# MaMe: Matrix-Based Token Merging

**Anonymous authors**
Paper under double-blind review

## Abstract

We introduce MaMe, a training-free, differentiable token merging method that relies entirely on matrix operations to accelerate vision transformers. When applied to pre-trained models, MaMe doubles ViT-B@224 throughput with a mere 2% drop in accuracy. For training from scratch, a ViT-T model with MaMe achieves 1.94x throughput with a 1.3% accuracy drop. As a downsampling layer in Swin architectures, MaMe reduces FLOPs by 2.4x for Swin-S backbones, achieving 47.0% mIoU on ADE20K semantic segmentation. In SigLIP2-B@512 zero-shot classification, MaMe provides 1.3× acceleration with negligible performance degradation (78.02 vs. 78.37). For multimodal reasoning, MaMe accelerates LLaVA-v1.5-7B inference by 36% on MME with minimal degradation (31.40 vs. 32.76). In video tasks, MaMe accelerates VideoMAE-L by 48.5% on Kinetics-400 with a 0.84% accuracy loss. Collectively, these results demonstrate MaMe's effectiveness in accelerating transformer-based vision and multimodal models.

## 1 Introduction

Vision Transformers (ViTs) (Dosovitskiy et al., 2021) have revolutionized computer vision by adopting the transformer architecture from natural language models (Vaswani et al., 2017). ViT's self-attention mechanism effectively captures long-range dependencies between image patches (i.e., "tokens"). However, the complexity of self-attention is quadratic $\mathcal{O}(N^2)$, where N represents the number of tokens. For applications requiring dense token representations, such as high-resolution images, this quadratic complexity presents a significant challenge, limiting the deployment of large-scale ViT models on resource-limited devices or in real-time applications.

To address the $\mathcal{O}(N^2)$ computational challenge, a straightforward yet effective strategy is to reduce the number of tokens $N$ involved in the process. The primary strategies that have emerged include token pruning, token merging, and hybrid methods that integrate both. Groundbreaking works like DynamicViT (Rao et al., 2021) introduced a dynamic token sparsification framework that employs a lightweight, learnable prediction module to hierarchically prune tokens at various stages of the network. EViT (Liang et al., 2022) utilizes the class token to assess token importance, retaining the most attentive tokens while merging the others. The main drawback of pruning is the irreversible loss of information. Token merging, on the other hand, combines similar tokens instead of discarding them. ToMe (Bolya et al., 2022) introduced a training-free method that uses a fast bipartite soft matching algorithm to progressively merge similar tokens. Some approaches, like ATM (Fayyaz et al., 2022) and DTEM (Duman & Kalkan, 2024), have been developed to overcome the limitations of static merging policies that rely on intermediate features not specifically designed for the merging task. Recognizing that pruning and merging address different types of redundancy, the hybrid method combines these strategies for a more adaptable and potentially optimal reduction. DiffRate (Chen et al., 2023) makes the compression rate differentiable to learn layer-wise rates, while Token Transforming (Zeng et al., 2025) generalizes both pruning and merging as specific cases of a broader matrix transformation, enabling more flexible, many-to-many mappings that can better preserve information in a training-free manner.

Despite recent advancements, existing token reduction methods face several challenges. A primary issue is the non-differentiable nature of the token selection process when using the Top-K operation, which often requires complex workarounds for end-to-end training. Some methods are slow due to their reliance on clustering techniques like k-means, which are computationally intensive. Additionally, many methods introduce extra learnable parameters for token selection or merging modules, leading to increased model complexity and training overhead.

To address these limitations, inspired by ToMe, we introduce a novel training-free token merging approach that overcomes the mentioned challenges through several key innovations:

**Fully Differentiable Design**: Our method employs only differentiable operations throughout the token merging process, enabling seamless end-to-end training and optimization. By avoiding discrete operations, we maintain gradient flow and allow the model to be trained from scratch to learn optimal merging strategies.

**Efficient Matrix Operations**: Instead of relying on computationally expensive clustering algorithms, we utilize efficient matrix operations based on normalized cosine similarity. This approach offers both theoretical efficiency and practical speedup.

**Parameter-Free Architecture**: Our approach introduces no additional learnable parameters, maintaining the original model's parameter, simplifying deployment, and reducing the complexity of model management.

**Plug-and-Play Integration**: Our approach can be directly applied to pre-trained models without requiring retraining, or seamlessly integrated during training from scratch. This flexibility significantly lowers the barrier to adoption.

## 2 Related Work

To address the quadratic computational complexity introduced by the self-attention mechanism in the Vision Transformer models, researchers have proposed a variety of acceleration and optimization methods. These methods aim to improve ViT inference and training efficiency without significantly sacrificing model performance.

**Token Pruning** Pruning methods hierarchically discard tokens deemed non-informative based on learned importance metrics. DynamicViT (Rao et al., 2021) pioneered this approach by attaching lightweight prediction heads at intermediate layers to score token relevance, using differentiable attention masking to enable end-to-end training. EViT (Liang et al., 2022) enhanced this framework by fusing pruned tokens into the class token, preserving partial information while reducing sequence length. Recent advancements include AdaViT (Meng et al., 2022), which extends pruning beyond tokens to attention heads and transformer blocks, implementing instance-adaptive computation graphs that allocate more resources to complex inputs. However, these methods face fundamental limitations: 1) Early pruning decisions risk irreversible information loss, 2) Discrete selection operations create optimization challenges, and 3) Task-specific tuning is required for optimal threshold calibration.

**Token Merging** Merging techniques combine similar tokens rather than discarding them, preserving information while reducing computational load. ToMe (Bolya et al., 2022) revolutionized this space with training-free bipartite soft matching, using attention weights to merge the most similar token pairs at each layer. Its efficiency stems from matching tokens within local neighborhoods rather than globally, achieving real-time performance. However, ToMe's fixed merge ratio per layer limits adaptability to varying input complexities. Since ToMe, several similar works were proposed. For example, ToFu(Song et al., 2024) diverges from ToMe's training-free methodology by proposing a learnable fusion module that is co-trained with the Vision Transformer to generate new, more expressive tokens. Hybrid approaches such as Pumer (Fu et al., 2024) and LTPM (Li et al., 2024) integrate token pruning and merging within a unified framework. Pumer introduces a learnable router to dynamically determine the number of tokens to prune and merge on a per-instance basis, whereas LTPM employs learnable parameters to decide whether a token should be pruned or which tokens should be merged, thereby establishing a flexible, end-to-end reduction policy. DiffRate (Chen et al., 2023) addresses the challenge of selecting an optimal compression ratio by rendering the rate itself differentiable. It utilizes a learnable budget controller to optimize this rate for each input, facilitating instance-adaptive efficiency through standard gradient descent.

**Clustering-Based Reduction** Clustering approaches utilize offline algorithms to group tokens based on similarity. TCFormer (Zeng et al., 2024) employs KNN-enhanced Density Peaks Clustering to adaptively group tokens, identifying cluster centers and merging redundant tokens through averaging for tasks centered on human activities, such as pose estimation. ClusTR (Xie et al., 2022)

implements hierarchical token merging with cosine similarity measures across Transformer layers, striking a balance between computation and feature preservation for vision tasks. However, its predefined merging ratios limit flexibility, and early token reduction may hinder small object detection. While these methods excel in preserving global context, they face three significant drawbacks: 1) Iterative clustering algorithms with O(nk) complexity often negate computational savings, 2) Discrete cluster assignments prevent gradient flow, and 3) Fixed cluster counts lack adaptability to input variations.

**Learnable Token Reduction** End-to-end trainable methods optimize reduction policies through differentiable architectures. ATS (Fayyaz et al., 2022) implements token merging via softmax-weighted averaging with gating mechanisms, enabling gradient-based optimization of merge decisions. Dynamic Token Morphing (Wang et al., 2023) employs cross-attention between original tokens and a small set of learnable proxy tokens that adaptively absorb information. Gumbel Token Selector(Kim et al., 2023) uses the Gumbel-Softmax trick to differentiably sample token subsets, maintaining information flow to discarded tokens through residual connections. These approaches show promise for task-specific optimization but increase model complexity (15-30% more parameters) and risk overfitting on small datasets.

**Challenges and Limitations** Across these methods, several common challenges persist. The **non-differentiable** nature of discrete token selection has been a recurring obstacle, necessitating sophisticated solutions like the Gumbel-Softmax trick, attention masking (Rao et al., 2021), or continuous relaxation (Duman & Kalkan, 2024) for end-to-end optimization. Furthermore, many approaches introduce **additional learnable parameters** via decision networks or selection modules (e.g., DynamicViT, AdaViT), which increases model complexity. In contrast, parameter-free methods like ToMe offer plug-and-play efficiency without requiring additional training but need to manually specify the compression ratio. Finally, some methods exhibit **architecture-specific dependencies**; for example, EViT's reliance on a token for importance scoring limits its direct applicability to dense prediction tasks like segmentation, where such a token may not be present.

# 3 METHODOLOGY

## 3.1 TOKEN PARTITIONING

Let the input sequence from a given layer be represented by the matrix $X \in \mathbb{R}^{L\times d}$, where $L$ is the number of tokens and $d$ is the feature dimension. We first partition this sequence into two disjoint sets: a set of $M$ destination tokens, denoted by $\mathbf{X}_{\text{dst}} \in \mathbb{R}^{M\times d}$, and a set of $N$ source tokens, $\mathbf{X}_{\text{src}} \in \mathbb{R}^{N\times d}$, where $L = M + N$.

$$\begin{aligned}\mathbf{X}_{\text{dst}} &= \{\mathbf{x}_i : i \in \mathcal{I}_{\text{dst}}\}\\ \mathbf{X}_{\text{src}} &= \{\mathbf{x}_j : j \in \mathcal{I}_{\text{src}}\}\end{aligned} \tag{1}$$

where $\mathcal{I}_{dst}$ and $\mathcal{I}_{src}$ represent the index sets for destination and source tokens, respectively, such that $\mathcal{I}_{dst} \cap \mathcal{I}_{src} = \emptyset$ and $\mathcal{I}_{dst} \cup \mathcal{I}_{src} = \mathcal{I}$ covers all token indices, excluding any special tokens (e.g., class tokens). The specific strategy for partitioning into $\mathcal{I}_{\text{dst}}$ and $\mathcal{I}_{\text{src}}$ can vary (e.g., fixed interleaved patterns or random selection).

## 3.2 SIMILARITY-BASED FUSION MATRIX

**Similarity Matrix.** We begin by computing the cosine similarity between each destination token and every source token. This yields a similarity matrix $S \in \mathbb{R}^{M\times N}$, where each element $S_{ij}$ is defined as:

$$S_{ij} = \frac{\mathbf{x}_i^{\text{dst}} \cdot \mathbf{x}_j^{\text{src}}}{\|\mathbf{x}_i^{\text{dst}}\| \cdot \|\mathbf{x}_j^{\text{src}}\|} \tag{2}$$

To isolate the most significant relationships, we apply a rectified linear unit (ReLU) activation with a shifting threshold $\tau$. This step filters out weak connections, producing a sparse similarity matrix $\tilde{S} \in \mathbb{R}^{M\times N}$:

$$\tilde{S}_{ij} = \text{ReLU}(S_{ij} - \tau) \tag{3}$$

**Adaptive Weight Pruning.** From the sparse similarity matrix $\tilde{S}$, we first compute an initial weight matrix $W \in \mathbb{R}^{M \times N}$ by normalizing its columns. This ensures the initial influence of each source token is properly distributed among its similar destination tokens.

$$W_{ij} = \frac{\tilde{S}_{ij}}{\sum_{i=1}^{M} \tilde{S}_{ij} + \epsilon} \tag{4}$$

where $\epsilon$ is a small constant for numerical stability.

To further refine these weights, we introduce a dynamic, column-specific thresholding mechanism. For each source token $j$, we define a threshold $\zeta_j$ as the average of its non-zero weights in $W$:

$$\zeta_j = \frac{\sum_{i=1}^{M} W_{ij}}{C_j + \epsilon} \tag{5}$$

where $C_j$ is the count of non-zero entries along the destination dimension and can be computed as

$$C_j = \sum_{i=1}^{M} \frac{W_{ij}}{W_{ij} + \epsilon} \tag{6}$$

The threshold $\zeta_j$ is to prune connections that are weak relative to a source token's other connections. We apply this threshold to obtain a pruned weight matrix $\tilde{W}$:

$$\tilde{W}_{ij} = \text{ReLU}(W_{ij} - \zeta_j) \tag{7}$$

Finally, the pruned matrix $\tilde{W}$ is re-normalized column-wise to produce the final fusion weights $W^{\text{F}} \in \mathbb{R}^{M \times N}$:

$$W^{\text{F}}_{ij} = \frac{\tilde{W}_{ij}}{\sum_{i=1}^{M} \tilde{W}_{ij} + \epsilon} \tag{8}$$

### 3.3 Token Aggregation and Preservation

The destination tokens are updated by aggregating the features from source tokens, guided by the final fusion weights. The fused destination tokens, $X'_{\text{dst}} \in \mathbb{R}^{M \times d}$, are computed as:

$$\begin{aligned} \mathbf{X}'_{\text{dst}} &= \mathbf{X}_{\text{dst}} + \mathbf{W}^{\text{F}} \mathbf{X}_{\text{src}} \\ \mathbf{x}''_{\text{dst},i} &= \frac{\mathbf{x}'_{\text{dst},i}}{1 + \sum_{j=1}^{N} W^{\text{F}}_{ij}} \end{aligned} \tag{9}$$

A key component of our methodology is the preservation of unique source tokens. A source token $x^{\text{src}}_j$ is preserved if it exhibits no significant similarity to any destination token after the initial filtering, which means the sum of its similarities to all destination tokens is zero:: $\sum_{i=1}^{M} \tilde{S}_{ij} = 0$.

The final, reduced sequence is formed by concatenating any special tokens $\mathbf{X}_{\text{spec}}$, the merged destination tokens $\mathbf{X}''_{\text{dst}}$, and the set of preserved source tokens $\mathbf{X}_{\text{pres}}$.

$$\mathbf{X}_{\text{final}} = \text{concat}(\mathbf{X}_{\text{spec}}, \mathbf{X}''_{\text{dst}}, \mathbf{X}_{\text{pres}}) \tag{10}$$

If $r$ source tokens satisfy the preservation condition and there are $l_{\text{spec}}$ special tokens, the resulting sequence will have a reduced length of $l_{\text{spec}} + M + r$. The algorithm is illustrated in Figure 1.

### 3.4 Integration with Transformer Blocks

$$\begin{aligned} x'_l &= \text{MSA}(\text{LN}(x_{l-1})) + x_{l-1} \\ x''_l &= \text{Merge}(x'_l) \\ x_l &= \text{MLP}(\text{LN}(x''_l)) + x''_l \end{aligned} \tag{11}$$

Where $x_{l-1}$ be the token sequence output by block $l-1$, and MSA, MLP, and LN denote Multi-head Self-Attention, Multi-Layer Perceptron, and Layer Normalization, respectively.

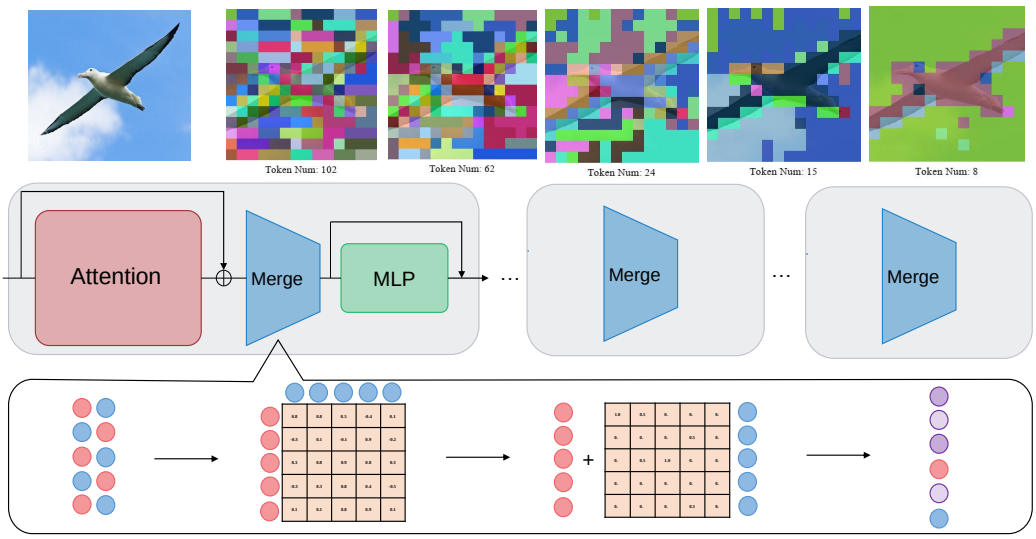



Figure 1: Illustration of MaMe Algorithm. The similarity threshold is 0.7 in this illustration.

# 4 Experiments

## 4.1 Image Classification

**Training-Free** The evaluation employs two representative Vision Transformer architectures: DeiT (Touvron et al., 2021) and MAE (He et al., 2022), utilizing their pre-trained weights without any fine-tuning. For these off-the-shelf experiments, we apply MaMe to the first 8 layers of each model, where we empirically set the similarity threshold to 0.8.

The results in Table 1 show that MaMe achieves 9015 img/s, 79% higher than baseline (5039 img/s) with 78.61% accuracy - 1.2 points below 79.82%. This outperforms EViT (8950 img/s, 73.83% accuracy) and ToMe (8874 img/s, 77.99% accuracy). For ViT-B (DeiT), MaMe delivers 4117 img/s (93% faster than baseline) with 2.03% accuracy drop (79.80% vs 81.83%). EViT shows higher throughput (4230 img/s) but lower accuracy (74.61%), while DiffRate has similar throughput but 78.98% accuracy. Comparing ViT-B (DeiT) and ViT-B (MAE) reveals key differences: MaMe achieves 4117 imgs/s with 79.80% accuracy on DeiT, but reaches 5418 imgs/s with 79.83% accuracy on MAE - a 31.6% improvement. MAE's self-supervised pre-training creates robust features, enabling aggressive token pruning. Static methods like EViT and ToMe show no throughput change between models. For ViT-L (MAE), MaMe achieves 2764 imgs/s with 84.81% accuracy. On ViT-H (MAE), it delivers 908 imgs/s, almost double EViT's speed, with minimal accuracy loss versus DiffRate.

**Visualization** Figure 2 illustrates our token visualization across successive transformer blocks in MaMe-enhanced ViT models. For slender elements like wooden slats or spider legs, MaMe effectively maintains distinct token assignments despite their thin morphology. Similarly, small objects in complex scenes retain dedicated token representation. For images with multiple targets, MaMe can also lock them correctly. The visualization also highlights MaMe's robust handling of geometrically challenging objects—cylindrical forms and irregular shapes maintain coherent token boundaries that faithfully follow their contours. This demonstrates consistent performance across different object scales and morphologies.

**Training-From-Scratch** For end-to-end training, we follow Swin Transformer (Liu et al., 2021) recipes while incorporating our compression strategy. In ViT architectures, we apply MAMe at layers 3, 6, and 9, with a 2:1 token reduction ratio at each compression point. The similarity threshold is 0.5. For hierarchical structures like Swin (Liu et al., 2021) and Iwin Transformer (Huo & Li, 2025), we replace downsampling modules with MaMe-based compression (reducing tokens to $\frac{1}{4}$ of original), eliminating embedding dimension doubling.

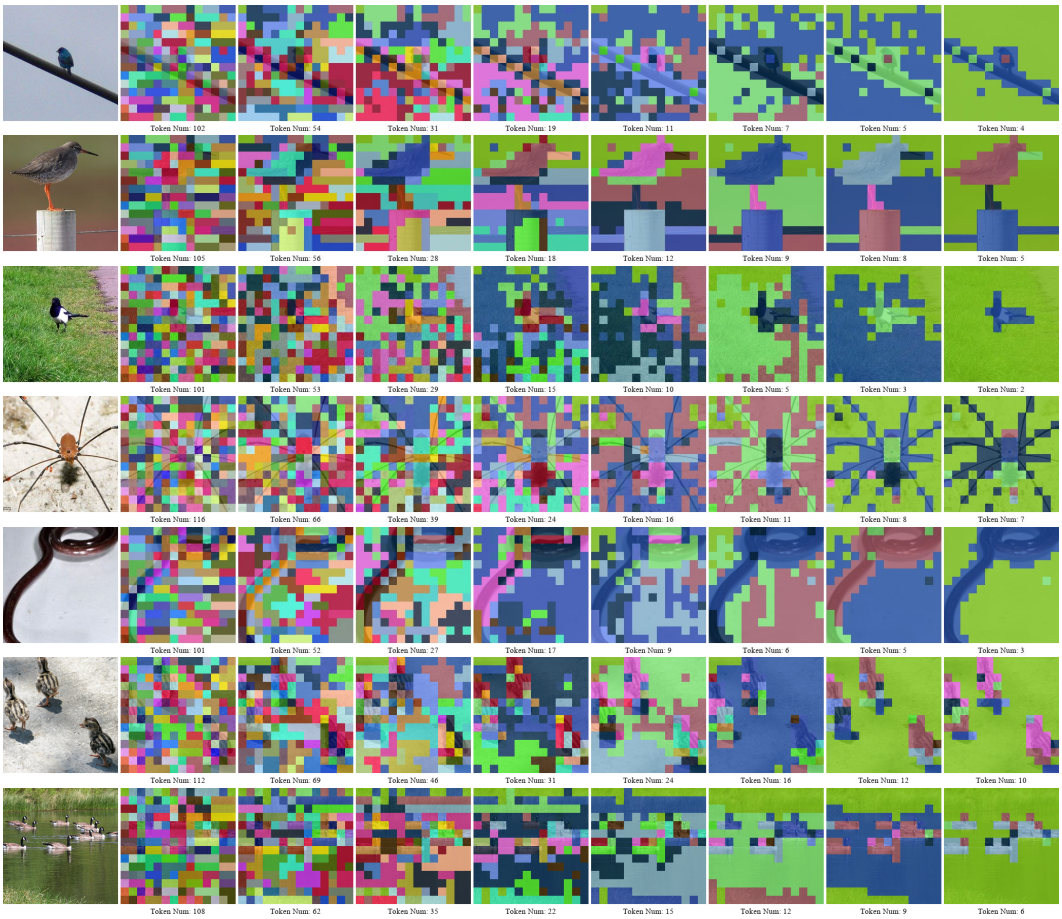



Figure 2: The visualization illustrates the progression of token count reduction in the first 8 blocks of the AugReg ViT-B/16 with MaMe. Each color represents a distinct type of token. See the Appendix 5 for more results.

The Table 3 shows metrics for ViT, Swin, and Iwin on ImageNet-1k (Russakovsky et al., 2015), with and without MaMe (marked with †). MaMe enhances throughput while maintaining competitive accuracy. ViT-T$^\dagger$ achieves 4462 img/s, doubling its baseline of 2291 img/s, with 1.3% accuracy drop. ViT-B$^\dagger$ shows 813 img/s (92% faster) at 5.8% accuracy cost, though its accuracy(76.0%) is lower than ViT-S$^\dagger$(77.0%). While Iwin-T (874 img/s) was slower than Swin-T (950 img/s), Iwin-T$^\dagger$ (1522 img/s) outperforms both alternatives. Similarly, Iwin-S$^\dagger$ achieves 1254 img/s versus Swin-S$^\dagger$'s 1043 img/s, while maintaining better accuracy (71.0% vs 65.8%). Accuracy-throughput tradeoffs vary across architectures. ViT shows minimal accuracy degradation (1-6 points), while Swin exhibits larger drops (15-20 points). Iwin achieves better balance - Iwin-S$^\dagger$ maintains 71.0% accuracy versus Swin-S$^\dagger$'s 65.8%. Iwin-T$^\dagger$ achieves higher accuracy (65.1%) than Swin-T$^\dagger$ (60.3%) with faster throughput.

### 4.2 Semantic Segmentation

To assess our compressed models to downstream dense prediction tasks, we evaluate the Swin and Iwin backbones on the ADE20K (Zhou et al., 2019) using UperNet (Xiao et al., 2018) in MMSegmentation (Contributors, 2020). Following (Liu et al., 2021) settings, results in Table 4 show MaMe marked as (†) reduces model complexity, with Swin-T$^\dagger$ decreasing parameters by 52% and FLOPs by 54% versus baseline, though with reduced segmentation performance. While compressed Iwin$^\dagger$ models achieve strong classification accuracy (Iwin-S$^\dagger$ at 71.0%), their semantic segmentation performance drops more than compressed Swin models, with Iwin-S$^\dagger$ declining 15.9 points versus Swin-S$^\dagger$'s 11.8 points.

| Model | Method | FLOPs (G) | Throughput (img/s) | Top-1 Acc (%) |
|---|---|---|---|---|
| **Training Free on ImageNet-1K (224×224)** | | | | |
| ViT-S (DeiT) | Baseline | 4.6 | 5039 | 79.82 |
| | EViT | 2.3 | 8950 | 73.83 |
| | ToMe | 2.3 | 8874 | 77.99 |
| | DiffRate | 2.3 | 8875 | 78.75 |
| | MaMe | - | 9015 | 78.61 |
| ViT-B (DeiT) | Baseline | 17.6 | 2130 | 81.83 |
| | EViT | 8.7 | 4230 | 74.61 |
| | ToMe | 8.8 | 4023 | 77.84 |
| | DiffRate | 8.7 | 4124 | 78.98 |
| | MaMe | - | 4117 | 79.80 |
| ViT-B (MAE) | Baseline | 17.6 | 2130 | 83.72 |
| | EViT | 8.7 | 4230 | 75.15 |
| | ToMe | 8.8 | 4023 | 78.86 |
| | DiffRate | 8.7 | 4150 | 79.96 |
| | MaMe | - | 5418 | 79.83 |
| ViT-L (MAE) | Baseline | 61.6 | 758 | 85.95 |
| | EViT | 29.7 | 1672 | 81.52 |
| | ToMe | 31.0 | 1550 | 84.24 |
| | DiffRate | 31.0 | 1580 | 84.65 |
| | MaMe | - | 2764 | 84.81 |
| ViT-H (MAE) | Baseline | 167.4 | 299 | 86.88 |
| | EViT | 99.1 | 512 | 85.54 |
| | ToMe | 92.9 | 500 | 86.01 |
| | DiffRate | 93.2 | 504 | 86.40 |
| | MaMe | - | 908 | 86.16 |

Table 1: Token compression on off-the-shelf models. Throughput measured on an A100 GPU (bs=1024, fp16).

| Model | Method | Input Size (px) | Throughput (img/s) | Top-1 Acc (%) |
|---|---|---|---|---|
| **Zero-Shot Classification on ImageNet-1K** | | | | |
| CLIP (ViT-L/14) | Baseline | 224 | 51.22 | 70.34 |
| | ToMe(r=8) | 224 | 51.33 | 68.98 |
| | ToMe(r=12) | 224 | 52.86 | 66.00 |
| | MaMe($\tau = 0.7$) | 224 | 69.09 | 67.60 |
| | MaMe($\tau = 0.8$) | 224 | 64.01 | 69.95 |
| SigLIP (ViT-B/16) | Baseline | 512 | 46.28 | 75.61 |
| | ToMe(r=32) | 512 | 55.10 | 74.33 |
| | ToMe(r=64) | 512 | 71.94 | 70.66 |
| | MaMe($\tau = 0.8$) | 512 | 79.25 | 71.17 |
| | MaMe($\tau = 0.9$) | 512 | 58.10 | 74.50 |
| SigLIP2 (ViT-B/16) | Baseline | 512 | 43.90 | 78.37 |
| | ToMe(r=32) | 512 | 50.89 | 76.46 |
| | ToMe(r=64) | 512 | 68.07 | 71.60 |
| | MaMe($\tau = 0.9$) | 512 | 76.15 | 75.09 |
| | MaMe($\tau = 0.95$) | 512 | 56.15 | 78.02 |

Table 2: Zero-shot results. Inference throughput measured on a 3090 GPU with bfp16.

| Model | Param (M) | FLOPs (G) | Throughput (img/s) | Top-1 Acc (%) |
|---|---|---|---|---|
| **Training From Scratch on ImageNet-1K (224×224)** | | | | |
| ViT-T | 5.72 | 1.3 | 2291 | 72.2 |
| ViT-T† | 5.72 | 0.6 | 4462 | 70.9 |
| Swin-T | 29.0 | 4.5 | 950 | 81.3 |
| Swin-T† | 1.45 | 1.5 | 1236 | 60.3 |
| Iwin-T | 30.2 | 4.7 | 874 | 82.0 |
| Iwin-T† | 1.46 | 1.5 | 1522 | 65.1 |
| ViT-S | 22.0 | 4.6 | 1157 | 79.8 |
| ViT-S† | 22.0 | 2.1 | 2257 | 77.0 |
| Swin-S | 50.0 | 8.7 | 548 | 83.0 |
| Swin-S† | 2.80 | 1.8 | 1043 | 65.8 |
| Iwin-S | 51.6 | 9.0 | 512 | 83.4 |
| Iwin-S† | 2.82 | 1.8 | 1254 | 71.0 |
| ViT-B | 86.4 | 17.6 | 422 | 81.8 |
| ViT-B† | 86.4 | 8.4 | 813 | 76.0 |

Table 3: Comparative evaluation with MaMe compression (†). Throughput measured on an A100 GPU (bs=64, fp32).

| Backbone | UperNet 160k | | | |
|---|---|---|---|---|
| | Param(M) | FLOPs(G) | mIoU(%) | mAcc(%) |
| Swin-T | 59.9 | 945 | 44.5 | 55.6 |
| Swin-T† | 28.4 | 432 | 33.1 | 44.1 |
| Iwin-T | 61.9 | 946 | 44.7 | 56.6 |
| Iwin-T† | 28.5 | 432 | 26.1 | 35.0 |
| Swin-S | 81.3 | 1038 | 47.6 | 58.8 |
| Swin-S† | 29.8 | 435 | 35.8 | 47.0 |
| Iwin-S | 83.2 | 1038 | 47.5 | 59.3 |
| Iwin-S† | 29.8 | 435 | 31.6 | 41.6 |

Table 4: Results for ADE20K semantic segmentation. FLOPs measured with input size 512×2048.

| Model | Method | Input (FxHW) | Throughput (videos/s) | Top-1 Acc (%) |
|---|---|---|---|---|
| **Action Recognition on Kinetics-400** | | | | |
| VideoMAE-B | Baseline | 16x224 | 13.24 | 76.81 |
| | ToMe(r=96) | 16x224 | 13.76 | 75.54 |
| | ToMe(r=128) | 16x224 | 14.06 | 73.34 |
| | MaMe($\tau = 0.85$) | 16x224 | 13.81 | 74.23 |
| | MaMe($\tau = 0.9$) | 16x224 | 13.33 | 76.03 |
| VideoMAE-L | Baseline | 16x224 | 6.25 | 82.31 |
| | ToMe(r=32) | 16x224 | 6.97 | 82.05 |
| | MaMe($\tau = 0.8$) | 16x224 | 9.28 | 81.47 |

Table 5: Results of VideoMAE on action recognition benchmarks. Inference throughput is measured on a 3090 GPU with fp16.

### 4.3 Multimodal Large Language Models

**Zero-shot Image Classification** We conducted zero-shot image classification on the ImageNet-1K validation set to evaluate token merging strategies across CLIP, SigLIP, and SigLIP2, focusing on inference throughput and accuracy. As shown in the Table 2, for CLIP, MaMe ($\tau = 0.8$) increased throughput by 25% (64.01 img/s) with a 0.39% decrease in Top-1 accuracy, while ToMe (r=12) offered a 3% throughput gain but resulted in a 4.34% accuracy drop. In the case of SigLIP, MaMe

| Method | MME | | MMMU | | ScienceQA | | SEED-Image | | MMStar | | CRPE | | MMBench | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Metric | Time(s) | Metric | Time(s) | Metric | Time(s) | Metric | Time(s) | Metric | Time(s) | Metric | Time(s) | Metric | Time(s) |
| LLaVA-1.5-7B (Baseline) | 32.76 | 597 | 32.22 | 481 | 65.43 | 625 | 60.17 | 3513 | 32.53 | 565 | 50.69 | 2076 | 62.80 | 1191 |
| + ToMe (r=8) | 31.40 | 509 | 30.11 | 440 | 63.42 | 554 | 58.19 | 3135 | 31.13 | 545 | 46.78 | 1794 | 61.00 | 1086 |
| + MaMe ($\tau$=0.8) | 31.40 | 447 | 30.56 | 422 | 64.47 | 478 | 57.20 | 2840 | 30.27 | 531 | 45.62 | 1659 | 60.48 | 1020 |

Table 6: Benchmark results for LLaVA-1.5-7B with different token merging methods. For each benchmark, we report the primary **Metric** (e.g., accuracy) and the total evaluation **Time** in seconds.

($\tau = 0.9$) boosted throughput by 25% (58.10 img/s) with an accuracy of 74.50% (a 1.11% reduction), whereas ToMe (r=32) achieved a 19% speedup with a 1.28% accuracy loss. For SigLIP2, MaMe ($\tau = 0.95$) improved throughput by 28% (56.15 img/s) with a 0.35% accuracy drop, while ToMe (r=32) increased throughput by 16% with a 1.91% accuracy loss. Notably, compared to SigLIP, SigLIP2 can still merge enough tokens at a stricter threshold $\tau = 0.95$ to achieve a 28% throughput gain, indicating that SigLIP2 lelarned highly confident and semantically clustered representations. The results demonstrate that MaMe offers a better balance between inference throughput and accuracy compared to the baseline and ToMe.

**Text-Image to Text** We evaluated the impact of token merging on the LLaVA-1.5-7B model (Liu et al., 2023a) using the VLMEvalKit framework (Duan et al., 2024) across various multimodal benchmarks (Fu et al., 2023; Yue et al., 2023; Lu et al., 2022; Li et al., 2023; Lin et al., 2024; Liu et al., 2024; 2023b), Merging was applied to the visual encoder to reduce the number of visual tokens fed to the language model. We compare the baseline against ToMe with a fixed reduction ratio of r=8 per layer and MaMe with a similarity threshold of $\tau = 0.8$. As shown in Table 6, both methods significantly reduce evaluation time. MaMe achieves greater acceleration while delivering metric scores that are competitive with or slightly better than ToMe on most of benchmarks. This demonstrates that token merging, particularly MaMe, is a highly effective strategy for accelerating large multimodal models with minimal impact on performance.

### 4.4 Video Classification

We apply token merging to VideoMAE (Tong et al., 2022) models' vision encoder and compare MaMe with ToMe on Kinetics-400 validation set (Kay et al., 2017). We sample 16 frames at $224 \times 224$ resolution per video clip. We report Top-1 accuracy and inference throughput in videos/s on a 3090 GPU with fp16 precision. Results in Table 5 show token merging effectively accelerates video transformer inference. For VideoMAE-B, MaMe with threshold $\tau = 0.9$ increases throughput from 13.24 to 13.33 videos/s with only 0.78% drop in Top-1 accuracy, outperforming ToMe(r=128) which shows 3.47% accuracy degradation for similar speedup.

### 4.5 Ablation Study

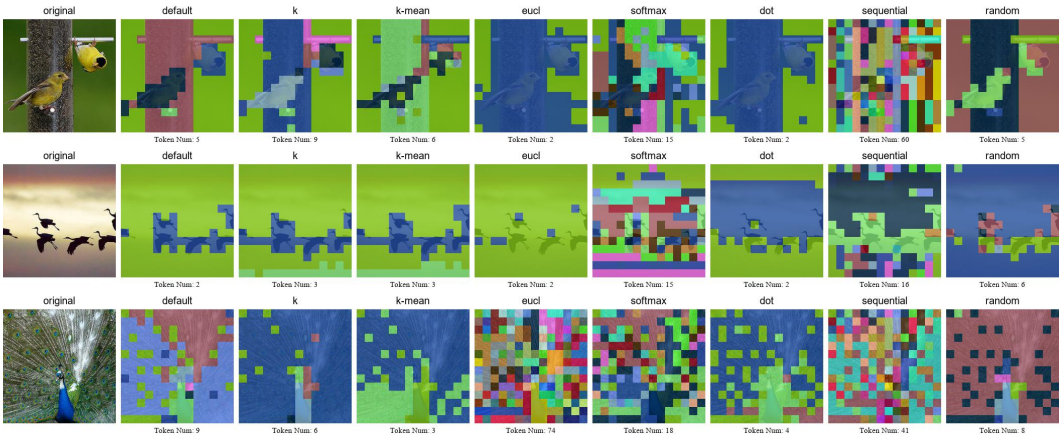
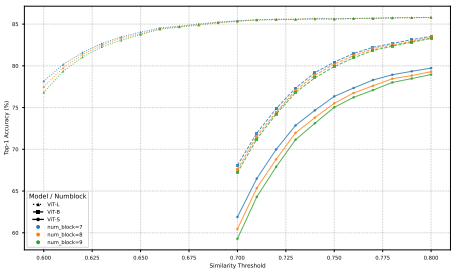
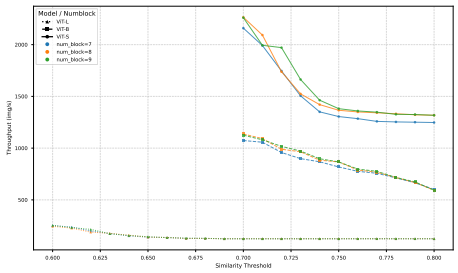


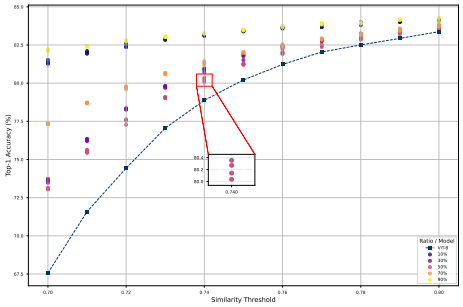

Figure 3: The visualization of the final token count in the 8th block of the AugReg ViT-B/16 using MaMe with different settings. Each color represents a distinct type of token.



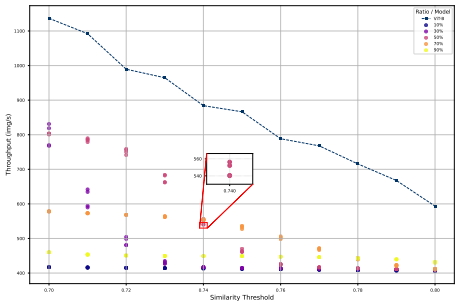
(a) Accuracy vs. Similarity Threshold



(b) Throughput vs. Similarity Threshold



(c) Accuracy vs. Similarity Threshold with different proportions of random partition



(d) Throughput vs. Similarity Threshold with different proportions of random partition

Figure 4: The accuracy and throughput curves vary with the similarity threshold under the alternating and random partition. Where $num\_block$ denotes the number of initial blocks (from block 0 onward) to which token merging is applied. The same configuration uses 5 different random seeds to perform 5 different random experiments. There is a non-linear, saturating relationship between similarity threshold and both accuracy and throughput across ViT architectures. Larger models like ViT-L are more robust to token merging, maintaining higher accuracy even at lower thresholds. (c) and (d) shows that the default curves represent a Pareto frontier.

| feature | acc | im/s |
|---|---|---|
| `x` | **83.35** | 73.06 |
| `k` | 71.63 | 72.45 |
| `k-mean` | 69.01 | 75.06 |

(a) **Feature Choice.** The `x` matrix has the most information within tokens.

| function | acc | im/s |
|---|---|---|
| eucl | 81.58 | 73.60 |
| cosine | 83.35 | 73.06 |
| dot | 80.43 | 81.14 |
| softmax | 61.90 | 80.33 |

(b) **Similarity Function.** Cosine similarity is the best choice for speed and accuracy.

| order | acc | im/s |
|---|---|---|
| sequential | 84.14 | 71.81 |
| alternating | 83.35 | 73.06 |
| random | 83.24 | 73.74 |

(c) **Partition Style.** Alternating is more reliable and faster.

Table 7: Ablation experiments using AugReg ViT-B/16. Our default settings are marked in green. We report Top-1 accuracy (acc) and fp32 model inference throughput (im/s) on a 3090 GPU. The visualization results of different methods are shown in Figure 3

## 5 Conclusion

In this work, we introduced MaMe, a token merging method composed entirely of efficient matrix operations to accelerate the inference of Vision Transformers. MaMe serves as a "plug-and-play" module that can be effortlessly incorporated into a wide array of existing architectures. We have demonstrated its effectiveness across various scenarios, including off-the-shelf models, end-to-end training, zero-shot classification with vision-language models, complex multimodal benchmarks, and video action recognition. Compared with ToMe, MaMe does not require intrusive modifications to the standard self-attention calculation module and can enjoy efficient attention computation. The primary limitation of MaMe is the need to manually set the similarity threshold. Future work will focus on developing an adaptive or learnable mechanism to automate the selection of this threshold, further enhancing the MaMe's autonomy and practical utility.

## REFERENCES


Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

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. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 17164–17174, 2023.

MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11198–11201, 2024.

Hasan Duman and Simon Kalkan. Dtem: Differentiable token-merging for vision transformers. *arXiv preprint arXiv:2404.09341*, 2024.

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*, pp. 396–414. Springer, 2022.

Chaoyou Fu, Yixuan Chen, Haotian Wang, Xinyu Liu, Mintong Ye, Bill Lin, David Han, and Gao Liu. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Zhaouihui Fu, Zikang Huang, Yu Liu, Siguang Han, Yixing Sun, Yitong Zhu, and Jun Yan. Pumer: Pruning and merging for efficient vision transformers. *arXiv preprint arXiv:2405.02835*, 2024.

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*, pp. 16000–16009, 2022.

Simin Huo and Ning Li. Iwin transformer: Hierarchical vision transformer using interleaved windows, 2025. URL https://arxiv.org/abs/2507.18405.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017.

Dong-Hwan Kim, Hyeong-Jun Kim, and Tae-Hyun Kim. Gumbel-gate: A gumbel-based gating network for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 1454–1462, 2023.

Bo Li, Wei Zhao, and Zhi Zhang. LTPM: A learnable token pruning and merging method for vision transformer. *arXiv preprint arXiv:2406.01289*, 2024.

Bohao Li, Yuanhan Zhang, Sheng Li, Gengyun Chen, Jing Yang, Guangzhi Chen, Ruisi He, Wene Liu, Huijuan Wang, Fang Wen, et al. SEED-Bench: Benchmarking multimodal LLMs with text-rich visual comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

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, 2022. URL https://arxiv.org/abs/2202.07800.

Hai-tian Lin, Zhe-Chen Feng, Can Xu, xiaogang Wang, and Hongsheng Yu. MM-Star: A large-scale and high-quality dataset for multimodal large language models. *arXiv preprint arXiv:2401.07849*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023a.

Shiyu Liu, Linjie Li, Zhe Gan He, Lijuan Wang, Kevin Sun, and Jianfeng Liu. CRPE: A dataset for composite reasoning and perception evaluation. *arXiv preprint arXiv:2405.08479*, 2024.

Yuan Liu, Haodong Li, Binyuan Li, Yuan He, Yong-Jae Zhang, Feng Sun, Yiyi Wang, Hao Zhang, Hong-xun Yang, Yu-Feng Li, et al. MMBench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023b.

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

Pan Lu, Swaroop Mishra, Tongshuang Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Singh. Learn to explain: Multimodal reasoning over structured knowledge for science question answering. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12309–12318, 2022.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

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. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Hyunsu Song, Young-Jae Kim, Seong-Woong Oh, and Seon-Ju Chun. ToFu: Token fusion for fast and accurate vision transformers. *arXiv preprint arXiv:2403.14950*, 2024.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

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 (NeurIPS)*, 30, 2017.

Yihua Wang, Yuxuan Chen, Lang Wang, and Jing Chen. Token morphing for vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16584–16593, 2023.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018.

Yutong Xie, Jianpeng Zhang, Yong Xia, Anton van den Hengel, and Qi Wu. Clustr: Exploring efficient self-attention via clustering for vision transformers, 2022. URL `https://arxiv.org/abs/2208.13138`.

Xiang Yue, Yuansheng Ni, Kai Zheng, Guanting Zhang, Yinan Cui, Bolin Li, Yuxiang Zhang, Chi Chen, Ziyue Zhang, Zhifeng Li, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023.

Fanhu Zeng, Deli Yu, Zhenglun Kong, and Hao Tang. Token transforming: A unified and training-free token compression framework for vision transformer acceleration, 2025. URL https://arxiv.org/abs/2506.05709.

Wang Zeng, Sheng Jin, Lumin Xu, Wentao Liu, Chen Qian, Wanli Ouyang, Ping Luo, and Xiaogang Wang. Tcformer: Visual recognition via token clustering transformer, 2024. URL https://arxiv.org/abs/2407.11321.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.

# A APPENDIX


Figure 5: The visualization illustrates the progression of token count reduction in the first 8 blocks of AugReg ViT-B/16 with MaMe. Each color represents a distinct type of token.

## LARGE LANGUAGE MODEL USAGE DECLARATION

In the preparation of this work, the authors utilized several large language models (LLMs) for specific tasks as detailed below. The authors are solely responsible for the content of the publication.

- **Literature Review Assistance:** Gemini 2.5 Pro was employed to assist in gathering and synthesizing relevant research literature. This assistance was primarily used in the preparation of the **Introduction** and **Related Work** sections to identify key developments and contextualize our contribution within the existing body of research.
- **Language Polishing:** Gemini 2.5 Pro, Gemini 2.5 Flash, and DeepSeek-R1 were used to refine English expression throughout the manuscript. This included improving grammatical accuracy, enhancing clarity of technical descriptions, and ensuring consistent academic tone.
- **Experimental Analysis Support:** Gemini 2.5 Pro was utilized as an analytical tool to assist in the interpretation of selected experimental results, particularly in identifying patterns and generating preliminary insights that were subsequently rigorously verified and expanded upon by the authors.
- **Declaration:** Deepseek-R1 was used to assist in writing the declaration.

All content generated with LLM assistance was carefully reviewed, critically evaluated, and substantially modified by the authors to ensure accuracy, originality, and adherence to scientific standards. The final manuscript represents the authors' own intellectual contribution and perspective.