MERGENET: DIFFERENTIABLE SPATIAL ROUTING WITH A PHYSICAL TOKEN BOTTLENECK

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision transformers preserve fine image detail but pay to process every patch through global blocks. Learned token merging must decide where information flows while producing a shorter sequence for later computation. We introduce MergeNet, a local-to-latent transformer that combines differentiable spatial mass transport with an exact-budget gather in one forward pass. Routing redistributes features and mass on the original patch grid; selected carriers then read full-grid features through cross-attention before latent encoding. Mass-aware latent attention retains the transported token mass, and an exact query/key augmentation implements its bias in an unmodified FlashAttention call. On ImageNet-1K with DeiT-S/8, MergeNet reaches 81.360% top-1 with 392 final patches and 82.618% after 512-pixel fine-tuning. Its 224-pixel configuration halves the latent patch sequence and reduces estimated multiply–accumulate operations by 37.0% relative to dense DeiT-S/8. In a separate synthetic A100 benchmark, compiled inference takes 27.24 ms at batch 64, compared with 27.22 ms for compiled ToMe. Evaluations across token budgets and image resolutions characterize the accuracy and implementation cost of this trainable physical bottleneck.

1 INTRODUCTION

Vision transformers use patch tokens to represent image detail and self-attention to integrate information across the image. Processing a fine patch grid through every block, however, makes this global reasoning expensive. Token merging reduces the sequence by combining patches rather than simply removing them (Bolya et al., 2023). A learned merger thus faces two linked requirements: learn where information should flow, and physically reduce the sequence seen by downstream blocks during training.

Differentiable merging does not automatically solve this execution problem. DTEM learns dedicated merge embeddings through continuous grouping and merging (Lee & Hong, 2024). Its relaxed operators retain all token slots; for end-to-end training, DTEM alternates soft updates of the merge embeddings with hard, token-reducing updates of the backbone. Thus DTEM uses both continuous policy learning and physical compression during training, but in separate update paths. We ask whether one forward pass can propagate gradients through soft routing while downstream blocks process an exact, physically shorter sequence.

We study this question with MergeNet[1], a local-to-latent vision transformer. Six local blocks encode the full patch grid, after which soft mass transport learns how patch information moves between slots. A hard gather then selects K carriers from N patches, so the subsequent latent blocks receive a physically shorter tensor during both training and inference. A residual cross-attention read lets those carriers access the full-grid local features before global latent reasoning. The gather indices are discrete; a straight-through weight connects the selected features to continuous routing scores without claiming to differentiate the indices themselves.

The routing support follows the original two-dimensional patch grid. A donor can send mass only to eligible receivers within radius R, while learned similarities determine the weights and the resulting mass determines which carriers survive the gather. Keeping these coordinates fixed gives the routing support a direct interpretation on the image grid at every step.

[1] An anonymized implementation is provided in the supplementary material.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Carrier mass also has a defined role after selection. Following proportional attention in ToMe (Bolya et al., 2023), MergeNet adds each carrier's $\log m$ to its attention key logits. This per-key bias is rank one across queries; the released operator reproduces it by adding one query/key dimension and using the original softmax scale in an unmodified FlashAttention call (Dao et al., 2022). We use an existing threshold-based smooth top-k relaxation (Su, 2024) for the continuous selection scores. Our contribution is the way these components connect soft spatial transport to an exact-budget latent path, together with the mass-bias implementation.

We evaluate the resulting architecture on ImageNet-1K across token budgets and image resolutions. At 224 pixels, the 300-epoch model reaches 81.360% top-1 with 392 final patches, compared with 81.932% for ToMe and 81.742% for PiToMe applied to a trained dense DeiT-S/8. Fine-tuning raises MergeNet to 82.582% at 384 pixels and 82.618% at 512 pixels. We also measure implementation cost separately: in a synthetic A100 benchmark, compiled MergeNet and ToMe take 27.24 and 27.22 ms per batch of 64, respectively. Together, these experiments describe how an exact latent budget interacts with classification accuracy, input resolution, and execution cost.

Our contributions are:

- a local-to-latent architecture that connects differentiable spatial mass transport to an exact-budget gather within one forward pass;
- an exact query/key reparameterization that retains the log-mass attention bias in an unmodified FlashAttention call; and
- ImageNet-1K evaluations across token budgets and resolutions, accompanied by measurements of arithmetic cost, latency, and memory.

2 Related Work

Token reduction. DynamicViT learns token retention (Rao et al., 2021), while A-ViT assigns adaptive depth through halting (Yin et al., 2022). ToMe uses bipartite soft matching and can compress pretrained or jointly trained transformers (Bolya et al., 2023); PiToMe (Tran et al., 2024), MCTF (Lee et al., 2024), and SAD-TM (Xie et al., 2026) modify its matching signal or merge schedule. DiffRate treats layerwise budgets as another learned variable (Chen et al., 2023). Equal final counts therefore do not imply equal layerwise computation.

Differentiable merging. DTEM is the closest differentiable predecessor. It learns lightweight embeddings dedicated to merge decisions, rather than choosing token pairs from the backbone features alone (Lee & Hong, 2024). A continuous relaxation of bipartite grouping and mass-weighted merging lets the task loss train these embeddings. DTEM supports both modular training with a frozen backbone and end-to-end training; in the latter, it alternates soft-merging updates for the embeddings with hard-merging updates for the backbone, then uses hard matching at inference. MergeNet also learns merge decisions through soft assignments, but restricts transport to the original patch grid and gathers a fixed number of carriers within the same training forward pass.

Spatial structure and recovery. DSM and ALGM combine local and global merging (Heo et al., 2024; Norouzi et al., 2024); GTP-ViT propagates tokens over spatial and semantic edges (Xu et al., 2024). Recent reducers constrain matching along rows/columns or after space-filling reordering (Gong & Lis, 2025; Li & Xu, 2025). MergeNet instead constrains eligible routes by Euclidean radius on the original grid. LookupViT uses cross-attention between compressed and high-resolution streams (Koner et al., 2024); MergeNet lets its selected carriers read the full-grid local features before latent processing.

3 Method

3.1 Overview

MergeNet connects differentiable routing to a physically shorter latent sequence within one forward pass. Given N image patches and a class token, a local encoder produces full-grid features $H^{(1)}, \ldots, H^{(L)}$. Starting from $X^{(0)} = H^{(L)}$ and unit patch masses, L soft routing steps redistribute information while retaining every patch slot. A hard gather then selects K patch carriers. The carriers

read the pre-transport features through cross-attention, and a latent encoder followed by a class head produces the prediction. Figure 1 sketches the pipeline.

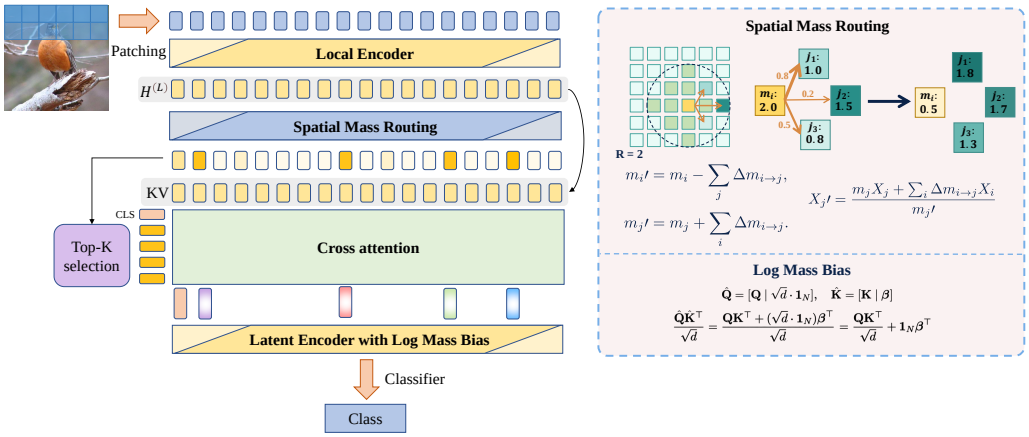

Figure 1: MergeNet architecture and routing. Soft mass transport retains the patch grid; top-K selection forms carriers that read full-grid features before latent encoding. The inset illustrates $R = 2$ (the 224-pixel experiments use $R = 3$) and the log-mass bias with $\beta = \log m$.

3.2 Spatial Mass Routing

Spatial routing restricts where patch information may move without fixing the routing weights. Patch i keeps its coordinate u_i on the original two-dimensional grid. At routing step ℓ, patches are partitioned into disjoint donor and receiver sets A_ℓ and B_ℓ; the class token participates in neither set. The eligible receivers for donor i are

$$\mathcal{E}_i^{(\ell)}(R) = \{j \in B_\ell : \|u_i - u_j\|_2 \le R\}. \tag{1}$$

Coordinates remain tied to original slots as features move. The radius bounds each transport step; information may pass through multiple steps.

The routing weights combine learned similarity with a soft donor decision. A learned projection of $H^{(\ell)}$ yields normalized metrics $z_i^{(\ell)}$. For eligible pairs, we compute

$$s_{ij}^{(\ell)} = \langle z_i^{(\ell)}, z_j^{(\ell)} \rangle / \tau, \qquad q_{ij}^{(\ell)} = \frac{\exp s_{ij}^{(\ell)}}{\sum_{r \in \mathcal{E}_i^{(\ell)}(R)} \exp s_{ir}^{(\ell)}}. \tag{2}$$

The row energy $e_i^{(\ell)} = \sum_{j \in \mathcal{E}_i^{(\ell)}(R)} \exp s_{ij}^{(\ell)}$ enters a threshold-based smooth top-k operator (Su, 2024). Given the step budget k_ℓ, it produces a continuous donor weight $g_i^{(\ell)}$; the transport weight is $a_{ij}^{(\ell)} = g_i^{(\ell)} q_{ij}^{(\ell)}$ on eligible edges and zero elsewhere; a donor with no eligible receiver has zero assignment. The smooth weights provide a gradient path from the task loss to the routing metrics. Appendix A gives the relaxation used here and for final carrier selection.

Soft transport updates mass and receiver features without removing the donor slot. With patch feature X_i and mass m_i, one step gives

$$\begin{aligned} m_i' &= m_i \left(1 - \sum_j a_{ij}\right), & X_i' &= X_i, \\ m_j' &= m_j + \sum_i a_{ij} m_i, & X_j' &= \frac{m_j X_j + \sum_i a_{ij} m_i X_i}{m_j'}. \end{aligned} \tag{3}$$

The sums cover eligible edges in the current donor–receiver partition. The implementation stabilizes residual fractions and denominators, while all patch slots remain present for the next routing step.

3.3 PHYSICAL BOTTLENECK AND RECOVERY

A hard gather turns the soft routing state into the shorter sequence used by later blocks. For compression factor λ, the patch-carrier budget is

$$K = N - \lfloor N(\lambda - 1)/\lambda \rfloor . \tag{4}$$

Let $\mathcal{I} = \mathrm{TopK}(m^{(L)}_{\mathrm{patch}}, K)$ contain the indices of the K largest patch masses, ordered before gathering. After normalizing the routed features, $\bar{X} = \mathrm{Norm}(X^{(L)})$, we form the carrier sequence $C = [\bar{X}_{\mathrm{cls}}; \bar{X}_{\mathcal{I}}]$. Thus C contains exactly $K + 1$ tokens, including the protected class token, in both training and inference.

The gather keeps its indices discrete while passing gradients to soft selection scores. A second smooth top-K operator applied to the final patch masses gives weights h_i. Each selected feature is multiplied by $1 + h_i - \mathrm{stopgrad}(h_i)$, which equals one in the forward pass. This connects the selected features to the continuous scores without differentiating the index choice.

Recovery lets the carriers read information still available on the full grid. Using $H^{(L)}$ as the pre-transport key and value sequence, define

$$Z = C + \mathrm{CrossAttn}(C, H^{(L)}, H^{(L)}). \tag{5}$$

The latent encoder processes Z, and its class-token output enters the classification head. Global latent self-attention therefore runs on $K + 1$ tokens, while recovery still reads the full-grid features.

3.4 MASS-AWARE LATENT ATTENTION

The selected carriers represent different amounts of transported mass. Following proportional attention in ToMe (Bolya et al., 2023), latent self-attention adds the log of each carrier mass to its key logit. For per-head query, key, and value matrices $Q_{\mathrm{a}}, K_{\mathrm{a}}, V_{\mathrm{a}}$, head width d_h, and selected masses m_j, this is

$$\mathrm{Attn}(Q_{\mathrm{a}}, K_{\mathrm{a}}, V_{\mathrm{a}}; m) = \mathrm{softmax}\left(\frac{Q_{\mathrm{a}} K_{\mathrm{a}}^{\top}}{\sqrt{d_h}} + \mathbf{1} \log(m)^{\top} \right) V_{\mathrm{a}}. \tag{6}$$

The class token retains unit mass. The log-mass bias is shared across query positions and varies with the mass represented by each key.

The per-key bias can be encoded as one extra query/key dimension. Set $\widehat{Q}_{\mathrm{a}} = [Q_{\mathrm{a}}, \sqrt{d_h}\mathbf{1}]$ and $\widehat{K}_{\mathrm{a}} = [K_{\mathrm{a}}, \log m]$. With the original scale $1/\sqrt{d_h}$,

$$\frac{\widehat{Q}_{\mathrm{a}} \widehat{K}_{\mathrm{a}}^{\top}}{\sqrt{d_h}} = \frac{Q_{\mathrm{a}} K_{\mathrm{a}}^{\top}}{\sqrt{d_h}} + \mathbf{1} \log(m)^{\top}. \tag{7}$$

The released FlashAttention-compatible operator applies this identity after padding the augmented head dimension.

4 EXPERIMENTS

We evaluate MergeNet on ImageNet-1K, study its accuracy across carrier budgets and image resolutions, and measure the cost of its implementation.

4.1 EXPERIMENTAL SETUP

Data and training. We train on ImageNet-1K and report top-1 accuracy on its 50,000-image validation set using exponential moving average (EMA) weights. At 224×224 resolution, DeiT-S/8 starts from 784 patch tokens. Dense DeiT and MergeNet train from scratch for 300 epochs with AdamW, an effective global batch of 1,024, cosine learning-rate decay, five warmup epochs, and the same augmentation family. The learning rate is 5×10^{-4} for DeiT and 7.5×10^{-4} for MergeNet. The 384-pixel models fine-tune the corresponding 224-pixel EMA weights for 30 epochs; the 512-pixel models receive a further 15 epochs. Appendix A gives the training settings.

Models and token budgets. MergeNet uses six local blocks followed by six latent blocks and gathers 392 patch carriers at its default $2\times$ compression setting. We compare it with dense DeiT-S/8 and with ToMe and PiToMe applied to the same trained dense checkpoint at inference. The compressed methods are matched by final patch count, excluding CLS. Their compression schedules remain different: ToMe and PiToMe merge across blocks, while MergeNet gathers between the local and latent stages. Dense DeiT-S/16 provides an additional coarse-patch reference with 196 patches.

Evaluation. The main compressed-model comparisons use complete validation evaluations of selected EMA checkpoints. Training-duration and resolution results report the best EMA top-1 recorded during each training stage. Carrier-budget sweeps evaluate fixed checkpoints without retraining for each budget. Latency and memory are measured separately on synthetic inputs with random weights, using fp16 autocast and device-resident tensors. Section 4.4 reports these implementation benchmarks.

4.2 Main Results

MergeNet reaches 81.360% top-1 at 224 pixels while passing 392 patch carriers to its latent encoder (Table 1). Dense DeiT-S/8 reaches 82.246%; its post-training ToMe and PiToMe variants reach 81.932% and 81.742% at the same final patch count as MergeNet. These results place the learned physical bottleneck within 0.886 percentage points of the dense fine-patch model. The coarse-patch DeiT-S/16 reference reaches 79.692% with 196 patches; its smaller input sequence is a distinct operating point.

Table 1: ImageNet-1K top-1 at 224 pixels after 300-epoch scratch training. ToMe and PiToMe reduce the trained DeiT-S/8 checkpoint at inference. Patch counts exclude CLS.

Model	Evaluation	Final patches	Top-1 (%)
DeiT-S/16	dense	196	79.692
DeiT-S/8	dense	784	82.246
DeiT-S/8 + ToMe	post-training	392	81.932
DeiT-S/8 + PiToMe	post-training	392	81.742
MergeNet ($R = 3$)	trained bottleneck	392	81.360

Longer scratch training improves both dense and compressed models. MergeNet increases from 79.316% after 150 epochs to 81.368% after 300 epochs, a gain of 2.052 percentage points in best EMA top-1. Dense DeiT-S/8 increases from 79.992% to 82.246% under the corresponding schedules. Appendix C reports these training-stage results, which are separate from the checkpoint re-evaluations in Table 1.

4.3 Token Budget and Resolution

Changing the carrier budget provides multiple inference operating points from one trained model (Figure 2). At 224 pixels, increasing the final budget from 196 to 523 patches raises MergeNet top-1 from 80.228% to 81.380%. At 384 pixels, the fine-tuned model reaches 82.562% with 1,152 carriers and 81.300% with 576. The best measured 384-pixel point occurs at 1,152 carriers; retaining more carriers does not necessarily improve a checkpoint trained at a fixed budget. ToMe retains higher top-1 across the displayed budgets, while the relative ordering of MergeNet and PiToMe varies with the operating point. Appendix B lists all budget evaluations.

Higher-resolution fine-tuning improves the trained classifier (Table 2). At 384 pixels, MergeNet reaches 82.582% with $R = 3$, compared with 82.976% for dense DeiT-S/8. The radius-scaled branch uses $R = 3 \times 384/224 \approx 5.143$ and reaches 82.574%; its checkpoint supplies the 384-pixel budget sweep. Fine-tuning that branch at 512 pixels with $R \approx 6.857$ reaches 82.618%, compared with 83.068% for dense DeiT-S/8. Table 2 therefore identifies the radius used by each reported model.

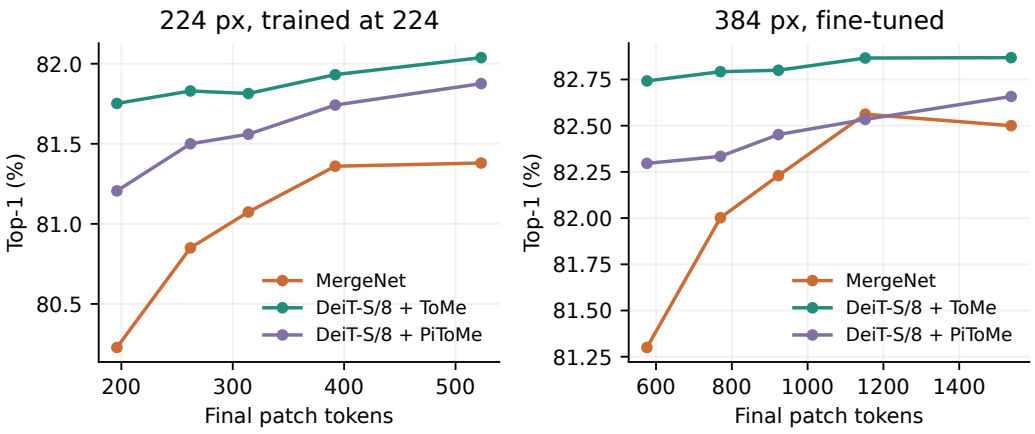

Figure 2: ImageNet top-1 across final patch budgets at 224 pixels (left) and after 384-pixel fine-tuning (right). The latter MergeNet checkpoint uses $R = 5.143$. Lines connect evaluations of fixed checkpoints.

Table 2: Best EMA ImageNet-1K top-1 (%) across resolutions. Gap is MergeNet minus dense DeiT-S/8 in percentage points. The 512-pixel MergeNet run initializes from the radius-scaled 384-pixel branch.

Pixels	Epochs	MN radius	DeiT-S/8	MergeNet	Gap (pp)
224	300	3	82.246	81.368	-0.878
384	300+30	3	82.976	82.582	-0.394
384	300+30	5.143	82.976	82.574	-0.402
512	300+30+15	6.857	83.068	82.618	-0.450

We also evaluate resolution changes without fine-tuning, using the 224-pixel checkpoints and unchanged radius $R = 3$. At the $2\times$ carrier budget, MergeNet reaches 81.400% at 288 pixels and 79.740% at 384 pixels. The corresponding 384-pixel fine-tuned checkpoint reaches 82.562%, indicating that higher resolution benefits from adaptation of the model and preprocessing. Appendix D gives the complete transfer results.

4.4 Efficiency

At 224 pixels, MergeNet passes 392 rather than 784 patches to the latent stage and requires an estimated 14.13 GMAC per image, compared with 22.41 GMAC for dense DeiT-S/8, a 37.0% reduction. The estimate excludes indexing and data movement; Table 3 measures their combined effect with the rest of the implementation on an A100.

Compiled batch-64 inference takes 27.24 ms for MergeNet and 27.22 ms for ToMe, with peak allocated memory of 0.627 and 0.614 GiB, respectively. Eager MergeNet takes 36.30 ms, compared with 34.72 ms for ToMe. Figure 3 shows both execution modes. These model-only benchmarks use random weights and synthetic inputs; Appendix E specifies the measurement and compilation settings.

Table 3: Synthetic A100 inference at 224 pixels with fp16 autocast and random weights. B1/B64 denote batch sizes 1 and 64; memory is batch-64 peak allocated and reserved GiB. The eager MergeNet B1 range spans two independent run medians.

Method	Patches	B1 (ms)	B64 (ms)	B64 (im/s)	Alloc.	Reserv.
Eager						
Dense DeiT-S/8	784	4.38	30.47	2100.5	0.744	0.957
ToMe	392	12.03	34.72	1843.5	0.724	1.004
PiToMe	392	14.17	70.28	910.7	1.774	3.104
MergeNet ($R = 3$)	392	18.27–21.00	36.30	1763.3	0.776	0.836
Compiled						
ToMe	392	9.24	27.22	2351.3	0.614	0.898
MergeNet ($R = 3$)	392	11.22	27.24	2349.3	0.627	0.715

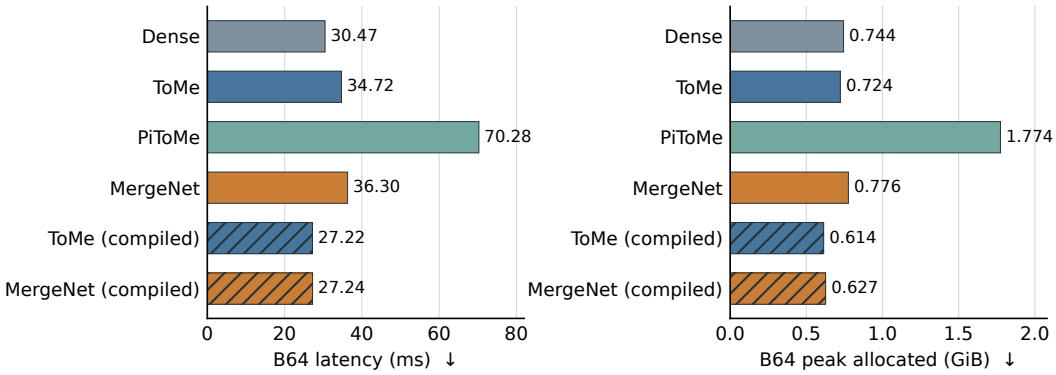

Figure 3: Batch-64 A100 inference with random weights. Solid bars denote eager execution; hatched bars denote compiled execution. Latency and peak allocated memory use the measurements in Table 3; lower is better.

A separate high-resolution benchmark on an H20 measures 136.54 ms for MergeNet and 170.61 ms for dense DeiT-S/8 at 384 pixels and batch 64. Peak allocated memory is 2.969 and 1.983 GiB, respectively. This operating point illustrates a latency reduction alongside a larger memory footprint. Appendix E.1 reports the training and inference measurements under that hardware setup.

5 Discussion

The physical bottleneck provides a fixed-size interface between local image processing and global latent reasoning. The carrier count can be changed at inference, and the resulting accuracy curves show how one checkpoint behaves across budgets. Higher-resolution fine-tuning further raises accuracy, while the resolution-transfer results show the value of adapting the model before operating on a substantially larger patch grid.

The full execution cost depends on more than the latent sequence length. Local encoding, routing, and recovery still process full-grid information. Compilation brings the measured batch-64 MergeNet latency close to ToMe on the A100, whereas the separate 384-pixel H20 benchmark shows lower latency than dense DeiT-S/8 with higher memory use. These observations concern the tested implementations and operating points. The classification results use individual training runs, and the synthetic efficiency measurements characterize model execution separately from checkpoint accuracy.

6 Conclusion

MergeNet connects differentiable spatial mass transport to an exact-budget latent sequence within one forward pass. Selected carriers retain their transported mass and recover full-grid information before global reasoning; an exact query/key augmentation supports the log-mass bias in FlashAttention. On ImageNet-1K, the model reaches 81.360% top-1 with 392 final patches at 224 pixels and 82.618% after 512-pixel fine-tuning. Budget sweeps and implementation benchmarks characterize the interaction between accuracy, resolution, and execution cost. The results motivate further work on reducing full-grid overhead while retaining the trainable physical bottleneck.

AI Use Statement

In this work, we used generative AI tools to assist with implementation refinement, experimental analysis, and interpretation of results. Additionally, we used these tools for literature retrieval, manuscript drafting and language polishing, and preparation of tables and figures. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the assistance of generative AI.

References

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token Merging: Your ViT but Faster. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.09461v3.

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*, 2023. URL https://arxiv.org/abs/2305.17997v1.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html.

Wenyi Gong and Mieszko Lis. CubistMerge: Spatial-Preserving Token Merging for Diverse ViT Backbones, 2025. URL https://arxiv.org/html/2509.21764v2. arXiv version 2, March 2, 2026; first posted September 26, 2025.

Jung Hwan Heo, Seyedarmin Azizi, Arash Fayyazi, and Massoud Pedram. Training-Free Acceleration of ViTs with Delayed Spatial Merging, 2024. URL https://arxiv.org/abs/2303.02331v2. ICML 2024 ES-FoMo Workshop; arXiv version 2, July 1, 2024.

Rajat Koner, Gagan Jain, Prateek Jain, Volker Tresp, and Sujoy Paul. LookupViT: Compressing Visual Information to a Limited Number of Tokens. In *European Conference on Computer Vision*, 2024. URL https://arxiv.org/html/2407.12753v1.

Dong Hoon Lee and Seunghoon Hong. Learning to Merge Tokens via Decoupled Embedding for Efficient Vision Transformers. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/html/2412.10569v1.

Sanghyeok Lee, Joonmyung Choi, and Hyunwoo J. Kim. Multi-criteria Token Fusion with One-step-ahead Attention for Efficient Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/papers/Lee_Multi-criteria_Token_Fusion_with_One-step-ahead_Attention_for_Efficient_Vision_Transformers_CVPR_2024_paper.pdf.

Yunge Li and Lanyu Xu. Neighbor-Aware Token Reduction via Hilbert Curve for Vision Transformers, 2025. URL https://arxiv.org/html/2512.22760v1. arXiv version 1, December 28, 2025.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/papers/Norouzi_ALGM_Adaptive_Local-then-Global_Token_Merging_for_Efficient_Semantic_Segmentation_with_CVPR_2024_paper.pdf.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *Advances in Neural Information Processing Systems*, 2021. URL https://papers.nips.cc/paper_files/paper/2021/file/747d3443e319a22747fbb873e8b2f9f2-Paper.pdf.

Jianlin Su. Softmax Backpropagation: Searching for a Smooth Approximation to Top-k, September 2024. URL https://spaces.ac.cn/archives/10373. Scientific Spaces blog; accessed September 22, 2026.

Hoai-Chau Tran, Duy M. H. Nguyen, Duy M. Nguyen, Trung-Tin Nguyen, Ngan Le, Pengtao Xie, Daniel Sonntag, James Y. Zou, Binh T. Nguyen, and Mathias Niepert. Accelerating Transformers with Spectrum-Preserving Token Merging. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.16148v2.

Weiying Xie, Xiaoyu Chen, Xin Zhang, Chenhe Hao, Jitao Ma, Yunsong Li, and Leyuan Fang. Saliency-Driven Token Merging for Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Xie_Saliency-Driven_Token_Merging_for_Vision_Transformers_CVPR_2026_paper.html.

Xuwei Xu, Sen Wang, Yudong Chen, Yanping Zheng, Zhewei Wei, and Jiajun Liu. GTP-ViT: Efficient Vision Transformers via Graph-Based Token Propagation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024. URL https://openaccess.thecvf.com/content/WACV2024/papers/Xu_GTP-ViT_Efficient_Vision_Transformers_via_Graph-Based_Token_Propagation_WACV_2024_paper.pdf.

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive Tokens for Efficient Vision Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/papers/Yin_A-ViT_Adaptive_Tokens_for_Efficient_Vision_Transformer_CVPR_2022_paper.pdf.

A Implementation Details

Table 4 gives the training settings for the main ImageNet experiments. All runs use an effective global batch of 1,024. Dense DeiT uses a learning rate of 5×10^{-4}; MergeNet uses 7.5×10^{-4} for the scratch-training results reported here. EMA top-1 is measured on the 50,000-image validation set.

Table 4: Training settings for the ImageNet-1K experiments.

Setting	224-pixel scratch	384/512-pixel fine-tune
Epochs	150 or 300	30 at 384; 15 at 512
Effective global batch	1024	1024
Optimizer	AdamW	AdamW
Learning rate	5×10^{-4} or 7.5×10^{-4}	10^{-5}
Minimum learning rate	10^{-5}	10^{-6}
Weight decay	0.05	10^{-8}
Warmup epochs	5	1
Warmup learning rate	10^{-6}	10^{-6}
EMA decay	0.99996	0.9996
Mixup / CutMix	0.8 / 1.0	0 / 0
Label smoothing	0.1	0.1
Random erasing	0.25	0.25
RandAugment	`rand-m9-mstd0.5-inc1`	same
Gradient clip norm	1.0	1.0
Validation crop percentage	0.9	1.0
MergeNet auxiliary-logit weight	ramp to 0.05	0.05
Teacher / feature / routing distillation	disabled	disabled

Classifier. The patch-8 model has width 384, six heads, six local blocks, and six latent blocks. At 224 pixels, the local encoder processes 784 patch slots. Local attention spans 16 sequence positions on either side of a patch query; the CLS query attends globally. Spatial routing uses radius 3, a $384 \to 64$ metric projection, similarity temperature 1, and metric input-gradient scale 0.1. At compression factor 2, the six soft donor budgets are $(65, 65, 65, 65, 65, 67)$; the hard gather retains 392 patches. The latent encoder does not merge further. Stochastic depth is 0.1, with separate local and latent ramps.

Routing and selection. Training partitions donors and receivers randomly for each image; evaluation alternates deterministic odd/even partitions across routing steps. The final gather indices are detached, and selected carriers are ordered by a detached, mass-weighted center of their original flattened indices. Spatial scoring uses sparse neighbor lists.

Smooth top-k. For donor energies e_i and step budget k_ℓ, let $v_i = 30e_i/(\max e - \min e)$. A threshold θ is determined from the sorted scores using prefix and suffix exponential sums. The continuous donor weight is

$$g_i = \begin{cases} 1 - \frac{1}{2}\exp(\theta - v_i), & v_i > \theta, \\ \frac{1}{2}\exp(v_i - \theta), & v_i \le \theta. \end{cases} \tag{8}$$

The threshold makes the weights sum to k_ℓ up to numerical precision. Donors without eligible receivers are assigned zero transport weight. Applying the same relaxation to the final patch masses supplies the straight-through carrier weights; the hard top-K indices still determine which slots are gathered.

At training time, an auxiliary full-grid readout contributes up to 0.05 to the latent CLS logits. Its weight begins ramping at epoch 10 for the 150-epoch runs and epoch 20 for the 300-epoch run, reaching 0.05 over 10 and 20 epochs, respectively. It is disabled at evaluation.

B Accuracy Budgets

Table 5 gives the values behind Figure 2. The MergeNet checkpoint uses radius 3 at 224 pixels and radius 5.143 after 384-pixel fine-tuning. Each row matches final patch counts, excluding CLS, but not

the layerwise sequence lengths: ToMe and PiToMe merge across blocks, whereas MergeNet gathers once between stages. All points evaluate fixed checkpoints at different inference budgets.

Table 5: ImageNet top-1 (%) at matched final patch counts. Δ is ToMe minus MergeNet in percentage points.

Checkpoint / pixels	Patches	MergeNet	ToMe	PiToMe	Δ ToMe
224 trained / 224	523	81.380	82.038	81.876	+0.658
224 trained / 224	392	81.360	81.932	81.742	+0.572
224 trained / 224	314	81.074	81.814	81.560	+0.740
224 trained / 224	262	80.850	81.830	81.500	+0.980
224 trained / 224	196	80.228	81.752	81.206	+1.524
384 tuned / 384	1537	82.500	82.868	82.658	+0.368
384 tuned / 384	1152	82.562	82.866	82.534	+0.304
384 tuned / 384	923	82.230	82.800	82.452	+0.570
384 tuned / 384	770	82.002	82.792	82.334	+0.790
384 tuned / 384	576	81.300	82.742	82.296	+1.442

C Training Duration

Table 6 compares the completed scratch-training schedules at 224 pixels. Each cell is the best recorded EMA top-1 on ImageNet-1K validation; it is a single-run training result. Dense DeiT uses learning rate 5×10^{-4} and MergeNet uses 7.5×10^{-4}. The remaining settings follow Table 4.

Table 6: Best EMA ImageNet-1K top-1 (%) under two scratch-training schedules. Δ is the 300-epoch minus 150-epoch result in percentage points.

Model	150 epochs	300 epochs	Δ (pp)
DeiT-S/16	77.196	79.692	+2.496
DeiT-S/8	79.992	82.246	+2.254
MergeNet ($R = 3$)	79.316	81.368	+2.052

D Resolution Transfer

Table 7 evaluates the 224-pixel checkpoints at 288 and 384 pixels without fine-tuning. The MergeNet radius remains 3 and the validation crop percentage remains 0.9. Position embeddings are adapted to the evaluation grid. These rows are separate from the fine-tuned 384-pixel results, which use validation crop percentage 1.0. Final patch counts match across the compressed methods in each row.

Table 7: ImageNet-1K top-1 (%) after changing input resolution without fine-tuning the 224-pixel checkpoints. Patch counts exclude CLS.

Pixels	Patches	MergeNet	ToMe	PiToMe
288	865	81.318	81.846	81.712
288	648	81.400	81.802	81.642
288	519	81.308	81.788	81.506
288	433	81.188	81.754	81.426
288	324	80.788	81.696	81.300
384	1537	79.476	80.268	79.998
384	1152	79.740	80.252	79.788
384	923	79.802	80.238	79.752
384	770	79.746	80.266	79.632
384	576	79.456	80.212	79.568

E Efficiency Details

A100 protocol. We measure inference on an A100-SXM4-80GB using randomly initialized models, device-resident 224×224 inputs, inference mode, and fp16 autocast. Each method and batch size runs in a fresh process with 50 warmups followed by five rounds of 200 CUDA-event samples. Latency is the median of all 1,000 samples and excludes model setup, data loading, and diagnostic hooks. Memory is the peak allocated or reserved GPU memory. Dense DeiT-S/8 has 784 patch tokens; each compressed method reaches 392 final patches, excluding CLS. ToMe and PiToMe enable proportional attention. The software environment is PyTorch 2.6.0 with CUDA 12.4 and timm 0.9.11.

The coefficient of variation of the five round medians is below 0.3% for every eager batch-64 measurement. Batch-1 latency is more sensitive to host scheduling: two independent eager MergeNet runs yield 18.27 and 21.00 ms. Table 3 reports both values.

Compiled execution. For ToMe, we compile the transformer-block forwards. For MergeNet, we compile local and latent transformer blocks, cross-attention, and selection and merge operations, while leaving the enclosing local-encoder forward eager. Compilation and its initial warmup are excluded from steady-state timing. The compiled rows in Table 3 use the same synthetic input shape and fp16 precision as the eager rows. The 0.02 ms difference between compiled batch-64 medians does not establish a meaningful speed ordering. These timings characterize the implementations independently of the trained-checkpoint accuracy measurements.

E.1 High-resolution benchmark

The separate 384-pixel benchmark uses an H20-3e and PyTorch 2.7.1 with CUDA 11.8. Models have random weights and synthetic fp16-autocast inputs; MergeNet uses $R = 3$ and retains 1,152 of 2,304 patches. The benchmark uses 30 warmups and 50 measured steps. Training time includes forward, cross-entropy loss, backward, and an AdamW update; inference measures forward only. Table 8 reports both batch sizes. These measurements describe this high-resolution hardware setup and are independent of the 224-pixel A100 benchmark.

Table 8: Synthetic H20-3e benchmark at 384 pixels with random weights. B denotes batch size; memory is peak allocated GiB. Train time includes forward, loss, backward, and an optimizer update.

B	Model	Train (ms)	Infer (ms)	Train GiB	Infer GiB
64	Dense DeiT-S/8	596.57	170.61	24.207	1.983
64	MergeNet ($R = 3$)	466.17	136.54	29.011	2.969
128	Dense DeiT-S/8	1135.48	324.36	47.990	3.776
128	MergeNet ($R = 3$)	889.76	262.02	57.676	5.757