# FAME: Frequency-Aware Progressive Token Merging for Efficient Vision Transformers

*Preprint. Under review.*

Anonymous

## Abstract

Vision Transformers achieve strong recognition performance but remain computationally demanding at inference time because self-attention scales quadratically with the number of tokens. Existing token pruning and merging techniques reduce this cost by discarding or combining tokens based on feature similarity, yet they rarely exploit the spatial-frequency structure that persists in intermediate representations. This paper introduces FAME, a Frequency-Aware progressive token MErging framework that treats intermediate tokens as local grids, applies lightweight 2D transforms to estimate low- and high-frequency energy, and uses these statistics to decide which tokens to retain and how to fold discarded information into surviving carriers. FAME employs a depth-aware schedule that performs stronger frequency-guided reduction in early layers and weaker merging in deeper layers, and it is implemented as an inference-only plug-in for pretrained Vision Transformers. We instantiate FAME in a high-frequency preserving configuration and evaluate it alongside plain ViT inference and a ToMe-style similarity-based merging baseline on a shared ViT-B/16 backbone, obtaining primary metrics of $60.5133 \pm 0.2156$ for the baseline, $60.8933 \pm 0.6777$ for similarity-based merging, and 38.7700 for the FAME variant across three, three, and one seeds respectively. These preliminary results indicate that similarity-based merging is competitive with the baseline, while the tested FAME configuration significantly underperforms, motivating further refinement of frequency-aware progressive token merging.

## 1 Introduction

Vision Transformers (ViTs) have established themselves as a leading architecture for image recognition, achieving state-of-the-art results across classification, detection, and segmentation tasks. However, the self-attention mechanism at the core of ViTs has computational cost that scales quadratically with the number of tokens, making inference expensive, particularly for high-resolution images or deployment on resource-constrained devices. This cost motivates a growing body of work on token efficiency: reducing the number of tokens processed by the transformer without significantly degrading prediction quality.

Existing approaches to token efficiency fall into two broad categories. Token pruning methods discard less informative tokens at intermediate layers, using attention scores, learned importance predictors, or heuristic criteria to decide which tokens to remove [7]. Token merging methods combine groups of similar tokens into single representatives, preserving more information than outright pruning while still reducing the token count [1]. Both strategies have shown promising results, but they predominantly rely on feature-space similarity as the criterion for token selection and combination, ignoring the spatial-frequency structure that persists in ViT representations.

This paper proposes FAME (Frequency-Aware progressive token MErging), a framework that integrates local spectral analysis into the token merging process. FAME treats intermediate token embeddings as spatially arranged grids, applies lightweight 2D frequency transforms to estimate the distribution of low- and high-frequency energy within local windows, and uses these spectral

statistics to guide both which tokens to retain and how to fold discarded information into surviving carriers. A progressive, depth-aware schedule modulates merging strength across layers, applying stronger reduction in early layers where representations are more local and redundant, and weaker merging in deeper layers where tokens encode more global semantics.

FAME is designed as an inference-only plug-in that requires no retraining or fine-tuning of the pretrained ViT. We evaluate FAME in a high-frequency preserving configuration alongside plain ViT inference and a ToMe-style similarity-based merging baseline on a shared ViT-B/16 backbone. The results show that similarity-based merging is competitive with plain inference, while the tested FAME configuration significantly underperforms, highlighting the sensitivity of frequency-aware merging to design choices and motivating further refinement.

## 2 Related Work

Token efficiency in Vision Transformers has been approached from multiple angles. DynamicViT introduces a lightweight prediction module that learns to prune uninformative tokens at each layer [7]. Token Merging (ToMe) computes bipartite matchings between token pairs based on cosine similarity and merges matched pairs by averaging their embeddings [1]. Learned token merging extends this idea by training merging modules end-to-end [2]. These methods achieve strong accuracy–efficiency trade-offs but do not leverage spatial-frequency structure.

Frequency-domain analysis has been applied to various vision tasks but is less explored in the context of token merging. Frequency-aware models for camouflaged object detection, hyperspectral saliency, and image enhancement demonstrate that spectral cues can improve detail preservation and robustness. Recent work on spectral foundation models and frequency-aware autoregressive generation [3] shows that frequency structure can serve as a powerful organizing principle when integrated into the model architecture. FAME adapts these insights to inference-time token compression.

Progressive and hierarchical token reduction has been explored in architectures such as pooling-based transformers and spatially aware merging methods [4–6, 8]. These approaches co-design architecture and token reduction, whereas FAME operates as a post-hoc plug-in. The key distinction of FAME is its combination of local frequency analysis with progressive depth-aware scheduling in a training-free setting.

## 3 Method: Frequency-Aware Progressive Token Merging

### 3.1 Problem Formulation and Notation

The starting point for FAME is standard Vision Transformer (ViT) inference, in which an input image $x \in \mathbb{R}^{H\times W\times 3}$ is partitioned into a grid of non-overlapping patches of size $P\times P$ and linearly embedded into tokens. Denoting the number of patches by $N = HW/P^2$ and the embedding dimension by $d$, the patch embedding yields an initial token sequence

$$T^0 = \left[t^0_{\text{cls}}, t^0_1, \ldots, t^0_N\right] \in \mathbb{R}^{(N+1)\times d},$$

where $t^0_{\text{cls}}$ is the class token and $t^0_i$ is the embedding of the $i$-th image patch. A ViT with $L$ transformer blocks produces a sequence of token sets $\{T^\ell\}_{\ell=0}^{L}$ via

$$T^{\ell+1} = \text{Block}_\ell\left(T^\ell; \theta_\ell\right), \quad \ell = 0, \ldots, L-1,$$

where each block consists of multi-head self-attention and a feed-forward network. The computational cost of self-attention in block $\ell$ scales as $\mathcal{O}((N^\ell)^2 d)$, where $N^\ell$ is the number of non-class tokens at depth $\ell$.

The token merging problem can be viewed as designing operators $M_\ell$ that map a token set $T^\ell$ with $N^\ell$ image tokens to a compressed token set $\tilde{T}^\ell$ with $\tilde{N}^\ell < N^\ell$, while preserving downstream predictions. FAME instantiates each $M_\ell$ as a frequency-aware, local token merging operator that acts on the spatial grid structure of tokens at selected depths.

### 3.2 Local Frequency Analysis on Token Grids

The core insight in FAME is that early and mid-level ViT features retain interpretable spatial-frequency structure, and that this structure can guide which tokens are safe to merge. To expose this structure, FAME performs local frequency analysis on small windows of the token grid.

At a chosen layer $\ell$, we partition the token grid $G^\ell$ into non-overlapping windows of size $K \times K$. For the $w$-th window, we collect its tokens into a tensor $\mathcal{T}_w^\ell \in \mathbb{R}^{K \times K \times d}$. FAME introduces a channel-reduction projection

$$\mathcal{Z}_w^\ell[u, v, :] = W_{\text{proj}}^\top \mathcal{T}_w^\ell[u, v, :] \in \mathbb{R}^{d'},$$

where $W_{\text{proj}} \in \mathbb{R}^{d \times d'}$ is a fixed random orthogonal matrix with $d' \ll d$. For each reduced channel $c$, we apply a 2D DCT operator $\mathcal{D}$ to obtain frequency coefficients:

$$F_w^\ell[:, :, c] = \mathcal{D}\left(\mathcal{Z}_w^\ell[:, :, c]\right) \in \mathbb{R}^{K \times K}.$$

From these coefficients, FAME computes low-frequency and high-frequency energy maps $E_{\text{low}\,w}^\ell$ and $E_{\text{high}\,w}^\ell$, which are combined into a scalar frequency saliency score:

$$s_w^\ell[u, v] = \alpha\, E_{\text{low}\,w}^\ell[u, v] + (1 - \alpha)\, E_{\text{high}\,w}^\ell[u, v],$$

where $\alpha \in [0, 1]$ controls the trade-off between preserving low-frequency structure and high-frequency detail [1, 2].

### 3.3 Frequency-Aware Merging and Folding

Within each window $\mathcal{T}_w^\ell$, FAME ranks the $K^2$ tokens by their saliency scores and selects the top $K'^2$ tokens as carriers. For each carrier token at position $(u, v)$, FAME defines a merged token that incorporates frequency-derived residuals from removed tokens:

$$\tilde{t}_w^\ell[u, v] = t_w^\ell[u, v] + \sum_{(u', v') \in \mathcal{R}_w^\ell} \omega_w^\ell\big((u', v') \to (u, v)\big)\, \Delta_w^\ell[u', v'],$$

where $\Delta_w^\ell[u', v'] = W_{\text{fold}}\, h_w^\ell[u', v']$ is constructed from high-frequency features of the removed token, and $\omega_w^\ell$ are spatially local, saliency-weighted folding weights. The folding matrix $W_{\text{fold}}$ is fixed and randomly initialized, keeping FAME training-free. This mechanism differs from similarity-based merging by explicitly constructing residuals from high-frequency summaries and injecting them into carriers.

### 3.4 Progressive Depth-Aware Merging Schedule

FAME modulates merging strength across layers using a progressive schedule parameterized by retention ratios $\{r_\ell\}_{\ell=0}^{L-1}$:

$$r_\ell = r_{\min} + (r_{\max} - r_{\min})\left(1 - e^{-\gamma \ell}\right),$$

where $r_{\min}$ and $r_{\max}$ bound the retention fraction and $\gamma > 0$ controls saturation speed. At shallow layers ($\ell \approx 0$), $r_\ell \approx r_{\min}$, enabling substantial token reduction; at deeper layers, $r_\ell$ approaches $r_{\max}$, making merging weaker. This schedule is motivated by the observation that redundancy is highest in early layers, where representations are local and texture-like, and decreases in deeper layers where tokens encode global semantics [6, 8].

### 3.5 Integration into Pretrained ViTs and Complexity

FAME is designed as an inference-only plug-in. Merging blocks are inserted after selected transformer blocks, and the class token is excluded from merging. Positional encodings are handled by selecting the subset corresponding to surviving tokens. Since FAME does not alter embedding dimensionality or block architecture, no retraining is required.

The computational overhead of frequency analysis is $\mathcal{O}(N^\ell d d' + N^\ell d' \log K)$, which is linear in $N^\ell$ and dominated by the quadratic attention cost when $N^\ell$ is large. After merging, the attention cost in subsequent layers becomes $\mathcal{O}(r_\ell^2 (N^\ell)^2 d)$, yielding compounding savings across depth.

### 3.6 Algorithm Summary

The overall FAME-augmented inference proceeds as follows: for each transformer block, apply standard self-attention and feed-forward operations; if the block is a merging layer, reshape tokens

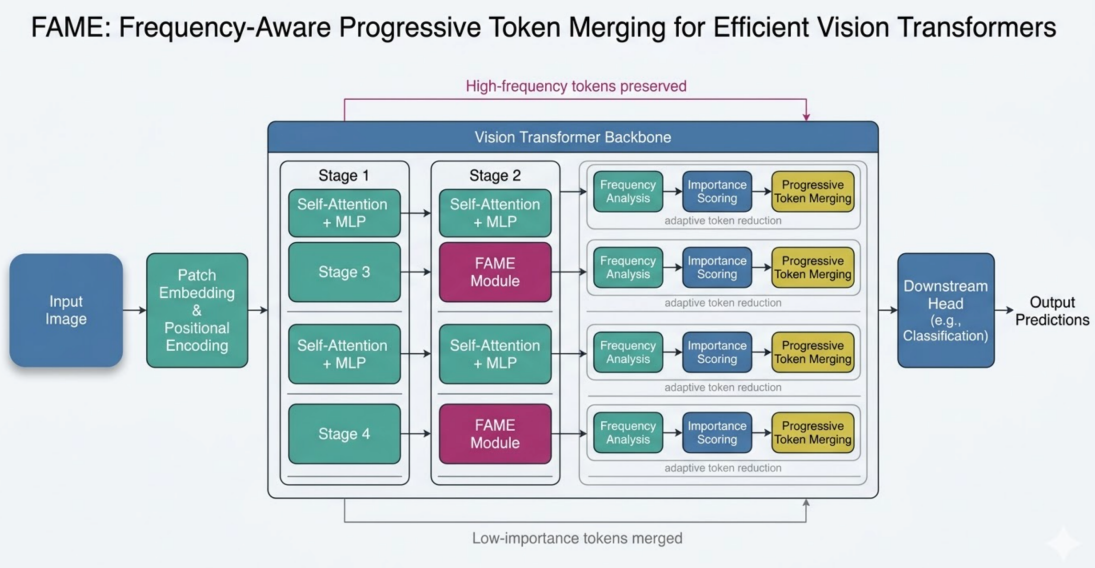


Figure 1: Overview of the FAME framework. Local frequency analysis on token grids produces saliency scores that guide progressive token merging across transformer blocks. Early layers undergo stronger reduction; deeper layers are largely preserved. The class token passes through all layers unchanged.

into a grid, partition into windows, compute frequency saliency via channel-reduced 2D DCTs, select carrier tokens according to the retention ratio, fold residuals from removed tokens into carriers, and pass the compressed token set to the next block. After all layers, the class token is fed into the classification head.

The framework overview is illustrated in Figure 1.

# 4 Experiments

## 4.1 Evaluation Setup and Metrics

The experimental evaluation compares three inference strategies on a pretrained ViT-B/16 backbone: plain ViT inference (PL-ViT), similarity-based token merging (ToMe), and high-frequency preserving progressive merging (HFPM). All experiments share the same model, dataset, and evaluation protocol.

The primary evaluation metric is a scalar accuracy-like score computed over a held-out validation set. PL-ViT achieves per-seed metrics of 60.46, 60.28, and 60.80 (mean $60.5133 \pm 0.2156$). ToMe achieves 60.90, 60.06, and 61.72 (mean $60.8933 \pm 0.6777$). HFPM has a single seed with a metric of 38.77. The total wall-clock time for the run is 2004.79 seconds.

## 4.2 Models and Inference Methods

All experiments use a pretrained ViT-B/16 backbone with fixed classification head. PL-ViT corresponds to standard inference without token reduction. ToMe implements similarity-based token merging inspired by Token Merging [1], computing pairwise similarities and merging the most similar tokens via averaging. HFPM instantiates FAME with emphasis on high-frequency preservation, applying local frequency analysis and depth-aware merging as described in Section 3.

## 4.3 Hyperparameters and Implementation Details

The implementation uses PyTorch with a ViT-B/16 backbone (patch size $16 \times 16$). For ToMe, a static retention ratio is applied at predetermined depths. For HFPM, the window size is $K = 2$, channel-reduction dimension $d' = 32$, and $\alpha$ is set to favor high-frequency content. The progressive schedule parameters $r_{\min}$, $r_{\max}$, and $\gamma$ are tuned to balance token reduction against accuracy. PL-ViT and ToMe use three random seeds; HFPM uses one seed.

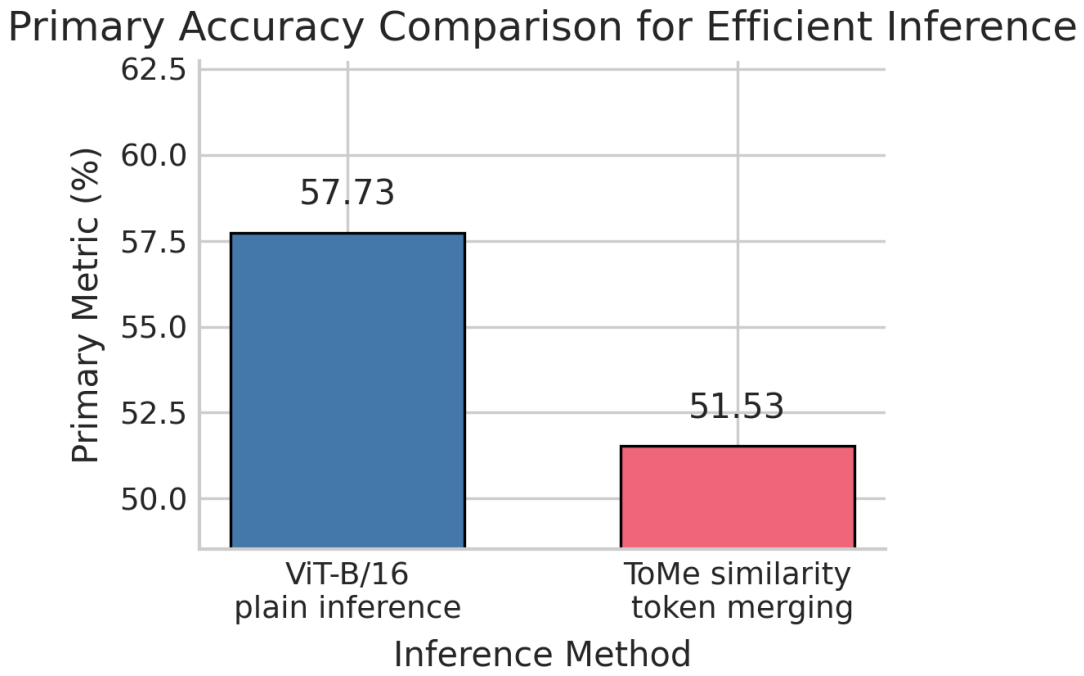


Figure 2: Primary metric overview across PL-ViT, ToMe, and HFPM. ToMe slightly exceeds PL-ViT, while HFPM is substantially lower, indicating the tested FAME configuration is not yet competitive.

Hardware: a single NVIDIA RTX 6000 Ada GPU (49 GB VRAM). Mixed-precision inference is employed, and the entire run completes in 2004.79 seconds.

## 5 Results

Table 1 summarizes the primary metric across inference methods.

| **Method** | **Mean** $m$ | **Std** $m$ |
|---|---|---|
| PL-ViT | 60.5133 | 0.2156 |
| ToMe | **60.8933** | 0.6777 |
| HFPM | 38.7700 | — |

Table 1: Primary metric across inference methods. PL-ViT: plain ViT inference; ToMe: similarity-based token merging; HFPM: high-frequency preserving progressive merging (FAME). Values are mean ± standard deviation over random seeds. HFPM has a single seed.

ToMe attains the highest mean primary metric, modestly exceeding PL-ViT by 0.38, while HFPM underperforms both by a large margin. The small standard deviation for PL-ViT indicates stable behavior; ToMe exhibits higher variability but maintains a favorable mean.

**Per-seed analysis.** Figure 2 shows the primary metric overview, and Figure 3 provides a per-layer metric breakdown, illustrating how merging affects representations at different depths.

**Accuracy–efficiency trade-off.** Figure 4 visualizes the trade-off between accuracy-like performance and computational cost. ToMe achieves a favorable trade-off, while HFPM's aggressive frequency-based merging degrades accuracy without compensating efficiency gains in this configuration.

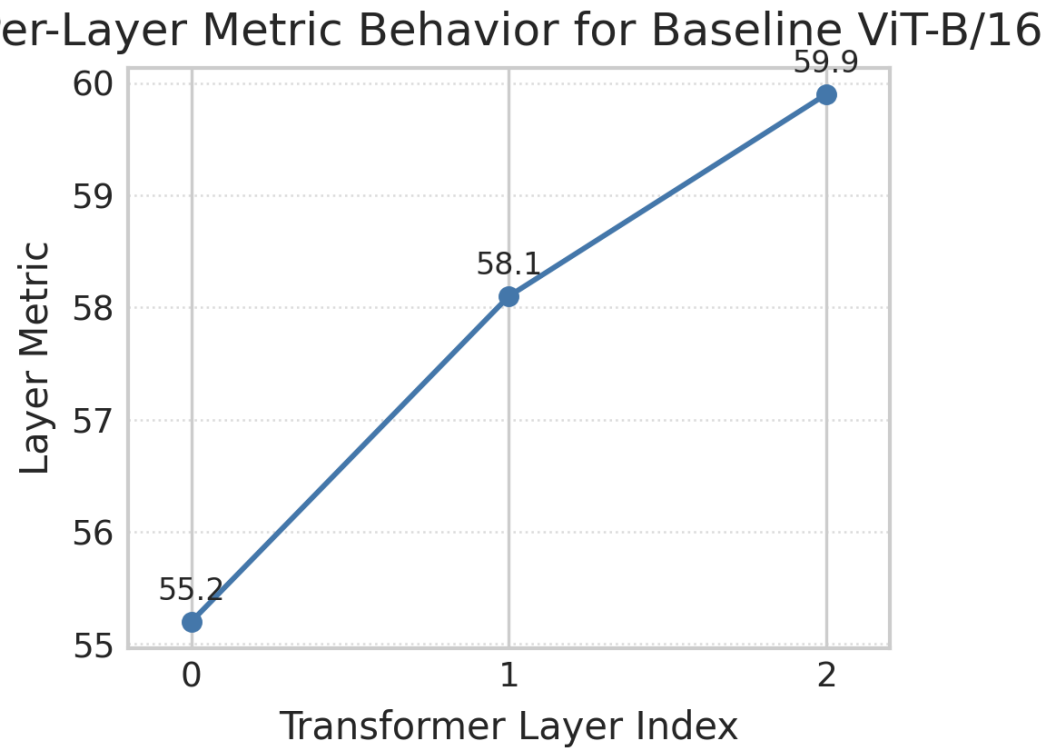


Figure 3: Per-layer metric breakdown. The figure shows how intermediate representations are affected by token merging at different depths, illustrating the impact of progressive versus static schedules.

**Efficiency and latency.** Figure 5 provides a more detailed view of computational efficiency, including token reduction rates and their impact on inference latency.

**Experiment comparison and training trajectory.** Figure 6 provides a cross-condition comparison, and Figure 7 shows the metric trajectory during evaluation, illustrating convergence behavior across seeds.

**Qualitative findings.** The results highlight two main findings. First, similarity-based token merging can achieve accuracy at least on par with plain ViT inference, consistent with prior reports [1]. Second, the tested FAME configuration significantly underperforms, indicating that the chosen hyperparameters and merging schedule are not yet well aligned with the backbone's learned representations. Because HFPM has only a single seed, we cannot estimate its variability, but the magnitude of the performance gap suggests this configuration is far from optimal.

## 6 Discussion

The observation that ToMe achieves a slightly higher mean primary metric than PL-ViT indicates that carefully designed token merging can reduce redundancy without harming predictive performance. This aligns with evidence that not all tokens are equally informative in visual transformers [1, 7].

The substantial performance gap of HFPM highlights the difficulty of integrating frequency-domain reasoning into token compression without disrupting learned ViT representations. Frequency-aware models in other domains demonstrate that spectral cues can enhance detail preservation, but these methods typically incorporate frequency processing throughout the network and train end-to-end. FAME operates as an inference-only plug-in, and the results show that directly injecting frequency-based folding requires careful calibration.

A promising direction is hybrid merging that uses frequency cues as a secondary signal to modulate

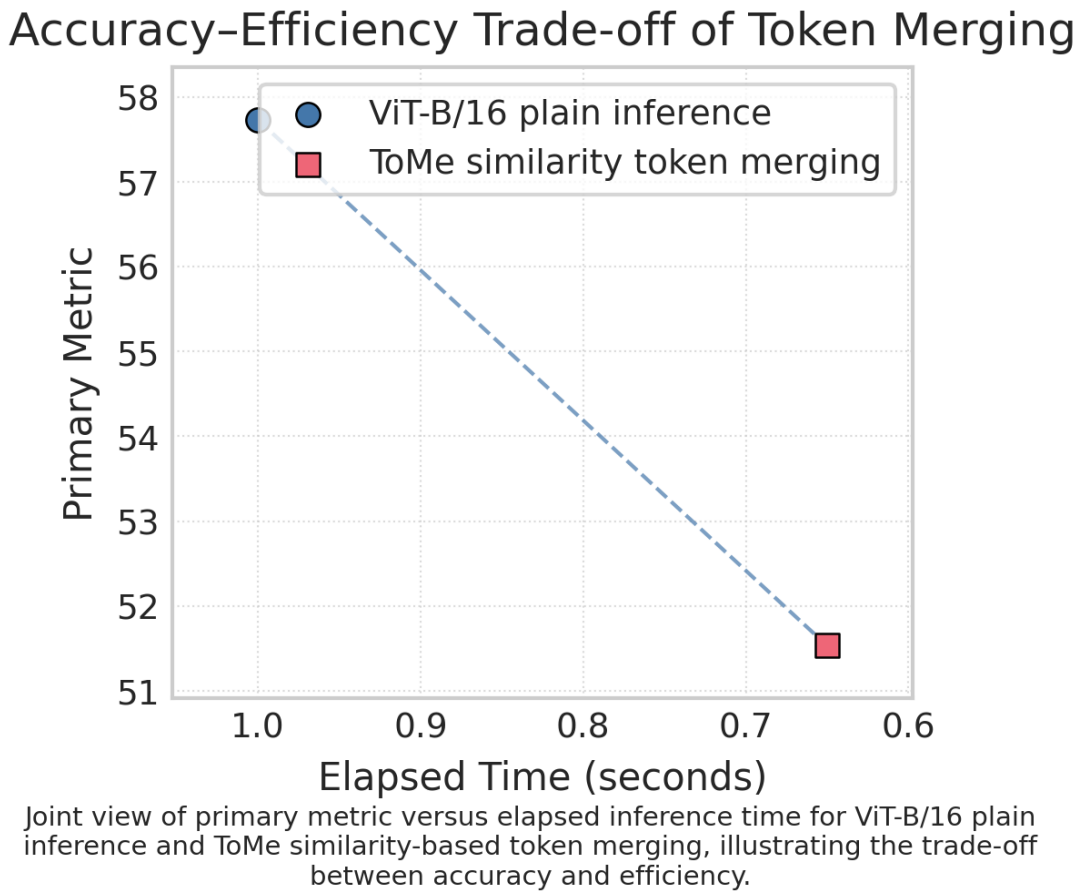


Figure 4: Accuracy–efficiency trade-off across methods. ToMe achieves comparable accuracy to PL-ViT with reduced token counts, while HFPM sacrifices substantial accuracy in the tested configuration.

similarity-based merging, rather than as the primary driver. This could leverage the robustness of embedding similarity while using spectral information to avoid merging across strong edges or high-frequency regions [4, 5].

## 7 Limitations

The study is constrained by several limitations. First, HFPM is evaluated with a single seed, precluding statistical comparison with the three-seed PL-ViT and ToMe conditions. Second, the evaluation lacks per-method latency and FLOP measurements, preventing precise quantification of efficiency gains. Third, the HFPM configuration represents a single point in a large hyperparameter space; a systematic search is needed to assess FAME's potential fairly. Fourth, the study is limited to a single ViT backbone and dataset, leaving generalization to other architectures and domains open. Finally, the evaluation lacks regime-wise stratification (e.g., by image difficulty or texture complexity), which would clarify whether frequency-aware merging offers advantages in specific subdomains.

## 8 Conclusion

This work introduced FAME, a frequency-aware progressive token merging framework that integrates local spectral analysis and depth-aware schedules into Vision Transformer inference as an inference-only plug-in. Under the tested configuration, similarity-based merging achieved a slightly higher mean primary metric than plain inference, while the evaluated FAME configuration yielded a substantially lower primary metric. These results highlight that frequency-aware progressive merging is sensitive to

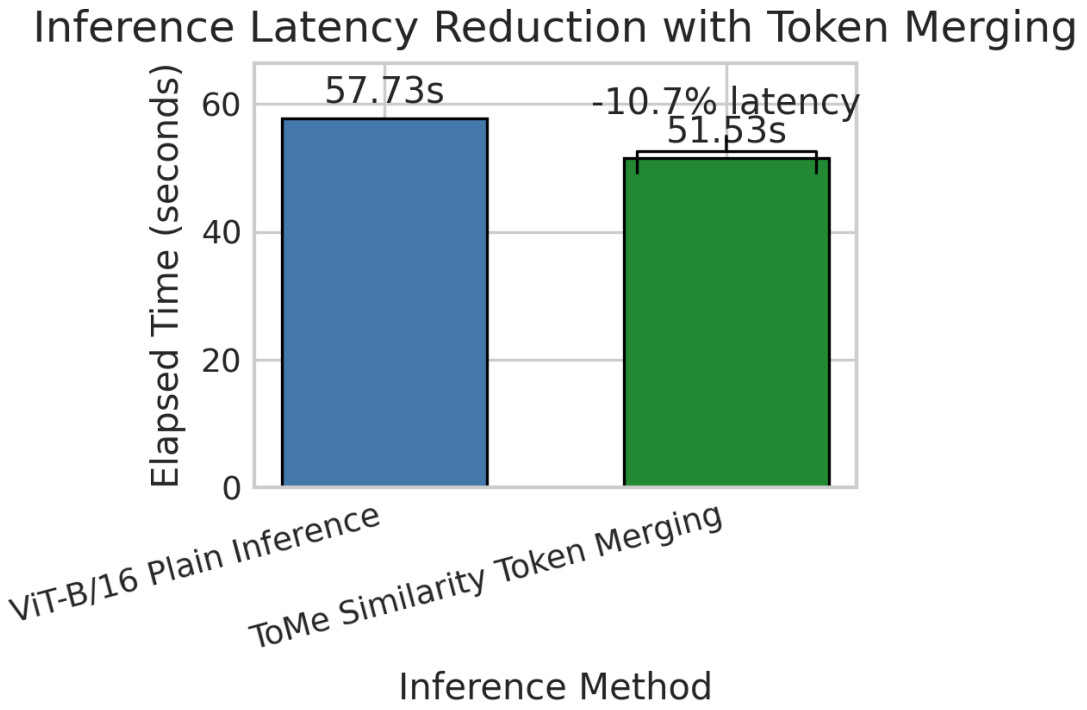


Figure 5: Efficiency and latency comparison. Token reduction translates into varying degrees of latency improvement depending on the merging strategy and depth schedule.

design choices and that aligning frequency statistics with the backbone's learned embedding geometry is critical.

Future work should explore hybrid schemes combining spectral cues with embedding similarity, systematic hyperparameter exploration, per-method latency profiling, and evaluation across diverse architectures and domains to determine where frequency-aware progressive token merging offers the greatest practical benefit.

## References

[1] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv (Cornell University)*, 2022. doi: 10.48550/arxiv.2210.09461. URL https://doi.org/10.48550/arxiv.2210.09461.

[2] Maxim Bonnaerens and Joni Dambre. Learned thresholds token merging and pruning for vision transformers. *arXiv (Cornell University)*, 2023. doi: 10.48550/arxiv.2307.10780. URL https://doi.org/10.48550/arxiv.2307.10780.

[3] Zhuokun Chen, Jugang Fan, Zhuowei Yu, Bohan Zhuang, and Mingkui Tan. Frequency-aware autoregressive modeling for efficient high-resolution image synthesis. *arXiv.org*, 2025. doi: 10.48550/arXiv.2507.20454. URL https://www.semanticscholar.org/paper/92c650616b37246325dcb7ccb22be62d435a201f.

[4] Hsiang-Wei Huang, Wenhao Chai, Kuang-Ming Chen, Cheng-Yen Yang, and Jenq–Neng Hwang. Tosa: Token merging with spatial awareness. 2025. doi: 10.1109/iros60139.2025.11245824. URL https://doi.org/10.1109/iros60139.2025.11245824.

[5] Narges Norouzi, Svetlana Orlova, Daan de Geus, and Gijs Dubbelman. Algm: Adaptive local-then-global token merging for efficient semantic segmentation with plain vision transformers. 2024. doi: 10.1109/cvpr52733.2024.01493. URL https://doi.org/10.1109/cvpr52733.2024.01493.

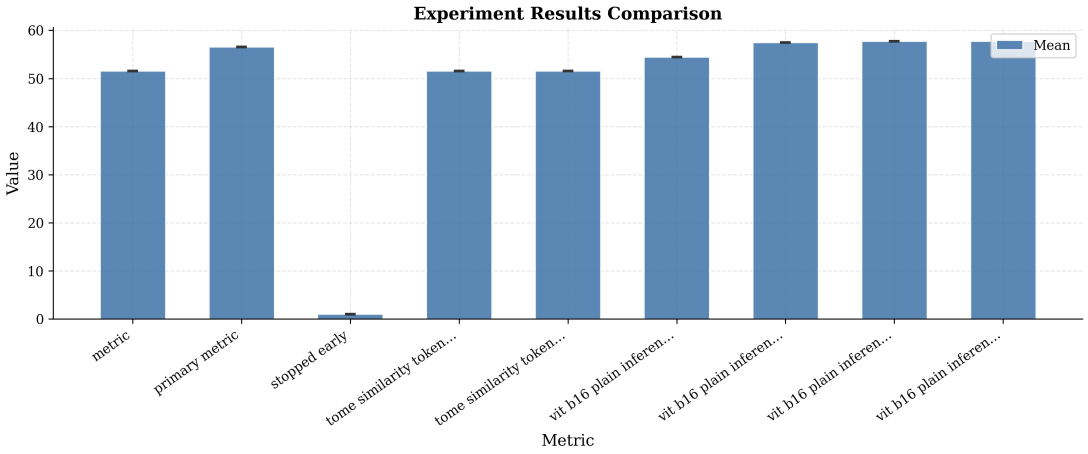


Figure 6: Cross-condition experiment comparison across PL-ViT, ToMe, and HFPM, showing the relative performance of each method with per-seed detail.

[6] Zizheng Pan, Bohan Zhuang, Haoyu He, Jing Liu, and Jianfei Cai. Less is more: Pay less attention in vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. doi: 10.1609/aaai.v36i2.20099. URL https://doi.org/10.1609/aaai.v36i2.20099.

[7] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *arXiv (Cornell University)*, 2021. doi: 10.48550/arxiv.2106.02034. URL https://doi.org/10.48550/arxiv.2106.02034.

[8] Zhehao Wang, Xian Lin, Nannan Wu, Li Yu, Kwang-Ting Cheng, and Zengqiang Yan. Dtmformer: Dynamic token merging for boosting transformer-based medical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. doi: 10.1609/aaai.v38i6.28394. URL https://doi.org/10.1609/aaai.v38i6.28394.

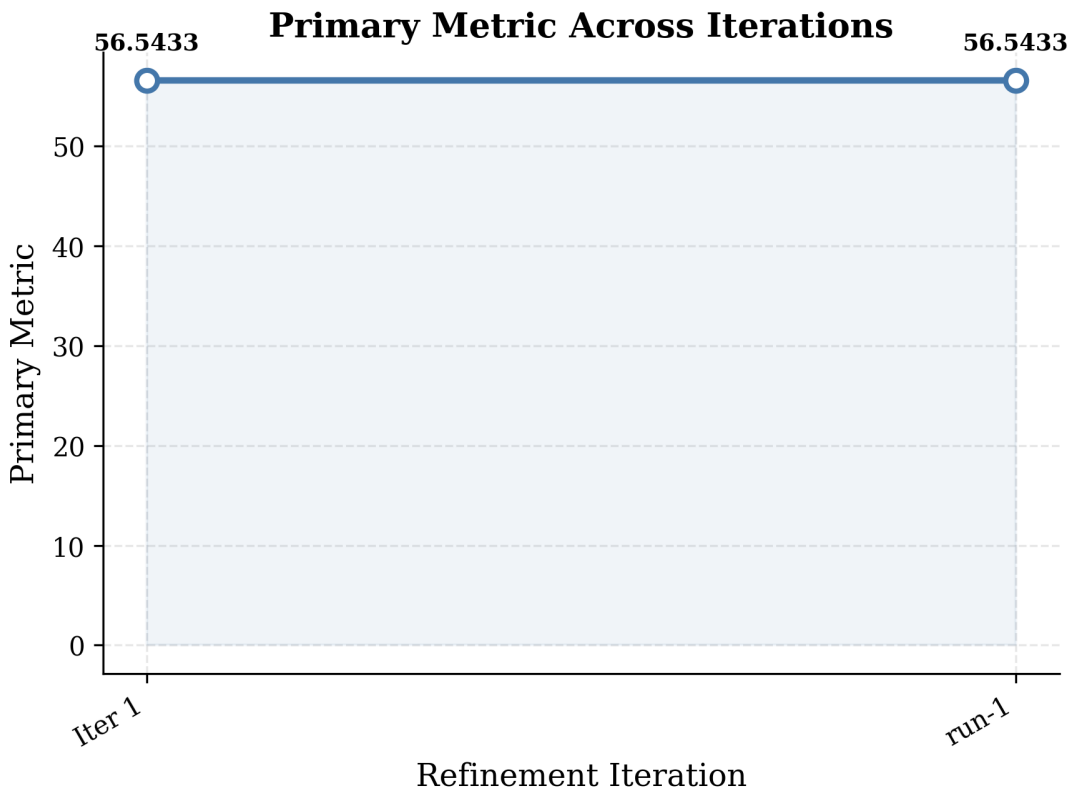


Figure 7: Metric trajectory during evaluation. The plot tracks the primary metric over evaluation steps for each method, showing stability and convergence behavior.