MergeVQ: A Unified Framework for Visual Generation and Representation with Disentangled Token Merging and Quantization

Siyuan Li[1,3*] **Luyuan Zhang**[2*] **Zedong Wang**[4] **Juanxi Tian**[3] **Cheng Tan**[1,3]
Zicheng Liu[1,3] **Chang Yu**[3] **Qingsong Xie**[5†] **Haonan Lu**[5] **Haoqian Wang**[2] **Zhen Lei**[6,7,8†]
[1]Zhejiang University [2]Tsinghua University [3]Westlake University
[4]The Hong Kong University of Science and Technology [5]OPPO AI Center
[6]CAIR, HKISI-CAS [7]MAIS CASIA [8]University of Chinese Academy of Sciences

Abstract

Masked Image Modeling (MIM) with Vector Quantization (VQ) has achieved great success in both self-supervised pre-training and image generation. However, most existing methods struggle to address the trade-off in shared latent space for generation quality vs. representation learning and efficiency. To push the limits of this paradigm, we propose MergeVQ, which incorporates token merging techniques into VQ-based generative models to bridge the gap between image generation and visual representation learning in a unified architecture. During pre-training, MergeVQ decouples top-k semantics from latent space with the token merge module after self-attention blocks in the encoder for subsequent Look-up Free Quantization (LFQ) and global alignment and recovers their fine-grained details through cross-attention in the decoder for reconstruction. As for second-stage generation, we introduce MergeAR, which performs KV Cache compression for efficient raster-order prediction. Extensive experiments on ImageNet verify that MergeVQ as an AR generative model achieves competitive performance in both visual representation learning and image generation tasks while maintaining favorable token efficiency and inference speed. Code and model will be available at https://apexgen-x.github.io/MergeVQ.

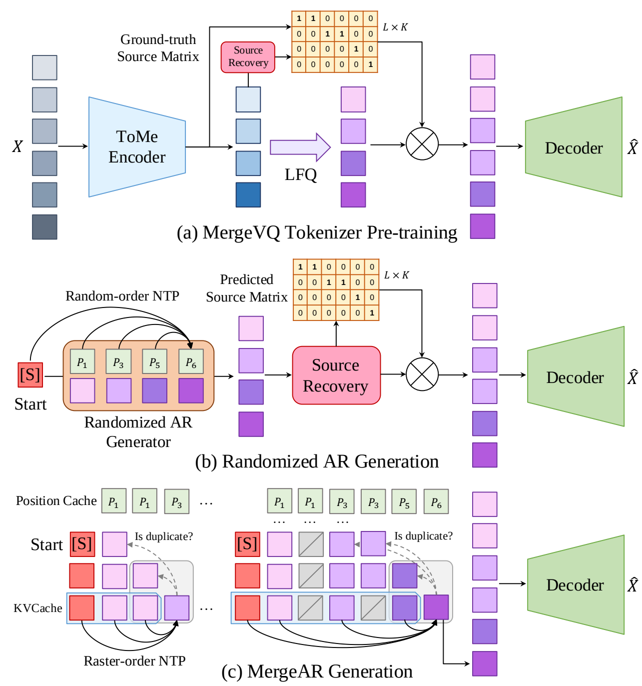

Figure 1. **MergeVQ learning paradigms**. **(a)** MergeVQ Tokenizer extracts K semantic tokens with decoupled positional information (retained in source matrix) by ToMe [6] while quantizing spatial details by LFQ [47, 70], which will be recovered and reconstructed correspondingly. **(b)** MergeVQ with random-order Generator [49] generates K discrete tokens with associated position instructions while trained Source Prediction and decoder restore position details. **(c)** MergeAR Generator predicts L tokens efficiently in raster-order by Next-token Prediction (NTP) [55] with KV Cache compression to remove the redundancy.

1. Introduction

Vector Quantization (VQ) [58] has garnered increasing attention for its ability to encode continuous visual signals into discrete tokens, enabling autoregressive (AR) models to process visual modalities. Since VQGAN [20], most visual AR generative models have adopted a two-stage design: first encode signals into discrete latent space for pretraining, then generate them with an autoregressive Transformer. Besides generation, BEiT [3] proposed Masked Image Modeling (MIM) based on the VQ framework, achieving successful latent-based pretraining [35, 37] and thus attracting growing interest in unifying visual representation learning and generation tasks in a *shared latent space* [78].

However, recent studies [43, 79] have shown that visual generation and representation capabilities often lack consistency [69] under VQ-based learning framework, *i.e.*, improvements in one task may not necessarily benefit the others. This inconsistency is conjectured to arise from the competing objectives for identical embedding space: **represen-**

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

tation learning tasks emphasize inter-class discrimination to maximize high-level semantics, while generative tasks prioritize the reconstruction of details. In addition, training obstacles brought by VQ itself further limit the optimization process. For example, the gradient approximation in canonical VQ (*e.g.*, VQGAN) sets an optimization bottleneck for the first-stage training. Moreover, the quantization of embedding space inevitably strips away fine-grained spatial information, which requires the models to reconstruct images with the loss of details and thus affects both the representation learning and generation.

As such, efforts have been made to extract rich semantic features from visual signals for quantization to improve the representation capacity of generative models [60, 81]. However, these coarse-grained semantics often sacrifice detailed information, making it difficult to support high-quality image reconstruction and generation, resulting in significant performance degradation. In this paper, we argue that representation learning and generation are not completely conflicting but with intrinsic complementarity. The crux lies in exploiting such complementarity while minimizing the information loss, which requires specific designs. To achieve this, we propose to *decouple coarse-grained semantics from latent space during training and recover them for reconstruction to meet the different needs while minimizing the information loss and overhead*. By leveraging token merging techniques [6], the encoder compresses latent space into K semantic tokens while preserving the fine-grained spatial information as positions within a source matrix, as illustrated in Figure 1. During reconstruction, the latent fine-grained details can be restored with this source matrix while the K compressed tokens serve as high-level semantics for global alignment [9, 78]. Built upon this intuition, we propose MergeVQ, which employs token merging and Look-up Free Quantization (LFQ) for spatial and channel compression. Extensive experiments show that MergeVQ as an AR generative model achieves competitive performance in both image generation and visual representation learning with favorable efficiency. Our contributions can be summarized as:

- We present a fresh learning paradigm that integrates token merging into a VQ-based AR generation framework, where high-level semantics are decoupled from patients in the first-stage training and can be restored with source matrix for details reconstruction, thus effectively reducing information loss while bridging the gap between representation learning and generation in a unified model.
- We offer two schemes for MergeVQ's second-stage generation. (i) We propose MergeAR, which performs KV-Cache compression for efficient raster-order prediction. (ii) With the source recovery module, existing random-order generators can also be directly used for generation.
- Experiments show MergeVQ's competitive performance in both visual representation learning and image generation, with favorable token efficiency and inference speed.

2. Related Work

2.1. Auto-regressive Image Generation

Tokenizer with Vector Quantization. Vector quantization, introduced by VQ-VAE [58] and enhanced by VQGAN [20] with adversarial loss and Transformer integration, faces three key challenges in traditional *cluster-based VQ* approach: (i) Gradient approximation issues: Straight-through estimator creates imprecise encoder gradients, addressed by MAGVIT-v2 [69] and OpenMAGVIT2 [44] through extended training. (ii) Inefficient codebook learning: Commitment loss causes uneven gradient distribution and codebook collapse. Solutions include RegVQ [74] and Kepler Codebook [40]'s priors, and BEiT.v2 [51] and ViT-VQGAN [66]'s EMA with normalization. (iii) The discrete bottleneck. Quantization eliminates fine-grained details, hampering generation and reconstruction. RQ [31] employs multi-level quantization to reduce this information loss. *Look-up Free Quantization* quantizes along channels, reducing overhead while improving codebook usage. Attempts like FSQ [47], MAGVIT-v2 [69], Open-MAGVIT2 [44], [30, 62, 77] demonstrate results that match or exceed vanilla VQ. Another research direction accelerates inference using *Adaptive-Length Quantization* to compress tokens. These methods utilize cross-attention [19, 72], attention-based token extraction [28], or token grouping [18], reducing token number for faster generation.

Auto-regressive Generation. VQGAN introduced autoregressive visual generation following the raster-order Next Token Prediction (NTP) in GPT [52, 53]. Subsequently, numerous works have built upon this raster generation paradigm, including LlamaGen [55] and Open-MAGVIT2 [44]. Concurrently, a line of research has explored parallel decoding methods to accelerate generation, exemplified by MaskGiT [10], which employs non-sequential generation to enhance generation speed. Recently, several studies have investigated randomized generation techniques, where token positions are predicted prior to token embeddings, or learnable positional encodings are utilized for position prediction, as demonstrated in works such as RandAR [49] and RAR [71].

2.2. Unifying Representation and Generation

Since BEiT [3] first combined Masked Modeling with VQ for pre-training, research unifying representation and generation within a latent space has gained increasing interest [33]. These studies, typically conducted within *cluster-based VQ* frameworks, fall into two categories: *(i) Using Pre-training Techniques in Quantized Space.* MQ-VAE [28] quantizes semantic tokens by masking important ones for reconstruction. MAGE performs Masked Modeling directly in latent space during second-stage generation training, while BEiT abandons second-stage generation, using Masked Modeling as the second stage itself. *(ii) Using representative tasks to enhance generation quality.* DiGIT [81] extracts semantic tokens from pre-trained models for rep-

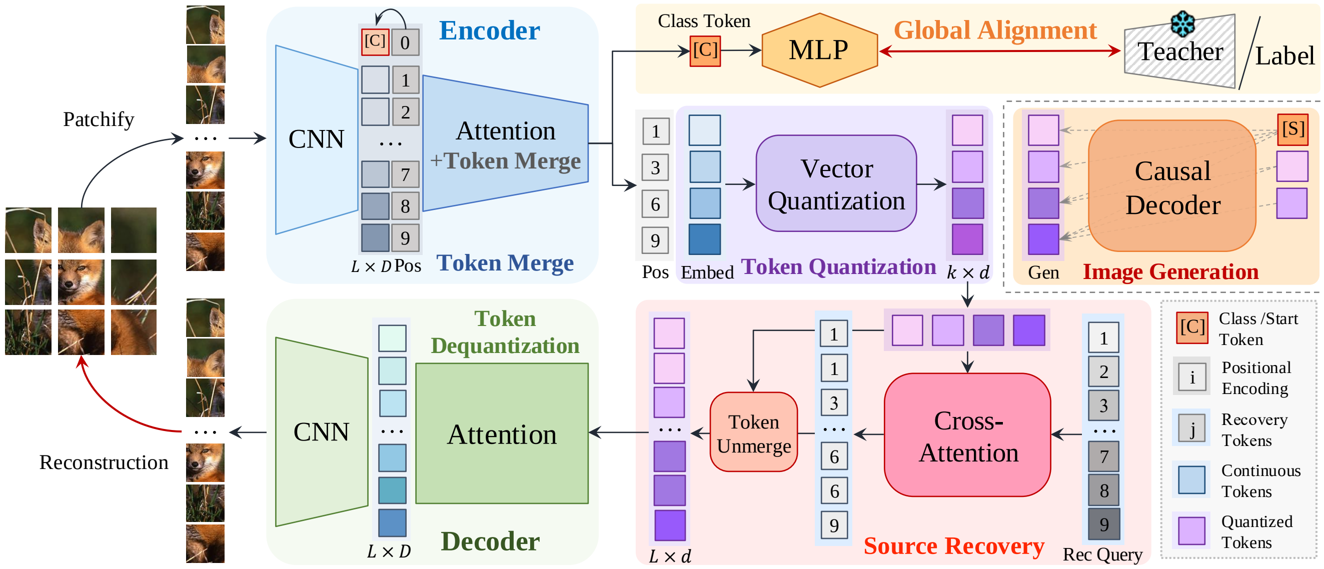

Figure 2. **Overview of MergeVQ framework**, which contains two stages and three groups of subtasks (Sec. 3.1). **(a)** As for representation learning (Sec. 3.2), K semantic tokens are extracted by the encoder with self-attention and token merging [6], which can be aligned globally with a pre-trained teacher while learning contextual information by predicting the source matrix. **(b)** As for reconstruction (Sec. 3.3), taking K merged and quantized tokens as the input, the positional information can be retained by the Source Recovery module, and then high-quality details will be reconstructed. **(c)** As for generation (Sec. 4), we utilize the source matrix to construct a causal mask for training and leverage the KV cache to prune repeated tokens during inference for efficient generation.

resentation learning while using a finely crafted decoder for generation. VQ-KD [60] employs a pre-trained teacher model to guide token reconstruction. REPA [73] proposes that representation alignment can significantly improve the training efficiency and generation quality of diffusion models. Some approaches align visual and text codebooks via CLIP-inspired methods [75]. SPAE [68] uses hierarchical codebooks to align visual representations with frozen LLMs, while V2L Tokenizer [79] employs global and local tokenizers for multimodal alignment.

2.3. Token Compression in Transformer

Token compression techniques have become essential for enhancing efficiency in Transformer-based architectures, particularly in ViTs and LLMs. ToMe and its advancements [5, 7, 8, 11] employ lightweight BSM techniques to achieve pruning-like efficiency, improving ViT throughput with minimal performance degradation. However, BSM-based methods often suffer from information token loss due to heuristic merging. For ViT encoders, token merging methods like k-means [46] and spectral clustering [4] have been explored to mitigate this issue, offering more controlled outputs. Yet, these techniques introduce complex iteration schemes that may conflict with reducing model complexity in ViT layers. For decoders, recent KV cache compression strategies like FastGen [23], SnapKV [39], and H_2O [76] optimize memory usage and inference speed through selective token retention and key-value pairs compression. While these methods effectively boost LLM inference efficiency, they remain inapplicable in training phase.

3. MergeVQ Learning Paradigm

3.1. MergeVQ Framework

In this section, we introduce the MergeVQ framework based on vector quantization and define key notations.

Token Merge Encoding: Given an image $X \in \mathbb{R}^{H\times W\times 3}$, we employ a two-stage downsampling encoder $\mathcal{E}$. First, a CNN layer $\mathcal{E}_1$ extracts features, producing a feature map $Z \in \mathbb{R}^{\frac{H}{f}\times\frac{W}{f}\times d}$, where f is the downsampling factor and d denotes the channel number. The feature Z is then flattened into L-length token sequence $Z_L \in \mathbb{R}^{L\times d}$ as:

$$Z_L = \mathcal{E}_1(X). \tag{1}$$

We then employ attention with the merging operation, denoted as $\mathcal{E}_2$, for second-stage extraction. During this process, we obtain a shorter sequence $Z_K \in \mathbb{R}^{K\times d}$ along with its source matrix $S \in \mathbb{R}^{K\times L}$ that preserves the spatial relationships of the sequence. This process is expressed as:

$$Z_K, S = \mathcal{E}_2(Z_L). \tag{2}$$

As such, the entire encoding process can be represented as:

$$Z_K, S = \mathcal{E}(X). \tag{3}$$

To ensure that Z_K owns rich semantics, we concurrently impose global alignment on Z_K as discussed in Sec. 3.2.

Quantization: We employ *LFQ* as MergeVQ's quantization module to minimize the loss of details. Specifically, the codebook is reduced to an integer set and could be denoted as: $\mathcal{C} = \times_{i=1}^{N}\{-1, 1\}, \quad |C| = 2^N$. Thus, the quantization can be summarized as follows: $z_{Ki} = \text{sign}(z_{Ki}) = -1 \cdot \mathbb{I}(z_{Ki} < 0) + \mathbb{I}(z_{Ki} > 0)$, where z_{Ki} denotes i-th vector in K semantic tokens Z_K. Then the index of the quantized feature z_{mi} is formulated as $\text{Index}(z_{Ki}) = \sum_{j=1}^{N} 2^{k-1} \cdot \mathbb{I}(z_{Kij} > 0)$. Finally, we obtain the quantized semantic tokens, denoted as Z_{Kq}:

$$Z_{Kq} = \mathcal{Q}(Z_K, \mathcal{C}). \tag{4}$$

Token Recovery and Reconstruction: we first perform token-level reconstruction with the recovery module $\mathcal{R}(\cdot,\cdot)$ and source matrix S, which yields a new L-length $\hat{Z}_L$ as:

$$\hat{Z}_L = \mathcal{R}(Z_{qK}, S). \tag{5}$$

This sequence is then fed into the decoder $\mathcal{D}$ for pixel-level reconstruction, which could be described as:

$$\hat{X} = \mathcal{D}(\hat{Z}_L). \tag{6}$$

3.2. Harmonize Reconstruction and Representation

Inspired by research on Masked Image Modeling in representation learning, we employ Token Merge to reduce the number of tokens and leverage Source Recovery to restore all tokens for contextual modeling, seamlessly integrating representation learning into our framework. To further enhance the representation capability, we impose global alignment constraints on the compressed visual tokens. Our framework is detailed and illustrated in Figure 2.

Attention with Token Merging: After encoding the input image X into Z_L as in Eq. (1), attention mechanisms further extract features. We employ the Token Merge Attention module proposed by ToMe [6] for token merging. Specifically, in each attention block, the top $2r$ tokens by attention score are merged into r tokens. With n attention blocks, the final token count K satisfies $K = L - nr$. The implementation details are provided in Appendix A.

$$Z_\text{K} = \text{ToMeAttention}(Z). \tag{7}$$

During this process, a source matrix S would be maintained to record the origin of each token. Starting with an original token sequence of length L, after N layers of ToMe Transformer Blocks, the sequence is reduced to length K. The source matrix owns the size of $K \times L$, where $S_{ij} \in \{0, 1\}$. The details are available in the Appendix.

$$Z_K, S = \mathcal{E}_2(X). \tag{8}$$

The source matrix preserves the positional and spatial information of the image during the encoding process.

Source Recovery Model: We introduce the Source Recovery Model to facilitate contextual modeling of tokens. In particular, for the K semantic tokens and the Source matrix that records their positional information, our aim is to design a module capable of recovering these tokens without relying on their source. To achieve this, we incorporate

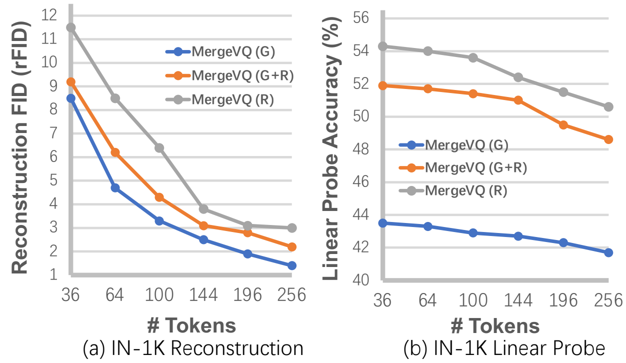

Figure 3. **Analysis of kept tokens in reconstruction and representation learning**. Three MergeVQ tokenizers are trained with 128 resolution for 30 epochs on ImageNet-1K. They keep 256, 144, and 36 tokens with ToMe [6] in the encoder during training. In inference, we evaluate rFID and linear probing top-1 accuracy with diverse merge ratios to show the trade-off between generation and representation. Please view Sec. 5 and Appendix B for details.

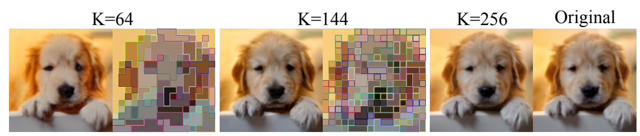

Figure 4. **Visualization of MergeVQ (G+R) reconstruction**. With the kept tokens varying from 64 to 256, clustering maps of ToMe Attention indicate that MergeVQ can extract discriminative semantic tokens while recovering contextual positions and details.

a lightweight Transformer decoder with L learnable positional embeddings Q to interact with the K tokens, as:

$$Q_r = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) Q^T. \tag{9}$$

Subsequently, we predict the source matrix $\hat{S}$, thereby enabling contextual modeling of tokens as:

$$\hat{S} = \arg\max(\text{softmax}\left(Q_r K^T\right)). \tag{10}$$

Overall, we employ cross-entropy to measure the difference between $\hat{S}$ and S to optimize this Source Recovery Model. The learning object source loss $\mathcal{L}_\text{src}$ is thus formulated as:

$$\mathcal{L}_\text{src} = -\sum_{i,j} S_{i,j} \log(\hat{S}_{i,j}) + (1 - S_{i,j}) \log(1 - \hat{S}_{i,j}). \tag{11}$$

Global Alignment: To further enhance the representation capability of semantic tokens, we perform global alignment on semantic tokens using the Self Distillation approach proposed by DINO [9]. We uniformly sample an image X from the training set, apply random augmentations to generate views u and v, and feed them into the DINOv2 encoder and MergeVQ. The predicted category distributions from the CLS tokens, $v_t = P_{\theta'}^{[CLS]}(v)$ and $u_t = P_{\theta'}^{[CLS]}(u)$, are aligned by minimizing the cross-entropy between them. The alignment loss is:

$$\mathcal{L}_{[CLS]} = -P_{\theta'}^{[CLS]}(v)^T \log P_{\theta'}^{[CLS]}(u). \tag{12}$$

3.3. Recovery and Reconstruction

Token Recovery For Reconstruction: Before reconstructing the image, we first perform token-level recovery to restore contextual information. Then, we reconstruct the image at the pixel level based on the recovered visual tokens, aiming to restore the fine details lost due to channel quantization. The Token Recovery process is achieved through the source matrix S. This process is denoted as:

$$\hat{Z}_L = \mathcal{R}(Z_{Kq}, S). \tag{13}$$

Specifically, we utilize the positional information recorded in S to expand Z_K back into a sequence of length L. For example, if the i-th row of S satisfies $S(i, j_1) = 1$ and $S(i, j_2) = 1$, we recover the L-length sequence $\hat{Z}_L$ such that $\hat{Z}_{Lj_1} = \hat{Z}_{Lj_2} = \hat{Z}_{Ki}$. This process can be implemented through matrix multiplication as:

$$\hat{Z}_L = [\hat{z}_l]_{l=1}^{L} = Z_{Kq}S = \left[\sum_{i=1}^{K} z_{qi} \times s_{il}\right]_{i=1}^{L}. \tag{14}$$

During training, we obtain the ground-truth source matrix through the encoder, allowing straightforward token recovery. In the inference phase, the predicted source matrix $\hat{S}$ is obtained by the Source Recovery Model in Sec. 2.2, enabling token recovery. Subsequently, we apply the decoder $\mathcal{D}$ to reconstruct the recovered $\hat{Z}_L$ as:

$$\hat{X} = \mathcal{D}(\hat{Z}_{\mathrm{L}}). \tag{15}$$

Hybrid Model with Weight Initialization: As for network architectures for generative tasks, feature extraction is typically performed using CNN, while pure Transformer-based backbones are relatively rare. However, in representation learning, Transformer-based architectures are prevalent. To bridge this gap, we employ a hybrid model that leverages the ToMe Attention mechanism of Transformers as a dynamic downsampling method. This approach not only enhances attention efficiency but preserves strong representation capabilities and flexibility. To further exploit these advantages, we integrate a pre-initialized Transformer backbone into our VQ architecture. The specific network architecture is detailed in Appendix A.1 and illustrated in Figure 2.

Adaptive Merge Ratios for Diverse Tasks: Unlike existing adaptive-length quantization methods [38, 72], our MergeVQ framework uses variable merge ratios r during training instead of fixed sequence lengths. The ToMe module provides flexibility for different tasks through adjustable merge ratios. Our experiments show that representation learning and reconstruction tasks benefit from different merge ratio settings. As Figure 3 illustrates, representation tasks (Sec. 3.2) favor larger merge ratios [26, 29], which help extract semantic features while preserving contextual information. Based on these findings, we offer three MergeVQ variants: Representation (R) version for enhanced generalization, Generation and Representation (R+G) version balancing both tasks, and Generation (G) version optimized for high-performance generalization and reconstruction. Meanwhile, we provide another trick sample merge ratios to expose the model to varying token counts, which could enhance the generalization and robustness of both stage-1 and stage-2 training. In practice, we retained three versions of semantic token counts: 256, 144, and 36, corresponding to pure Generation (G), Generation and Representation (R+G), and pure Representation (R), respectively. During training, we determine the corresponding merge ratio r by sampling the number of tokens retained, focusing on a range around the target token count for each version. We use exponential distribution sampling for the G and R versions, and discrete Gaussian distribution sampling for the G+R version. The sample details are available in Appendix A.

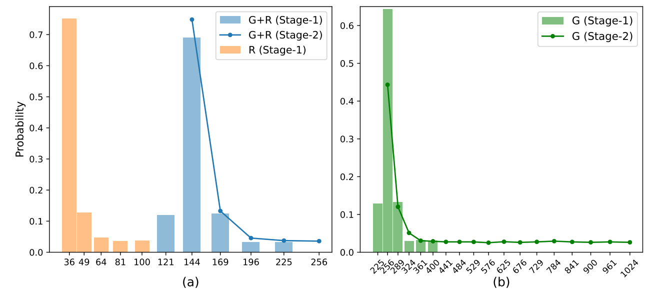

Figure 5. **Distribution of merge ratios sampling in training**. (a) With 256 tokens in total, MergeVQ (R) and (G+R) sample the square number as kept token numbers in $[36, 100]$ and $[121, 225]$ with exponential and Gaussian distributions for stage-1 training, while the G+R version sampling from $[144, 256]$ for stage-2 training. (b) With 1024 tokens in total, MergeVQ (G) samples the square kept number in $[225, 400]$ and $[256, 1024]$ with Gaussian and exponential distributions in both stage-1 and stage-2 training.

4. MergeVQ for Efficient Generation

4.1. MergeAR with KV Cache Compression

We introduce MergeAR for efficient autoregressive generation based on our Merged Tokens framework. Unlike common approaches, MergeAR leverages intrinsic token sparsity to accelerate generation. It compares each new token against the existing sequence, pruning similar tokens while preserving essential information through strategic copying. A position-recording system ensures output coherence.

During training, we first sample a merge ratio r as Appendix. A, which determines the number of merged visual tokens and results in K discretized tokens along with their ground-truth source matrix S. To regulate the level of sparsity, we introduce a Merge Instruction Token M, which serves as an indicator of merging extent. Using the source S and target Z_K, we construct a causal mask to guide the training process. Concretely, suppose the causal mask

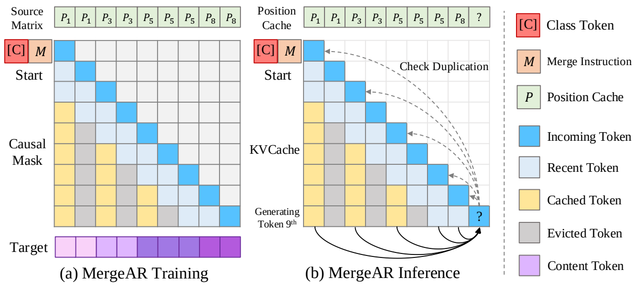

Figure 6. **Illustration of MergeAR pipeline**. **(a)** MergeAR training with the source matrix and target sequences from MergeVQ tokenizer to build a causal mask with duplicated tokens masked out, taking a class token and a merge instruction token as the starting conditions. **(b)** MergeAR inference that generates L tokens in the raster order with duplicated tokens detected and removed.

$M \in \mathbb{R}^{L\times L}$, $M(i,j) \in \{0,1\}$ are denoted as:

$$M(i,j) = 1, \text{when } S(i,j) = 1 \text{ and } 1 \notin \bigcup_{k=1}^{i-1} S(k,j). \quad (16)$$

As such, efficient generation can be achieved while enabling the model to effectively capture contextual information. In the inference phase, we construct the KV cache using the same method as for the causal mask. When generating the t-th token, we compare it with the previously generated tokens. If the generated token is a duplicate, it is pruned and not stored in the cache. Otherwise, if a new token is generated, it is added to the KV cache. View Figure 6 for details.

4.2. Randomized Auto-regressive with Source Recovery

Concurrently, Randomized AR techniques like RandAR [49] introduce positional encoding prediction, whose objective $p_\theta(\mathbf{x}|\mathbf{P})$ could be formulated as:

$$\prod_{n=1}^{N} p_\theta\left(x_n^{\pi(n)} \mid P_1^{\pi(1)}, x_1^{\pi(1)}, \ldots, x_{n-1}^{\pi(n-1)}, P_n^{\pi(n)}\right), \quad (17)$$

where $x_i^{\pi(i)}$ is the i-th token in this randomly shuffled N-length sequence, and $\pi(i)$ denotes its original position in raster order. We then insert a positional instruction token $P_i^{\pi(i)}$ before each image token $x_i^{\pi(i)}$.

MergeVQ can be also implemented by this RandsAR generative framework, where the K quantized tokens Z_{Kq} obtained in the first stage serve as target image tokens, and source matrix is used as the corresponding positional information. The model is trained and optimized through RandAR paradigm. After generating K generated tokens, we invoke the source recovery model and decoder, as described in Eq.(6) and Eq.(5), to recover all tokens for generation.

5. Experiments

5.1. Implementation Details

Visual Tokenizer Setup. We offer three MergeVQ versions for visual representation learning and generation: MergeVQ (G) for pure generation, MergeVQ (G+R) for both generation and representation, and MergeVQ (R) for representation learning only. As detailed in Appendix A.1, we present three architectures of these versions with the latent embedding dimension of 512, whose encoders have 63M, 62M, and 86M parameters. As discussed in Sec. 3.2, we apply the hybrid model that contains 4 and 5 hierarchical stages of ResNet blocks [25] with 12-layer of ToMe Attention blocks [6] at the last stage for the encoder networks in MergeVQ (G) and MergeVQ (R+G), as well as LFQ layer [70] with the dimension of 18. The corresponding decoder shares a similar architecture as encoders without ToMe modules. For fair comparisons, MergeVQ (R) adopts ViT-B [17] with random initialization as encoder but still adopts an identical decoder and LFQ as MergeVQ (G+R). As for the token number after quantization, the raw output number of three versions are 1024, 256, and 256, and we merge them to 256, 144, and 36 tokens during training and inference. All versions are trained by AdamW optimizer [42] with (β_1, β_2) of $(0.5, 0.9)$, a default learning rate of $1e-4$, and a total batch size of 256 for 200∼300 epochs on ImageNet-1K without annotations. As for reconstruction, models are trained in 256×256 resolutions with a combination of ℓ_i reconstruction loss, GAN loss, perceptual loss, entropy penalty, commitment loss, and LeCAM regularization as MAGVITv2, combined with our proposed source recovery loss $\mathcal{L}_{\text{src}}$ and alignment loss $\mathcal{L}_{[CLS]}$.

Visual Generator Setup. Following LlamaGen [55] and the concurrent work RandAR [49][1], we conduct three versions of AR generators with MergeVQ tokenizers: MergeVQ with vanilla LlamaGen for classical raster-order generation, MergeVQ with MergeAR (built upon LlamaGen) for efficient generation, and MergeVQ with RandAR for random-order generation. As for the third version, it requires the pre-trained Source Recovery module to predict the source matrix with the generated sequences as mentioned in Sec. 4.2, which can be a 2-layer standard Transformer decoder with 512 embedding dimensions at 7M parameters. We adopt LlamaGen-L as the generator architecture, which is a 24-layer Transformer decoder [53] in LLaMA-based architecture [57] and trained by AdamW optimizer [42] with a weight decay of 0.05, a basic learning rate of 4×10^4, and a batch size of 1024 for 300 epochs. View Appendix A.2 for more details.

5.2. Self-supervised Pre-training

We evaluated self-supervised pre-trained models by linear probing (Lin.) [26] and end-to-end fine-tuning (FT) [3] protocols on ImageNet-1K. Table 1 shows that MergeVQ variants substantially outperform prior models like BYOL, MoCoV3, and DINOv2 in performance and efficiency, notably with fewer tokens achieving superior accuracy. MergeVQ (R), which focuses on representation learning, achieves impressive results with only 36 tokens. With fewer tokens

[1]More studies of MergeAR and combination of MergeVQ with concurrent AR works [49, 71] will be updated in the arXiv preprint.

Table 1. **Comparsion of self-supervised pre-training on ImageNet-1K**. Top-1 accuracy of linear probing (Lin.) and fully fune-tuning (FT) results are reported. ‡ denotes using the multi-crop augmentation or additional data. We summarize the target for alignment (Align.) and reconstruction (Rec.), the pre-training epochs, the encoder architecture type, and the number of learnable parameters (#Param) of the encoder and latent tokens (#Tokens), where MIM and TMM denote Masked Image Modeling and Token-merge Modeling.

Support Tasks	Method	Date	Align. Target	Rec. Target	Epochs	Encoder Type	#Param	#Tokens	Accuracy↑ Lin.	FT
Contrastive Pre-training	BYOL [24]	NeurIPS'2020	MSE	✗	800	R50-W2	94M	7×7	75.6	–
	MoCoV3 [12]	ICCV'2021	InfoNCE	✗	300	ViT-B	86M	196	76.7	83.2
	DINO‡ [9]	ICCV'2021	CE	✗	300	ViT-B	86M	196	78.2	83.6
	DINOv2‡ [48]	TMLR'2024	CE	✗	1000	ViT-B	86M	196	**84.5**	**85.7**
MIM Pre-training	BEiT [3]	ICLR'2022	✗	DALLE	800	ViT-B	86M	196	56.7	83.2
	iBOT‡ [78]	ICLR'2022	CE	EMA	800	ViT-B	86M	196	76.0	84.0
	MAE [26]	CVPR'2022	✗	RGB	1600	ViT-B	86M	196	68.0	83.6
	SimMIM [64]	CVPR'2022	✗	RGB	800	ViT-B	86M	196	67.9	83.8
	CAE [13]	IJCV'2023	✗	DALLE	1600	ViT-B	86M	196	70.4	83.6
	PeCo [16]	AAAI'2023	✗	VQVAE	800	ViT-B	86M	196	–	**84.5**
	A^2MIM [32]	ICML'2023	✗	RGB	800	ViT-B	86M	196	68.8	84.2
	I-JEPA [1]	CVPR'2023	✗	RGB	600	ViT-B	86M	196	**72.9**	–
	EVA-02 [21]	CVPR'2024	✗	EVA-CLIP	300	ViT-B	86M	196	–	84.0
Generative	ViT-VQGAN [66]	ICLR'2022	✗	RGB	100	VIM-Base	650M	1024	65.1	–
	MaskGIT [10]	CVPR'2022	✗	RGB	200	BERT	227M	256	57.4	–
	LlamaGen [55]	NeurIPS'2024	✗	RGB	40	CNN	72M	1024	47.6	–
	Titok-B [72]	NeurIPS'2024	✗	VQGAN	200	Titok-B	86M	64	53.9	–
	REPA [73]	ICLR'2025	DINOv2	Velocity	100	SiT-L/2	458M	1024	**71.1**	–
Generative & Pre-training	MAGE-C [35]	CVPR'2023	InfoNCE	VQGAN	1600	ViT-B	24+86M	196	78.2	82.9
	DiGIT [81]	NeurIPS'2024	DINOv2	RGB	200	ViT	219M	256	71.7	–
	MergeVQ (G+R)	**Ours**	DINOv2	RGB+TMM	200	Hybrid	63M	144	77.9	82.0
	MergeVQ (R)	**Ours**	DINOv2	RGB+TMM	300	ViT-B	86M	36	**79.8**	**84.2**

than DINOv2 (196), MergeVQ (R) achieves 79.8% Lin. Accuracy and 84.2% FT accuracy, leveraging a flexible and discriminative latent space for both efficiency and performance. MergeVQ (G+R) performs slightly lower than MergeVQ (R) due to its inclusion of generation alongside representation learning, highlighting the trade-off between tasks, which require more tokens, and pretraining, which benefits from coarse-grained latent. Despite this, MergeVQ (G+R) remains competitive, reaching 77.9% in Lin. and 82.3% in FT, demonstrating competitive results while handling both generative and representation objectives.

5.3. Image Generation

Reconstruction. Table 2 compares the reconstruction performance of VQ-based tokenizers on 256 × 256 ImageNet-1K. MergeVQ (G+R) achieves an effective balance between reconstruction and token efficiency (nearly a 100%-utilized LFQ codebook with dynamic token lengths), leading to an rFID of 1.48. This outperforms methods that use larger codebooks and more tokens, such as RQ-VAE and LlamaGen. MergeVQ (G), applying the same codebook but with 256 tokens, hits an even lower rFID of 0.54, excelling in reconstruction quality. Overall, MergeVQ variants show high performance by optimizing codebook and token usage. While MergeVQ (G+R) slightly sacrifices rFID for handling both generation and representation, it remains competitive, highlighting the trade-off between these objectives.

Class Conditional Generation. As shown in Table 3, MergeVQ (G+R) and MergeVQ (G) stand out as competitive models. MergeVQ (G+R) uses 144 latent tokens and our MergeAR generator and achieves a gFID of 3.27 and an IS of 253.8 without CFG. When CFG and RandAR generator are applied, it improves to a gFID of 2.63 and an IS of 279.5, surpassing many auto-regressive models. On the other hand, MergeVQ (G) with a MergeAR generator, which uses 256 tokens and 1024 steps, demonstrates even

Table 2. **Comparison of reconstruction on 256×256 ImageNet-1K** with reconstruction FID (rFID) of VQ tokenizers. We sum up the types, sizes, and dims of the codebook with its usage ratio. Ratio and #Tokens denote the downsampling rate and token number.

Method	VQ Codebook Type	Size	Dim	Usage↑	Ratio	#Tokens ↓	rFID ↓
Taming-VQGAN [20]	Cluster	2^{10}	256	49%	16	16^2	7.94
SD-VQGAN [54]	Cluster	2^{10}	4	–	16	16^2	5.15
RQ-VAE [31]	Cluster	2^{14}	256	–	16	16^2	3.20
MaskGIT [10]	Cluster	2^{10}	256	–	16	16^2	2.28
LlamaGen [55]	Cluster	2^{14}	8	97%	16	16^2	2.19
TiTok-L-32 [72]	Cluster	2^{12}	16	–	–	32	2.21
TiTok-B-64 [72]	Cluster	2^{12}	12	–	–	64	1.70
VQGAN-LC [80]	CLIP	10^5	8	99%	16	16^2	2.62
VQ-KD [60]	DINO	2^{13}	32	100%	16	16^2	3.41
MAGVIT-v2 [69]	LFQ	2^{18}	1	100%	16	16^2	1.16
OpenMAGVIT2 [44]	LFQ	2^{18}	1	100%	16	16^2	1.17
MaskBiT [62]	LFQ	2^{14}	1	100%	16	16^2	1.37
MergeVQ (R)	LFQ	2^{18}	1	86%	16	144	4.67
MergeVQ (G+R)	LFQ	2^{18}	1	99%	16	144	1.48
MergeVQ (G+R)	LFQ	2^{18}	1	99%	16	256	**1.12**
ViT-VQGAN [66]	Cluster	2^{13}	8	96%	8	16^2	1.28
OmiTokenizer [59]	Cluster	2^{13}	8	–	8	16^2	1.11
LlamaGen [55]	Cluster	2^{14}	8	97%	8	16^2	0.59
TiTok-S-128 [72]	Cluster	2^{12}	16	–	–	128	1.71
VQGAN-LC [80]	CLIP	10^5	8	99%	8	16^2	1.29
MergeVQ (G)	LFQ	2^{18}	1	100%	8	256	1.06
MergeVQ (G)	LFQ	2^{18}	1	100%	8	1024	**0.54**

Table 3. **System comparsion of class-conditional generation on 256×256 ImageNet-1K**. Generation Fréchet inception distance (gFID) and inception score (IS) are reported with ADM [15]. "# P" means the parameter number, step means sampling steps, and ‡ denotes training tokenizers on OpenImages. Note that "-cfg" or "-re" denotes using classifier-free guidance or rejection sampling, and "-384" denotes for generating images at 384×384 resolutions and resize back to 256×256 for evaluation.

Type	Tokenizer	Generator	# P.	Step	gFID↓	IS↑
Diff.	VAE‡	LDM-4 [54]	400M	250	3.60	247.7
		UViT-L/2 [2]	287M	250	3.40	219.9
		UViT-H/2 [2]	501M	250	2.29	263.9
		DiT-XL/2 [50]	675M	250	2.27	278.2
		MDTv2-XL/2 [22]	676M	250	**1.58**	**314.7**
		SiT-XL [45]	675M	250	2.06	270.3
		DiMR-XL/2R [41]	505M	250	1.70	289.0
Mask.	VQGAN	MaskGIT [10]	177M	8	6.18	182.1
	TiTok-B-64‡	MaskGIT-ViT [10]	177M	8	2.48	262.5
	TiTok-S-128‡	MaskGIT-UViT-L [2]	287M	64	1.97	281.8
	MAR	MAR-B-cfg [36]	208M	100	2.31	281.7
	MAR	MAR-L-cfg [36]	479M	100	**1.78**	**296.0**
VAR	VAR‡	VAR-d16 [56]	310M	10	3.30	274.4
		VAR-d20 [56]	600M	10	2.57	302.6
		VAR-d24 [56]	1.0B	10	**2.09**	**312.9**
AR (raster)	VQGAN	GPT2 [53]	1.4B	256	15.78	74.3
	VQGAN	GPT2-re [53]	1.4B	256	5.20	280.3
	VIT-VQGAN	VIM-L [66]	1.7B	1024	4.17	175.1
	ViT-VQGAN	VIM-L-re [66]	1.7B	1024	3.04	227.4
	RQ-VAE	RQ-Trans.-re [31]	3.8B	64	3.80	323.7
	MAGVIT-v2	MAGVIT-cfg [67]	307M	256	1.78	319.4
	LlamaGen	LlamaGen-L [55]	343M	256	3.80	248.3
	LlamaGen	LlamaGen-L-384 [55]	343M	576	3.07	256.1
	LlamaGen	LlamaGen-XL [55]	775M	256	3.39	227.1
	LlamaGen	LlamaGen-XL-384 [55]	775M	576	2.62	244.1
	OpenMAGVIT2	OpenMAGVIT2-B[44]	343M	256	3.08	258.3
	OpenMAGVIT2	Open-MAGVIT2-L[44]	804M	256	2.51	271.7
	MaskBit	LlamaGen-cfg [55]	305M	256	**1.52**	**328.6**
AR & PT	VQGAN	MAGE-L [35]	230M	20	6.93	195.8
	VQGAN	DiGIT [81]	732M	256	3.39	206.0
	MergeVQ (G+R)	LlamaGen-L [55]	343M	256	3.28	251.6
	MergeVQ (G+R)	**MergeAR (Ours)**	343M	256	3.25	253.8
	MergeVQ (G)	**MergeAR (Ours)**	343M	1024	**3.05**	**260.9**
AR (random)	LlamaGen	RandAR-L-cfg [49]	343M	88	2.55	288.8
	LlamaGen	RandAR-L-cfg [49]	775M	88	2.25	317.8
	MergeVQ (G+R)	RandAR-L-cfg [49]	343M	64	2.63	279.5
	MergeVQ (G)	RandAR-L-cfg [49]	343M	88	**2.24**	**320.4**

better performance, with a gFID of 3.05 and an IS of 260.9 without CFG, and achieving a gFID of 2.24 and IS of 320.4 with CFG and RandAR generator. Despite using fewer tokens than several computationally expensive models (*e.g.*, large VQGAN and ViT-VQGAN), MergeVQ variants excel in class-conditional image generation by balancing generative quality and computational efficiency, setting a new benchmark for models in this domain. By using fewer tokens while maintaining high image quality, the MergeVQ models show that it is possible to achieve state-of-the-art results with a more streamlined and efficient approach compared to some of the most advanced diffusion and GAN-based models. This makes the proposed MergeVQ particularly promising for real-world applications where efficiency and generation quality are both crucial.

Table 4. **Ablation of three versions of MergeVQ tokenizers** with the number of kept tokens during training for pre-training (linear probing Acc.) and reconstruction (rFID) tasks on ImageNet-1k.

	G	G+R				R
#Tokens	rFID (↓)	rFID (↓)	# Step (↓)	Acc. (↑)	FLOPs (↓)	Acc. (↑)
256	**1.41**	2.15	64	48.6	76.2G	–
196	1.89	2.53	49	49.5	74.8G	51.2
144	2.03	3.07	36	51.0	73.4G	52.5
100	2.96	4.62	25	51.2	72.4G	53.9
64	4.74	6.51	16	51.8	71.5G	54.1
36	–	8.94	9	52.1	71.7G	**54.3**

Table 5. **Ablation of main modules for MergeVQ generation** with reconstruction (rFID) and generation (gFID) evaluation.

Version	$\mathcal{R}$	$\mathcal{G}$	rFID	gFID	# Token
(G+R)	Ground-truth $\mathcal{S}$	✗	1.48	–	144
(G+R)	2-layer Cross-Attention	✗	1.71	–	144
(G+R)+RandAR	2-layer Cross-Attention	LlamaGen-L	1.71	2.63	144
(G+R)+LlamaGen	✗	LlamaGen-L	–	3.28	256
(G)+LlamaGen	✗	LlamaGen-L	–	3.14	1024
(G)+MergeAR	✗	LlamaGen-L	–	3.05	1024

5.4. Ablation Study

We conduct ablation studies of key modules on ImageNet-1K. As for the tokenizer, Table 4 shows that MergeVQ (G) and MergeVQ (R) could achieve the best reconstruction and pre-training performances with 256 tokens (*i.e.*, adaptive downsampling instead of convolution projection) and 36 tokens (*i.e.*, a small number of semantic tokens for better global alignment). MergeVQ (G+R) could well balance the reconstruction performance with the pre-training task and efficiency (fewer steps and FLOPs) by 144 tokens. As for generation, we validate the three designed versions in Sec. 4. In Table 5, the Source Recovery module is essential to restore the positional information for MergeVQ (G+R) with RandAR, which could approximate the ground-truth $\mathcal{S}$ recover positional information for the generator. As shown in Table 3 and Table 5, the proposed KV Cache compression in MergeAR could be more useful when the generated sequence is redundant, which improves the vanilla LlamaGen by 0.09 *vs.* 0.03 gFID with the MergeVQ (G) and MergeVQ (G+R).

6. Conclusion

This paper presents MergeVQ, a unified framework that bridges the competing objectives of representation learning and image generation. It incorporates flexible token merging-based designs to balance compact latent space and fine-grained generation. In addition, we propose MergeAR, a second-stage KVCache compressive technique that yields considerable speed gains while retaining high-quality image generation ability. Experiments demonstrate that MergeVQ achieves competitive performance across tasks, outperforming existing methods in both representation learning and image generation. The results highlight MergeVQ's versatility and robustness, showcasing its ability to adapt to both generative and discriminative demands.

Acknowledgement

This work was supported in part by Chinese National Natural Science Foundation Projects U23B2054, 62276254, 62306313, the Beijing Science and Technology Plan Project Z231100005923033, Beijing Natural Science Foundation L221013, and the InnoHK program. This work was done when Juanxi Tian interned at Westlake University. We also thank OPPO AI Center and AI Station of Westlake University for the support of GPUs.

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, pages 15619–15629, 2023. 7

[2] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, pages 22669–22679, 2023. 8

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022. 1, 2, 6, 7

[4] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *Proceedings of the 37th International Conference on Machine Learning*, pages 874–883. PMLR, 2020. 3

[5] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4599–4603, 2023. 3

[6] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *ICLR*, 2023. 1, 2, 3, 4, 6

[7] Maxim Bonnaerens and Joni Dambre. Learned thresholds token merging and pruning for vision transformers, 2023. 3

[8] Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. PuMer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12890–12903, Toronto, Canada, 2023. Association for Computational Linguistics. 3

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2, 4, 7

[10] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 2, 7, 8

[11] Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. Diffrate : Differentiable compression rate for efficient vision transformers, 2023. 3

[12] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9640–9649, 2021. 7

[13] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022. 7, 2

[14] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 1

[15] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021. 8

[16] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Peco: Perceptual codebook for bert pre-training of vision transformers. *arXiv preprint arXiv:2111.12710*, 2021. 7

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 6, 1

[18] Zhihao Duan, Ming Lu, Jack Ma, Yuning Huang, Zhan Ma, and Fengqing Zhu. Qarv: Quantization-aware resnet vae for lossy image compression. *IEEE TPAMI*, 2023. 2

[19] Shivam Duggal, Phillip Isola, Antonio Torralba, and William T Freeman. Adaptive length image tokenization via recurrent allocation. *arXiv preprint arXiv:2411.02393*, 2024. 2

[20] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 1, 2, 7

[21] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. In *CVPR*, 2024. 7

[22] Shanghua Gao, Pan Zhou, Mingg-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *ICCV*, pages 23107–23116, 2023. 8

[23] Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive kv cache compression for llms, 2024. 3

[24] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 7

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[26] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 5, 6, 7, 2

[27] Jonathan Ho. Classifier-free diffusion guidance. *ArXiv*, abs/2207.12598, 2022. 2

[28] Mengqi Huang, Zhendong Mao, Quan Wang, and Yongdong Zhang. Not all image regions matter: Masked vector quantization for autoregressive image generation. In *CVPR*, pages 2002–2011, 2023. 2

[29] Ziyu Jiang, Yinpeng Chen, Mengchen Liu, Dongdong Chen, Xiyang Dai, Lu Yuan, Zicheng Liu, and Zhangyang Wang. Layer grafted pre-training: Bridging contrastive learning and masked image modeling for label-efficient representations. In *ICLR*, 2023. 5

[30] Ahmed Khalil, Robert Piechocki, and Raul Santos-Rodriguez. Ll-vq-vae: Learnable lattice vector-quantization for efficient representations. *arXiv preprint arXiv:2310.09382*, 2023. 2

[31] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *CVPR*, pages 11523–11532, 2022. 2, 7, 8

[32] Siyuan Li, Di Wu, Fang Wu, Zelin Zang, Kai Wang, Lei Shang, Baigui Sun, Haoyang Li, and Stan.Z.Li. Architecture-agnostic masked image modeling - from vit back to cnn. In *ICML*, 2023. 7, 2

[33] Siyuan Li, Luyuan Zhang, Zedong Wang, Di Wu, Lirong Wu, Zicheng Liu, Jun Xia, Cheng Tan, Yang Liu, Baigui Sun, et al. Masked modeling for self-supervised representation learning on vision and beyond. *arXiv preprint arXiv:2401.00897*, 2023. 2

[34] Siyuan Li, Zedong Wang, Zicheng Liu, Cheng Tan, Haitao Lin, Di Wu, Zhiyuan Chen, Jiangbin Zheng, and Stan Z. Li. Moganet: Multi-order gated aggregation network. In *ICLR*, 2024. 2

[35] Tianhong Li, Huiwen Chang, Shlok Kumar Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *CVPR*, 2023. 1, 7, 8

[36] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *NeurIPS*, 2024. 8

[37] Xiaotong Li, Yixiao Ge, Kun Yi, Zixuan Hu, Ying Shan, and Ling yu Duan. mc-beit: Multi-choice discretization for image bert pre-training. In *ECCV*, 2022. 1

[38] Xiang Li, Hao Chen, Kai Qiu, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens. *arXiv preprint arXiv:2410.01756*, 2024. 5

[39] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation, 2024. 3

[40] Junrong Lian, Ziyue Dong, Pengxu Wei, Wei Ke, Chang Liu, Qixiang Ye, Xiangyang Ji, and Liang Lin. Kepler codebook. In *ICML*, 2024. 2

[41] Qihao Liu, Zhanpeng Zeng, Ju He, Qihang Yu, Xiaohui Shen, and Liang-Chieh Chen. Alleviating distortion in image generation via multi-resolution diffusion models. In *NeurIPS*, 2024. 8

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[43] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. In *CVPR*, pages 26429–26445, 2024. 1

[44] 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. 2, 7, 8, 1

[45] 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. In *ECCV*, 2024. 8

[46] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 12–21, 2023. 3

[47] Fabian Mentzer, David C. Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. In *ICLR*, 2024. 1, 2

[48] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024. 7

[49] 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 *CVPR*, 2025. 1, 2, 6, 8

[50] William S. Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4172–4182, 2023. 8

[51] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *ArXiv*, abs/2208.06366, 2022. 2

[52] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. 2

[53] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. 2, 6, 8

[54] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685, 2022. 7, 8

[55] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. In *NeurIPS*, 2024. 1, 2, 6, 7, 8

[56] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *NeurIPS*, 2024. 8, 1

[57] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, 2023. 6

[58] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *ArXiv*, 2017. 1, 2

[59] Junke Wang, Yi Jiang, Zehuan Yuan, Binyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *arXiv preprint arXiv:2406.09399*, 2024. 7

[60] Luting Wang, Yang Zhao, Zijian Zhang, Jiashi Feng, Si Liu, and Bingyi Kang. Image understanding makes for a good tokenizer for image generation. In *NeurIPS*, 2024. 2, 3, 7

[61] Yuqing Wang, Shuhuai Ren, Zhijie Lin, Yujin Han, Haoyuan Guo, Zhenheng Yang, Difan Zou, Jiashi Feng, and Xihui Liu. Parallelized autoregressive visual generation. In *CVPR*, 2025. 2

[62] Mark Weber, Lijun Yu, Qihang Yu, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. Maskbit: Embedding-free image generation via bit tokens. *TMLR*, 2024. 2, 7

[63] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In-So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *CVPR*, 2023. 2

[64] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022. 7, 2

[65] Zhiqiu Xu, Yanjie Chen, Kirill Vishniakov, Yida Yin, Zhiqiang Shen, Trevor Darrell, Lingjie Liu, and Zhuang Liu. Initializing models with larger ones. *ArXiv*, 2023. 1

[66] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 2, 7, 8

[67] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *CVPR*, pages 10459–10469, 2023. 8

[68] Lijun Yu, Yong Cheng, Zhiruo Wang, Vivek Kumar, Wolfgang Macherey, Yanping Huang, David A. Ross, Irfan Essa, Yonatan Bisk, Ming Yang, Kevin P. Murphy, Alexander G. Hauptmann, and Lu Jiang. Spae: Semantic pyramid autoencoder for multimodal generation with frozen llms. In *ArXiv*, 2023. 3

[69] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. In *ICLR*, 2023. 1, 2, 7

[70] Lijun Yu, Jose Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, 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. 1, 6, 2

[71] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. *ArXiv*, abs/2411.00776, 2024. 2, 6

[72] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *NeurIPS*, 2024. 2, 5, 7

[73] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025. 3, 7

[74] Jiahui Zhang, Fangneng Zhan, Christian Theobalt, and Shijian Lu. Regularized vector quantization for tokenized image synthesis. In *CVPR*, pages 18467–18476, 2023. 2

[75] Qian Zhang, Xiangzi Dai, Ninghua Yang, Xiang An, Ziyong Feng, and Xingyu Ren. Var-clip: Text-to-image generator with visual auto-regressive modeling. *arXiv preprint arXiv:2408.01181*, 2024. 3

[76] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2o: Heavy-hitter oracle for efficient generative inference of large language models, 2023. 3

[77] Yue Zhao, Yuanjun Xiong, and Philipp Krahenbuhl. Image and video tokenization with binary spherical quantization. *ArXiv*, 2024. 2

[78] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 1, 2, 7

[79] Lei Zhu, Fangyun Wei, and Yanye Lu. Beyond text: Frozen large language models in visual signal comprehension. In *CVPR*, pages 27037–27047, 2024. 1, 3

[80] Lei Zhu, Fangyun Wei, Yanye Lu, and Dong Chen. Scaling the codebook size of vqgan to 100,000 with a utilization rate of 99%. In *NeurIPS*, 2024. 7

[81] Yongxin Zhu, Bocheng Li, Hang Zhang, Xin Li, Linli Xu, and Lidong Bing. Stabilize the latent space for image autoregressive modeling: A unified perspective. In *NeurIPS*, 2024. 2, 7, 8

MergeVQ: A Unified Framework for Visual Generation and Representation with Disentangled Token Merging and Quantization

Supplementary Material

A. Implementation Details

A.1. Stage 1: MergeVQ Tokenizer

Tokenizer Network. MergeVQ introduces hybrid encoders with self-attention blocks [17] using ToMe modules [6], built after the bottom of the pure convolution tokenizer (using Residual blocks) proposed in MAGVITv2 [70]. We provide three versions of MergeVQ tokenizers, where the G and G+R versions use the hybrid encoders while the R version uses the vanilla ViT-B [17]. The specific network configurations, experimental settings, and training details are thoroughly described in Table A1. The corresponding decoder shares a similar architecture as encoders except for using ToMe modules. MergeVQ (G+R) and (G) versions initialize the parameters in the Transformer encoder with DINOv2 pre-trained model (*i.e.*, DINOv2-Base) by weight selection [65], while MergeVQ (R) adopts ViT-B [17] without pre-training as the encoder. Following the setup of OpenMAGVIT2 [44] codebase, we also remove the gradient penalty loss and replace StyleGAN with PatchGAN as the discriminator (not employing DINO discriminator as VAR [56] in the current version). During training, we apply the reconstruction loss, the GAN loss, the perceptual loss, and the commitment loss, combined with the proposed source recovery loss $\mathcal{L}_{\text{src}}$ as Eq. (11) and the alignment loss $\mathcal{L}_{[CLS]}$ as Eq. (12).

Source Recovery Model. The network details of the Source Recovery module in MergeVQ are shown in Table A2, where we utilize two cross-attention blocks to predict the source matrix based on the quantized tokens. As for implementation, we utilize the standard Transformer decoder to compute from the K quantized tokens (as KV) and L learnable recovery queries (as the query position embeddings) similar to Maskformer [14]. As for MergeVQ with Randomized AR generators, we further fine-tuned this module with the generator obtained after stage-2 training. Despite the fact that Source Recovery can be optimized during the stage-1 training (regarded as the contextual representation learning task), the additional fine-tuning could further enhance its robustness and generalization abilities for the generation task. As for MergeAR, it does not require the assistance of the Source Recovery module, which achieves speed-up by the proposed KV Cache compress.

Token Merge Module. Token Merge Module follows the design principles of ToMe [6]. It reduces the number of tokens to improve efficiency while maintaining accuracy. Unlike token pruning, which drops tokens, Token Merge combines similar tokens into one representation, preserving more information and reducing accuracy loss, making it a practical, lightweight approach for both inference and training. Specifically, the token merging process consists of the following four steps:

- Tokens are evenly divided into two groups, A and B, based on their odd or even positions.
- Each token in A is paired with most similar token in B.
- The r most similar pairs are selected for merging.
- The features of tokens in these pairs are averaged to create a single representation.

Token similarity is determined using the keys (K) from the self-attention mechanism, with metrics like cosine similarity or dot product to measure similarity between tokens in A and B. Since merged tokens represent multiple originals, attention computation is affected. To address this, the softmax attention scores are adjusted by adding $\log s$, where s

Table A1. Configuration of the network, weights of loss functions, and training settings for the three versions of MergeVQ tokenizers on ImageNet-1k. Note that the network designs are specified for the encoder, and the reported FLOPs are calculated for the encoder and decoder with ToMe [6] on 256×256 resolutions.

Settings	G	G+R	R
Base channels	64	64	768
CNN Stage number	4	5	–
Channel multiplier	[1, 2, 4, 8]	[1, 1, 2, 4, 8]	[1]
Residual Blocks	[4, 4, 4, 4]	[4, 4, 4, 4, 4]	–
Attention Blocks	[0, 0, 0, 12]	[0, 0, 0, 0, 12]	[12]
Downsampling ratio	[1, 1/2, 1/4, 1/8]	[1, 1/2, 1/4, 1/8, 1/16]	[1/16]
Vocabulary size		2^{18}	
Keep token number	256	144	36
Discriminator loss		0.8	
Perceptual loss		0.7	
LeCam regularization		0.01	
L2 reconstruction		1.0	
Commitment loss		0.25	
LFQ Entropy loss		0.1	
Source recovery loss	0.5	0.5	1.0
Alignement loss	0.1	1.0	1.0
Optimizer		AdamW	
(β_1, β_2)		(0.5, 0.9)	
Weight decay		0.0	
Training epochs	200	200	300
Base learning rate		1e-4	
Batch size		256	
LR scheduler	Step	Step	Cosine
Gradient clipping	–	–	5.0
EMA decay		0.999	
#Param. of Encoder	62.3M	62.7M	86.6M
FLOPs of Encoder	97.5G	46.4G	9.5G
#Param. of Decoder	82.8M	83.4M	83.4M
FLOPs of Decoder	169.2G	65.6G	65.6G

Table A2. Configuration of generators and Source Recovery model in MergeVQ or MergeAR for image generation on ImageNet-1k.

Settings	LlamaGen-L	RandAR-L	Source Recovery
Base channels	1024	1024	384
Depth	24	24	2
Attention heads	16	16	8
FFN dimension	4096	4096	1536
Dropout	0.1	0.1	0
Mask schedule	Arccos	Arccos	–
Label smoothing	0.1	0.1	–
# Parameter	343M	343M	7M
Optimizer	AdamW	AdamW	AdamW
(β_1, β_2)	(0.9, 0.99)	(0.9, 0.95)	(0.9, 0.95)
Weight decay	5e-2	5e-2	1e-2
Training epochs	300	300	5 (optional)
Base learning rate	4×10^{-4}	4×10^{-4}	1×10^{-4}
Batch size	1024	1024	256
LR scheduler	Step	Step	Step
Gradient clipping	1.0	1.0	–

is the token size, ensuring merged tokens have the correct influence and maintain consistency in representation.

$$A = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}} + \log s\right), \tag{18}$$

where A denotes the attention weight matrix, Q denotes the query matrix, derived from the input tokens, K denotes the key matrix, also derived from the input tokens, $\log s$ denotes the size adjustment term, where s represents the size of the token, indicating the number of original patches it represents after merging In practice, two types of merging schedules are provided: (1) **Linearly Decreasing Schedule**. The number of merged tokens linearly decreases as the layer depth increases. (2) **Square Decreasing Schedule**. The number of merged tokens decreases as the layer depth increases in the squared schedule. These strategies allow flexibility in balancing computational efficiency and model performance. We choose the square decreasing schedule.

A.2. Stage 2: MergeVQ Generation

We conducted raster-order and random-order autoregressive (AR) generation experiments based on LlamaGen [55] (modified by OpenMAGVIT2 [44]) and RandAR [49]. Using the LlaMA-based architecture, we adopted 2D RoPE, SwiGLU, and RMSNorm, which have been shown to be effective in previous works and thoroughly described in Table A2. The class embedding, indexed from a set of learnable embeddings, serves as the starting token. As for MergeAR, we also insert a Merge Instruction token, which is a learnable embedding token with a given merge number. For MergeVQ with RandAR [49], the classifier-free guidance (CFG) [27] with a linear sampling schedule is adopted as randomized AR variants [61, 71], where the optimal CFG weight is determined through a sweep with a step size of 0.1 across all methods.

A.3. Merge Ratio Sampling Strategy

Although our proposed MergeVQ framework can target certain tasks (representation learning or generation) by choosing a certain merge ratio, it can also benefit from a wide range of merge ratios, a kind of data augmentation that enhances the generation abilities with dynamic merge ratios. During training, we determine the corresponding merge ratio r by sampling the number of tokens retained, focusing on a range around the target token count for each version. For the versions with 256 and 36 semantic tokens, we use a discrete exponential distribution to sample the varying token counts as follows:

$$P(T = k) = (1 - \exp(-\lambda))\exp(-\lambda k), \tag{19}$$

where T represents the variation in the number of tokens with the index $k \geq 0$. As for the G and R versions, the number of retained tokens is $K = (16 - T)^2$ and $(6 + T)^2$. As for the (R+G)-version in Figure 5, we use a discrete Gaussian distribution for sampling.

$$P(T = k) = \frac{\exp(-\frac{(k-\mu)^2}{2\sigma^2})}{Z}, \quad k \in \mathbb{Z}, \tag{20}$$

where retained semantic tokens in the training are $(12+T)^2$.

A.4. Evaluation of Representation Learning

As for the linear probing protocol, we follow MAE variants [13, 26] to evaluate the linear classification performance in the latent token space and intermediate features of trained SiT models. Specifically, we train a parameter-free batch normalization layer and a linear layer for 90 epochs by AdamW optimizer with a batch size of 1024, the Cosine annealing learning rate scheduler, where the initial learning rate is set to 1×10^{-3}. As for the fine-tuning protocol, we follow SimMIM variants [32, 64] to fully fine-tune the pre-trained encoder for 100 epochs with AdamW optimizer and a batch size of 1024, which requires advanced augmentations and training strategies for modern architectures [34, 63].

B. More Experiment Results

We evaluate the reconstruction of MergeVQ (G) and MergeVQ (G+R) tokenizers at different merging ratios A1. The specific results can be seen in the figure, where we compare our experimental results with those of MAGVIT2 [70]. We also visualize the generation results of MergeVQ variants in Figure A2. It can be seen that as the merge ratio decreases, the reconstruction quality progressively improves. The G+R Version also achieves competitive results with 144 tokens.

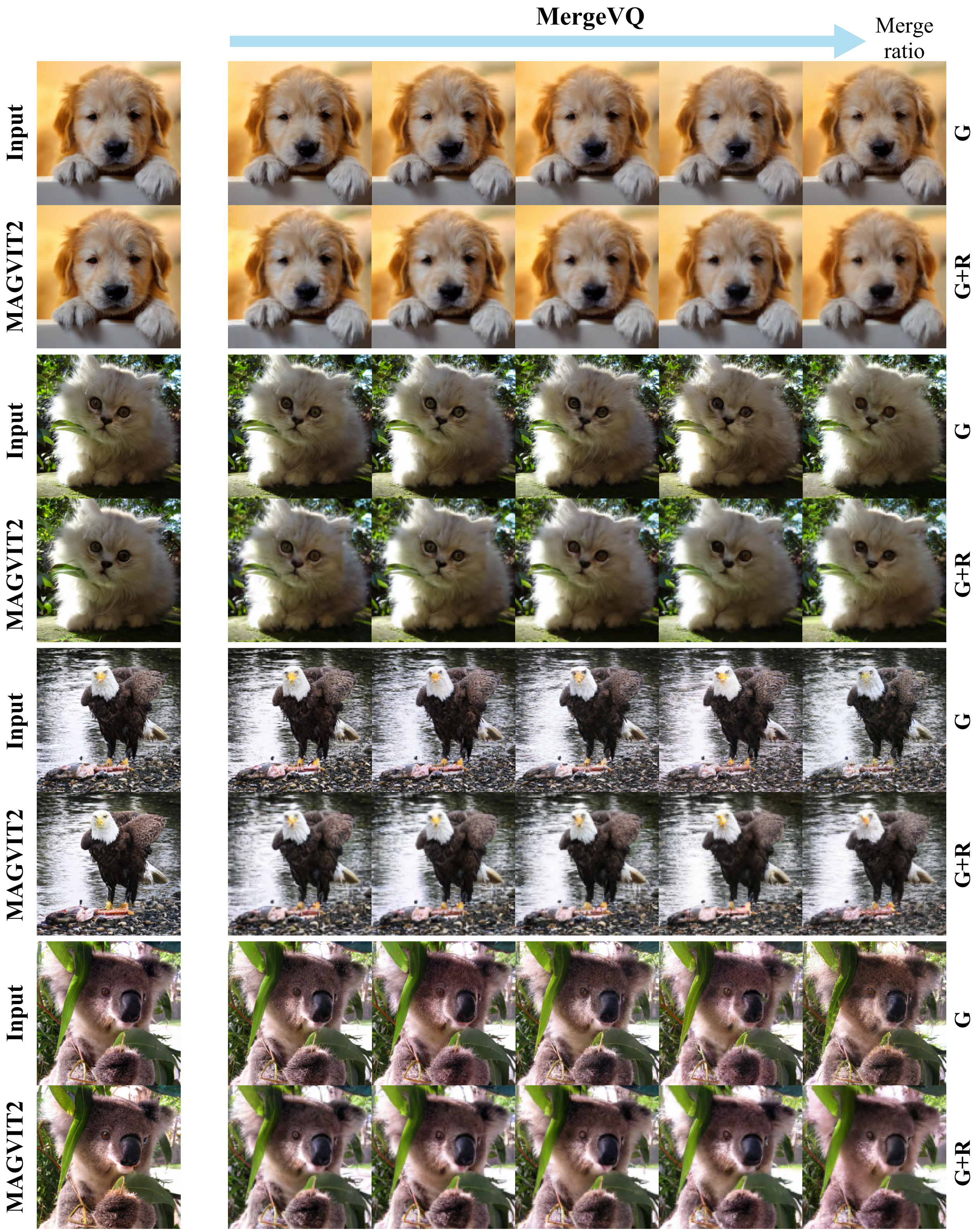

Figure A1. **Visualization of tokenizer reconstruction on ImageNet-1k.** We conducted reconstruction experiments with our G version using 1024, 576, 400, 256, and 144 tokens and with our G+R version using 256, 196, 144, 100, 64, and 36 tokens. The reconstruction results are shown in the figure. As the number of retained tokens increases, the reconstruction becomes more realistic.

Figure A2. **Visualization of class conditional generation** with MergeVQ variants on ImageNet-1k. The G version performs generation on 256 tokens, and the G+R version performs generation on 144 tokens.