AdaptMerge: Inference Time Adaptive Visual and Language-Guided Token Merging for Efficient Large Multimodal Models

Zahidul Islam
University of Saskatchewan, Canada
eqm971@usask.ca

Mrigank Rochan
University of Saskatchewan, Canada
mrochan@cs.usask.ca

Abstract

Recent advances in Large Multimodal Models (LMMs) have showcased impressive visual understanding and vision-language reasoning capabilities, yet their computational cost hinders practical deployment, especially in resource-constrained settings. A key bottleneck is the large number of visual tokens generated by its vision encoders, which increases latency and memory demands. Existing token reduction methods often require costly fine-tuning or apply fixed token reduction ratios, ignoring image complexity and vision-language interactions. We propose *AdaptMerge*, a training-free, inference-time token merging strategy that adaptively reduces visual tokens by leveraging feature diversity and language-guided relevance. By dynamically adjusting to image complexity and ensuring multimodal coherence, AdaptMerge significantly lowers floating-point operations while improving performance. Extensive experiments on Google's latest Gemma 3 models (4B and 12B parameters) across four challenging benchmarks demonstrate that AdaptMerge outperforms state-of-the-art token reduction techniques, achieving both reduced computational costs and improved performance, thereby providing a practical pathway to more efficient LMMs.

1 Introduction

Recent advances in Large Multimodal Models (LMMs) (Liu et al., 2023; Beyer et al., 2024; Team et al., 2025) have demonstrated remarkable capabilities in visual understanding and vision-language reasoning. LMMs typically consist of a vision encoder (Dosovitskiy et al., 2021; Radford et al., 2021; Zhai et al., 2023) responsible for processing visual information such as images into visual tokens or embeddings and a Large Language Model (LLM) that processes the combined visual and text tokens to perform sophisticated tasks requiring complex multimodal understanding, such as visual question answering and image captioning. However, the computational demands of LMMs during inference remain a significant barrier to their practical adoption and efficient deployment, particularly in resource-constrained environments. A primary contributor to this challenge is the large number of visual tokens generated by vision encoders, which scales with image resolution and model size, leading to increased latency and substantial computational requirements (Bolya et al., 2023; Shang et al., 2025; Sun et al., 2025; Alvar et al., 2025).

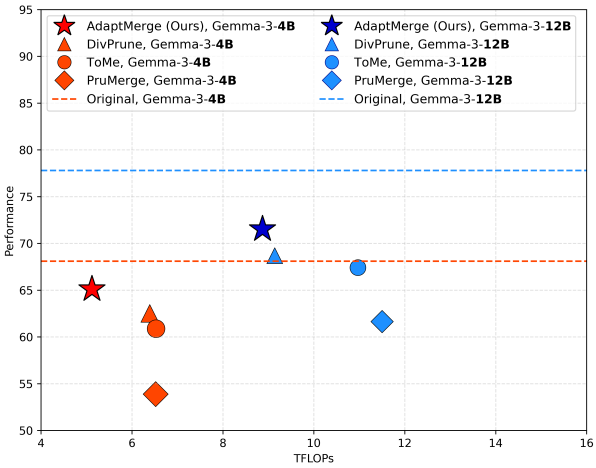

Figure 1: Comparison of different token reduction methods for Google's latest high-performing Gemma-3-4B and Gemma-3-12B LMMs during inference on the AI2D dataset. Our method, AdaptMerge, reduces computational cost (measured in TFLOPs) while surpassing state-of-the-art methods by a substantial margin.

To mitigate these challenges, token reduction techniques have been developed to streamline visual processing in LMMs while preserving performance. However, existing methods face several limitations. Many approaches require fine-tuning or calibration on large datasets, which is computationally intensive and impractical due to substantial GPU memory and training time requirements (Cai et al., 2025; Li et al., 2024; Ye et al., 2025; Lin

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 7352–7361
November 4-9, 2025

et al., 2025; Sun et al., 2025). Additionally, most token reduction methods apply a fixed reduction ratio across all images, lacking adaptiveness to image complexity. This one-size-fits-all approach can lead to suboptimal outcomes: images with sparse details, such as natural scenes, could tolerate aggressive token reduction, whereas images with fine-grained details, like wildlife or satellite images, require a more conservative strategy to preserve critical information. Furthermore, the majority of current methods (Bolya et al., 2023; Ye et al., 2025; Shang et al., 2025; Alvar et al., 2025) focus solely on visual information for token reduction, overlooking the multimodal nature of LMMs and the relevance and relationship of visual tokens to the textual input, which is vital for optimizing performance in vision-language tasks.

In this paper, we propose *AdaptMerge*, a novel inference-time adaptive token reduction strategy that integrates visual and language guidance to enhance the efficiency of LMMs. AdaptMerge is entirely training-free, eliminating the need for costly retraining or fine-tuning. For the unimodal vision encoder, AdaptMerge employs diversity-guided visual token merging, which dynamically reduces tokens by prioritizing significant visual tokens while merging redundant or less informative ones in each encoder layer based on their feature diversity, adapting to the complexity of input images. Subsequently, it performs language-guided visual token merging on the visual tokens fed into the LLM, prioritizing those most relevant to the text prompt to reduce LLM input tokens. By dynamically consolidating visual tokens during inference, AdaptMerge strikes an optimal balance between computational cost and performance, reducing floating-point operations (FLOPs) while improving accuracy across multiple multimodal benchmarks compared to existing token reduction techniques (Fig. 1). As a plug-and-play solution, AdaptMerge paves the way for broader adoption of LMMs in real-world applications.

In summary, we make the following key contributions:

- We highlight the importance of adaptive visual token reduction in LMMs, which integrates both visual and language inputs to improve its efficiency.
- We introduce *AdaptMerge*, a novel inference-time visual token reduction strategy that enhances the efficiency of LMMs through a training-free, adaptive, and multimodal approach guided by both visual and language cues.
- We conduct extensive experiments on the recently released 4B and 12B versions of Google's powerful Gemma 3 models across four standard multimodal benchmarks, showing that AdaptMerge outperforms state-of-the-art token reduction methods by reducing floating-point operations while achieving notable performance gains.

2 Related Work

2.1 Efficient Large Multimodal Models (LMMs)

Recent state-of-the-art LMMs (Team et al., 2025; Beyer et al., 2024; Liu et al., 2023; Bai et al., 2023) have demonstrated impressive performance on various mutimodal tasks including image captioning (Agrawal et al., 2019), visual question answering (Goyal et al., 2017; Marino et al., 2019; Kembhavi et al., 2016), OCR (Singh et al., 2019; Mathew et al., 2021), and so on. These models have to process a large number of visual tokens, driving up both inference time and computational costs. A number of recent works therefore aim to improve efficiency of LMMs with minimal performance degradation. For example, Mamba (Gu and Dao, 2024) uses a selective state-space architecture to reduce compute per token, while BLIP-2 (Li et al., 2023) uses a specific vision encoder called Q-Former (Zhang et al., 2024) that condenses image features into a smaller set of visual tokens. Orthogonal approaches such as weight quantization (Gong et al., 2014), layer skipping (Shukor and Cord, 2024), knowledge distillation (Wang et al., 2020) into smaller LLM backbones can also cut inference cost. However, these methods rely on costly re-training or fine-tuning. In contrast, various token reduction methods (Alvar et al., 2025; Chen et al., 2024) including ours achieve computational savings while leaving the underlying model architecture and pretrained weights untouched.

2.2 Visual Token Reduction in LMMs

As the computational demand of LMMs scales quadratically with the token count, the sheer number of visual tokens, especially when processing high-resolution images, hampers the practicality of LMMs in real-world applications. Consequently, recent studies has focused on reducing these visual tokens through pruning and merging to boost LMM efficiency while minimizing performance sacrifice

(Alvar et al., 2025; Chen et al., 2024; Lin et al., 2025; Shang et al., 2025). Token pruning methods select a subset of the visual tokens based on some importance criteria and simply discard the rest (Alvar et al., 2025), whereas token merging methods show that preserving the less important tokens by merging them back into the selected ones is beneficial for fine-grained analysis (Shang et al., 2025; Bolya et al., 2023). ToMe (Bolya et al., 2023), a seminal work on token merging, proposes to merge visual tokens in ViT vision encoder (Dosovitskiy et al., 2021) using bipartite matching. Our AdaptMerge framework adopts token merging as a token reduction strategy. Fine-tuning and calibration based methods achieve token reduction (Sun et al., 2025; Zhang et al., 2025; Ye et al., 2025) at the expense of additional computational cost. FitPrune (Ye et al., 2025) formulates a pruning recipe based on the attention divergence statistics while M^3 (Cai et al., 2025) employs model fine-tuning for training nested visual token representations at various scales. To forgo additional fine-tuning or calibration, several works have proposed training-free token reduction methods. Among them, PruMerge (Shang et al., 2025) and FastV (Chen et al., 2024) employs attention scores to prune their visual tokens. However, it is shown that this can lead to a selection of redundant visual tokens (Lin et al., 2025). DivPrune (Alvar et al., 2025) prunes a fixed ratio of visual tokens and uses an algorithm based on MMDP (Resende et al., 2010) to prune tokens. Whereas, PruMerge selects their reduction ratio using a simple outlier detection algorithm which may lead to suboptimal results. However, these methods do not reduce tokens in the vision encoder and solely rely on visual information to make token reduction decisions while ignoring the highly informative text prompt and its relevance to the visual tokens. In contrast, our AdaptMerge strategy reduces visual tokens in both the vision encoder and the LLM input leading to substantial compute savings. In addition, it is an adaptive approach that is guided by both visual and language cues.

3 Our Approach

We present *AdaptMerge*, an inference-time, plug-and-play adaptive token reduction strategy that can be easily retrofitted into LMMs. We begin by describing the general architecture of LMMs, followed by a description of AdaptMerge.

3.1 Large Multimodal Models (LMMs)

Given an image I and a text prompt or query Q, an LMM, such as Gemma 3, generates a text response R in an auto-regressive manner, leveraging the visual information in the image and the input text. The LMM processes input images using a vision encoder E, which is typically a transformer network (Vaswani et al., 2017) consisting of multiple sequential transformer layers. E first divides the input image I into p patches of fixed size. These p image patches are then processed to produce their corresponding visual tokens $V^E = \{v_1^E, \ldots, v_p^E\}$. These visual tokens pass through a projection module, which typically includes fully-connected and adaptive average pooling layers, to produce a fixed number of n visual tokens $V = \{v_1, \ldots, v_n\}$. Meanwhile, the associated text query is mapped to m text embeddings $Q = \{q_1, \ldots, q_m\}$. The LLM decoder receives the concatenated sequence $[V; Q]$ as input and generates the text response R.

3.2 AdaptMerge

Due to the high dimensionality of visual input, the number of visual tokens often far exceeds the number of text tokens. However, many of these visual tokens may be redundant and irrelevant to the associated text prompt. As a result, effectively compressing visual information by reducing the number of visual tokens has significant potential to reduce the computational load of LMMs during inference. Reducing a fixed number or ratio of visual tokens is suboptimal, as it does not account for the complexity of input images. For instance, information-dense inputs, such as wildlife images containing complex animal behavior or environmental features, require more tokens to capture fine details. In contrast, sparse natural scenes or images dominated by irrelevant regions can be processed with far fewer tokens. To address this, we propose AdaptMerge, an adaptive approach that reduces visual tokens during inference in LMMs by merging them based on their visual redundancy and relevance to the text prompt. AdaptMerge performs token reduction in two stages (Fig. 2): Adaptive Visual Token Merging (AVTM) and Adaptive Language-Guided Visual Token Merging (ALVTM).

3.2.1 Adaptive Visual Token Merging (AVTM)

Depending on the patch size and image, the vision encoder processes thousands of visual tokens across multiple sequential transformer layers, often resulting in significant redundant computation. To

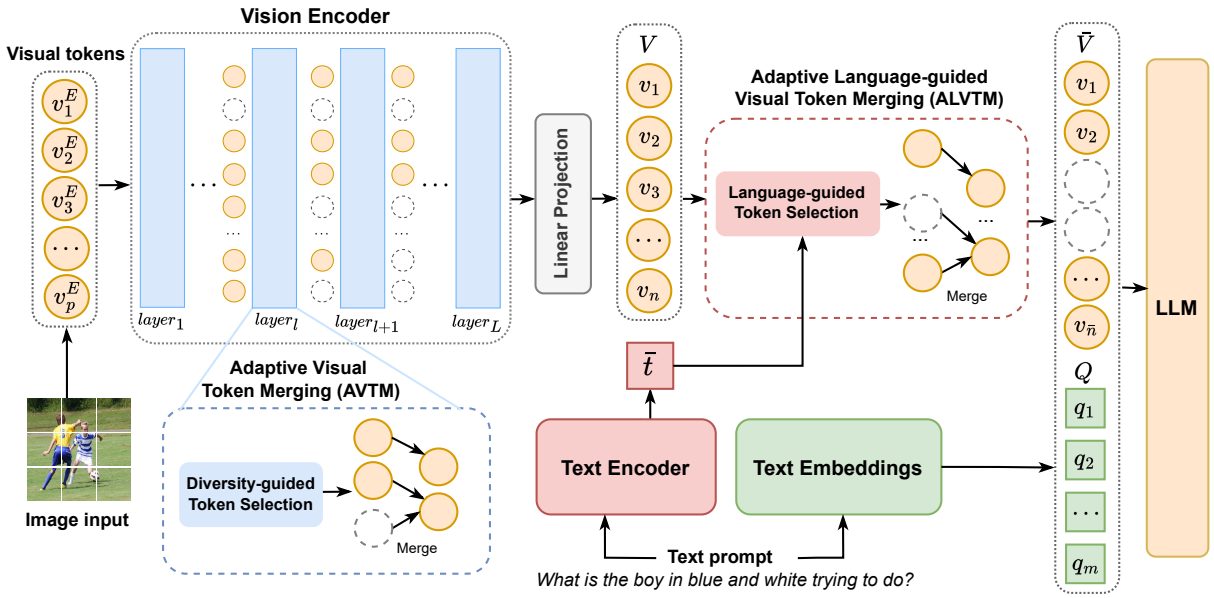

Figure 2: Overview of our AdaptMerge strategy for adaptive visual token reduction during inference in the LMM. AdaptMerge reduces visual tokens at two stages: within the vision encoder using Adaptive Visual Token Merging (AVTM), and in the input to the LLM using Adaptive Language-Guided Visual Token Merging (ALVTM). AVTM selects and merges tokens based on visual diversity at each layer of the vision encoder. ALVTM performs language-guided merging by leveraging text prompt features from a text encoder jointly trained with the vision encoder. The final reduced set of visual tokens is concatenated with the text prompt tokens and fed to the LLM for text generation.

address this, we propose prioritizing important visual tokens while merging redundant ones at each layer of the vision encoder, progressively reducing their number. Specifically, we reduce visual tokens based on their visual diversity, which helps mitigate redundancy and enables more efficient processing.

Let the vision encoder consist of L layers, and denote the input to layer l as $X^l = \{x_1^l, \ldots, x_{p^l}^l\}$, where p^l represents the number of visual tokens at layer l. At the first layer, the input is $X^1 = V^E$, with $p^1 = p$, where p is the number of image patches, and V^E is the set of p visual tokens corresponding to these patches. Applying adaptive token merging at each layer progressively reduces the number of visual tokens; therefore, for any layer l, it holds that $p^{l+1} < p^l$.

We measure the diversity score d_i of the i-th visual token x_i^l by computing its average similarity to all other visual tokens in the same layer. To reduce computational overhead, instead of using the token embeddings directly, we use their corresponding key vectors from the attention mechanism of the transformer layer, as these key vectors serve as compressed representations of the tokens. Hence, the diversity score d_i for the i-th token in layer l can be written as:

$$d_i = 1 - \frac{1}{p^l - 1} \sum_{i \neq j} \text{sim}(x_i^l, x_j^l) \quad (1)$$

$$\text{sim}(x_i^l, x_j^l) = \text{cosine}\big(k_i^l,\, k_j^l\big) \quad (2)$$

A low value of d_i suggests that the visual token x_i^l carries less informative content. We use the corresponding attention key vector k_i^l as a compact representation of x_i^l. We select a subset of tokens $\mathcal{K}$ with high diversity scores (i.e., $d_i \geq \rho$) as they contain significant information. For each unselected token in the remaining set $\mathcal{M}$, we merge it into its most similar token in $\mathcal{K}$ by averaging. This strategy minimizes information loss and helps maintain accurate predictions.

AVTM retains more visual tokens for images with fine-grained details due to higher token diversity, while it aggressively merges tokens for simpler images with lower diversity. This way, AVTM adaptively reduces computational cost based on input image complexity. We apply AVTM between the Attention and MLP blocks in each vision encoder layer, significantly reducing the computational load by lowering the quadratic cost of attention computations.

3.2.2 Adaptive Language-Guided Visual Token Merging (ALVTM)

In the LMM, there is a linear projection module next to the vision encoder that performs average pooling to produce a fixed number of visual tokens for the LLM. While AVTM reduces the token count within the vision encoder, the projection module still outputs that fixed number of tokens for the LLM. Hence, to reduce the computational load of the LLM, we next apply ALVTM, an additional adaptive merging step before feeding the tokens into the LLM. Unlike AVTM, ALVTM leverages multimodal information by using the text prompt to guide token reduction. It selectively retains visual tokens most relevant to the prompt, while merging the less relevant ones.

Given a set of n visual tokens $V = \{v_1, \ldots, v_n\}$ as input to the LLM, our goal is to reduce it to a smaller set $\bar{V}$ containing $\bar{n}$ tokens, where $\bar{n} < n$. By forwarding only $\bar{n}$ visual tokens to the LLM, we reduce its overall computational load and inference time. We select tokens based on their relevance to the text prompt. LMMs typically use vision encoders (SigLIP (Zhai et al., 2023) in Gemma 3) pretrained jointly with a corresponding text encoder via contrastive learning, aligning visual and textual features from paired image-text datasets. We utilize this jointly trained text encoder (i.e., SigLIP text encoder) to generate a robust text feature representation $\bar{t}$ from the text prompt and use it to compute a text-relevance score r_i for each visual token v_i as follows:

$$r_i \;=\; \text{cosine}\big(\, v_i,\; \bar{t}\big) \quad \text{for } i = 1 \ldots n \qquad (3)$$

Here, $\bar{t}$ denotes the pooled text feature obtained from the text encoder for the prompt and cosine refers to the cosine similarity function.

Next, we select a subset of visual tokens $\mathcal{K}$ that are highly relevant to the text prompt, i.e., those satisfying $r_i \geq \tau$. To prevent the loss of important visual information, particularly crucial for fine-grained, detail-sensitive tasks, we adopt a merging strategy similar to AVTM. Each unselected token in the remaining set $\mathcal{M}$ is merged into its most similar token in $\mathcal{K}$ by averaging. This preserves essential image context while reducing the number of visual tokens passed to the LLM. Finally, the resulting $\bar{n}$ visual tokens are then concatenated with the LMM text embeddings Q and fed into the LLM to generate the output text.

ALVTM is language-guided and adaptive, as it selects prompt-relevant visual tokens while merging less informative ones, resulting in a variable-length set of visual tokens for different input images. Additionally, since the text encoder used for ALVTM is explicitly trained via contrastive learning to align textual and visual features, extracting the text feature $\bar{t}$ from it produces more meaningful text-relevance scores while introducing only negligible computational overhead (less than 0.01 TFLOPs). Nevertheless, we experiment with using LMM text embeddings directly to compute text-relevance scores (see Table 4). Although this approach achieves reasonable performance, features from the jointly trained text encoder perform better, as they are explicitly optimized for alignment with visual features.

4 Experiments

In this section, we first lay out our experimental settings and implementation details. Then, we showcase the effectiveness of our method on several standard multimodal benchmarks.

4.1 Datasets and Settings

Datasets. We evaluate our method on four challenging image-language benchmarks focused on multimodal reasoning and understanding tasks, namely AI2D (Kembhavi et al., 2016), SQA (Lu et al., 2022) (ScienceQA-IMG), OKVQA (Marino et al., 2019), and TextVQA (Singh et al., 2019). These datasets originally consist of around 15k, 14k, 21k, and 45k questions, respectively. Since our method is applied at inference time, we use only the respective test or validation splits to report the performance. SQA, AI2D, and OKVQA contains around 2k, 3k, and 5k test samples, respectively. We evaluate on the TextVQA validation set which contains around 5k samples.

AI2D (Kembhavi et al., 2016) is a popular benchmark for visual question answering related to science diagrams. SQA (Lu et al., 2022) includes a wide range of questions across topics in natural sciences, social studies, and linguistics. Both datasets provide an associated multiple-choice question for each image. On the other hand, Outside Knowledge VQA (OKVQA) (Marino et al., 2019) is used to evaluate open-ended question answering based on images, whereas TextVQA (Singh et al., 2019) focuses on evaluating a language-and-vision model's ability to read and understand textual information from images.

Model	Method	TFLOPs ↓	OKVQA ↑	AI2D ↑	SQA ↑	TextVQA (val) ↑
Gemma-3-4B	Original	7.91	53.67	68.10	70.55	65.50
	PruMerge	6.52	47.82	53.89	62.77	40.56
	ToMe	6.43	48.64	60.88	68.42	50.14
	DivPrune	6.39	49.62	62.53	69.11	57.58
	AdaptMerge (Ours)	**5.12**	**53.47**	**65.12**	**70.02**	**61.44**
Gemma-3-12B	Original	12.87	59.77	77.82	76.45	75.96
	PruMerge	11.50	52.17	61.64	73.55	60.55
	ToMe	10.97	55.58	67.42	74.47	64.31
	DivPrune	9.14	59.51	68.69	76.0	68.20
	AdaptMerge (Ours)	**8.87**	**59.71**	**71.53**	**76.60**	**71.56**

Table 1: Comparison of performance and inference TFLOPs for the Original Gemma-3-4B and Gemma-3-12B models, baseline methods, and AdaptMerge across four multimodal benchmarks.

Models and baseline comparisons. We evaluate the performance of our token reduction strategy, *AdaptMerge*, on two high-performing multimodal models from Google's latest Gemma 3 family (Team et al., 2025): Gemma-3-4B and the larger Gemma-3-12B, using their official checkpoints. For comparison, we implement three state-of-the-art token reduction baselines using their public code: (i) *PruMerge* (Shang et al., 2025), a training-free method that adaptively selects key visual tokens augmented with uniformly sampled ones; (ii) *ToMe* (Bolya et al., 2023), which merges tokens iteratively within the vision encoder using bipartite matching; and (iii) *DivPrune* (Alvar et al., 2025), an inference-time method that selects diverse output tokens from the projection layer. Additionally, we compare against the original Gemma-3 models without any token reduction, which serve as upper-bound baselines and are referred to as *Original* in our results. Note that DivPrune and ToMe are not adaptive and use a fixed token reduction mechanism, whereas PruMerge is adaptive. For a fair comparison with fixed token reduction methods, specifically DivPrune and ToMe, we follow prior work (Alvar et al., 2025) and configure their token reduction mechanisms to have a slight computational advantage over ours.

Metrics. Following the evaluation protocols of prior work (Alvar et al., 2025; Team et al., 2025; Shang et al., 2025), we use the Exact Match (EM) metric for performance evaluation. For Gemma 3 models (Team et al., 2025), we use their standard evaluation settings and employ few-shot prompting with four example image-question-answer pairs from training data and apply pan and scan, which converts images with wide aspect ratio in multiple dynamically cropped sqaure regions to enhance their performance. We measure efficiency via teraFLOPs (i.e., TFLOPs), inference time, and throughput.

Implementation details. The Gemma-3-4B and Gemma-3-12B models, with approximately 4 billion and 12 billion parameters respectively, process 896×896 input images using the SigLIP vision encoder (Zhai et al., 2023), which generates 4,096 initial visual tokens from 4,096 image patches. The linear projector following the vision encoder in Gemma 3 outputs a fixed set of 256 visual tokens as input to the LLM. The AVTM component (Sec. 3.2.1) of AdaptMerge adaptively reduces the 4,096 initial visual tokens within each layer of the vision encoder using a diversity threshold of $\rho = 0.05$, while ALVTM (Sec. 3.2.2) reduces the 256 projected tokens for LLM input based on a text relevance threshold of $\tau = 0.15$. We conduct experiments on NVIDIA L40 GPUs for Gemma-3-4B and A100 GPUs for Gemma-3-12B.

4.2 Comparisons with State-of-the-Art

In Table 1, we compare AdaptMerge with baseline methods for Gemma-3-4B and Gemma-3-12B, reporting performance and computational cost in TFLOPs across the four multimodal benchmarks. Note that TFLOPs are measured on the AI2D dataset, as discussed in the following section on the detailed efficiency analysis. By adaptively merging tokens in the vision encoder and LLM input based on visual redundancy and text relevance, AdaptMerge significantly reduces TFLOPs while achieving higher accuracy than state-of-the-art baseline methods on all datasets. Notably, on the TextVQA dataset, a key benchmark which evaluates the challenging ability to read and reason about text in

Method	TFLOPs ↓	Inference Time ↓ (sec.)	Vision Enc. Time ↓ (sec.)	LLM Time ↓ (sec.)	Throughput ↑ (sample/sec.)
Original	7.91	0.711	0.532	0.172	1.41
PruMerge	6.52	0.645	0.420	0.115	1.55
ToMe	6.43	0.481	0.316	0.122	2.08
DivPrune	6.39	0.503	0.423	**0.059**	1.99
AdaptMerge (Ours)	**5.12**	**0.399**	**0.203**	0.102	**2.51**

Table 2: Detailed inference efficiency comparison among Original Gemma-3-4B, baseline methods, and AdaptMerge.

images for answering questions, AdaptMerge significantly minimizes performance degradation compared to the Original models, demonstrating its ability to enhance computational efficiency while maintaining strong performance. It also shows that AdaptMerge preserves critical visual details necessary for fine-grained analysis and understanding.

4.3 Ablation Studies

To analyze the contribution of various components of our method, we perform several ablation experiments. We use the Gamma-3-4B model and the AI2D dataset for all ablation experiments and efficiency analysis.

Detailed efficiency analysis. Table 2 summarizes the detailed inference efficiency of AdaptMerge, comparing to the Original Gemma-3-4B model and baselines, reporting TFLOPs, run-time metrics including inference time, vision encoder time, LLM time, and throughput on an NVIDIA L40 GPU. We report the average of TFLOPs and run-time metrics for each test image instance of AI2D dataset without adding additional images from few-shot prompting or pan and scan. AdaptMerge reduces total TFLOPs from 7.91 to 5.12 ($\sim$35% reduction) and end-to-end latency from 0.71 s to 0.40 s ($\sim$44% improvement), surpassing baselines. Despite higher computational loads, DivPrune, ToMe, and PruMerge achieve lower accuracy. AdaptMerge halves vision encoder time and boosts throughput from 1.41 to 2.51 samples/sec., outperforming all baselines. This demonstrates AdaptMerge's ability to minimize computational cost through adaptive token merging while preserving accuracy.

Impact of AVTM and ALVTM. AdaptMerge proposes reducing visual tokens in both the vision encoder and the LLM input. AVTM adaptively merges tokens in each vision encoder layer based on visual redundancy, while ALVTM reduces LLM input tokens based on text prompt relevance. Table 3 quantifies their contributions. With larger number of tokens in the vision encoder, AVTM achieves a $\sim$23% TFLOPs reduction with minimal performance loss, whereas ALVTM offers smaller savings. Importantly, combining both offers over $\sim$35% TFLOPs reduction with only marginal performance loss, demonstrating their complementary efficiency gains.

AVTM	ALVTM	TFLOPs	AI2D
✗	✗	7.91	68.10
✓	✗	6.08	**65.42**
✗	✓	7.17	65.38
✓	✓	**5.12**	65.12

Table 3: Ablation study on the relative impact of AVTM and ALVTM in AdaptMerge.

SigLIP text features vs. LLM text embeddings: Table 4 compares our ALVTM under two choices for computing the text-relevance scores (Eq. 3). When using raw LMM text embeddings to compute text-relevance with the visual tokens from the linear projection, we achieve similar compute reduction in TFLOPs but observe a larger performance drop. This occurs because the alignment between the LLM embeddings and the projected visual tokens is achieved only indirectly through joint training for auto-regressive text generation, and their embedding space is not explicitly optimized for alignment with visual tokens. In contrast, guiding AdaptMerge's ALVTM using SigLIP text encoder features achieves nearly the same compute savings while maintaining higher performance. This indicates that SigLIP's text encoder representations are more effective for computing the text-relevance scores in ALVTM.

Token merging vs. token dropping. Table 5 compares the effects of visual token merging (*Merge*) in AdaptMerge versus discarding (*Drop*) unselected

Method	TFLOPs	AI2D
Original	7.910	68.10
LMM text embeddings	**5.110**	62.85
SigLIP text features (Ours)	5.118	**65.12**

Table 4: Comparison of SigLIP text encoder features (Ours) vs. LLM text embeddings for computing text-relevance scores in ALVTM.

visual tokens with low diversity or text relevance. Merging unselected tokens into similar selected ones preserves higher task performance by retaining fine-grained details and preventing information loss critical for multimodal tasks, unlike dropping, which leads to information loss and lower performance.

Method	TFLOPs	AI2D
Original	7.91	68.10
Drop	5.12	62.98
Merge (Ours)	**5.12**	**65.12**

Table 5: Comparison of token reduction strategies in AdaptMerge: simply dropping them (Drop) vs. merging unselected visual tokens into retained ones (Merge).

Importance of selected tokens. To assess the importance of the tokens selected by AdaptMerge, we conduct the following experiment. In AVTM, we multiply each token's diversity score by -1, causing it to select less diverse tokens instead of the most diverse ones. Similarly, in ALVTM, we invert each token's text-relevance score by multiplying it by -1, leading it to select less relevant tokens. Additionally, we invert the respective selection thresholds for both AVTM and ALVTM. We observe a significant drop in performance under these inverted settings (Table 6), where unimportant tokens are selected (AdaptMerge (inverted)). This performance degradation provides quantitative evidence that the selection criteria introduced in AdaptMerge effectively identify important tokens.

Method	TFLOPs	AI2D
AdaptMerge (Ours)	**5.12**	**65.12**
AdaptMerge (inverted)	5.78	52.91

Table 6: Comparison of AdaptMerge and its inverted variant for token selection.

Effectiveness of AdaptMerge on another LMM. In our paper, we primarily focus on Google's latest Gemma 3 models (Team et al., 2025) due to their state-of-the-art performance. However, to further evaluate the generalizability of AdaptMerge across different models, we also conduct experiments on another popular LMM, LLaVA-1.5-7B (Liu et al., 2023). Once again, AdaptMerge outperforms existing state-of-the-art token reduction approaches on this model (Table 7), achieving superior compute efficiency and performance, thereby demonstrating its effectiveness on other models.

Method	TFLOPs	AI2D
Original	9.95	53.50
PruMerge	8.23	41.45
ToMe	7.91	49.64
DivPrune	7.68	49.31
AdaptMerge (Ours)	**7.56**	**51.30**

Table 7: Performance comparison of AdaptMerge on another LMM, LLaVA-1.5-7B.

Qualitative analysis. Figure 3 visualizes token selection by PruMerge, DivPrune, and AdaptMerge (Ours) for three sample images, with each row representing one method and red masks indicating key selected tokens. AdaptMerge dynamically selects visual tokens based on image complexity and text relevance, assigning more to information-dense images and fewer to sparse ones. Unlike the less adaptive or static strategies of PruMerge and DivPrune, AdaptMerge's adaptive approach preserves fine-grained and relevant visual details, enhancing multimodal task performance while reducing TFLOPs.

5 Conclusion

We propose AdaptMerge, a novel inference-time, training-free adaptive visual token merging strategy that represents a significant advancement in the pursuit of efficient LMMs, facilitating the broader adoption of multimodal AI in real-world applications. By addressing the computational challenges of LMMs through an innovative integration of adaptiveness with both visual and language guidance, AdaptMerge outperforms state-of-the-art token reduction methods, achieving lower computational cost while enhancing task performance across diverse image-language benchmarks.

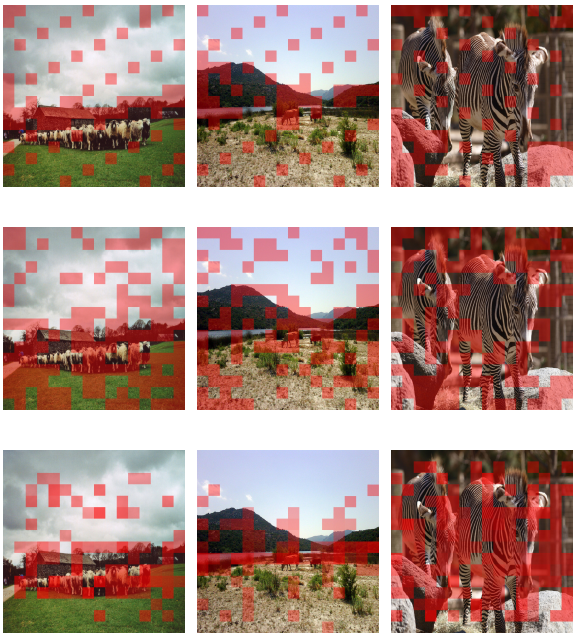

Figure 3: Visualization examples of important visual tokens identified by baseline methods and our AdaptMerge on three sample images. First row: PruMerge; second row: DivPrune; third row: AdaptMerge. Red masks on each image indicate the tokens deemed important by each method. Compared to the baselines, AdaptMerge adaptively adjusts token selection based on image complexity and text relevance.

Limitations

While AdaptMerge presents a promising approach to improving efficiency in LMMs, it is not without limitations. It relies heavily on the availability and quality of visual and language guidance, which may limit its effectiveness in scenarios where either modality is noisy or ambiguous, potentially leading to suboptimal token merging. Additionally, the reliance on adaptive token merging could introduce challenges in terms of model interpretability and consistency, as the merging process might obscure how individual tokens contribute to the final output.

Acknowledgments

We acknowledge the support of the University of Saskatchewan, the Natural Sciences and Engineering Research Council of Canada (NSERC), and Google Cloud. We also acknowledge the use of AI assistants (e.g., ChatGPT, Copilot) to improve the language and clarity of our writing. We thank Sujoy Paul (Google DeepMind) for valuable discussions.

References

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. 2019. Nocaps: Novel object captioning at scale. In *IEEE/CVF International Conference on Computer Vision*.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. 2025. Divprune: Diversity-based visual token pruning for large multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, and 1 others. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your vit but faster. In *International Conference on Learning Representations*.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. 2025. Matryoshka multimodal models. In *International Conference on Learning Representation*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*.

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. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. 2014. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *European Conference on Computer Vision*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. 2024. Tokenpacker: Efficient visual projector for multimodal llm. *arXiv preprint arXiv:2407.02392*.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. 2025. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *AAAI Conference on Artificial Intelligence*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *IEEE/CVF Winter Conference on Applications of Computer Vision*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

Mauricio GC Resende, Rafael Martí, Micael Gallego, and Abraham Duarte. 2010. Grasp and path relinking for the max–min diversity problem. *Computers & Operations Research*, 37(3):498–508.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2025. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *IEEE/CVF International Conference on Computer Vision*.

Mustafa Shukor and Matthieu Cord. 2024. Skipping computations in multimodal llms. *arXiv preprint arXiv:2410.09454*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yizheng Sun, Yanze Xin, Hao Li, Jingyuan Sun, Chenghua Lin, and Riza Batista-Navarro. 2025. LVPruning: An effective yet simple language-guided vision token pruning approach for multi-modal large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4299–4308, Albuquerque, New Mexico. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. 2025. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *AAAI Conference on Artificial Intelligence*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision*.

Qiming Zhang, Jing Zhang, Yufei Xu, and Dacheng Tao. 2024. Vision transformer with quadrangle attention. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 46(05):3608–3624.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. 2025. Llava-mini: Efficient image and video large multimodal models with one vision token. In *International Conference on Learning Representations*.