MaMe: Matrix-Based Token Merging

Simin Huo
Shanghai Jiao Tong University
Shanghai, China
sameenhuo@sjtu.edu.cn

Ning Li
Shanghai Jiao Tong University
Shanghai, China
ningli@sjtu.edu.cn

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 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 Iwin models, MaMe dramatically reduced Iwin-S' GFLOPs from 9.0 to 1.8 with a 12.4% accuracy drop. 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. Furthermore, MaMe achieves simultaneous improvements in both performance and speed on the COCO Caption task, significantly boosting CIDEr to 2.71 compared to the baseline's 0.71 with a speedup of 16%. Collectively, these results demonstrate MaMe's effectiveness in accelerating transformer-based vision models.

1. Introduction

Vision Transformers (ViTs)[6] have revolutionized computer vision by adopting the transformer architecture from natural language models[35]. 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 approach is to reduce the number of tokens N involved in the process. The strategies that have emerged include token pruning, token merging, and hybrid methods that integrate both. Pioneering works like DynamicViT[29] introduced a dynamic token sparsification framework that uses a lightweight, learnable prediction module to hierarchically prune tokens at various stages of the network. EViT[18] uses the class token to evaluate token importance, keeping the most attentive tokens while merging the others. Pruning's main drawback is irreversible information loss. Token merging combines similar tokens instead of discarding them. ToMe[1] introduced a training-free method, using a fast bipartite soft matching algorithm to progressively merge similar tokens. Token Pooling[26] uses cluster analysis to aggregate information from neighboring tokens. DiffRate[3] makes compression rate differentiable to learn layer-wise rates, while Token Transforming[39] 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.

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** in practice. Additionally, many methods introduce **extra learnable parameters** for token selection or merging modules, leading to increased model complexity and training overhead. Lastly, an issue is the **dependency on specific architectures**; for example, EViT's reliance on a class token restricting its use in models where a class token might not be available.

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

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

Efficient Matrix Operations: Instead of relying on operations such as clustering algorithms, sorting or explicit maximum selection,we utilize efficient, GPU-friendly full-matrix operations. This approach offers both theoretical efficiency and practical speedup.

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

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

2. Related Work

2.0.1. Token Pruning

Pruning methods discard non-informative tokens based on importance metrics. DynamicViT[29] pioneered this by using lightweight prediction heads to score token relevance, enabling end-to-end training. EViT[18] enhanced this by fusing pruned tokens into the class token while reducing sequence length. AdaViT[27] extends pruning to attention heads and transformer blocks, creating instance-adaptive computation graphs for complex inputs. However, these methods face limitations: 1) Early pruning risks information loss, 2) Discrete selection creates optimization challenges, and 3) Task-specific tuning is needed for threshold calibration.

2.0.2. Token Merging

Merging techniques combine similar tokens rather than discarding them, preserving information while reducing computational load. ToMe[1] revolutionized this area with training-free bipartite soft matching to merge the most similar token pairs at each layer. However, ToMe's fixed merge ratio per layer limits adaptability to varying input complexities. DiffRate[3] addresses the challenge of selecting an optimal merge 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 but increasing complexity. ToFu[31] diverges from ToMe's training-free methodology by proposing a learnable fusion module that is co-trained with the models to generate new, more expressive tokens. Hybrid approaches such as Pumer[10] and LTPM[17] 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.

2.0.3. Clustering-Based Reduction

Clustering approaches use offline algorithms to group similar tokens. TCFormer[40] employs KNN-enhanced Density Peaks Clustering to group tokens and merge redundant ones through averaging for human activity tasks like pose estimation. ClusTR[37] uses hierarchical token merging with cosine similarity across Transformer layers for vision tasks, but its fixed ratios limit flexibility and may hinder small object detection. While these methods preserve global context, they face three drawbacks: 1) Iterative clustering algorithms with O(nk) complexity offset computational gains, 2) Discrete cluster assignments prevent gradient flow, and 3) Fixed cluster counts lack input adaptability.

2.0.4. Learnable Token Reduction

End-to-end trainable methods optimize reduction policies through differentiable architectures. ATS[8] implements token merging via weighted averaging with gating mechanisms. Dynamic Token Morphing[36] uses cross-attention between original and learnable proxy tokens for information absorption. Gumbel Token Selector[15] employs Gumbel-Softmax to sample token subsets through residual connections. These approaches show promise but increase model complexity (15-30% more parameters) and risk overfitting on small datasets.

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., alternating or random partition, see 2c).

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 τ. 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 ϵ 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 ζ_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

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

For differentiability, we don't apply boolean operations $C_j = \sum_{i=1}^{M} W_{ij} > 0$.

The threshold ζ_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:

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

where $W^{F}_{i,j}$ represents the similarity between the i-th destination token and the j-th source token.

A key component of our methodology is the preservation of unique source tokens $\mathbf{X}_{\text{pres}}$. A source token x^{src}_j is preserved if it exhibits no similarity to any destination token, which means the sum of its similarities to all M destination tokens is zero: $m_j = \mathbb{I}(\sum_{i=1}^{M} W^{F}_{ij} = 0)$, where $\mathbb{I}(\cdot)$ is the indicator function. So $\mathbf{X}_{\text{pres}} = \{\mathbf{x}^{src}_j \mid m_j = 1\}$.

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

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

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

Batch Processing Implementation. For efficient implementation on batched data, the preservation decision must be consistent across all samples in a batch. Given the fusion matrix for a batch be $\mathbf{W}^{F} \in \mathbb{R}^{B \times M \times N}$. A per-sample preservation mask $m^{(b)} \in \{0,1\}^N$ is computed for each sample b, where $m^{(b)}_j = \mathbb{I}(\sum_{i=1}^{M} W^{F}_{bij} = 0)$. To ensure batch consistency, a source token j is preserved if it is marked for preservation in *any* sample, yielding a final batch-wide mask $m^{\text{final}}_j = \bigvee_{b=1}^{B} m^{(b)}_j$. So the preserved source tokens $\mathbf{X}_{\text{pres}} = \{\mathbf{x}^{src}_j \mid m^{\text{final}}_j = 1\}$. Subsequently, to prevent preserved tokens from fusing, the corrected fusion matrix $\tilde{\mathbf{W}}^{F}$ is obtained by $\tilde{\mathbf{W}}^{F}_{b,i,j} = \mathbf{W}^{F}_{b,i,j} \cdot (1 - m^{\text{final}}_j)$, zeroing out columns corresponding to preserved tokens and keeping others unchanged.

3.4. Integration with Transformer Blocks

MaMe can be seamlessly integrated into the transformer-based architecture. Let 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. The operations within a modified Transformer block l are formalized as follows:

$$x'_l = \text{MSA}(\text{LN}(x_{l-1})) + x_{l-1} \tag{11}$$

$$x''_l = \text{MaMe}(x'_l) \tag{12}$$

$$x_l = \text{MLP}(\text{LN}(x''_l)) + x''_l \tag{13}$$

4. Experiments

4.0.1. Implementation Details

Training-Free The evaluation employs two representative architectures: DeiT[33] and MAE[12], utilizing their pre-trained weights without any fine-tuning. For these off-the-shelf experiments, we apply MaMe to the first 8 layers of

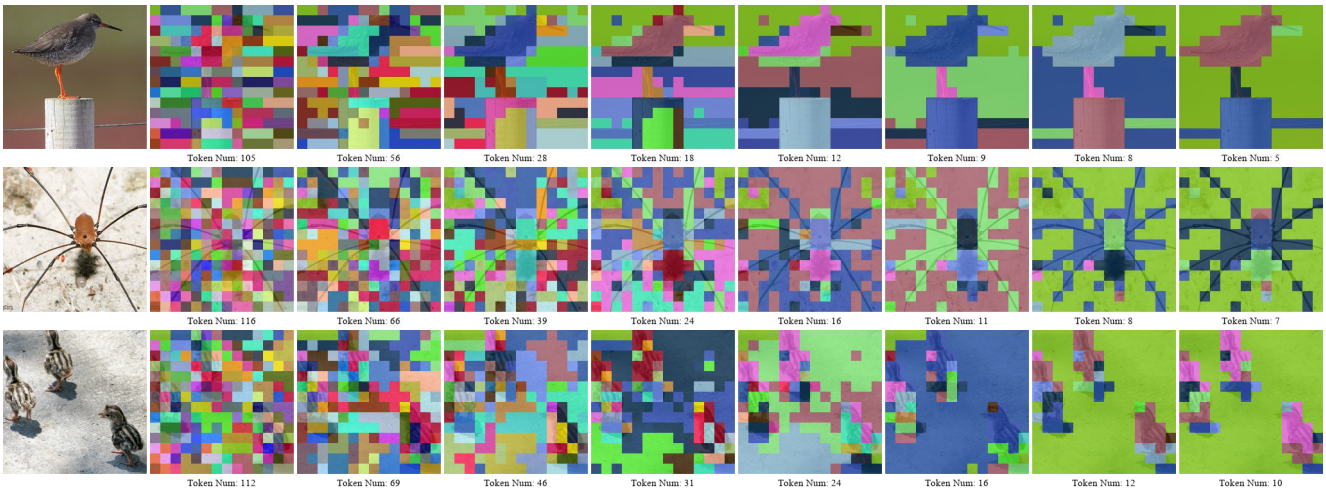


Figure 1. The visualization illustrates the progression of token count reduction in the first 8 blocks of the AugReg ViT-B/16 with MaMe. Each color represents a distinct type of token. More results are in the supplementary materials.

each model, where we empirically set the similarity threshold to 0.8. All other things remain identical to [3].

Training-From-Scratch For end-to-end training experiments, we follow the training recipes[24] while incorporating our compression strategy. In standard ViT architectures, we apply merging at layers 3, 6, and 9, implementing a consistent 2:1 token reduction ratio (reducing token count, except for the special class token, to half of original) at each compression point. The similarity threshold is 0.5. For hierarchical pyramid structure like Swin[24] and Iwin Transformer[13], we replace the original downsampling layers with MaMe by setting 25% tokens as destination tokens, reducing the token count to $\frac{1}{4}$ of the original. In order to maintain a regular shape for window partition, we had to discard the "preserved source tokens". ToMe is unsuitable for this task as its reliance on sorted similarity rankings would disrupt spatial relationships. This modification **eliminates the need for embedding dimension doubling** after downsampling, maintaining consistent feature dimensions throughout the network and reducing parameters and flops dramatically, facilitating deployment on edge devices. All other training settings including optimization (AdamW), and learning rate schedule (cosine decay with 20-epoch warmup) remain identical to [24].

4.0.2. Results

Training-Free

Table 1 evaluates several token compression methods on ViT models. For ViT-S (DeiT), MaMe achieves 9015 img/s, 79% higher than baseline while maintaining 78.61% accuracy (1.2 points below original), surpassing 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) with 79.80% accuracy. EViT shows higher throughput (4230 img/s) but lower accuracy (74.61%), while DiffRate[3] has similar speed but 78.98% accuracy.

Notably, comparing ViT-B (DeiT) and ViT-B (MAE), which share identical architecture, reveals significant differences in MaMe's performance. On ViT-B (DeiT), MaMe achieves 4117 imgs/s with 79.80% accuracy, while on MAE, throughput increases to 5418 imgs/s, representing a 31.6% throughput improvement, while maintaining 79.83% accuracy. This highlights MaMe's ability to leverage MAE's self-supervised robust representations effectively for token merging. EViT and ToMe show no throughput change between ViT-B models. MaMe's advantage grows with model size: for ViT-L (MAE), it achieves 2764 imgs/s (EViT's 1.63x) while maintaining the highest accuracy (84.81%). On ViT-H (MAE), MaMe delivers 908 imgs/s (almost EViT's 2x), with only a marginal accuracy decrease compared to DiffRate.

Training From Scratch

The Table 2 presents the metrics for ViT, Swin, and Iwin models, on ImageNet-1k [30], both without and with MaMe compression (marked with †). MaMe consistently enhances throughput across all architectures while maintaining competitive accuracy. For example, ViT-T† achieves 4462 img/s, nearly doubling its baseline of 2291 img/s, with only a 1.3 percentage point drop in accuracy (70.9% vs. 72.2%). For larger models, as seen with ViT-B†, which shows 813 img/s (92% faster than the baseline) at a 5.8-point accuracy cost. However, its accuracy (76.0%) is lower than the small variant ViT-S†(77.0%).

Accuracy-throughput tradeoffs differ across architectures. ViT variants show minimal accuracy loss with compression (1-6 points), while Swin transformers drop more significantly (15-20 points). Iwin achieves a balanced middle ground - Iwin-S† maintains 71.0% accuracy versus base-

Table 1. The Results of Token compression on the off-the-shelf models. Throughput is measured on an A100 GPU with a batch-size of 1024 using FP16 precision.

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	2.3	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	8.7	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	8.7	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	31.0	2764	84.81
ViT-H (MAE)	Baseline	167.4	299	86.88
	EViT	92.9	500	86.01
	ToMe	99.1	512	85.54
	DiffRate	93.2	504	86.40
	MaMe	93.2	908	85.51

line 83.4%, while Swin-S† drops to 65.8% from 83.0%.

In summary, this part of the work on using MaMe as downsampling in a hierarchical pyramid structure is exploratory and sacrifices some performance due to the need to discard "preserved source tokens" limited by the model architecture. This work demonstrates the feasibility of MaMe as downsampling, achieving a significant reduction in parameters and computation by removing the doubling of embedding dimension as the network deepens. There is room for improvement, which will be left for future work.

4.1. Multimodal Large Language Models

Zero-shot Image Classification We conducted zero-shot image classification on ImageNet-1K validation set to evaluate token merging strategies across CLIP[28], SigLIP[41], and SigLIP2[34]. For CLIP, MaMe ($\tau = 0.8$) increased throughput by 25% (64.01 img/s) with 0.39% accuracy drop, while ToMe (r=12) gave 3% throughput gain with 4.34% accuracy loss. For SigLIP, MaMe ($\tau = 0.9$) improved throughput by 25% (58.10 img/s) with 1.11% accuracy reduction, while ToMe (r=32) achieved 19% speedup with 1.28% accuracy loss. For SigLIP2, MaMe ($\tau = 0.95$) increased throughput by 28% (56.15 img/s) with 0.35% accuracy drop, while ToMe (r=32) gave 16% throughput gain with 1.91% accuracy loss. SigLIP2's ability to merge tokens at $\tau = 0.95$ indicates its confident semantic representations. MaMe demonstrates better balance between throughput and accuracy versus baseline and ToMe.

Table 2. Comparative Evaluation of Vision Architectures with MaMe Compression. Throughput is measured on an A100 GPU with a batchsize 64 and FP32. The mark † means using MaMe.

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. Zero-shot image classification results. Inference throughput measured on a 3090 GPU with FP32 and batchsize 1.

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

Text-Image to Text We thoroughly investigated the impact of token merging on the LLaVA-1.5-7B model[21] within the VLMEvalKit framework[7], evaluating its performance

across a diverse suite of multimodal benchmarks[9, 16, 19, 22, 23, 25, 38]. The token merging techniques were applied to the visual encoder, aiming to **reduce the number of visual tokens fed into the large language model**, thereby enhancing computational efficiency. We compare the baseline LLaVA-1.5-7B against two distinct token merging strategies: ToMe, employing a fixed reduction number of $r = 8$ per layer, and MaMe, which utilizes a similarity threshold of $\tau = 0.8$.

Results in Table 4 demonstrate that ToMe and MaMe significantly accelerate evaluation time across benchmarks. MaMe achieves greater acceleration than ToMe while maintaining competitive or marginally superior metric scores across most benchmarks. On the COCO Caption[20] task, ToMe increased CIDEr to 1.60, while MaMe achieved **2.71**, representing a 3.8x improvement over baseline and a 69% gain compared to ToMe, indicating MaMe is helpful for better human consensus alignment.

Why does MaMe enhance CIDEr? We think MaMe acts as a **high-pass filter** by adaptively compressing redundant, low-frequency information (e.g., smooth image regions) while preserving and passing unique, high-frequency tokens (e.g., edges and textures) unchanged through attention layers. And then the reduction of token count happens to mitigate the "attention dilution" effect, also known as "rank-collapse" or "token uniformity,"[2, 5, 11] where a large softmax denominator in MSA disperses finite attention capacity, suppressing significant tokens. Unique tokens, previously diluted, now compete within a smaller candidate set, thereby receiving higher attention scores, which means **amplifying high-frequency signals.** The preserved and enhanced details improve visual responses, indicating the MaMe's potential for enhancing the Vision Language Model / Vision Language Action.

4.2. Video classification

We apply token merging to VideoMAE[32] models' vision encoder and compare MaMe with ToMe on Kinetics-400 validation set[14]. For evaluation, 16 frames were sampled per video clip, each resized to a 224×224 resolution.

Results in Table 6 show token merging accelerates video transformer inference. For VideoMAE-B, the baseline achieved 76.81% accuracy with 13.24 videos/s throughput. MaMe($\tau = 0.9$) maintained 76.03% accuracy while reaching 13.33 videos/s, outperforming ToMe(r=128) at 14.06 videos/s but 3.47% lower accuracy. MaMe($\tau = 0.85$) achieved 13.81 videos/s comparable to ToMe(r=96). With VideoMAE-L, the baseline achieved 82.31% accuracy at 6.25 videos/s. While ToMe(r=32) increased throughput to 6.97 videos/s, MaMe($\tau = 0.8$) reached 9.28 videos/s with 81.47% accuracy, showing a 49% speed increase. This demonstrates MaMe's effectiveness for larger models in balancing throughput gains with performance.

Table 6. Results on Kinetics-400. Inference throughput is measured on a 3090 GPU with FP16 precision and batchsize 1.

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

4.3. Ablation Study

4.3.1. Algorithmic Design Choices

Our ablation studies extend to the core algorithmic components of MaMe, with results summarized in Table 2.

Feature Choice: The raw token matrix `x` achieves optimal accuracy (83.35%) with competitive throughput. Features `k` and `k-mean` show lower accuracy (71.63% and 69.01%), confirming `x` preserves the most discriminative information for merging decisions.

Similarity Function: Cosine similarity achieves the best balance (83.35% accuracy, 73.06 im/s throughput). Dot product improves throughput (81.14 im/s) but sacrifices accuracy (80.43%), while Euclidean and softmax-based methods underperform in either metric.

Partition Style: Sequential ordering maximizes accuracy (84.14%) but reduces throughput (71.81 im/s). Alternating order offers the best trade-off (83.35% accuracy, 73.06 im/s), outperforming random ordering, which slightly boosts throughput at the cost of accuracy.

Adaptive Weight Pruning: AugReg models require pruning to achieve best 83.35% accuracy. MAE models show moderate gains but remain less dependent on pruning.

4.3.2. Where and What

To investigate where MaMe should be applied within models and what similarity threshold yields optimal performance, we examine the joint impact of similarity threshold (τ) and the number of blocks applying token merging (num_block) on both accuracy and throughput on ViT models as shown in Figure 4.

Accuracy The relationship between similarity threshold and accuracy shows a non-linear pattern across ViT architectures, influenced by MaMe applied block depth. As the threshold increases from 0.6 to 0.8, accuracy improves rapidly before plateauing, indicating diminishing returns

Table 4. 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.

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 (τ=0.8)	31.40	447	30.56	422	64.47	478	57.20	2840	30.27	531	45.62	1659	60.48	1020

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.

src	pruning	acc	im/s
mae		77.51	85.59
mae	✓	80.02	78.96
augreg		72.47	78.59
augreg	✓	83.35	73.06

d **Adaptive Weight Pruning.** AugReg models need pruning.

Figure 2. Ablation experiments using AugReg ViT-B/16. Our default settings are marked in Gray. We report Top-1 accuracy (acc) with FP32 precision and model inference throughput (im/s) on a 3090 GPU. The visualization results of different methods are shown in Figure 3.

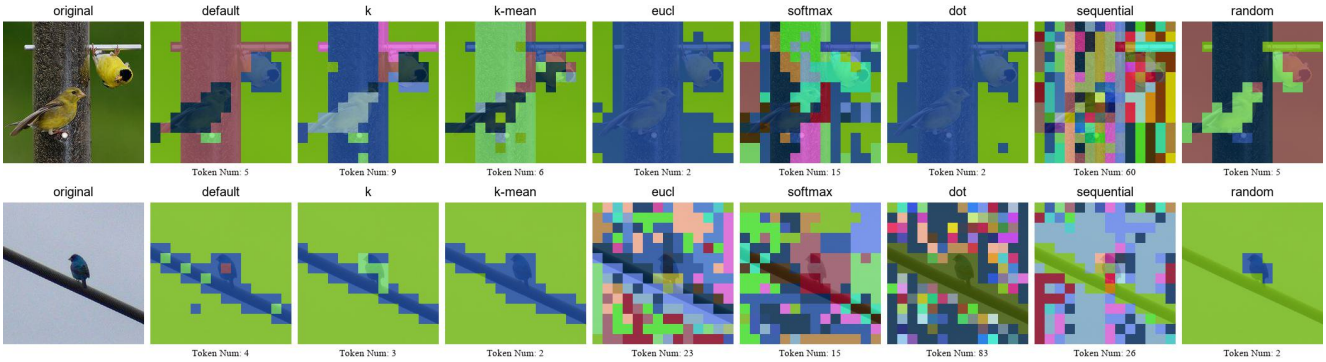


Figure 3. The visualization in the 8th block of the AugReg ViT-B/16 using MaMe with different settings. Each color square represents a distinct type of token. Default is our default method. More results are in the supplementary materials.

Table 5. COCO caption task performance of the LLaVA-1.5-7B model and its modified versions by ToMe and MaMe.

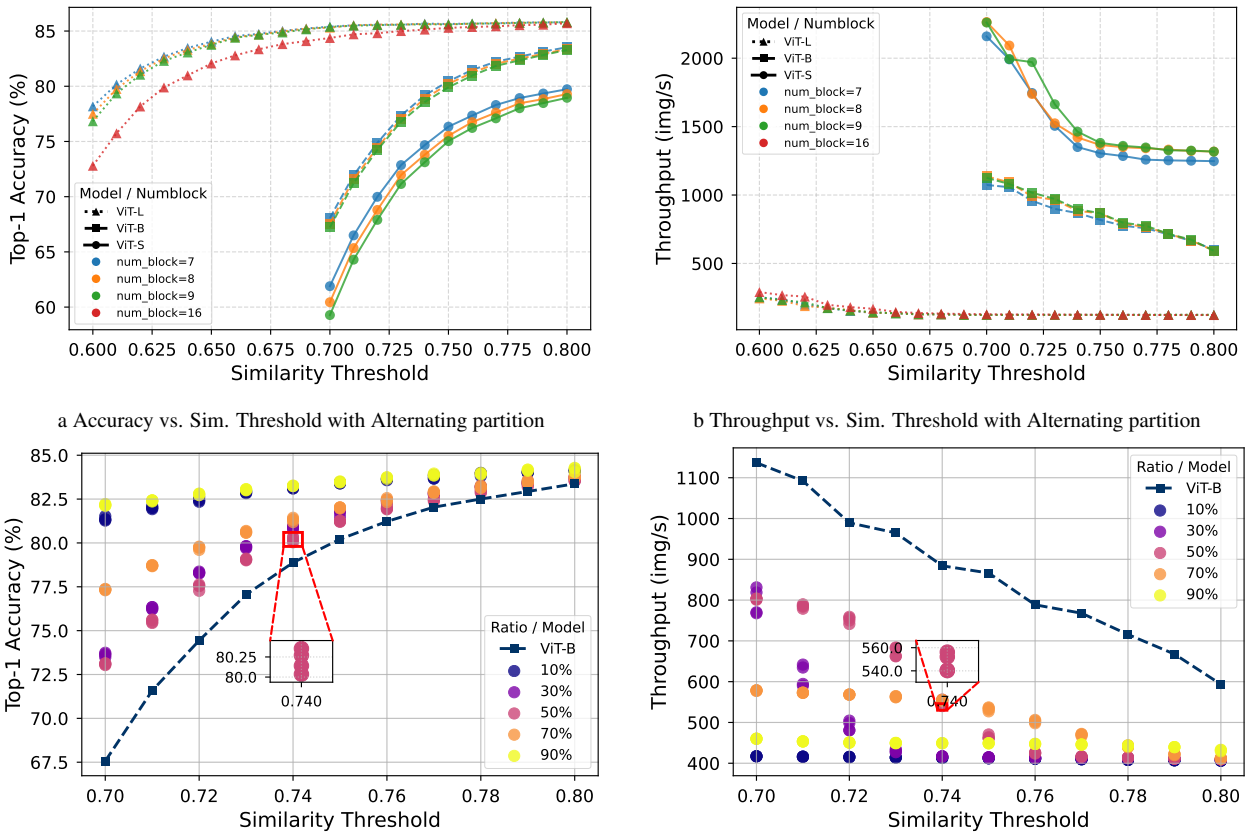
Method	Latency(s)	Bleu-1	Bleu-2	Bleu-3	Bleu-4	ROUGE_L	CIDEr
LLaVA-1.5-7B	3.12	20.72	13.28	8.08	4.93	20.94	0.71
+ ToMe	2.30	20.15	12.90	7.90	4.89	21.67	1.60
+ MaMe	2.61	20.10	12.87	7.87	4.83	21.69	**2.71**

from stricter token retention and suggesting that exceeding a critical threshold sufficiently distinguishes features.

Throughput Throughput monotonically decreases with similarity threshold, dropping sharply at low thresholds due to rising computational costs. Threshold sensitivity inversely correlates with model size: ViT-S experiences the steepest decline, followed by ViT-B and ViT-L, indicating larger models better mitigate merging overhead.

Model-Scale Sensitivity Model sensitivity to token merging varies by size: ViT-L maintains >85% accuracy across thresholds (0.6–0.8) with throughput gains, ViT-B shows moderate sensitivity, and ViT-S is most vulnerable (accuracy drops from 79% to 60% with aggressive merging). Larger models exhibit greater representational redundancy, allowing coarser merging with minimal performance loss.

Random Partition The comparison between deterministic default configurations (dashed lines) and stochastic trials (scatter points) indicates that the default settings define a Pareto frontier: stochastic partitions yield higher accuracy (points above the dashed line) but lower throughput (points below the dashed line). This trade-off suggests that while stochasticity enhances accuracy, it undermines computational efficiency. The deterministic, alternating partition thus serves as a robust baseline, balancing performance and efficiency.



a Accuracy vs. Sim. Threshold with Alternating partition

b Throughput vs. Sim. Threshold with Alternating partition

c Accuracy vs. Sim. Threshold with Random partition

d Throughput vs. Sim. Threshold with Random partition

Figure 4. The accuracy and throughput change with the similarity threshold under both alternating and random partitions. There is a non-linear, saturating relationship between the similarity threshold and accuracy across ViT architectures. Five different random seeds are employed to conduct five experiments under different ratios of tokens as source tokens as shown in the box in the Figures (c) and (d), illustrating that the default, determined alternating partition curves represent a Pareto frontier.

5. Discussion

Pros and Cons One of the primary advantages of MaMe is its non-intrusive nature with respect to standard attention mechanisms. Unlike ToMe, which requires attention modifications, MaMe preserves the standard attention calculation and easily integrates with optimized implementations like Flash Attention[4]. Its full-matrix operations are GPU-friendly, avoiding hard-to-optimize sorting operations. However, the optimal similarity threshold (τ) requires manual determination. Future work will develop automated methods to determine it.

Migrating to LLMs Token merging approaches like ToMe break causality in autoregressive LLMs by allowing tokens to merge with future tokens. MaMe enables **causal token merging** by partitioning tokens into odd (destination) and even (source) sets, then applying a causality mask M ($M_{i,j} = 1$ if $j \leq i$, else 0) to zero upper triangular part of fusion weights W_{ij}^{F} by $W_{ij}^{\mathrm{F}} \odot M_{ij}$. This restricts each destination token (e.g., token 5) to merge only with preceding source tokens (e.g., tokens 0, 2, 4), preserving causality and allowing reduce KV cache length in LLMs.

6. Conclusion

In this work, we introduced MaMe, a differentiable, training-free token merging method based on full-matrix operations. Through extensive experimentation, including accelerating off-the-shelf and from-scratch trained ViT models, as well as video classification models, we demonstrated that MaMe, like ToMe, achieves significant inference speedups, typically with a trade-off in performance metrics. However, MaMe achieves simultaneous improvements in both performance and speed on COCO caption task, indicating MaMe's potential for optimizing multimodal models. Lastly, we discussed MaMe's causal token merging potential for application in large language models to compress KV cache while preserving causality.

References

[1] 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. 1, 2

[2] Giuseppe Bruno, Federico Pasqualotto, and Andrea Agazzi. Emergence of meta-stable clustering in mean-field transformer models. In *International Conference on Learning Representations*, 2025. 6

[3] 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*, pages 17164–17174, 2023. 1, 2, 4

[4] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024. 8

[5] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International conference on machine learning*, pages 2793–2803. PMLR, 2021. 6

[6] 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. 1

[7] 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*, pages 11198–11201, 2024. 5

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

[9] 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. 6

[10] 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. 2

[11] Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. *Advances in Neural Information Processing Systems*, 36:57026–57037, 2023. 6

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 3

[13] Simin Huo and Ning Li. Iwin transformer: Hierarchical vision transformer using interleaved windows, 2025. 4

[14] 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. 6

[15] 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*, pages 1454–1462, 2023. 2

[16] Bohao Li, Yuanhan Zhang, Sheng Li, Gengyun Chen, Jing Yang, Guangzhi Chen, Ruisi He, Wen-e Liu, Huijuan Wang, Fang Wen, et al. SEED-Bench: Benchmarking multimodal LLMs with text-rich visual comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 6

[17] Bo Li, Wei Zhao, and Zhi Zhang. LTPM: A learnable token pruning and merging method for vision transformer. *arXiv preprint arXiv:2406.01289*, 2024. 2

[18] 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. 1, 2

[19] 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. 6

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014. 6

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 5

[22] 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. 6

[23] 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*, 2023. 6

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 4

[25] 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. 6

[26] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel.

Token pooling in vision transformers. *arXiv preprint arXiv:2110.03860*, 2021. 1

[27] 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*, pages 12309–12318, 2022. 2

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 5

[29] 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. 1, 2

[30] 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. 4

[31] 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. 2

[32] 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. 6

[33] 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*, pages 10347–10357. PMLR, 2021. 3

[34] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. 5

[35] 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. 1

[36] 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*, pages 16584–16593, 2023. 2

[37] Yutong Xie, Jianpeng Zhang, Yong Xia, Anton van den Hengel, and Qi Wu. Clustr: Exploring efficient self-attention via clustering for vision transformers, 2022. 2

[38] 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. 6

[39] Fanhu Zeng, Deli Yu, Zhenglun Kong, and Hao Tang. Token transforming: A unified and training-free token compression framework for vision transformer acceleration, 2025. 1

[40] 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. 2

[41] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. 5