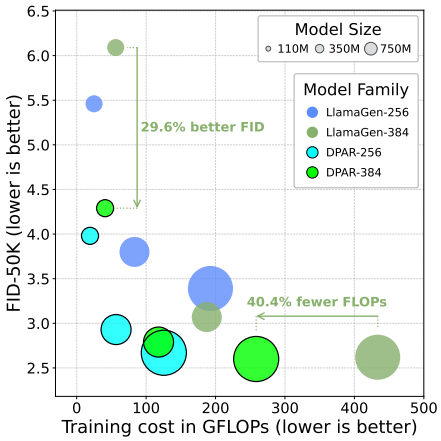
This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

# DPAR: Dynamic Patchification for Efficient Autoregressive Visual Generation

Divyansh Srivastava[1†] Akshay Mehra[2*] Pranav Maneriker[2*] Debopam Sanyal[3†]
Vishnu Raj[2] Vijay Kamarshi[2] Fan Du[2] Joshua Kimball[2]
[1]University of California San Diego [2]Dolby Laboratories [3]Georgia Tech [*]Equal Contribution



(a)




(b)

Figure 1. **Autoregressive image generation with DPAR.** (a) Selected samples from our class-conditional DPAR-384-XL model trained on ImageNet. (b) FID vs. FLOPs for DPAR variants at 256 × 256 and 384 × 384 resolution. DPAR reduces training FLOPs by up to 40.4% while improving FID by up to 29.6% over baseline models.

## Abstract

*Decoder-only autoregressive image generation typically relies on fixed-length tokenization schemes whose token counts grow quadratically with resolution, substantially increasing the computational and memory demands of attention. We present **DPAR**, a novel decoder-only autoregressive model that dynamically aggregates image tokens into a variable number of patches for efficient image generation. Our work is the first to demonstrate that next-token prediction entropy from a lightweight and unsupervised autoregressive model provides a reliable criterion for merging tokens into larger patches based on information content. DPAR makes minimal modifications to the standard decoder architecture, ensuring compatibility with multimodal generation frameworks and allocating more compute to the generation of high-information image regions. Further, we demonstrate that training with dynamically sized patches yields representations that are robust to patch boundaries, allowing DPAR to scale to larger patch sizes at inference. DPAR reduces token count by 1.81× and 2.06× on ImageNet 256 and 384 generation resolution, respectively, leading to a reduction of up to 40.4% FLOPs in training costs. Further, our method exhibits faster convergence and improves FID by up to 29.6% relative to baseline models.*

## 1. Introduction

The large-scale success of autoregressive (AR) decoder-only language models [1, 6, 11, 28, 38, 47, 48, 54, 55, 83, 84, 86, 88, 89] has sparked a growing interest in extending next-token prediction paradigm to image generation, for a seamless integration with language models for unified multimodal generation [10, 18, 80, 87, 90, 92, 98]. Recent works [36, 51, 56, 77, 78, 91, 94] demonstrate that AR approaches can achieve performance comparable to, and in some cases surpass, diffusion models [12, 16, 23, 25, 57, 60, 64, 72–74], which have long dominated the image generation [5, 19, 26, 27, 29–31, 67, 68, 99] landscape. Typical AR image generation methods, including Llama-

† Work done during internship at Dolby Laboratories.
Code: https://github.com/dsrivastavv/DPAR

Gen [77], employ VQ-VAE [59, 77, 85] to tokenize images into discrete 2D tokens that are flattened into 1D sequences, followed by next-token prediction training with minimal changes to a decoder-only transformer model. However, a fundamental scalability challenge persists: the number of tokens scales quadratically with image resolution, leading to substantial increases in computational and memory costs for attention [8, 79, 97]. For instance, generating a 256×256 image with a standard 16× downsampling requires generating 256 tokens, whereas a 1024×1024 image requires 4096 tokens — a 16× increase in token count and context length for attention.

Existing methods have attempted to reduce token counts for AR image generation through 1D tokenization [14, 44, 70, 96] and token compression techniques [2, 3, 9, 17, 20, 32, 37, 43, 58, 63]. While 1D tokenizers reduce the number of tokens, they are often not favored due to the loss of 2D spatial structure, which is essential for zero-shot editing capabilities such as extrapolation and outpainting [51]. Moreover, token compression methods typically merge statically by a fixed factor, often combining high-information regions, leading to information loss and degraded generation quality. In this work, we ask the question: *Can we dynamically merge image tokens based on their information content while preserving the 2D spatial structure for efficient AR image generation?*

To this end, we propose **DPAR**, a novel autoregressive image generation model that dynamically aggregates discrete 2D image tokens into a variable number of *patches* for efficient visual generation. Images often contain low-information regions, such as homogeneous areas like sky or walls, that can be represented with fewer tokens without information loss [21]. Inspired by recent advances in NLP [45, 46, 49, 76], we leverage next-token prediction entropy from a lightweight unsupervised AR model to merge tokens into larger units called *patches* (see Figure 2). Image regions with homogeneous details have few potential choices and patterns for the next token, and vice versa, leading to low and high next-token prediction entropy, respectively. This allows merging tokens in low-information regions while preserving token-level granularity in high-information areas, resulting in a more balanced allocation of compute during generation. Overall, our method unifies the strengths of 2D tokenization, 1D tokenization, and token-merging approaches by preserving 2D spatial structure while dynamically merging tokens based on their information content.

Our method leverages a lightweight encoder that aggregates tokens into patches based on next-token prediction entropy, and a corresponding decoder that reconstructs tokens from generated patches. The autoregressive transformer operates on a reduced number of patches instead of tokens, reducing the computational cost of attention. We evaluate our proposed method on the ImageNet-256 [62] class-conditional image generation benchmark and demonstrate that DPAR achieves a 1.81× and 2.06× reduction in token count for 256×256 and 384×384 image generation, respectively, leading to a significant reduction of up to 40.4% GFLOPs in training costs and improves FID by up to 29.6% relative to baselines. Finally, we show that DPAR's training with a dynamic patch-based representation yields robust representations, enabling DPAR to scale to larger patch sizes at inference. Our contributions are summarized below:

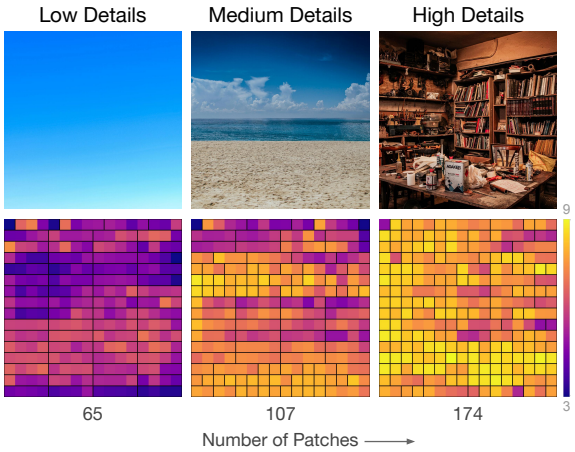



Figure 2. **Entropy-guided Token Aggregation.** *(Top)* Input images with increasing information content. *(Bottom)* Corresponding entropy heatmaps, where warmer colors indicate higher next-token prediction entropy. Low-information regions yield low-entropy tokens that are merged into patches, while high-information regions retain fine-grained token-level granularity. Black outlines denote final patch boundaries for 256×256 images.

- We present **DPAR**, a novel autoregressive image generation model that dynamically aggregates image tokens into a variable number of patches based on their information content, enabling efficient generation.
- DPAR achieves 1.81× and 2.06× reduction in token count on ImageNet 256×256 and 384×384 generation, respectively, leading to a significant reduction of up to 40.4% GFLOPs in training costs. Further, our method exhibits faster convergence and improves FID by up to 29.6% relative to baseline models.
- We demonstrate that DPAR's training with dynamically sized patches makes its learned representations robust to patch boundaries, enabling DPAR to scale to larger patch sizes during inference for further efficiency gains.

## 2. Related Work

**Autoregressive Image Generation** Seminal work LlamaGen [77] adopts a decoder-only Llama [83] architecture and is trained with standard next-token prediction (NTP) loss on discrete 2D VQ-VAE [85] tokens rasterized left-to-right to 1D sequence, achieving performance compara-

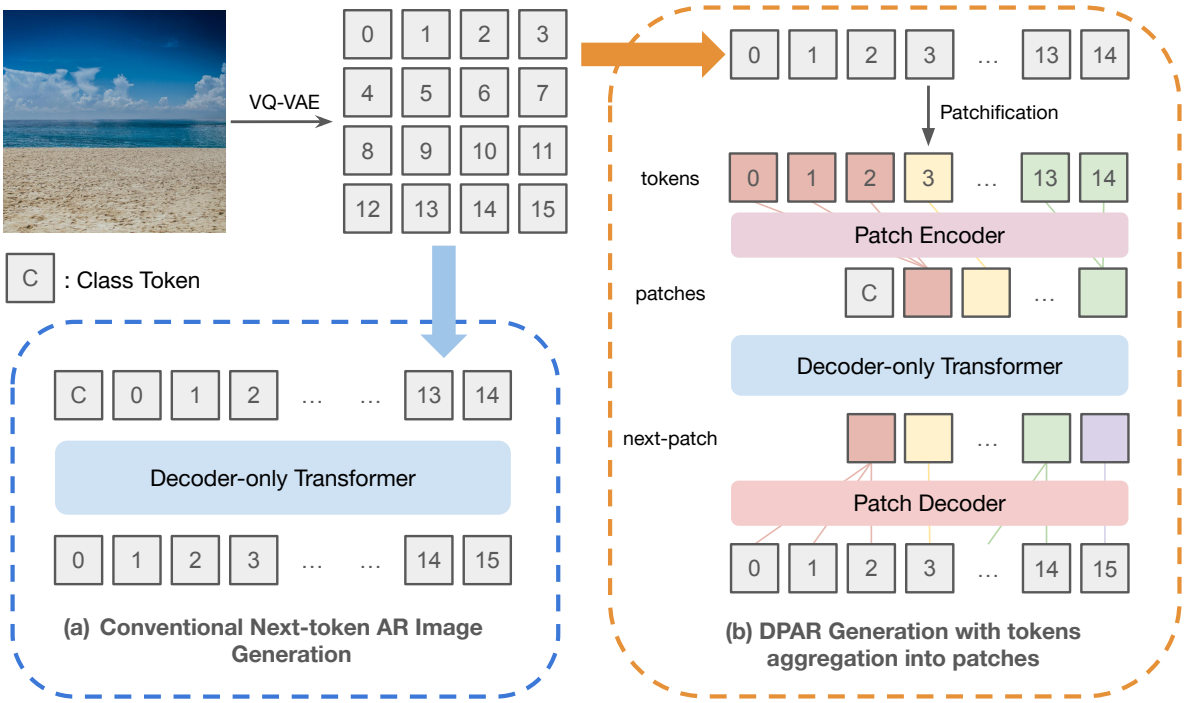



Figure 3. **Overview of DPAR**. (a) Conventional AR image generation employs decoder-only transformers that operate on a fixed number of tokens per image, where the token count scales quadratically with image resolution. (b) DPAR dynamically aggregates image tokens based on information content, producing a variable number of patches per image. Decoder-only transformers then operate on this reduced set of patches, lowering both computational and memory overhead of attention. DPAR requires minimal modifications to the standard decoder architecture, preserving compatibility with multimodal generation frameworks.

ble to diffusion models [12, 16, 23, 25, 57, 60, 64, 72–74]. Later works have proposed improvements to vanilla AR image generation along two major directions: i) **Random-order Generation** RAR [94] modifies standard NTP training by randomly reordering the raster-order sequence and then linearly lowers the reordering probability, guiding training back to raster order NTP. RandAR [51] removes the raster-order sequencing of decoder-only models and generates images via random-order next-token prediction, both during training and inference. SAR [39] proposes next-set generation, which is obtained by splitting the sequence into arbitrary sets of multiple tokens. ii) **Modified Training Paradigms** VAR [82] trains a multi-scale tokenizer and generates images by autoregressively producing next-resolution token maps, progressing from coarse to fine resolutions with parallel decoding within each scale step. NPP [50] starts training with large static patches and gradually transitions to standard NTP, and does not use patches at inference time, unlike our method. While these approaches focus on improving the fidelity of AR Image Generation, our work addresses the scalability challenge posed by quadratic token growth in 2D tokenizers with increasing resolution, while maintaining generation quality. Further, our approach is orthogonal and complementary to these works and we leave this exploration to future work.

**Token Reduction in NLP** Tokenization of text forms a fundamental part of the pipeline for processing text data used in training models for processing natural language. Despite known limitations [4, 65], algorithms utilizing count-based merging of text units called subwords are present in the majority of Large Language Models (LLMs) today [71]. The most common of these are the Byte Pair Encoding (BPE) [69] tokenizer, which starts by treating each word as a sequence of characters and iteratively merges the most frequent adjacent pair to create larger tokens. Recently, Byte Latent Transformer (BLT) [49] introduced a dynamic byte-merging strategy based on next-byte prediction entropy from a lightweight autoregressive model, effectively reducing the token count for efficient language modeling from bytes. Our work draws inspiration from BLT, but unlike BLT, which tokenizes text bytes, we demonstrate that next-token prediction entropy can be applied at the VQ-VAE token level to merge low-information spatial tokens into larger patches for efficient AR image generation.

**Token Reduction for Image Generation** Recent works have explored 1D tokenizers, which aim for better compression ratios and fewer tokens at the expense of 2D spatial structure: CAT [70] primarily focuses on continuous latent space and adjusts token counts from a fixed set of

**Algorithm 1** DPAR Training Algorithm

**Input:** Image tokens $I_{\text{tok}} = [x_0, \ldots, x_{T-1}]$, condition $C$, entropy model $\mathcal{E}_\phi$, threshold $\mathcal{H}_{\text{Th}}$, max patch length $P_{\max}$
**Output:** Cross-entropy loss $\mathcal{L}_{\text{CE}}$

$I_{\text{patch}} \leftarrow \textsc{Patchify}(I_{\text{tok},\,0:T-2},\, \mathcal{E}_\phi,\, \mathcal{H}_{\text{Th}},\, P_{\max})$
$\mathbf{H}^{\text{tok}}, \mathbf{H}^{\text{patch}} \leftarrow \textsc{Encoder}(I_{\text{tok},\,0:T-2},\, I_{\text{patch}})$
$\widehat{\mathbf{H}}^{\text{patch}} \leftarrow \text{GPT}\big(C,\, \mathbf{H}^{\text{patch}}\big)$ ▷ AR on patches
$\widehat{\mathbf{P}}^{\text{tok}} \leftarrow \textsc{Decoder}(\mathbf{H}^{\text{tok}},\, \widehat{\mathbf{H}}^{\text{patch}})$
$\mathcal{L}_{\text{CE}} \leftarrow \textsc{CrossEntropyLoss}(\widehat{\mathbf{P}}^{\text{tok}},\, I_{\text{tok}})$
**return** $\mathcal{L}_{\text{CE}}$

compression ratios (typically 3) predicted by LLM based on the image complexity. TiTok [95] directly compresses images into 1D latent sequences and shows successful representation of 256×256 images with just 32 tokens. One-D-Piece [44] extends TiTok and proposes tail-token drop to concentrate information at the head of the sequence, allowing for variable length representation. Adaptive Length Image Tokenization [14] recurrently distills 2D discrete tokens into 1D tokens, with each iteration adding more tokens and representation capability. While 1D tokenizers result in fewer tokens, they are often less favored due to the loss of spatial structure, which is crucial for zero-shot editing capabilities such as outpainting and inpainting. Another line of work [20, 43] focuses on token-merging techniques to reduce the token count: Token-Shuffle [43] leverages the low dimensionality of visual codes to merge local tokens along the channel dimension, effectively reducing the overall token count. However, these methods rely on fixed-scale merging, which can combine high-information regions and result in the loss of fine details during generation. In contrast, our approach combines strengths of both directions: it supports variable-length representations, as in 1D tokenizers, based on image information content, while retaining the 2D spatial structure required for zero-shot editing tasks.

# 3. Methodology

## 3.1. Preliminaries

**Autoregressive Image Generation** Decoder-only autoregressive models represent an image $I \in \mathbb{R}^{H\times W\times 3}$ as a sequence of discrete 1D tokens and predict each token sequentially, conditioned on the previous tokens. Formally, given an image-condition pair $(I, C) \in \mathcal{D}$ and its tokenized representation $I_{\text{tok}} = [x_0, x_1, \ldots, x_{T-1}]$, where $x_i \in \{0, 1, \ldots, V-1\}$ is the $i$-th token in an ordered sequence from a vocabulary of size $V$ and $T$ is the total number of tokens. The model learns the likelihood of $I$ as a product of autoregressive factors:

$$P(I_{\text{tok}} \mid C) = \prod_{t=0}^{T-1} P_\theta(x_t \mid C, x_{<t}), \tag{1}$$

**Algorithm 2** DPAR Inference Algorithm

**Input:** Condition $C$, entropy model $\mathcal{E}_\phi$, threshold $\mathcal{H}_{\text{Th}}$, max patch length $P_{\max}$, target token length $T$
**Output:** Generated image tokens $I_{\text{tok}}$

$I_{\text{tok}} \leftarrow [\,]$
**for** $t = 0$ to $T-1$ **do**
  $I_{\text{patch}} \leftarrow \textsc{Patchify}(I_{\text{tok}},\, \mathcal{E}_\phi,\, \mathcal{H}_{\text{Th}},\, P_{\max})$
  $e_t \leftarrow \mathcal{H}(x_{<t};\, \mathcal{E}_\phi)$
  **if** $e_t \le \mathcal{H}_{\text{Th}}$ **and** $|P_m| < P_{\max}$ **then**
    $\mathbf{H}^{\text{tok}}, _ \leftarrow \textsc{Encoder}(I_{\text{tok}},\, I_{\text{patch}})$
  **else**
    $\mathbf{H}^{\text{tok}}, \mathbf{H}^{\text{patch}} \leftarrow \textsc{Encoder}(I_{\text{tok}},\, I_{\text{patch}})$
    $\widehat{\mathbf{H}}^{\text{patch}} \leftarrow \text{GPT}\big(C,\, \mathbf{H}^{\text{patch}}\big)$ ▷ next-patch
  **end if**
  $\widehat{\mathbf{P}}^{\text{tok}} \leftarrow \textsc{Decoder}(\mathbf{H}^{\text{tok}},\, \widehat{\mathbf{H}}^{\text{patch}})$
  $x_t \sim \hat{p}_t$ ▷ $\hat{p}_t$ is the $t$-th distribution in $\widehat{\mathbf{P}}^{\text{tok}}$
  $I_{\text{tok}} \leftarrow I_{\text{tok}} \cup [x_t]$
**end for**
**return** $I_{\text{tok}}$

where $C$ denotes an optional conditioning signal (e.g., class label or text prompt), and $P_\theta$ is parameterized by a decoder-only transformer with parameters $\theta$. The model is trained to minimize the cross-entropy loss between the predicted next-token probabilities $\widehat{\mathbf{P}}^{\text{tok}} = [\hat{p}_0, \ldots, \hat{p}_{T-1}]$, where $\hat{p}_t = P_\theta(\cdot \mid C, x_{<t})$ and the ground-truth tokens $I_{\text{tok}}$.

$$\mathcal{L}_{\text{CE}} = -\sum_{t=0}^{T-1} \log \hat{p}_t(x_t). \tag{2}$$

During inference, the model generates images by sampling tokens sequentially from the learned distribution until the token length $T$ is reached.

**Tokenization** One popular choice for image tokenization is VQ-VAE which encodes images into discrete 2D grid of tokens. Formally, given an image $I \in \mathbb{R}^{H\times W\times 3}$, VQ-VAE downsamples images by a factor of $K$ and maps each latent to a discrete token $x_i$ in the codebook of size $V$, i.e $x_i \in \{0, 1, \ldots, V-1\}$, resulting in a total of $T = \lceil \frac{H}{K} \rceil \times \lceil \frac{W}{K} \rceil$ tokens. The tokenized 1D sequence is obtained by raster-scanning the 2D tokens from top-left to bottom-right. While this approach has shown promising results in generating high-fidelity images, it suffers from scalability issues as the number of tokens $T$ scales quadratically with image resolution. For instance, increasing the image resolution from 256px to 1024px results in a 16× increase in token count, substantially increasing the computational and memory demands of attention [8, 79, 97] in decoder-only AR models. DPAR aims to address a quadratic increase in token count by dynamically merging low-information tokens into patches guided by next-token prediction entropy, using simple encoder-decoder modules, resulting in efficient AR image generation.

| Model | Params | Layers | | | Hidden | Heads |
|---|---|---|---|---|---|---|
| | | Enc. | Global T. | Dec. | | |
| DPAR-B | 120M | 1 | 8 | 3 | 768 | 12 |
| DPAR-L | 352M | 1 | 19 | 4 | 1024 | 16 |
| DPAR-XL | 789M | 1 | 30 | 5 | 1280 | 20 |
| DPAR-XXL | 1.4B | 1 | 41 | 6 | 1536 | 24 |

Table 1. **Architectural specifications of DPAR model variants.** Enc., Global T., and Dec. refer to patch encoder, global transformer, and patch decoder, layers respectively.

## 3.2. Dynamic Patchification for Efficient Autoregressive Image Generation

**DPAR** dynamically aggregates discrete 1D token sequences into a variable length *patch* sequence based on image information content for efficient generation. Our method comprises four main components (Figure 3): (1) a lightweight entropy model that computes next-token prediction entropy for tokens; (2) a patch encoder that aggregates tokens within the same patch into a patch representation of the same dimensionality as each token; (3) a decoder-only transformer that operates on patch representations for efficient autoregressive generation; and (4) a patch decoder that reconstructs individual tokens from the generated patches. We use uppercase symbols to denote sequence-level quantities and hyperparameters, and lowercase symbols to denote individual token or patch states and scalar values.

### 3.2.1. Patchification

The goal of patchification is to assign each token in the 1D sequence $I_{\text{tok}} = [x_0, \ldots, x_{T-1}]$ to a patch index such that all token indices within a patch remain contiguous. Formally, we construct a patch sequence $I_{\text{patch}} = [P_0, \ldots, P_{M-1}]$, where each patch $P_m = [s_m, \ldots, f_m]$ is a contiguous non-overlapping span of token indices starting at $s_m$ and ending at $f_m$ in the original token sequence. The number of patches $M$ varies per image and is typically smaller than the total token count $T$.

Inspired by BLT [49], we propose to use next-token prediction entropy as a measure of information content for patch formation. We train a lightweight *unconditional* GPT-style AR model with $C = \varnothing$ following Eqs. (1) and (2) and refer to this model as the *entropy model* $\mathcal{E}_\phi$ parameterized by $\phi$. We set the next-token prediction entropy for first token, $e_0 = +\infty$ to ensure it starts a new patch. The next-token prediction entropy for the token $i \in [1, T-1]$ is computed as:

$$\begin{aligned} e_i &= \mathcal{H}(x_{<i};\, \mathcal{E}_\phi) \\ &= -\sum_{c=0}^{V-1} \mathcal{E}_\phi(x_i = c \mid x_{<i}) \log \mathcal{E}_\phi(x_i = c \mid x_{<i})\,. \end{aligned} \tag{3}$$

and $\mathbf{E}^{\text{tok}} = [e_0, e_1, \ldots, e_{T-1}] = \mathcal{H}(I_{\text{tok}};\, \mathcal{E}_\phi)$ denotes the entropy values for all tokens in the image. We add a token $x_i$ to the current patch $P_m$ if $e_i \leq \mathcal{H}_{\text{Th}}$, and start a new patch $P_{m+1}$ when $e_i > \mathcal{H}_{\text{Th}}$. We further limit the patch to a maximum length of $P_{\max}$ tokens to prevent excessive aggregation that could lead to information loss, and start a new patch at the end of each row to account for discontinuities in image features at row boundaries. $\mathcal{H}_{\text{Th}}$ and $P_{\max}$ are hyperparameters chosen via ablation, and together they determine the average patch length $P_{\text{avg}} = \mathbb{E}[T/M]$ for a given dataset.

### 3.2.2. Patch Encoder: Tokens to Patches

The encoder is a lightweight module that encodes tokens within each patch into a patch representation and builds upon the BLT [49] local encoder architecture. Each patch encoder block consists of causal self-attention (SA) among tokens with 2D Rotary Positional Embedding (RoPE) [75] for spatial positional encoding:

$$\mathbf{H}^{\text{tok}} = \left[h_{x_0}, \ldots, h_{x_{T-1}}\right] = \text{SA}\left(\mathbf{Z}^{\text{tok}}\right) \tag{4}$$

where $\mathbf{Z}^{\text{tok}} = \left[z_{x_0}, \ldots, z_{x_{T-1}}\right]$, $z_{x_i}$ is the embedding of token $x_i$, and $h_{x_i}$ is its latent representation after self-attention. This is followed by cross-attention (CA) [86] with patches as query and tokens as keys/values. Let $\mathbf{Z}^{\text{patch}} = \left[z_{P_0}, \ldots, z_{P_{M-1}}\right]$ denote the initial patch queries, and let $\mathbf{H}^{\text{tok}}_{s_m:f_m} = \left[h_{x_{s_m}}, \ldots, h_{x_{f_m}}\right]$ denote the token span corresponding to patch $P_m$. Then,

$$h_{P_m} = \text{CA}\left(z_{P_m},\, \mathbf{H}^{\text{tok}}_{s_m:f_m}\right), \quad \forall\, P_m \in I_{\text{patch}} \tag{5}$$

and the resulting patch representations are $\mathbf{H}^{\text{patch}} = \left[h_{P_0}, h_{P_1}, \ldots, h_{P_{M-1}}\right]$.

### 3.2.3. Patch Transformer

The patch transformer is a decoder-only model that operates on patch representations conditioned on a class label or prompt token $C$. By processing patches instead of individual tokens, it substantially reduces the computational cost of attention while remaining the most compute-intensive component of the pipeline. Following LlamaGen [77], we adopt the LLaMA [83] architecture as our autoregressive decoder backbone and follows typical transformer design with causal self-attention and MLP layers. Given $\mathbf{H}^{\text{patch}}$, the patch transformer predicts the next-patch representations $\widehat{\mathbf{H}}^{\text{patch}}$. However, unlike standard transformers that operate on tokens and utilize 2D RoPE for positional encoding, our patch transformer employs a Dynamic RoPE mechanism (see Appendix) for encoding patches with variable token lengths.

### 3.2.4. Patch Decoder: Patches to Tokens

The decoder is a lightweight module that maps patches back to individual tokens, and is inspired by the local decoder architecture in BLT [49]. Given predicted patch states $\widehat{\mathbf{H}}^{\text{patch}} = \left[\hat{h}_{P_0}, \ldots, \hat{h}_{P_{M-1}}\right]$, each decoder block consists of a copy operation where patches are copied to their corresponding tokens followed by norm and linear projection:

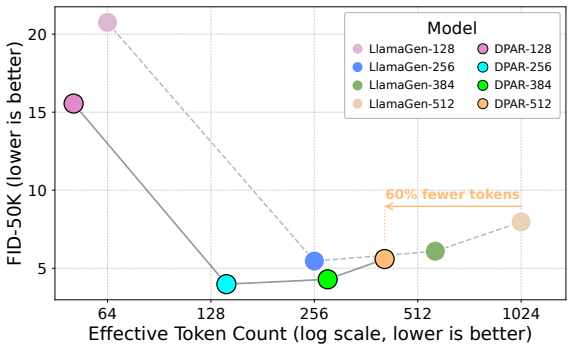



Figure 4. **DPAR scales more efficiently than LlamaGen across resolutions**. Comparison of DPAR-B and LlamaGen-B models trained at resolutions of 128, 256, 384, and 512. DPAR yields sub-quadratic growth in patch count with resolution, and achieves consistently lower FID at every resolution — demonstrating favorable quality-efficiency scaling.

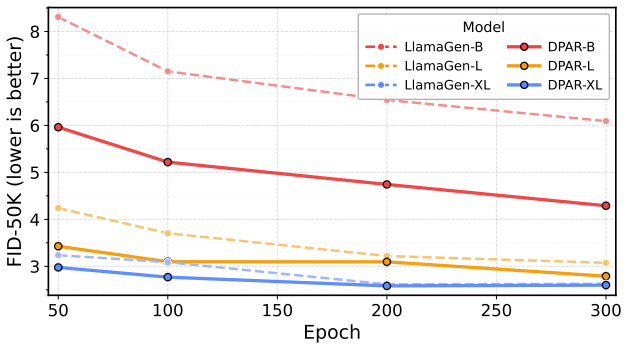



Figure 5. **DPAR converges faster than LlamaGen across model sizes.** FID vs. training epochs on ImageNet-384 for B, L, and XL variants. DPAR consistently achieves lower FID throughout training — demonstrating faster convergence and quality.

$$\tilde{h}_{x_i} = h_{x_i} + \mathrm{Linear}\big(\mathrm{Norm}(\hat{h}_{P_m})\big), \quad \forall\, i \in P_m \tag{6}$$

Let $\widetilde{\mathbf{H}}^{\mathrm{tok}} = \big[\tilde{h}_{x_0}, \ldots, \tilde{h}_{x_{T-1}}\big]$ denote the token states after this update. We then apply causal self-attention among tokens with 2D RoPE for positional encoding:

$$\bar{\mathbf{H}}^{\mathrm{tok}} = \big[\bar{h}_{x_0}, \ldots, \bar{h}_{x_{T-1}}\big] = \mathrm{SA}\big(\widetilde{\mathbf{H}}^{\mathrm{tok}}\big) \tag{7}$$

Finally, a linear output head maps $\bar{\mathbf{H}}^{\mathrm{tok}}$ to the next-token probability sequence $\widehat{\mathbf{P}}^{\mathrm{tok}}$ used for training and sampling.

# 4. Experiments

## 4.1. Implementation Details

**Tokenizer** We use LlamaGen [77] VQ tokenizer trained on ImageNet [62] with codebook size of $V = 16384$ and downsampling of $K = 16$, resulting in $T = 256$ and $T = 576$ tokens for 256×256 and 384×384 image respectively.

**Entropy Model** We train an unconditional 111M LlamaGen-B on ImageNet using the same training setup as the patch transformer. For DPAR training with 256×256 images, we use $\mathcal{H}_{\mathrm{Th}} = 7.8$ and $P_{\max} = 4$, resulting in an average patch length of $P_{\mathrm{avg}} = 1.81$ on the ImageNet training set. For 384×384 images, we use $\mathcal{H}_{\mathrm{Th}} = 7.9$ and $P_{\max} = 4$, resulting in $P_{\mathrm{avg}} = 2.06$.

**DPAR Architecture** We implement DPAR with minimal modifications on the LlamaGen architecture. Across all DPAR model variants, we use a single-layer encoder, as our ablations did not show any meaningful gains from deeper encoders (see Appendix). We start with 3 decoder layers for the Base model and increase the number of decoder layers at the same rate as the hidden dimensions (i.e., doubling the hidden dimension from 768 to 1536 increases the decoder layers from 3 to 6). Overall, the aim is to keep both the encoder and decoder shallow, so that the majority of the computational budget is allocated to the patch transformer, which operates on variable-length patches, thereby enabling efficient computation. Furthermore, for each variant, we set the number of patch transformer layers to ensure the total layer count matches that of LlamaGen models of comparable size for a fair comparison. Detailed DPAR architectural specifications are provided in Table 1.

**Training and Sampling** We train class-conditional DPAR models on ImageNet for resolutions 256×256 and 384×384. Our models are trained on 8 A100 GPUs with a global batch size of 256 for 300 epochs (≈1.5M steps) using the AdamW [40] optimizer with a learning rate of 1e-4, weight decay of 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and gradient clipping of 1.0 with PyTorch DDP. Other settings, including data augmentation, follow LlamaGen [77]. We do not use any learning rate scheduling and maintain a constant learning rate throughout training. We also use classifier-free guidance [24] with a drop probability of 0.1 during training and preprocess tokens and entropy values for each image in the dataset to avoid on-the-fly computation during training. Furthermore, since the patch transformer operates on a variable patch sequence, we implement a packed variant of LlamaGen with xformers [35] to efficiently process a batch without padding to the maximum sequence length. For sampling, we follow the sampling strategy of LlamaGen [77] and use top-k sampling with $k = 0$, top-p of 1.0, and temperature of 1.0 for all our experiments.

## 4.2. ImageNet-256 Benchmark

We evaluate the performance of our method on the class-conditional ImageNet-256 generation benchmark, a popular benchmark for image generation [51, 53, 77]. We train class-conditional DPAR model variants on ImageNet at resolutions 256×256 and 384×384 and measure model per-

| Type | Model | #Params | FID↓ | IS↑ | Prec.↑ | Rec.↑ | Steps |
|---|---|---|---|---|---|---|---|
| Diffusion | ADM [13] | 554M | 4.59 | 186.70 | 0.82 | 0.523 | 250 |
| | LDM-4 [61] | 400M | 3.60 | 247.70 | – | – | 250 |
| | DiT-XL [52] | 675M | 2.27 | 278.20 | 0.83 | 0.57 | 250 |
| | SiT-XL [42] | 675M | 2.06 | 270.30 | 0.82 | 0.59 | 250 |
| Bidirectional AR | MaskGIT-re [7] | 227M | 4.02 | 355.60 | – | – | 8 |
| | MAGVIT-v2 [93] | 307M | 1.78 | 319.40 | – | – | 64 |
| | MAR-H [36] | 943M | 1.55 | 303.70 | 0.81 | 0.62 | 256 |
| | TiTok-S-128 [95] | 287M | 1.97 | 281.80 | – | – | 64 |
| Causal AR (non-raster order / modified-training) | VAR [81] | 600M | 2.57 | 302.60 | 0.83 | 0.56 | 10 |
| | VAR [81] | 2.0B | 1.92 | 350.20 | 0.82 | 0.59 | 10 |
| | SAR-XL [39] | 893M | 2.76 | 273.80 | 0.84 | 0.55 | 256 |
| | RAR-L [94] | 461M | 1.70 | 299.50 | 0.81 | 0.60 | 256 |
| | RAR-XXL [94] | 955M | 1.50 | 306.90 | 0.80 | 0.62 | 256 |
| | RAR-XL [94] | 1.5B | 1.48 | 326.00 | 0.80 | 0.63 | 256 |
| | RandAR-L [51] | 343M | 2.55 | 288.82 | 0.81 | 0.58 | 88 |
| | RandAR-XL [51] | 775M | 2.25 | 317.77 | 0.80 | 0.60 | 88 |
| | RandAR-XL [51] | 775M | 2.22 | 314.21 | 0.80 | 0.60 | 256 |
| | RandAR-XXL [51] | 1.4B | 2.15 | 321.97 | 0.79 | 0.62 | 88 |
| Causal AR (raster order) | VQGAN-re [15] | 1.4B | 5.20 | 280.30 | – | – | 256 |
| | RQTran.-re [34] | 3.8B | 3.80 | 323.70 | – | – | 64 |
| | Open-MAGVIT2-XL [41] | 1.5B | 2.33 | 271.80 | 0.84 | 0.54 | 256 |
| Causal AR (raster order with LlamaGen tokenizer) | LlamaGen-B [77] | 111M | 5.46 | 193.61 | 0.84 | 0.46 | 256 |
| | LlamaGen-L [77] | 343M | 3.80 | 248.30 | 0.83 | 0.52 | 256 |
| | LlamaGen-XL [77] | 775M | 3.39 | 227.10 | 0.81 | 0.54 | 256 |
| | LlamaGen-384-B [77] | 111M | 6.09 | 182.53 | 0.84 | 0.42 | 576 |
| | LlamaGen-384-L [77] | 343M | 3.07 | 256.06 | 0.83 | 0.52 | 576 |
| | LlamaGen-384-XL [77] | 775M | 2.62 | 244.08 | 0.80 | 0.57 | 576 |
| | LlamaGen-384-XXL [77] | 1.4B | 2.34 | 253.91 | 0.81 | 0.60 | 576 |
| | DPAR-B (cfg=2.1) | 120M | 3.98 | 250.62 | 0.83 | 0.49 | 142 |
| | DPAR-L (cfg=1.9) | 352M | 2.93 | 269.34 | 0.81 | 0.56 | 142 |
| | DPAR-XL (cfg=2.0) | 789M | 2.67 | 281.65 | 0.82 | 0.56 | 142 |
| | DPAR-384-B (cfg=2.10) | 120M | 4.29 | 254.54 | 0.83 | 0.47 | 280 |
| | DPAR-384-L (cfg=1.90) | 352M | 2.79 | 283.84 | 0.81 | 0.55 | 280 |
| | DPAR-384-XL (cfg=1.90) | 789M | 2.60 | 285.43 | 0.81 | 0.57 | 280 |
| | DPAR-384-XXL (cfg=1.75) | 1.4B | 2.30 | 287.37 | 0.82 | 0.59 | 280 |

Table 2. **DPAR model comparisons on class-conditional ImageNet 256×256 benchmark.** We report FID [22], Inception Score(IS) [66], and precision/recall [33] and the average number of sampling steps used for generation. DPAR model outperforms prior raster-order autoregressive models with similar parameter counts, achieving significantly better FID scores. Models containing '-384' in their names are trained on $384 \times 384$ and resized to $256 \times 256$ for evaluation.

formance primarily with FID-50K [22], which computes FID between 50,000 generated samples and the ImageNet validation dataset. For 384×384 resolution, the generated images are resized to 256×256 before computing FID, as in prior work [51, 77]. We also report Inception Score (IS) [66], and precision/recall [33] for completeness. We primarily compare our model with LlamaGen, as other works either utilize different tokenizers [81], token orderings [39, 51, 94], and training paradigms, making direct comparisons unfair. We leave the extension of our method to these approaches for future work. Our results are summarized in Table 2. We observe that DPAR outperforms LlamaGen at both resolutions across all model sizes, improving the FID score by up to 29.6% on the Base model at 384×384 resolution. We further compare the convergence speed of DPAR and LlamaGen by plotting FID against training epochs in Figure 5 for models trained on resolution 384×384 due to the availability of intermediate epoch results from LlamaGen. We observe that DPAR consistently outperforms LlamaGen throughout training, demonstrating faster convergence and better final performance. Finally, we study the scaling behavior of DPAR-B across resolutions 128, 256, 384, and 512. As shown in Figure 4, DPAR attains lower FID than LlamaGen while using up to 60% fewer tokens, highlighting better quality-efficiency scaling.

### 4.3. Training FLOPs

We estimate the mean training cost by measuring FLOPs over 500 training steps and averaging the total FLOP measurements across the entire run to obtain a per-sample FLOP estimate. This ensures a consistent, profile-based estimate of the end-to-end compute required by DPAR with variable-length patches. The results are summarized in Fig. 1(b). We observe that DPAR significantly reduces training FLOPs across all model sizes compared to LlamaGen, primarily due to its packed patch transformer implementation, which

| Entropy Gating | Max Patch Len. | Row-Bound. Reset | FID ↓ |
|---|---|---|---|
| × | × | × | 3.58 |
| ✓ | × | × | 3.91 |
| ✓ | ✓ | × | 3.45 |
| ✓ | ✓ | ✓ | **3.32** |

Table 3. **Patchification strategies**. Entropy-based patchification with max patch length and row-boundary reset constraint leads to the best FID score on ImageNet 256×256 benchmark.

efficiently handles variable-length patches without padding. Notably, DPAR-XL achieves a 40.4% reduction in training FLOPs compared to LlamaGen-XL at a 384px generation resolution on ImageNet.

### 4.4. Ablation Studies

We use the DPAR-L model trained on ImageNet at 256px resolution for 50 epochs for ablations, unless specified otherwise, and compare ablation choices using FID-50K.

**Patchification Strategies** We consider three binary design choices for patchification: i) starting a new patch when the entropy exceeds a threshold $\mathcal{H}_{\text{Th}}$ (entropy gating), ii) limiting the maximum patch length to $P_{\max}$ (maximum patch length), and iii) resetting patches at row boundaries (row-boundary resets). When all three design choices are disabled, this corresponds to the static patchification scheme with fixed patch length $P_{\text{static}}$, and we set $P_{\text{static}} = 1.81$ to match the average patch length of our full method. The results are summarized in Table 3. We observe that enabling all three design choices leads to the best FID score of 3.32. Furthermore, entropy alone is insufficient to achieve optimal performance, as the absence of a limit on the maximum patch length can lead to excessive aggregation, potentially resulting in information loss.

**Effect on Entropy Threshold $\mathcal{H}_{\text{Th}}$** We study the impact of the entropy threshold $\mathcal{H}_{\text{Th}}$ on FID with a fixed maximum patch length $P_{\max} = 4$ (see Table 4). Lower thresholds produce smaller patches and increase computational cost, whereas higher thresholds produce larger patches but degrade image quality. An intermediate value of $\mathcal{H}_{\text{Th}} = 7.8$ provides the best balance, achieving the lowest FID of 3.32.

**Effect of Max Patch Length $P_{\max}$** We examine the effect of the maximum patch length $P_{\max}$ on FID with fixed entropy threshold $\mathcal{H}_{\text{Th}} = 7.8$ (see Table 4). We observe that FID first improves as we increase $P_{\max}$ from 1 to 4, achieving the best FID of 3.32 at $P_{\max} = 4$. However, further increasing $P_{\max}$ results in performance degradation, likely due to information loss from excessive aggregation.

### 4.5. Adaptive Patch Lengths at Inference

We investigate whether DPAR models trained with a fixed entropy threshold $\mathcal{H}_{\text{Th}}^{\text{train}}$ can generalize to different inference thresholds $\mathcal{H}_{\text{Th}}^{\text{inf}}$ at inference time. To evaluate this, we take our DPAR-L model trained with $\mathcal{H}_{\text{Th}}^{\text{train}} = 7.8$ and $P_{\max} = 4$ for 50 epochs and vary $\mathcal{H}_{\text{Th}}^{\text{inf}}$ during inference from 7.8 to 8.1, and compare with a static model trained with fixed patch length $P_{\text{static}} = 1.81$ in the ablation studies. The results are summarized in Table 5. We observe that as $\mathcal{H}_{\text{Th}}^{\text{inf}}$ increases from 7.8 to 8.1, DPAR FID degrades marginally from 3.32 to 3.39, while the static model's FID degrades significantly, from 3.58 to 25.59.

We argue that DPAR with dynamic patchification yields stronger global representations, as a patch must keep track of future tokens across variable-length sequences. This enables better generalization, resulting in adaptive patch sizes at inference without significant degradation in FID. We further test our hypothesis by linearly probing the global image feature obtained by averaging the patch/token representations of the last hidden layer of the DPAR-L patch transformer and the LlamaGen-L transformer. DPAR-L achieves 37.82% top-1 and 61.15% top-5 accuracy versus 32.62% and 56.61% for LlamaGen-L — a ∼5 pp. improvement in both metrics — suggesting that DPAR learns stronger global representations.

| $\mathcal{H}_{\text{Th}}$ | **7.6** | **7.7** | **7.8** | **7.9** | **8.0** |
|---|---|---|---|---|---|
| **FID-50K (↓)** | 3.36 | 3.41 | **3.32** | 3.46 | 3.43 |

(a) Varying $\mathcal{H}_{\text{Th}}$ with $P_{\max} = 4$ fixed.

| $P_{\max}$ | **1** | **2** | **4** | **6** | **8** | **16** |
|---|---|---|---|---|---|---|
| **FID-50K (↓)** | 3.73 | 3.42 | **3.32** | 3.49 | 3.41 | 3.61 |

(b) Varying $P_{\max}$ with $\mathcal{H}_{\text{Th}} = 7.8$ fixed.

Table 4. **Ablation on entropy threshold $\mathcal{H}_{\text{Th}}$ and maximum patch length $P_{\max}$.** $\mathcal{H}_{\text{Th}} = 7.8$ and $P_{\max} = 4$ jointly achieve the best FID of 3.32.

| $\mathcal{H}_{\text{Th}}^{\text{inf}}$ | **7.8** | **7.9** | **8.0** | **8.1** |
|---|---|---|---|---|
| $P_{\text{avg}}$ | 1.81 | 1.92 | 2.03 | 2.16 |
| Static | 3.58 | 7.85 | 17.91 | 25.59 |
| DPAR-L | 3.31 | 3.32 | 3.38 | 3.39 |

Table 5. **Adaptive Patch Length at Inference.** DPAR-L trained with $\mathcal{H}_{\text{Th}}^{\text{train}}$=7.8 maintains stable FID as $\mathcal{H}_{\text{Th}}^{\text{inf}}$ increases from 7.8 to 8.1, whereas the static model degrades substantially.

## 5. Conclusion

In this work, we introduced DPAR, a novel autoregressive image generation model that leverages next-token prediction entropy to dynamically aggregate tokens into variable-length patches, reducing token count and training FLOPs for efficient image generation. While this work focuses on raster-order generation, our proposed strategy is compatible with random-order methods [51, 94], and extending it to these settings as well as developing KV-Cache strategies tailored to the encoder-decoder architecture for inference speedups are promising directions for future work.

## References


[1] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 1

[2] Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14496–14506, 2023. 2

[3] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4599–4603, 2023. 2

[4] Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, 2020. 3

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

[6] 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. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[7] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 7

[8] 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. 2, 4

[9] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36:2252–2274, 2023. 2

[10] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining, 2025. 1

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1, 3

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 7

[14] Shivam Duggal, Phillip Isola, Antonio Torralba, and William T Freeman. Adaptive length image tokenization via recurrent allocation. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2024. 2, 4

[15] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 7

[16] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. 1, 3

[17] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *European conference on computer vision*, pages 396–414. Springer, 2022. 2

[18] Yuying Ge, Yixiao Ge, Ziyun Zeng, Xintao Wang, and Ying Shan. Planting a seed of vision in large language model, 2023. 1

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[20] Jakob Drachmann Havtorn, Amélie Royer, Tijmen Blankevoort, and Babak Ehteshami Bejnordi. Msvit: Dynamic mixed-scale tokenization for vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 838–848, 2023. 2, 4

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

[22] 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, 2018. 7

[23] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1, 3

[24] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 6

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 3

[26] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 1

[27] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023. 1

[28] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec

Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

[29] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1

[30] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[31] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 1

[32] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European conference on computer vision*, pages 620–640. Springer, 2022. 2

[33] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019. 7

[34] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *CVPR*, 2022. 7

[35] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022. 6

[36] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024. 1, 7

[37] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022. 2

[38] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 1

[39] Wenze Liu, Le Zhuo, Yi Xin, Sheng Xia, Peng Gao, and Xiangyu Yue. Customize your visual autoregressive recipe with set autoregressive modeling. *arXiv preprint arXiv:2410.10511*, 2024. 3, 7

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 6

[41] Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024. 7

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

[43] Xu Ma, Peize Sun, Haoyu Ma, Hao Tang, Chih-Yao Ma, Jialiang Wang, Kunpeng Li, Xiaoliang Dai, Yujun Shi, Xuan Ju, et al. Token-shuffle: Towards high-resolution image generation with autoregressive models. *arXiv preprint arXiv:2504.17789*, 2025. 2, 4

[44] Keita Miwa, Kento Sasaki, Hidehisa Arai, Tsubasa Takahashi, and Yu Yamaguchi. One-d-piece: Image tokenizer meets quality-controllable compression, 2025. 2, 4

[45] Piotr Nawrot, Szymon Tworkowski, Michał Tyrolski, Łukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. Hierarchical transformers are more efficient language models, 2022. 2

[46] Piotr Nawrot, Jan Chorowski, Adrian Lancucki, and Edoardo Maria Ponti. Efficient transformers with dynamic token pooling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 6403–6417. Association for Computational Linguistics, 2023. 2

[47] R OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5):1, 2023. 1

[48] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 1

[49] Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srinivasan Iyer. Byte latent transformer: Patches scale better than tokens, 2024. 2, 3, 5

[50] Yatian Pang, Peng Jin, Shuo Yang, Bin Lin, Bin Zhu, Zhenyu Tang, Liuhan Chen, Francis E. H. Tay, Ser-Nam Lim, Harry Yang, and Li Yuan. Next patch prediction for autoregressive visual generation, 2025. 3

[51] Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T Freeman, and Yu-Xiong Wang. Randar: Decoder-only autoregressive visual generation in random orders. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 45–55, 2025. 1, 2, 3, 6, 7, 8

[52] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 7

[53] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. 6

[54] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *article*, 2018. 1

[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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 1

[56] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 1

[57] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 1, 3

[58] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification, 2021. 2

[59] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 2

[60] 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. 1, 3

[61] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 7

[62] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. 2, 6

[63] Michael S Ryoo, A Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? arxiv 2021. *arXiv preprint arXiv:2106.11297*, 2021. 2

[64] 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. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1, 3

[65] Jonne Sälevä and Constantine Lignos. What changes when you randomly choose bpe merge operations? not much. In *Proceedings of the Fourth Workshop on Insights from Negative Results in NLP*, pages 59–66, 2023. 3

[66] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016. 7

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

[68] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. *arXiv preprint arXiv:2301.09515*, 2023. 1

[69] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1715–1725, 2016. 3

[70] Junhong Shen, Kushal Tirumala, Michihiro Yasunaga, Ishan Misra, Luke Zettlemoyer, Lili Yu, and Chunting Zhou. Cat: Content-adaptive image tokenization. *arXiv preprint arXiv:2501.03120*, 2025. 2, 3

[71] Aaditya K Singh and DJ Strouse. Tokenization counts: the impact of tokenization on arithmetic in frontier llms. *arXiv preprint arXiv:2402.14903*, 2024. 3

[72] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 3

[73] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[74] Divyansh Srivastava, Xiang Zhang, He Wen, Chenru Wen, and Zhuowen Tu. Lay-your-scene: Natural scene layout generation with diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17909–17919, 2025. 1, 3

[75] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. 5

[76] Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 331–335, 2019. 2

[77] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 1, 2, 5, 6, 7

[78] Haotian Tang, Yecheng Wu, Shang Yang, Enze Xie, Junsong Chen, Junyu Chen, Zhuoyang Zhang, Han Cai, Yao Lu, and Song Han. Hart: Efficient visual generation with hybrid autoregressive transformer. *arXiv preprint arXiv:2410.10812*, 2024. 1

[79] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022. 2, 4

[80] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2025. 1

[81] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024. 7

[82] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction, 2024. 3

[83] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 2, 5

[84] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[85] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. 2

[86] 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. 1, 5

[87] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 1

[88] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 1

[89] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 1

[90] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024. 1

[91] 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. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022. 1

[92] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, Candace Ross, Adam Polyak, Russell Howes, Vasu Sharma, Puxin Xu, Hovhannes Tamoyan, Oron Ashual, Uriel Singer, Shang-Wen Li, Susan Zhang, Richard James, Gargi Ghosh, Yaniv Taigman, Maryam Fazel-Zarandi, Asli Celikyilmaz, Luke Zettlemoyer, and Armen Aghajanyan. Scaling autoregressive multi-modal models: Pretraining and instruction tuning, 2023. 1

[93] Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion–tokenizer is key to visual generation. In *ICLR*, 2024. 7

[94] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. *arXiv preprint arXiv:2411.00776*, 2024. 1, 3, 7, 8

[95] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024. 4, 7

[96] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation, 2024. 2

[97] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12104–12113, 2022. 2, 4

[98] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024. 1

[99] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 1