techniques [92, 88, 91], and efforts to reduce redundant computation [42, 79, 1]. The last approach bears some resemblance to our computation reuse mechanism in Sec. 4.2, but with notable differences. Firstly, the subjects of research differ: we focus on NAT models. This focus introduces unique properties, *e.g.*, NATs incrementally decode new tokens at specific spatial locations, resulting in feature maps that are only significantly updated in those areas during generation. This contrasts with diffusion models, where feature map similarity between adjacent steps does not follow such predictable spatial patterns; instead, some layers show high overall similarity within a certain range of timesteps, while others may not. Secondly, these characteristic differences lead to distinct methodologies. Diffusion models typically require manually fine-tuned, and sometimes layer-specific caching schedules [42] to reuse previously computed features. This process can be labor-intensive and may struggle with generalization. In contrast, our method prioritizes model computation on newly decoded tokens in NATs and reuses the final representations of previously computed tokens without manually fine-tuned caching schedules.

Masked image modeling (MIM) methods like MAE [29] are widely used for *learning image representations* by predicting missing patches, with the encoder processing visible tokens and the decoder attending to both visible and masked tokens for reconstruction. CrossMAE [15] extends this by adopting a more disentangled architecture for handling both token types separately. In contrast, our work focuses on *image generation*, applying masked image modeling in discrete image token space, where token prediction and reconstruction are required at every step. This introduces key differences, such as SC-Attention and computation reuse mechanisms (see Sec. 4) which are not explored in these MIM approaches.

Non-autoregressive Transformers (NATs) originated in machine translation for their fast inference capabilities [19, 20]. Recently, they have been adapted for image synthesis, enabling efficient high-quality image generation as evidenced by various studies [7, 33, 35, 6, 53, 86]. MaskGIT [7] was the first to show NAT's effectiveness on ImageNet. This approach has been expanded for text-to-image generation, scaling up to 3B parameters in Muse [6] and achieving outstanding performance. Token-critic [33] and MAGE [35] enhance NATs further: Token-critic uses an auxiliary model for guided sampling, while MAGE integrates representation learning with image synthesis using NATs. Recent studies [46, 47] have also explored techniques for further improving the training and inference process of NATs. In contrast to these works, we aim to better understand the mechanisms behind NATs' effectiveness, uncovering findings that naturally lead to a more efficient and effective design for NAT models.

3 Preliminaries of Non-autoregressive Transformers (NATs)

In this section, we provide an overview of Non-Autoregressive Transformers (NATs) [7, 6, 35] for image generation. NATs operate with a pre-trained VQ-Autoencoder [71, 57, 13], which maps images to discrete visual tokens and reconstructs images from these tokens. The VQ-Autoencoder consists of three components: an encoder $\mathcal{E}^{\mathrm{VQ}}$, a quantizer $\mathcal{Q}$ with a learnable codebook e, and a decoder $\mathcal{D}^{\mathrm{VQ}}$. The encoder and quantizer transform an image into a sequence of visual tokens:

$$\boldsymbol{v} = \mathcal{Q}(\mathcal{E}^{\mathrm{VQ}}(\boldsymbol{x})), \tag{1}$$

where $\boldsymbol{v} = [v_i]_{i=1:N}$ is the sequence of visual tokens, and N is the sequence length. Each token v_i corresponds to a specific entry in the VQ-Autoencoder codebook. The above process is known as *tokenization*. After tokenization, NATs learn to generate visual tokens in the latent VQ space.

During training, NATs optimize the masked language modeling (MLM) objective [9]. Specifically, a random subset of tokens is replaced with a special `[MASK]` token, and the model is trained to predict the original tokens based on the unmasked ones. Formally, let $\boldsymbol{M}$ be the mask vector, where $m_i = 1$ indicates the i-th token is masked. The training objective minimizes the negative log-likelihood of the masked tokens:

$$L_{MLM} = -\sum_{i\in[1,N], m_i=1} \log p(v_i|\boldsymbol{v}_{\bar{\boldsymbol{M}}}), \tag{2}$$

where $p(v_i|\boldsymbol{v}_{\bar{\boldsymbol{M}}})$ is the predicted probability of token v_i given the unmasked tokens $\boldsymbol{v}_{\bar{\boldsymbol{M}}}$.

To generate images, NATs follow an iterative decoding strategy [7]. Starting with a fully masked token map, the model predicts all masked positions and samples a portion of the most confident predictions to replace the mask tokens in each iteration. The number of masked tokens to be replaced follows a cosine function, with fewer tokens replaced in the early iterations and more tokens replaced in later iterations. The finally decoded token sequence $\hat{\boldsymbol{v}}$ is then decoded into an image by the VQ-Autoencoder decoder:

$$\hat{\boldsymbol{x}} = \mathcal{D}^{\mathrm{VQ}}(\hat{\boldsymbol{v}}). \tag{3}$$

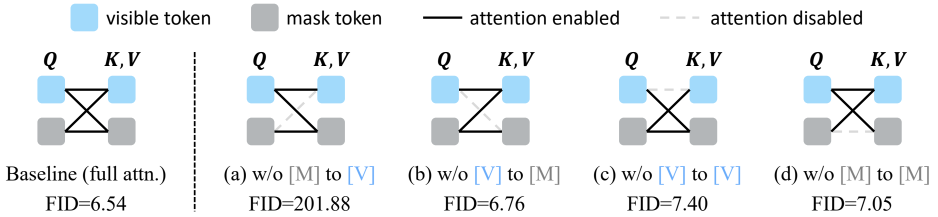
Due to space limitations, we refer readers to [7] for more details.

4 EfficientNAT (ENAT)

In this section, we design several analytical experiments (details in Appendix A.1) to advance the understanding of the mechanisms behind the effectiveness of NATs, aiming to accordingly improve the design of NAT models. Specifically, we uncover the critical spatial and temporal interaction patterns that naturally emerge within NATs under the goal of image generation. Inspired by our findings, we further propose to gradually re-design NATs towards maximally exploiting these characteristics.

4.1 Spatial Level Interaction

Motivation: an ablation study. A notable characteristic of NATs is the concurrent processing and interaction (through attention layers) of visible ([V]) and `[MASK]` ([M]) tokens when inferring the unknown image content. To better understand this mechanism, we consider an ablation study on four types of spatial interactions: a) [M] to [V] attention, b) [V] to [M] attention, c) [V] to [V] attention, and d) [M] to [M] attention. We find these four types of spatial interactions have significantly different impacts on the generation performance. As shown in Figure 2, the most important spatial interaction is the [M] to [V] attention (*i.e.*, [V]→[M] information propagation), without which the model is unable to converge at all. Moreover, both [M] to [M] and [V] to [V] attentions (*i.e.*, self-attention within the representation-extraction processes of visible and `[MASK]` tokens, respectively) moderately improve the model. The most intriguing fact is that removing the [V] to [M] attention (*i.e.*, [M]→[V] information propagation) only marginally hurts the model's performance.



Baseline (full attn.)
FID=6.54

(a) w/o [M] to [V]
FID=201.88

(b) w/o [V] to [M]
FID=6.76

(c) w/o [V] to [V]
FID=7.40

(d) w/o [M] to [M]
FID=7.05

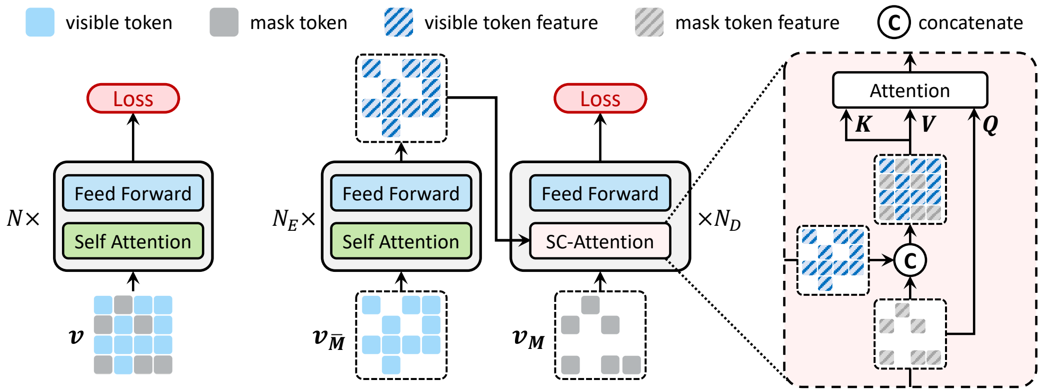
Figure 2: **An ablation study on four types of spatial interactions.** The essential spatial interaction is the [M] to [V] attention. In contrast, the [V] to [M] attention only marginally affects the model.

This imbalanced importance of four spatial interactions highlights the distinct roles of visible and `[MASK]` tokens. Specifically, the processing of the visible tokens primarily establishes certain internal representations based on the currently available and reliable information, and propagates them to the `[MASK]` tokens. In fact, their corresponding deep representations can be built mainly on top of themselves. In contrast, `[MASK]` tokens progressively gather information from visible tokens to predict the proper token values corresponding to the unknown parts of the images. In other words, *NATs naturally separate the role of visible and mask tokens when learning to generate images effectively, even though the two types of tokens are designed to be processed equally in NAT models.*

This phenomenon raises an intriguing question: can we **improve NATs by explicitly encouraging the naturally emergent spatial-level token-interaction patterns**? Actually, this idea is feasible. For example, we can consider a disentangled architecture that explicitly differentiates the roles of visible and `[MASK]` tokens. As shown in Figure 3b, we may process visible tokens *independently* of `[MASK]` tokens, with the sole purpose of encoding the current visible and reliable information. In contrast, the computation allocated to `[MASK]` tokens may only focus on predicting unknown image contents correctly with

Table 1: Effectiveness of disentangled architecture.

Arch.	GFLOPs	FID↓
Baseline	39.6	6.54
Disentangled	40.2	**5.50**



(a) Existing Works

(b) **Ours: Disentangled Architecture**

Figure 3: (a) **Existing works of NATs** process visible and [MASK] tokens equivalently. (b) **Our disentangled architecture** independently encodes visible tokens and integrates their fully contextualized features into the [MASK] token decoding process. $\boldsymbol{M}$ is the indicator of [MASK] tokens while $\bar{\boldsymbol{M}}$ is the indicator of visible tokens. The SC-Attention concatenates the visible and mask token features to produce keys and values, providing a complete context for the mask token decoding.

the help of fully contextualized visible token representations. Such a disentangled architecture significantly improves the performance of NATs at a similar computational cost (see Table 1 for evidence). Inspired by [2], we efficiently integrate the encoded visible tokens into the decoding process of [MASK] tokens with a tailored SC (SelfCross)-attention mechanism (see Figure 3b). The SC-attention simultaneously handles the interactions within [MASK] tokens and the interactions between [MASK] tokens and visible tokens, and it outperforms other possible designs like stacking self-attention and cross-attention layers alternately (see Table 6b).

Moreover, further explorations of our disentangled architecture yield an interesting finding: **prioritizing visible tokens results in an enhanced efficiency**. As shown in Table 2, the paradigm of equal computation allocation across all tokens derived from existing NATs may be far from optimal. Instead, allocating more computation to visible tokens yields notably better performance without sacrificing efficiency, while the computation on masked tokens can be reduced to only a single layer. This observation further underscores the importance of our proposed disentangled paradigm of processing visible tokens from masked ones in enabling advanced network architecture design.

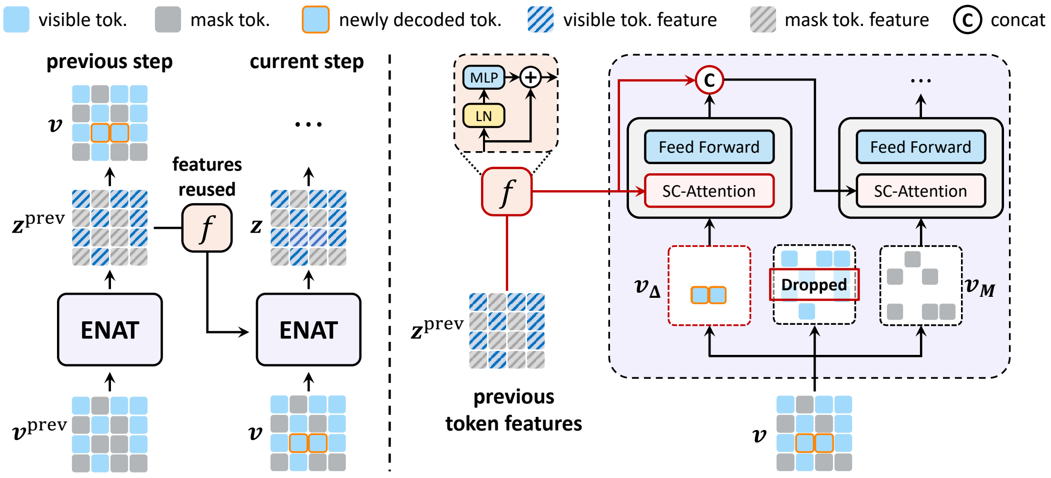
Table 2: Effects of prioritizing visible tokens. N_E, N_D: encoder/decoder layers (for visible/[MASK] tokens). Network width is slightly adjusted to make GFLOPs approximately unchanged.

N_E	N_D	GFLOPs	FID↓
8	8	40.2	5.50
12	4	38.2	4.98
15	1	39.8	**4.78**

4.2 Temporal Level Interaction

Feature similarity across generation steps. Another critical characteristic of NATs is their incremental revelation of unknown parts of the image upon previous steps. Beyond this straightforward procedure of progressive generation, here we are interested in whether there exist some interpretable temporal interaction patterns in NATs' behaviors. For instance, how do a NAT's computation results at the current step relate to those at the previous step? To investigate this, we conduct a similarity analysis of NATs' output features between two adjacent generation steps.

In Figure 5a, we randomly select two generated samples in NATs and visualize their token feature similarity at two adjacent steps (steps 2 & 3 and steps 6 & 7). We compare token-wise similarity and adopt cosine similarity as the metric: $\mathrm{Sim}(\boldsymbol{z}^{(t-1)}, \boldsymbol{z}^{(t)})_{ij} = \frac{\boldsymbol{z}_{ij}^{(t-1)} \cdot \boldsymbol{z}_{ij}^{(t)}}{\|\boldsymbol{z}_{ij}^{(t-1)}\| \|\boldsymbol{z}_{ij}^{(t)}\|}$, where $\boldsymbol{z}_{ij}^{(t)}$ denotes the feature of the token at position (i, j) and timestep t. The similarity map exhibits a highly polarized pattern: token representations undergo drastic changes at some "critical positions", while other positions remain highly similar between adjacent steps. When comparing with the positions of newly decoded tokens, we find that these "critical positions" correspond precisely to where the newly decoded tokens are located. In other words, *the major significance of each time step lies in updating*

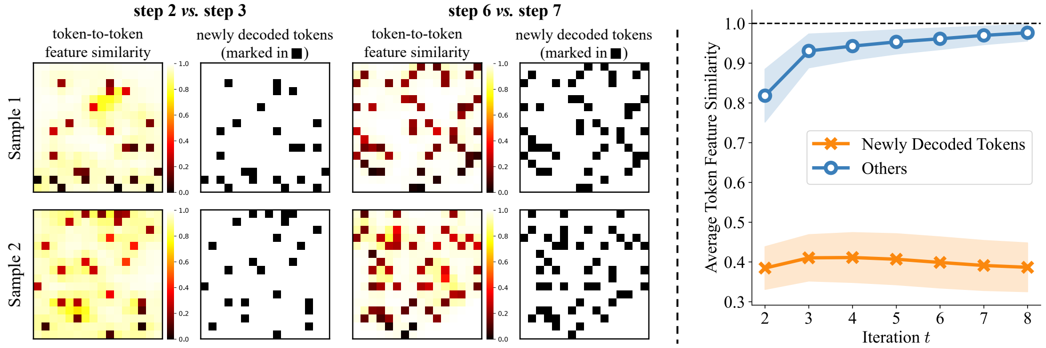


(a) Inference process of ENAT

(b) Detail of ENAT (diffs from Fig. 3b marked in **red**)

Figure 4: **Overview of ENAT.** Based on the disentangled architecture in Fig. 3b, we further propose to only encode the critical (*i.e.*, newly decoded) tokens and maximally reuse previously extracted features to supplement necessary information. $\boldsymbol{\Delta}$ is the indicator of newly decoded tokens. Only one transformer block is illustrated for simplicity.

the representations of newly decoded tokens, while the computation for the remaining majority of tokens is generally repetitive. In Figure 5b, we plot the average token similarity over 50,000 generated samples in each pair of adjacent steps ($t = 1 \to 2, t = 2 \to 3, \ldots, t = 7 \to 8$). The results show that this temporal interaction pattern remains consistent for different timesteps/samples.



(a) Token feature undergoes drastic change **only** at newly decoded positions

(b) Avg. similarity (step $t-1$ *vs.* step t)

Figure 5: **Feature similarity analysis.** (a) We randomly choose two samples and visualize the token-to-token feature similarity between adjacent steps (2 & 3 and 6 & 7), with the positions of newly decoded tokens visualized on the right. (b) The token feature similarity averaged over 50,000 generated samples in each pair of adjacent steps ($t = 1 \to 2, t = 2 \to 3, \ldots, t = 7 \to 8$).

Computation reuse. Driven by these observations, our key insight is: *during the generation of NATs, not all tokens need to be re-computed from scratch at each step*. Instead, only the newly decoded tokens need to be re-encoded to inject new knowledge about the image, while the previously encoded information can be maximally reused to supplement necessary details.

To implement this idea, we slightly modify the inference process upon our disentangled architecture (Sec. 4.1) by only encoding the newly decoded tokens at each step, while integrating the previously computed features to assist the current step's decoding:

$$\text{without reuse (Fig. 3b)}: \quad \boldsymbol{z} = \text{Forward}(\boldsymbol{v}_{\bar{M}}, \boldsymbol{v}_M) \tag{4}$$

$$\textbf{with reuse} \text{ (Fig. 4b)}: \quad \boldsymbol{z} = \text{Forward}(\boldsymbol{v}_{\boldsymbol{\Delta}}, \boldsymbol{v}_M, f(\boldsymbol{z}^{\text{prev}})) \tag{5}$$

where $\boldsymbol{z}^{\text{prev}}$ is the feature computed on the previous step, which is projected by a light-weight projection module $f(\cdot)$, and $\boldsymbol{v}_{\boldsymbol{\Delta}}$ denotes the newly decoded tokens. We adopt the SC-Attention mechanism (Sec. 4.1) to integrate the previous features into the current step's decoding process. At the end of the encoder, we simply concatenate the projected previous features with the computed features on the newly decoded tokens, and feed them into the decoder. In this way, the previously computed features are maximally reused to supplement both the encoding and decoding process of the current step, significantly reducing the computation cost and accelerating the generation process. The detailed inference process is illustrated in Figure 4b.

To equip the model with the ability to utilize previously computed features, we introduce minor modifications to the training process by alternating between normal forward mode and a "reuse forward mode" during training (each with 50% probability).

More specifically, the "reuse mode forward" during training is achieved through the following steps:

1. **Masking Tokens**: Given the current input token map $\mathbf{v}$, we mask a random subset of visible tokens in $\mathbf{v}$ to create $\mathbf{v}^{\text{prev}}$.
2. **Feature Extraction**: Feed $\mathbf{v}^{\text{prev}}$ into the NAT model to obtain its features $\mathbf{z}^{\text{prev}}$.
3. **Forward Pass and Loss Computation**: Use Eq. (5) to forward the current input token map $\mathbf{v}$ along with $\mathbf{z}^{\text{prev}}$ obtained in Step 2, and compute the loss. In practice, a stop gradient operation is applied before feeding $\mathbf{z}^{\text{prev}}$ into Eq. (5).

At other times, we use the original forward mode without incorporating previous features: Eq. (4), where the previous features are empty and the SC-Attention on the left of Figure 4b naturally reduces to the original self-attention mechanism. The reuse mechanism significantly accelerates the generation process, as shown in Table 6a. Additionally, we find in practice that only feeding the visible token feature of the previous step is sufficient and achieves better efficiency, as shown in Table 6d.

5 Experiments

Setups. Following [7, 35, 6], we utilize a pretrained VQGAN [13] with a codebook of size 1024 for image and visual token conversion. We employ three NAT models: ENAT-S (15 encoder layers, 1 decoder layer, 366 embedding dimensions, primarily for ablations), ENAT-B (15 encoder layers, 1 decoder layer, 768 embedding dimensions), and ENAT-L (22 encoder layers, 2 decoder layers, 1024 embedding dimensions). For class-conditional generation, we use adaptive layer normalization [80, 49] for conditioning. For text-to-image generation, we concatenate text embeddings with visual tokens for conditioning. Our training configurations follow [3] with minor adjustments to batch sizes and learning rates to accommodate different model sizes. For system-level comparisons in Sec. 5.1, we measure the TFLOPs of the entire generation process (including the decoder part for latent space generation models) to ensure fair comparisons.[4] All our experiments are conducted with 8 × A100 80G GPUs. We generally follow the approach described in [3] with minor modifications. More details on the training and inference setups, and the choice of our baselines can be found in Appendix A.2.

5.1 Main Results

Class-conditional generation on ImageNet 256×256 and 512×512. In Table 3, we compare our approach with other generative models on ImageNet 256×256. Our ENAT achieves superior performance with significantly lower computational cost. For instance, our ENAT-B model, despite having an extremely low inference cost, attains competitive FID scores of 3.53 in 8 steps. With a slightly increased computational budget, our ENAT-L model achieves a FID of 2.79 with only 0.3 TFLOPs, surpassing leading models with substantially less computational effort. For example, compared to the most performant baseline, *i.e.*, U-ViT-H [3], our ENAT-L model achieves a lower FID score (2.79 *vs.* 3.37) while requiring **8×** lower computational cost (0.3 TFLOPs *vs.* 2.4 TFLOPs). We further evaluate our ENAT on ImageNet 512×512 in Table 4. Our ENAT-L model also achieves a superior FID of 4.00 with only 1.3 TFLOPs, outperforming leading models with much lower inference cost. Qualitative results of our method are presented in Figure 7 and Appendix B.

[4] This differs from the GFLOPs reported in our ablation studies in Tabs. 1, 2, 6, where VQ-decoder costs are excluded to better compare the efficiency of different NAT designs.

Table 3: **Results on ImageNet 256×256** . TFLOPs quantify the total computational cost for generating a single image. For DPM-Solver [40] augmented diffusion models (†), we follow [40] to tune configurations and report the lowest FID. Diff: diffusion, AR: autoregressive.

Method	Type	#Params	Steps	TFLOPs↓	FID↓	IS↑
BigGAN-deep [4] (*ICLR'19*)	GAN	-	1	-	6.95	171.4
StyleGAN-XL [63] (*SIGGRAPH'22*)	GAN	-	1	1.5	2.30	265.1
VQVAE-2 [57] (*NeurIPS'19*)	AR	13.5B	5120	-	31.1	∼ 45
VQGAN [13] (*CVPR'21*)	AR	1.4B	256	-	15.78	78.3
ADM-G [10] (*NeurIPS'21*)	Diff.	554M	250	334	4.59	186.7
LDM [59] (*CVPR'22*)	Diff.	400M	250	52.3	3.60	247.7
LDM† [59] (*CVPR'22*)	Diff.	400M	4	1.2	11.74	-
			8	2.0	4.56	262.9
U-ViT-H† [3] (*CVPR'23*)	Diff.	501M	4	1.4	8.45	-
			8	2.4	3.37	235.9
DiT-XL† [49] (*ICCV'23*)	Diff.	675M	4	1.3	9.71	-
			8	2.2	5.18	213.0
MDT-XL† [16] (*ICCV'23*)	Diff.	676M	4	1.3	11.36	-
			8	2.2	4.00	-
USF [38] (*ICLR'24*)	Diff.	554M	8	10.7	9.72	-
MaskGIT [7] (*CVPR'22*)	NAT	227M	12	1.22	4.92	-
Token-Critic [33] (*ECCV'22*)	NAT	422M	36	1.9	4.69	174.5
Draft-and-revise [32] (*NeurIPS'22*)	NAT	1.4B	72	-	3.41	224.6
MAGE [35] (*CVPR'23*)	NAT	230M	20	1.0	6.93	-
MaskGIT-FSQ [43] (*ICLR'24*)	NAT	225M	12	0.8	4.53	-
AdaNAT [47] (*ECCV'24*)	NAT	206M	8	0.9	2.86	265.4
ENAT-B	NAT	219M	4	0.1	5.86	-
			8	0.2	3.53	302.4
ENAT-L	NAT	574M	4	0.2	4.13	-
			8	0.3	**2.79**	**326.7**

Table 4: **Results on ImageNet 512×512**. †: DPM-Solver [40] augmented diffusion models.

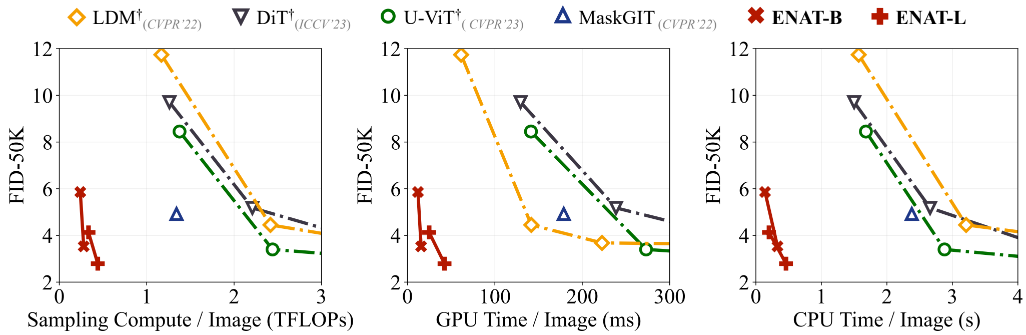
Method	Type	#Params	Steps	TFLOPs↓	FID↓	IS↑
VQGAN [13] (*CVPR'21*)	AR	227M	1024	-	26.52	66.8
ADM-G [10] (*NeurIPS'21*)	Diff.	559M	250	579	7.72	172.7
U-ViT-H† [3] (*CVPR'23*)	Diff.	501M	8	3.4	4.60	**286.8**
DiT-XL† [49] (*ICCV'23*)	Diff.	675M	8	9.6	5.44	275.0
MaskGIT [7] (*CVPR'22*)	NAT	227M	12	3.3	7.32	156.0
MaskGIT-RS [7] (*CVPR'22*)	NAT	227M	12	13.1	4.46	-
Token-Critic [33] (*ECCV'22*)	NAT	422M	36	7.6	6.80	182.1
Token-Critic-RS [33] (*ECCV'22*)	NAT	422M	36	34.8	4.03	-
ENAT-L	NAT	574M	8	**1.3**	**4.00**	285.7

Text-to-image generation on MS-COCO. We further assess the efficacy of ENAT for text-to-image generation on MS-COCO [36]. Table 5 shows that ENAT-B surpasses competing baselines with just 0.3 TFLOPs, achieving a FID score of 6.82. Compared to the competitive diffusion model U-ViT [3] with a fast sampler [41], ENAT-B requires similar computational resources to its 4-step variant while significantly outperforming it (6.82 *vs.* 16.20), and it also surpasses the 8-step sampling results of U-ViT with lower computational costs.

Table 5: **Results on MS-COCO**; all models are trained and evaluated on MS-COCO. †: DPM-Solver [40] augmented diffusion models.

Method	#Params	Steps	TFLOPs↓	FID↓
VQ-Diffusion [21]	370M	100	-	13.86
Frido [14]	512M	200	-	8.97
U-Net† [3]	53M	50	-	7.32
U-ViT† [3]	44M	4	0.4	16.20
		8	0.5	6.92
ENAT-B	116M	8	**0.3**	**6.82**

Practical efficiency. We provide more comprehensive comparisons of the trade-off between generation quality and computational cost in Figure 6. Both theoretical TFLOPs and the practical GPU/CPU latency for generating an image are reported. Our results show that ENAT consistently outperforms other baselines in terms of both generation quality and computational cost.



(a) **FID *vs.* TFLOPs** (b) **FID *vs.* GPU Time** (c) **FID *vs.* CPU Time**

Figure 6: **Practical efficiency of ENAT** . As a reference, we also plot the TFLOPs for generating a single image in (a). GPU time is measured on an A100 GPU with batch size 50. CPU time is measured on Xeon 8358 CPU with batch size 1. †: DPM-Solver [40] augmented diffusion models.

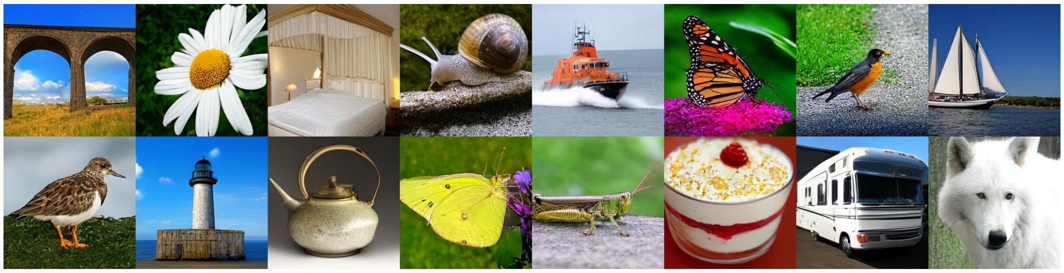


Figure 7: **Selected samples of ENAT-L** with 8 generation steps on ImageNet 256×256.

5.2 Ablation Studies

In this section, we present additional ablation studies on Imagenet 256×256 to validate the effectiveness of our proposed mechanisms. We use ENAT-S with 8 generation steps as our default setting, and report the FID score as well as the computational cost in GFLOPs for each NAT model.

Main ablation. Disentangled architecture and computation reuse are the two fundamental mechanisms in ENAT. The former separates the processing of visible and `[MASK]` tokens, and prioritizes computation on visible ones, while the latter eliminates repetitive processing of non-critical tokens. In Table 6a, we demonstrate the effectiveness of these two mechanisms. The results show that the disentangled architecture significantly improves NAT's performance, with a 1.76 improvement in FID score at a similar computational cost. Computation reuse, on the other hand, significantly reduces computational cost (1.8× fewer GFLOPs) while preserving most of the gains from disentanglement.

Effectiveness of SC-Attention. The SC-Attention mechanism adopted in our work serves dual roles: handling interactions of input tokens while simultaneously incorporating necessary additional information. Theoretically, the same functionality can be achieved with a stack of one self-attention layer and one cross-attention layer. However, as shown in Table 6b, SC-Attention outperforms the stack of self-attention and cross-attention layers with a lower FID (4.97 *vs*. 5.85) and a lower computational cost (22.6 *vs*. 25.0), demonstrating its effectiveness in our ENAT model.

Effectiveness of reuse projection module. In our computation reuse mechanism, a lightweight reuse projection module first processes the previous feature before integrating it into the current generation step. As shown in Table 6c, this design is highly important to our reuse mechanism. Without this module, the FID is 5.96, which is much worse than the 4.78 FID achieved without reuse. An intuitive explanation is that the reuse projection module learns the minimal necessary updates for the features of non-crucial tokens, preventing them from becoming too stale for more distant subsequent steps.

Which token features to reuse? Our basic reuse formulation integrates all previous token features into the current step. However, as shown in Table 6d, reusing only visible token features is equally effective while being much more efficient. As discussed in Section 4.1, encoding visible tokens

Table 6: **Ablation studies on ImageNet 256×256**. We use ENAT-S with 8 generation steps as our default setting, which is marked in gray . We report FID-50K following [3, 49] and total GFLOPs for each NAT model throughout the generation process.

(a) **Main ablation.** Our disentangled architecture and reuse mechanism significantly improves NATs.

Disentangle	Reuse	FID↓	GFLOPs↓
		6.54	39.6
✓		**4.78**	39.8
✓	✓	4.97	**22.6**

(b) **SC-Attention** outperforms alternately stacking self&cross attention layers with fewer GFLOPs.

Attn. Type	FID↓	GFLOPs↓
SC	**4.97**	**22.6**
self + cross	5.85	25.0

(c) **Reuse projection** is lightweight yet critical for maintaining performance.

Proj.	FID↓	GFLOPs↓
✓	**4.97**	22.6
✗	5.96	**20.8**

(d) **Which token features to reuse?** Reusing only visible token features of previous step is sufficient and much more efficienct.

Prev. Token Features	FID↓	GFLOPs↓
all	**4.95**	37.5
visible only	4.97	**22.6**

(e) **Which layer of feature to reuse?** Reusing last layer prev. features for all current layers is better than reusing in a layer-to-layer correspondence manner.

Prev. Feature Pos.	FID↓	GFLOPs
last layer	**4.97**	**22.6**
layer-to-layer	5.77	36.1

is most critical for NAT, and thus our ENAT model focuses most computation on these tokens. Therefore, using only visible token features suffices to provide the necessary information for reuse.

Which layer of feature to reuse? We compared reusing the last layer's features from the previous step with reusing features in a layer-by-layer manner, where the i-th layer of the current step reuses the features from the i-th layer of the previous step. As shown in Table 6e, reusing features from the last layer of the previous step outperforms the layer-by-layer approach, achieving a lower FID of 4.97. Additionally, it requires fewer GFLOPs (22.6 vs. 36.1), as the layer-by-layer approach needs to project features of each previous layer, while the last layer approach only projects once.

6 Conclusion

In this paper, we explored the underlying mechanisms of non-autoregressive Transformers (NATs) and uncovered key spatial and temporal token interaction patterns exist within NATs. Our findings highlight that spatially, visible tokens primarily provide information for [MASK] tokens, while temporally, updating the representations of newly decoded tokens is the main focus across generation steps. Driven by these findings, we propose ENAT, a NAT model that explicitly encourages these critical interactions. We spatially disentangle the computations of visible and [MASK] tokens by independently encoding visible tokens and conditioning [MASK] tokens on fully encoded visible tokens. Temporally, we focus computation on newly decoded tokens at each step, while reusing previously computed representations to facilitate decoding. Experiments on ImageNet and MS-COCO demonstrate that ENAT enhances NATs' performance with significantly reduced computational cost.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China under Grants 42327901.

References

[1] Shubham Agarwal, Subrata Mitra, Sarthak Chakraborty, Srikrishna Karanam, Koyel Mukherjee, and Shiv Kumar Saini. Approximate caching for efficiently serving text-to-image diffusion models. In *NSDI*, 2024.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[3] Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a vit backbone for score-based diffusion models. In *CVPR*, 2023.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

[6] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. MUSE: Text-to-image generation via masked generative transformers. In *ICML*, 2023.

[7] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. MaskGIT: Masked Generative Image Transformer. In *CVPR*, 2022.

[8] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *JMLR*, 2023.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. google ai language. In *NAACL*, 2019.

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021.

[11] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. CogView: Mastering text-to-image generation via transformers. In *NeurIPS*, 2021.

[12] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *ICML*, 2022.

[13] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.

[14] Wan-Cyuan Fan, Yen-Chun Chen, DongDong Chen, Yu Cheng, Lu Yuan, and Yu-Chiang Frank Wang. Frido: Feature pyramid diffusion for complex scene image synthesis. In *AAAI*, 2023.

[15] Letian Fu, Long Lian, Renhao Wang, Baifeng Shi, Xudong Wang, Adam Yala, Trevor Darrell, Alexei A Efros, and Ken Goldberg. Rethinking patch dependence for masked autoencoders. *arXiv preprint arXiv:2401.14391*, 2024.

[16] Shanghua Gao, Pan Zhou, Mingg-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *ICCV*, 2023.

[17] Yuying Ge, Yixiao Ge, Ziyun Zeng, Xintao Wang, and Ying Shan. Planting a seed of vision in large language model. In *ICLR*, 2024.

[18] Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. Making llama see and draw with seed tokenizer. In *ICLR*, 2024.

[19] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP*, 2019.

[20] Jiatao Gu and Xiang Kong. Fully non-autoregressive neural machine translation: Tricks of the trade. In *ACL*, 2021.

[21] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 2022.

[22] Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *ICCV*, 2023.

[23] Dongchen Han, Yifan Pu, Zhuofan Xia, Yizeng Han, Xuran Pan, Xiu Li, Jiwen Lu, Shiji Song, and Gao Huang. Bridging the divide: Reconsidering softmax and linear attention. In *NeurIPS*, 2024.

[24] 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. In *NeurIPS*, 2024.

[25] Dongchen Han, Tianzhu Ye, Yizeng Han, Zhuofan Xia, Shiji Song, and Gao Huang. Agent attention: On the integration of softmax and linear attention. In *ECCV*, 2024.

[26] Yizeng Han, Zeyu Liu, Zhihang Yuan, Yifan Pu, Chaofei Wang, Shiji Song, and Gao Huang. Latency-aware unified dynamic networks for efficient image recognition. *TPAMI*, 2024.

[27] Yizeng Han, Yifan Pu, Zihang Lai, Chaofei Wang, Shiji Song, Junfeng Cao, Wenhui Huang, Chao Deng, and Gao Huang. Learning to weight samples for dynamic early-exiting networks. In *ECCV*, 2022.

[28] Chunming He, Yuqi Shen, Chengyu Fang, Fengyang Xiao, Longxiang Tang, Yulun Zhang, Wangmeng Zuo, Zhenhua Guo, and Xiu Li. Diffusion models in low-level vision: A survey. *arXiv preprint arXiv:2406.11138*, 2024.

[29] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[30] 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. In *NeurIPS*, 2017.

[31] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[32] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and WOOK SHIN HAN. Draft-and-revise: Effective image generation with contextual rq-transformer. In *NeurIPS*, 2022.

[33] José Lezama, Huiwen Chang, Lu Jiang, and Irfan Essa. Improved masked image generation with token-critic. In *ECCV*, 2022.

[34] Shiyao Li, Xuefei Ning, Ke Hong, Tengxuan Liu, Luning Wang, Xiuhong Li, Kai Zhong, Guohao Dai, Huazhong Yang, and Yu Wang. Llm-mq: Mixed-precision quantization for efficient llm deployment. In *NeurIPS Workshop*, 2023.

[35] Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. MAGE: Masked generative encoder to unify representation learning and image synthesis. In *CVPR*, 2023.

[36] 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 *ECCV*, 2014.

[37] Enshu Liu, Xuefei Ning, Zinan Lin, Huazhong Yang, and Yu Wang. Oms-dpm: Optimizing the model schedule for diffusion probabilistic models. In *ICML*, 2023.

[38] Enshu Liu, Xuefei Ning, Huazhong Yang, and Yu Wang. A unified sampling framework for solver searching of diffusion probabilistic models. In *ICLR*, 2024.

[39] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. 2018.

[40] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-Solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022.

[41] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[42] Xinyin Ma, Gongfan Fang, and Xinchao Wang. DeepCache: Accelerating diffusion models for free. In *CVPR*, 2024.

[43] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. In *ICLR*, 2023.

[44] David Mizrahi, Roman Bachmann, Oguzhan Kar, Teresa Yeo, Mingfei Gao, Afshin Dehghan, and Amir Zamir. 4m: Massively multimodal masked modeling. In *NeurIPS*, 2023.

[45] Zanlin Ni, Yulin Wang, Jiangwei Yu, Haojun Jiang, Yue Cao, and Gao Huang. Deep incubation: Training large models by divide-and-conquering. In *ICCV*, 2023.

[46] Zanlin Ni, Yulin Wang, Renping Zhou, Jiayi Guo, Jinyi Hu, Zhiyuan Liu, Shiji Song, Yuan Yao, and Gao Huang. Revisiting non-autoregressive transformers for efficient image synthesis. In *CVPR*, 2024.

[47] Zanlin Ni, Yulin Wang, Renping Zhou, Rui Lu, Jiayi Guo, Jinyi Hu, Zhiyuan Liu, Yuan Yao, and Gao Huang. AdaNAT: Exploring adaptive policy for token-based image generation. In *ECCV*, 2024.

[48] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018.

[49] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[50] Yifan Pu, Weicong Liang, Yiduo Hao, Yuhui Yuan, Yukang Yang, Chao Zhang, Han Hu, and Gao Huang. Rank-detr for high quality object detection. In *NeurIPS*, 2024.

[51] Yifan Pu, Yiru Wang, Zhuofan Xia, Yizeng Han, Yulin Wang, Weihao Gan, Zidong Wang, Shiji Song, and Gao Huang. Adaptive rotated convolution for rotated object detection. In *ICCV*, 2023.

[52] Yifan Pu, Zhuofan Xia, Jiayi Guo, Dongchen Han, Qixiu Li, Duo Li, Yuhui Yuan, Ji Li, Yizeng Han, Shiji Song, et al. Efficient diffusion transformer with step-wise dynamic attention mediators. In *ECCV*, 2024.

[53] Shengju Qian, Huiwen Chang, Yuanzhen Li, Zizhao Zhang, Jiaya Jia, and Han Zhang. StraIT: Non-autoregressive generation with stratified image transformer. *arXiv preprint arXiv:2303.00750*, 2023.

[54] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

[55] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

[56] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[57] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2, 2019.

[58] Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

[59] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[60] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. In *IJCV*, 2015.

[61] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

[62] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

[63] Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling stylegan to large diverse datasets. In *SIGGRAPH*, 2022.

[64] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

[65] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[66] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[67] Yansong Tang, Zanlin Ni, Jiahuan Zhou, Danyang Zhang, Jiwen Lu, Ying Wu, and Jie Zhou. Uncertainty-aware score distribution learning for action quality assessment. In *CVPR*, 2020.

[68] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[69] Changyao Tian, Chenxin Tao, Jifeng Dai, Hao Li, Ziheng Li, Lewei Lu, Xiaogang Wang, Hongsheng Li, Gao Huang, and Xizhou Zhu. ADDP: Learning general representations for image recognition and generation with alternating denoising diffusion process. In *ICLR*, 2024.

[70] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *NeurIPS*, 2024.

[71] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017.

[72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[73] Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective works best for zero-shot generalization? In *ICML*, 2022.

[74] Yulin Wang, Zhaoxi Chen, Haojun Jiang, Shiji Song, Yizeng Han, and Gao Huang. Adaptive focus for efficient video recognition. In *ICCV*, 2021.

[75] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[76] Yulin Wang, Kangchen Lv, Rui Huang, Shiji Song, Le Yang, and Gao Huang. Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. In *NeurIPS*, 2020.

[77] Yulin Wang, Zanlin Ni, Shiji Song, Le Yang, and Gao Huang. Revisiting locally supervised learning: an alternative to end-to-end training. In *ICLR*, 2021.

[78] Yulin Wang, Yang Yue, Rui Lu, Yizeng Han, Shiji Song, and Gao Huang. EfficientTrain++: Generalized curriculum learning for efficient visual backbone training. *TPAMI*, 2024.

[79] Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache Me if You Can: Accelerating Diffusion Models through Block Caching. *CVPR*, 2024.

[80] Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. *NeurIPS*, 2019.

[81] Le Yang, Yizeng Han, Xi Chen, Shiji Song, Jifeng Dai, and Gao Huang. Resolution adaptive networks for efficient inference. In *CVPR*, 2020.

[82] Le Yang, Haojun Jiang, Ruojin Cai, Yulin Wang, Shiji Song, Gao Huang, and Qi Tian. CondenseNet v2: Sparse feature reactivation for deep networks. In *CVPR*, 2021.

[83] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. *arXiv preprint arXiv:2311.18828*, 2023.

[84] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[85] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *TMLR*, 2022.

[86] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. In *ICLR*, 2024.

[87] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2023.

[88] Zhihang Yuan, Pu Lu, Hanling Zhang, Xuefei Ning, Linfeng Zhang, Tianchen Zhao, Shengen Yan, Guohao Dai, and Yu Wang. DiTFastAttn: Attention compression for diffusion transformer models. In *NeurIPS*, 2024.

[89] Yang Yue, Yulin Wang, Bingyi Kang, Yizeng Han, Shenzhi Wang, Shiji Song, Jiashi Feng, and Gao Huang. Dynamic inference of multimodal large language models for efficient robot execution. In *NeurIPS*, 2024.

[90] Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. Anygpt: Unified multimodal llm with discrete sequence modeling. *arXiv preprint arXiv:2402.12226*, 2024.

[91] Tianchen Zhao, Tongcheng Fang, Enshu Liu, Wan Rui, Widyadewi Soedarmadji, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, et al. ViDiT-Q: Efficient and accurate quantization of diffusion transformers for image and video generation. *arXiv preprint arXiv:2406.02540*, 2024.

[92] Tianchen Zhao, Xuefei Ning, Tongcheng Fang, Enshu Liu, Guyue Huang, Zinan Lin, Shengen Yan, Guohao Dai, and Yu Wang. MixDQ: Memory-efficient few-step text-to-image diffusion models with metric-decoupled mixed precision quantization. In *ECCV*, 2024.

[93] Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dynamic diffusion transformer, 2024.

[94] Wangbo Zhao, Jiasheng Tang, Yizeng Han, Yibing Song, Kai Wang, Gao Huang, Fan Wang, and Yang You. Dynamic tuning towards parameter and inference efficiency for vit adaptation. In *NeurIPS*, 2024.

[95] Ziwei Zheng, Le Yang, Yulin Wang, Miao Zhang, Lijun He, Gao Huang, and Fan Li. Dynamic spatial focus for efficient compressed video action recognition. *TCSVT*, 2024.

[96] Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, et al. A survey on efficient inference for large language models. *arXiv preprint arXiv:2404.14294*, 2024.

A Implementation Details

A.1 Detailed model configurations.

Here we present the detailed configurations of all our NAT models appeared within this paper in Table 7. We provide the number of encoder layers (N_E), decoder layers (N_D), the dimension of the hidden states (embed dim.), the number of attention heads (# attn. heads):

Table 7: **Summary of model configurations.** N_E: encoder layers (for visible token encoding), N_D: decoder layers (for `[MASK]` token decoding). *: In conventional NAT models, the layers for visible token encoding are shared with the layers for `[MASK]` token decoding.

arch.	reuse?	N_E	N_D	embed dim.	# attn. heads
baseline	✗	8^*		288	6
disentangled	✗	8	8	288	6
disentangled	✗	12	4	318	6
disentangled	✗	15	1	366	6
ENAT-S	✓	15	1	366	6
ENAT-B	✓	15	1	768	8
ENAT-L	✓	22	2	1024	16

A.2 Details of training and evaluation.

For ImageNet 256×256, we use a batch size of 2048 and a learning rate of 4e-4. For ImageNet 512×512, to manage the increased sequence length, we reduce the batch size to 512 and linearly scale down the learning rate to 1e-4. For MS-COCO, we train for 150k steps instead of the 1000k steps used in [3].

For our ablation studies in Sec. 5.2 and explorative experiments in Sec. 4, we train the models for 300k steps instead of the 500k steps used in [3], while keeping the other settings the same as above.

For data preprocessing, we perform center cropping and resizing to 256×256 for ImageNet 256×256 and MS-COCO, and to 512×512 for ImageNet 512×512. Additionally, we adopt random horizontal flipping as data augmentation, following [3, 49].

Our evaluation on FID follows the same evaluation protocol as [10, 3, 49]. We adopt the pre-computed dataset statistics from [3] and generate 50k samples for ImageNet (30k for MS-COCO) to compute the statistics for the generated samples, using the following formula to calculate FID [30]:

$$\text{FID} = ||\mu_{\text{real}} - \mu_{\text{fake}}||_2^2 + \text{Tr}(\Sigma_{\text{real}} + \Sigma_{\text{fake}} - 2(\Sigma_{\text{real}}\Sigma_{\text{fake}})^{1/2}), \tag{6}$$

where μ and Σ are the mean and covariance of the real and fake samples, respectively. The evaluation on Inception Score (IS) follows the same protocol as [3, 49], using a pre-trained InceptionV3 model [66] to compute the IS.

For the choice of baselines in our work, since ENAT focuses on inference efficiency, we aim to compare ENAT with other models in a lightweight, low-FLOPs scenario. However, while the inference efficiency of generative models is important, it is generally under-explored in the original papers of state-of-the-art diffusion models (e.g., DiT [49], MDT [16]), which mostly focus on enhancing generation performance. The official results of them are primarily obtained with hundreds of inference steps, making direct comparisons with ENAT challenging. For instance, as shown in Fig. 8, the official results of DiT, MDT, etc. all concentrate at the high end of overall inference costs, requiring hundreds of times more computation than ENAT.

Fortunately, there are well-established fast sampling techniques (e.g. DPM-Solver [40]) for accelerating diffusion models, which allows us to reduce their sampling steps and compare them with ENAT in a fairer setting.

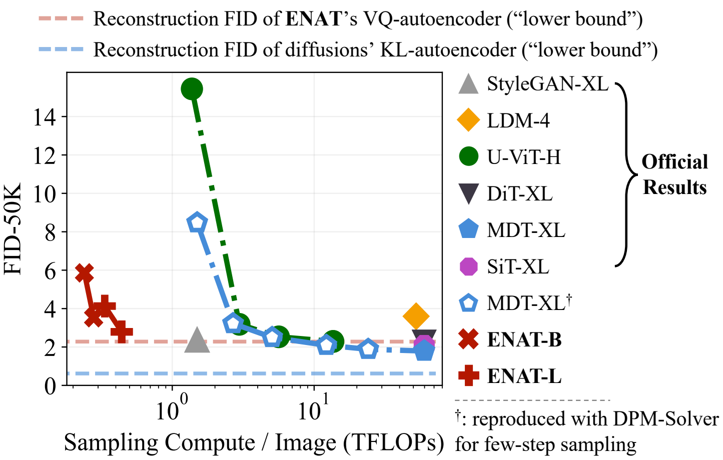


Figure 8: **System-level comparisons on ImageNet 256×256**. All baseline results are sourced from their original papers, except for the few-step MDT results (†).

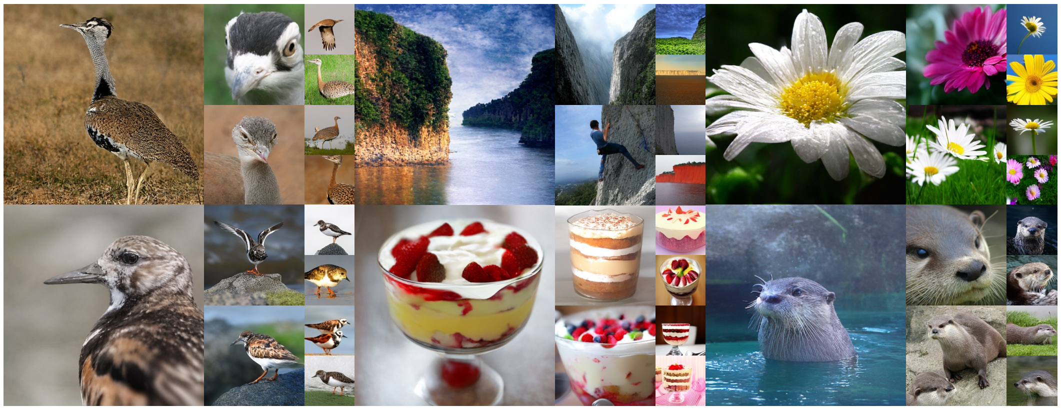


Figure 9: **Selected samples of ENAT-L** with 8 generation steps on ImageNet 256×256 and 512×512.

B More Qualitative Results.

Here we present more qualitative results in Figure 9. For each class, the first two columns contain 3 ImageNet 512×512 samples and the last column contains 4 ImageNet 256×256 samples.

C Limitations and Future Work

Although our experiments have covered two fundamental types of generative models, namely class-conditional and text-to-image generation, and utilized three datasets, investigating the efficacy of ENAT on more diverse datasets, such as the widely used CelebA [39] and LSUN [84], and exploring additional generation types like unconditional generation, constitute valuable directions for future research. Moreover, scalability, both in terms of model size and dataset volume, is a crucial capability for current generative models. Our largest model scales up to approximately 0.6 billion parameters, and our experiments utilized datasets with a maximum size of 1.2 million images (ImageNet dataset). Evaluating the performance of ENAT on even larger-scale datasets, such as LAION-5B [64], and further scaling the model to surpass 1 billion parameters, could provide deeper insights into its scalability and robustness.

To further enhance the applicability and efficiency of non-autoregressive Transformers, integrating other adaptive inference methods [76, 75, 26, 95] and learning techniques [67, 82, 77] will be essential. For instance, methods like dynamic neural network [74, 27, 89, 93, 94, 52] and resolution-adaptive models [81] offer promising pathways to explore. Additionally, examining ENAT across

Table 8: **Licenses for existing assets.**

dataset / code	source	license
MS-COCO	[36]	New BSD License
ImageNet	[60]	Custom (research, non-commercial)
MaskGIT	[7]	Apache-2.0 license
U-ViT	[3]	MIT license

diverse tasks and domains [51, 50, 28] and leveraging advances in model training and inference techniques [45, 22, 25, 23, 78, 24] can strengthen its performance and expand its scope.

D Broader Impacts

On the positive side, the proposed EfficientNAT (ENAT) models significantly reduce computational costs, making advanced visual generation technology more accessible. This democratization can benefit diverse sectors, including education, healthcare, and creative industries. However, as with any AI-generated content technology, there are potential ethical considerations such as creating misleading content or spreading misinformation. Additionally, like other data-driven approaches, the model may inadvertently reinforce biases present in the training data. Possible mitigation strategies for these concerns include developing robust detection methods for generated content, promoting transparency in AI-generated content, and ensuring diverse and representative training data.

E Licenses

The Table 8 outlines the assets used in our work, their sources and licenses. Our models, data and code will be open-sourced under the MIT License upon paper acceptance.

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: The abstract and introduction have stated our contributions and scope, which are supported by the results in Section 5.1. Additionally, the ablation studies in Section 5.2 provide insights into the mechanisms behind the effectiveness of ENAT.

   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: Limitations are discussed in Appendix C

   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: Implementation details are provided in Section 5 and Appendix A.

   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: [No]

Justification: Data and code will be available upon paper acceptance.

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: Detailed experimental setups are provided in Section 5 and Appendix A.

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: Like most works [59, 49] in the field, training generative models on large-scale datasets typically involves high computational costs, making it impractical to run multiple trials for each experiment.

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: Compute resource information is provided in Appendix A.2.

   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: The research conforms with the NeurIPS Code of Ethics.

   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: Social impacts are discussed in Appendix D.

    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: [NA]

    Justification: The paper poses no such risks.

    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: We cited their original papers and included their license in Appendix E.

    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: [Yes]

    Justification: The code will be released under the MIT License. Implementation details are provided in Section 5 and Appendix A.

    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.