QuarterMap: Efficient Post-Training Token Pruning for Visual State Space Models

Tien-Yu Chi [1 2] **Hung-Yueh Chiang** [3] **Diana Marculescu** [3] **Kai-Chiang Wu** [1]

Abstract

State space models (SSMs) reduce the quadratic complexity of transformers by leveraging linear recurrence. Recently, VMamba has emerged as a strong SSM-based vision backbone, yet remains bottlenecked by spatial redundancy in its four-directional scan. We propose QuarterMap, a post-training activation pruning method that removes redundant spatial activations before scanning and restores dimensions via nearest-neighbor upsampling. Our method improves throughput without retraining. On ImageNet-1K, QuarterMap achieves up to 11% speedup on VMamba with less than 0.9% accuracy drop, and yields similar gains on ADE20K segmentation. Beyond VMamba, we validate QuarterMap on MedMamba, a domain-specific model that shares the same four-directional scanning structure, where it consistently improves throughput while preserving accuracy across multiple medical imaging tasks. Compared to token merging methods like ToMe, QuarterMap is tailored for SSMs and avoids costly merge-unmerge operations. Our method offers a plug-and-play tool for deployment-time efficiency without compromising transferability.

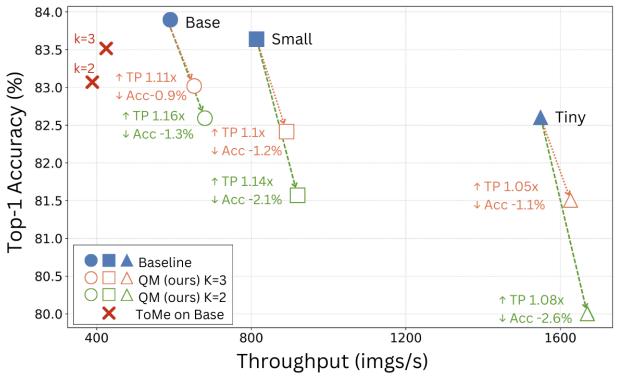

Figure 1. The accuracy-throughput trade-off when applying QuarterMap and Token Merging (ToMe). Our method demonstrates that QuarterMap not only increases throughput but also maintains comparable accuracy. In contrast, ToMe experiences a drop in throughput due to the overhead of merge and unmerge operations.

1. Introduction

Advancements in computer vision have been significantly driven by deep learning and the availability of large-scale datasets. Convolutional Neural Networks (CNNs) have served as the basis for tasks such as image classification (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; He et al., 2016) and object detection (Girshick et al., 2014; Girshick, 2015; Redmon et al., 2016). However, CNNs exhibit limitations in capturing long-range dependencies. Vision Transformers (ViTs) (Dosovitskiy et al., 2021; Liu et al., 2021a; Touvron et al., 2021), with their self-attention mechanisms, effectively overcome these limitations, but incur high computational costs due to their quadratic complexity. To alleviate these computational demands, recent research has focused on reducing the complexity of ViTs (Wang et al., 2020; Beltagy et al., 2020; Liu et al., 2021a; 2022; 2023), applied model compression techniques (Liu et al., 2021b; Lin et al., 2021; Zhu et al., 2021; Touvron et al., 2021; Lin et al., 2023), and investigated alternative architectures such as RWKV (Peng et al., 2023) and State Space Models (SSMs) (Gu et al., 2021; Fu et al., 2022; Gu & Dao, 2023).

State Space Models (SSMs) were initially introduced in the natural language processing (NLP) domain to reduce the computational cost associated with maintaining hidden states during the decoding phase. In contrast, computer vision tasks typically interpret the hidden state as a representation of the entire image's information. Recently, SSM have emerged as efficient alternatives to ViTs in computer vision, demonstrating competitive performance across multiple tasks (Liu et al., 2024; Zhu et al., 2024; Yang et al., 2024; Li et al., 2024; Teng et al., 2024). For instance, VMamba

[1]National Yang Ming Chiao Tung University [2]Canva [3]The University of Texas at Austin. Correspondence to: Tien-Yu Chi <b03902059@ntu.edu.tw>.

Proceedings of the 3^{rd} Workshop on Efficient Systems for Foundation Models at the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. 2025.

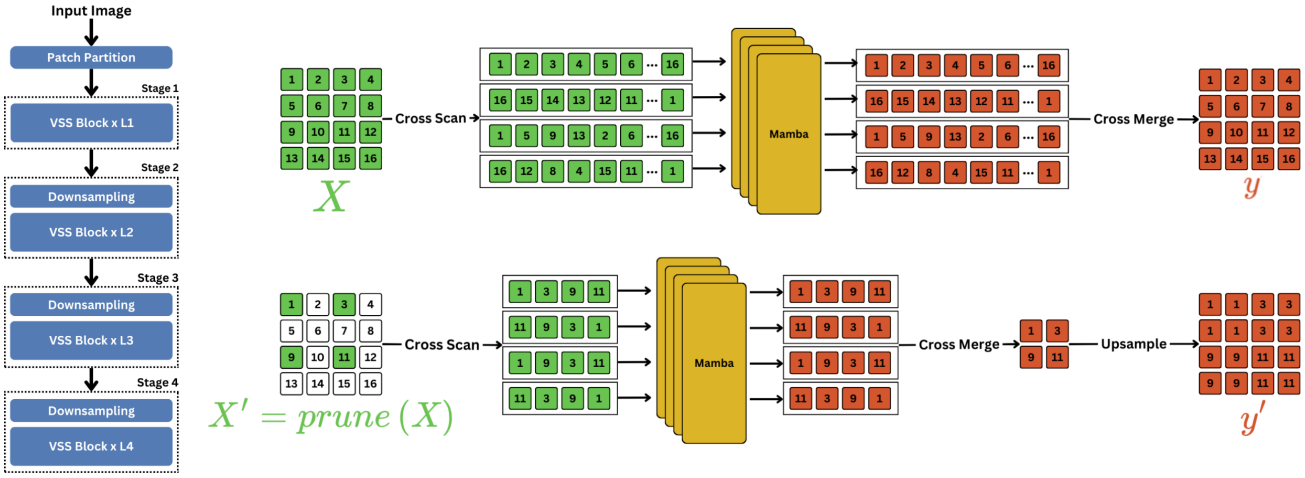

Figure 2. Illustration of the VMamba model architecture (**left**) and the proposed QuarterMap applied to the SS2D mechanism (**right**). The top flow shows the original cross-scan and cross-merge operations, while the bottom applies QuarterMap, pruning activations before scan and restoring spatial dimensions via nearest-neighbor upsampling. This reduces spatial redundancy and improves runtime efficiency without retraining.

(Liu et al., 2024) achieves a top-1 accuracy of 82.6% on the ImageNet-1K benchmark (Deng et al., 2009), outperforming the Swin Transformer (Liu et al., 2021a) by 1.3% with comparable FLOPs. However, within VMamba, the kernel responsible for selective scanning, an operation analogous to attention in transformer models, both serving as mechanisms to capture global context, still accounts for 18.3% of the total kernel execution time, highlighting a notable efficiency bottleneck. To address this, we explored transformer optimization techniques but identified a lack of optimization methods specifically designed for SSMs. We prove that conventional methods, such as token merging (Bolya et al., 2023) widely used in transformers, are suboptimal for VMamba due to computational tradeoffs arising from frequent merging and unmerging operations, as illustrated in Figure 1. Other recent approaches, such as Top-ViM (Zhan et al., 2024) and R-MeeTo (Shi et al., 2025), introduce token pruning and merging strategies tailored to ViM-based (Zhu et al., 2024) models. However, both rely on retraining to maintain performance, making them less applicable with limited computational resources in post-training deployment scenarios.

Motivated by these challenges, we explore whether a technique similar to token merging could be adapted for activation pruning in SSMs, and more broadly, how efficiency can be further improved in already efficient linear SSMs without *retraining*. We begin by analyzing VMamba's cross-scan and cross-merge mechanism, and the effective receptive field (ERF) in VMamba (Liu et al., 2024). Our analysis reveals that the four-directional traversal in VMamba, which introduces substantial spatial redundancy, with some tokens accumulating excessive and potentially unnecessary information. This led us to hypothesize that specialized activation pruning, aligned with the scanning structure, could reduce the latency while preserving accuracy.

To this end, we propose QuarterMap, a training-free activation pruning method specifically designed to improve VMamba's efficiency by reducing the feature map to one quarter of its original spatial size before scanning. As illustrated in Figure 2, QuarterMap introduces a lightweight two-stage pipeline: spatial pruning is applied before the cross-scan module, and nearest-neighbor upsampling is used to restore resolution after cross-merge. During pruning, QuarterMap retains every alternate element in both spatial dimensions, leveraging the spatial redundancy inherent in VMamba's four-way scanning. We adopt nearest-neighbor interpolation under the hypothesis that adjacent spatial positions convey similar information, making it an efficient yet effective reconstruction strategy. This design significantly reduces computation across the cross-scan, selective scan, and cross-merge, all without modifying model weights or requiring retraining.

We evaluate QuarterMap on both image classification and semantic segmentation tasks, showing improved throughput with minimal accuracy degradation. On ImageNet-1K, it achieves up to a $1.11\times$ speedup with less than a 0.9% drop in top-1 accuracy. Similar trends are observed in semantic segmentation and medical imaging benchmarks. QuarterMap proves particularly effective for VMamba and its variants, as confirmed by comparisons with CNNs, ViTs, and other SSMs like PlainMamba (Yang et al., 2024) and ViM (Zhu et al., 2024). Visualizations of attention maps and effective receptive fields (ERF) show that QuarterMap removes redundant activations while preserving key spatial

signals. Comprehensive ablation studies further explore pruning intervals, layer selection strategies, and upsampling methods, providing practical guidance for training-free deployment in the real world.

2. Method

We introduce QuarterMap, a post-training pruning function specifically designed to enhance the efficiency of VMamba by reducing spatial redundancy in activation feature maps, as illustrated in Figure 2. Formally, we define QuarterMap as a function T that operates on an input activation map x by selectively retaining spatial information. Given an activation map $x \in \mathbb{R}^{H \times W \times D}$, where H and W represent the spatial dimensions and D denotes the channel dimension, QuarterMap operates through the following stages:

Block Selection QuarterMap applies pruning selectively to specific blocks within the VMamba architecture, determined by a block selection interval k. This high-level strategy governs how frequently pruning is applied, balancing computational efficiency and accuracy. Applying QuarterMap to every three blocks (*i.e.,* k=3), excluding the first layer, yields the best accurarcy-latency trade-off (*ref.* Appendix E.) The early layers are critical for encoding fundamental features, whereas the deeper layers are more resilient to pruning, making them suitable candidates for optimization.

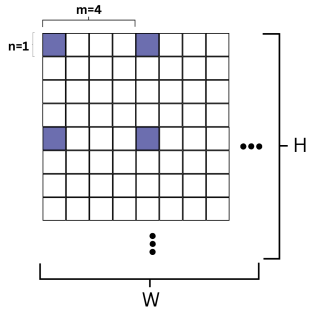

Figure 3. Illustration of the pruning stage, where m represents the interval size and n indicates the number of elements retained in the spatial dimensions.

Pruning Stage Within each selected block, QuarterMap performs a downsampling operation on the spatial dimensions of x before cross-scan. For a specified interval m, the function T retains every n elements in both the H and W dimensions, as depicted in Figure 3, resulting in a pruned activation map $x' = T(x) \in \mathbb{R}^{\lceil H*n/m \rceil \times \lceil W*n/m \rceil \times D}$. This process leverages the VMamba cross-scan mechanism, which aggregates information from four directions, combined with the SSM recurrence function. Together, these ensure that each element of the feature map $x_{i,j}$ incorporates information from neighboring spatial positions, improving pruning effectiveness while minimizing the loss of accuracy. We find that the setting $m = 2$ and $n = 1$ achieves the optimal trade-off between computational efficiency and accuracy. As the pruning stage is applied prior to cross-scan, the computational savings primarily stem from the reduced size of the input x for both the cross-scan and Mamba operations. Notably, in Mamba, this reduction leads to savings in the input length for the SSM (Equation (3)) as well as in the linear computation through the selective mechanism (Equation (5)).

Upsampling Stage After processing through cross-scan, selective scan, and cross-merge, QuarterMap restores the spatial dimensions of the activation map using an upsampling function U applied to y, the cross-merge output. Nearest-neighbor interpolation is used to reconstruct the original spatial dimensions, producing an output $y' = U(y) \in \mathbb{R}^{H \times W \times D}$. This approach aligns with the assumption that adjacent spatial elements contain similar information, enabling QuarterMap to balance computational efficiency with minimal accuracy degradation.

3. Experiments

3.1. Classification and Segmentation

The results in Table 1 show that applying QuarterMap to the *base* VMamba model yields a slight accuracy drop of 0.86% when $k = 3$, while improving throughput by $1.11\times$. This gain primarily stems from reduced sequence length within the selective scan mechanism, with minimal overhead introduced by our pruning and upsampling stages. Profiling reveals that QuarterMap reduces the scan kernel time from 1.4 to 0.6 ms, and introduces only 0.2 ms of additional overhead. In contrast, ToMe incurs significantly higher additional overhead (9.7 ms), despite a similar scan kernel runtime (0.7 ms), due to the cost of merge and unmerge operations. These comparisons highlight the efficiency of QuarterMap in minimizing unnecessary computation while retaining accuracy. Additional results on *small* and *tiny* VMamba configurations, as well as detailed segmentation metrics, are included in Appendix D.

3.2. QuarterMap on MedMamba for Medical Imaging

To evaluate QuarterMap's applicability beyond VMamba, we apply it to MedMamba-T (Yue & Li, 2024), a domain-specific model built on VMamba's architecture. We benchmark performance across four MedMNIST classification datasets: BoodMNIST, OrganMNIST, RetinaMNIST, and PathMNIST (Yang et al., 2021; 2023). As shown in Table 2, QuarterMap consistently delivers a $1.21\times$ throughput improvement (from 854 to 1034 images/sec) with no degradation in classification accuracy. These results demonstrate QuarterMap's generalization to VMamba-based models and reinforce its utility for domain-specific deployments where

Table 1. Performance of VMamba models on ImageNet-1K with and without QuarterMap (QM). QM improves throughput with minimal accuracy drop, while incurring lower overhead than ToMe.

Model	Method	K	Acc@1 (%)	Throughput	Speedup	Scan Kernel	Add. overheadd
VMamba-B	Baseline	-	83.88	590	1×	1.4ms	-
	ToMe	3	83.52 (-0.36)	424	0.72×	0.7ms	9.7ms
	ToMe	2	83.07 (-0.81)	389	0.66×		
	QM (ours)	3	83.02 (-0.86)	654	1.11×	0.6ms	0.2ms
	QM (ours)	2	82.58 (-1.30)	682	1.16×		

Table 2. QuarterMap on MedMamba-T across MedMNIST tasks. TP = Throughput (img/s).

Dataset	Base Acc.	QM (Ours) Acc.	Base TP	QM (Ours) TP
BloodMNIST	97.72%	97.78%	854.3	1033.7
OrganMNIST	81.85%	81.90%	854.5	1033.8
RetinaMNIST	54.25%	54.25%	853.6	1033.8
PathMNIST	29.44%	29.44%	854.0	1033.9

post-training efficiency and accuracy preservation are critical. For a breakdown of class-wise performance on BloodMNIST, see Appendix D.

Table 3. Accuracy comparison of QuarterMap (QM) on different model types for ImageNet-1K classification.

Model	Type	Baseline	QM (Ours)
ConvNeXtv2-B	Conv	84.89	45.71
DeiT-B	Transformer	81.80	79.50
Swin-B	Transformer	85.17	82.91
ViM-B	Mamba	80.40	71.00
VMamba-B	Mamba	83.88	83.02

3.3. QuarterMap on Other Architectures

We evaluate QuarterMap on CNNs, ViTs, and SSMs to assess whether its design is specifically suited for VMamba. We focus on base variants of ConvNeXtv2 (Woo et al., 2023), DeiT (Touvron et al., 2021), Swin Transformer (Liu et al., 2021a), and ViM (Zhu et al., 2024), all pretrained on ImageNet-1K with weights from Hugging Face (Wolf et al., 2020). QuarterMap is applied to every third (*i.e.* k=3) blocks after the first two. In CNNs, pruning disrupts spatial continuity and significantly harms accuracy. ViTs and ViM are more resilient, but still show non-trivial accuracy drops. As shown in Table 3, our method reduces the latency on VMamba with its four-directional scanning mechanism while maintaining the accuracy. QuarterMap is less compatible with CNNs and 1D-scanning SSMs like ViM, as these models lack the redundant activation patterns presented in VMamba. See Appendix D for results on other variants.

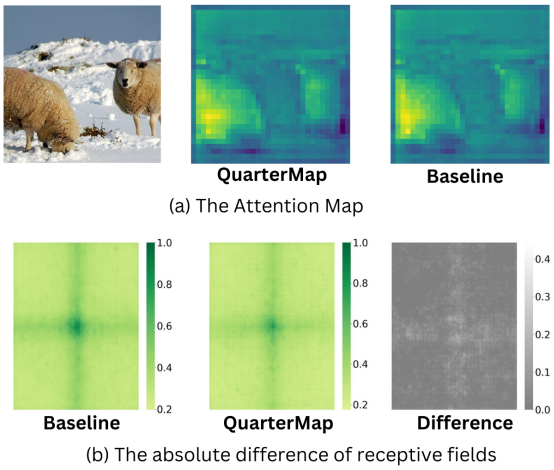

Figure 4. Comparison of (a) attention maps and (b) effective receptive fields before and after applying QuarterMap. The visualizations highlight the differences introduced by QuarterMap, demonstrating its selective pruning of redundant information while preserving the model's essential functionality.

3.4. Attention Map and Effective Receptive Field (ERF)

We visualize the attention maps and ERF before and after applying QuarterMap to analyze its spatial impact. Following the formulation in VMamba, we extract the attention map of the 12^{th} block (*i.e.,* the deepest layer), after several preceding blocks have been pruned. As shown in Figure 4, the attention patterns remain largely unchanged, indicating that QuarterMap preserves key contextual behavior. For the ERF visualization, the gray regions represent the activations removed by QuarterMap. These areas often overlap with receptive field hotspots that were spatially redundant. This supports our hypothesis that QuarterMap effectively eliminates redundant information while preserving the functional structure of the model.

4. Conclusion

We propose QuarterMap, a post-training pruning method for VMamba that improves runtime efficiency with minimal accuracy loss and requires no retraining. Our experiments show that QuarterMap aligns particularly well

with VMamba's four-directional scan and is also compatible with derived applications based on its backbone. Although designed for VMamba, QuarterMap is orthogonal to techniques such as quantization, enabling further efficiency gains. While this study focuses on VMamba, our findings open new directions to understanding and extending pruning strategies in SSMs.

Acknowledge

This work was supported in part by the ONR Minerva program, NSF CCF Grant No. 2107085, iMAGiNE - the Intelligent Machine Engineering Consortium at UT Austin, UT Cockrell School of Engineering Doctoral Fellowships

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work. By making ML models more efficient, our work contributes to democratizing access to advanced AI technologies and enabling their wider adoption.

References

Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Bolya, D. and Hoffman, J. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4599–4603, 2023.

Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., Feichtenhofer, C., and Hoffman, J. Token merging: Your vit but faster, 2023. URL https://arxiv.org/abs/2210.09461.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Everingham, M., Gool, L. v., Williams, C., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2008, 2008. URL http://host.robots.ox.ac.uk/pascal/VOC/voc2008.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Franklin, G. F., Powell, J. D., Emami-Naeini, A., and Powell, J. D. *Feedback control of dynamic systems*, volume 4. Prentice hall Upper Saddle River, 2002.

Fu, D. Y., Dao, T., Saab, K. K., Thomas, A. W., Rudra, A., and Re, C. Hungry hungry hippos: Towards language modeling with state space models. In *ICLR*, 2022.

Girshick, R. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. Hippo: Recurrent memory with optimal polynomial projections. *NeurIPS*, 33:1474–1487, 2020.

Gu, A., Goel, K., and Re, C. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2021.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, Y., Zhang, X., and Sun, J. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 1389–1397, 2017.

He, Y., Kang, G., Dong, X., Fu, Y., and Yang, Y. Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv:1808.06866*, 2018.

Hu, V. T., Baumann, S. A., Gui, M., Grebenkova, O., Ma, P., Fischer, J., and Ommer, B. Zigma: Zigzag mamba diffusion model. *arXiv preprint arXiv:2403.13802*, 2024.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *NeurIPS*, pp. 1106–1114, 2012.

Kálmán, R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1): 35–45, 1960.

Langley, P. Crafting papers on machine learning. In *ICML*, pp. 1207–1216, 2000.

LeCun, Y., Denker, J., and Solla, S. Optimal Brain Damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper_files/paper/1989/hash/6c9882bbac1c7093bd25041881277658-Abstract.html.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

Li, K., Li, X., Wang, Y., He, Y., Wang, Y., Wang, L., and Qiao, Y. Videomamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*, 2024.

Li, K., Li, X., Wang, Y., He, Y., Wang, Y., Wang, L., and Qiao, Y. Videomamba: State space model for efficient video understanding. In *European Conference on Computer Vision*, pp. 237–255. Springer, 2025.

Lin, H., Han, G., Ma, J., Huang, S., Lin, X., and Chang, S.-F. Supervised masked knowledge distillation for few-shot transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19649–19659, 2023.

Lin, M., Ji, R., Wang, Y., Zhang, Y., Zhang, B., Tian, Y., and Shao, L. Hrank: Filter pruning using high-rank feature map. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1529–1538, 2020.

Lin, Y., Zhang, T., Sun, P., Li, Z., and Zhou, S. Fq-vit: Post-training quantization for fully quantized vision transformer. *arXiv preprint arXiv:2111.13824*, 2021.

Liu, X., Peng, H., Zheng, N., Yang, Y., Hu, H., and Yuan, Y. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14420–14430, 2023.

Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., and Liu, Y. VMamba: Visual State Space Model, April 2024. URL http://arxiv.org/abs/2401.10166. arXiv:2401.10166 [cs].

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 10012–10022, 2021a.

Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., and Gao, W. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34: 28092–28103, 2021b.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, pp. 12009–12019, 2022.

Ma, J., Li, F., and Wang, B. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024.

Nguyen, E., Goel, K., Gu, A., Downs, G., Shah, P., Dao, T., Baccus, S., and Ré, C. S4nd: Modeling images and videos as multidimensional signals with state spaces. *Advances in neural information processing systems*, 35:2846–2861, 2022.

Pei, X., Huang, T., and Xu, C. Efficientvmamba: Atrous selective scan for light weight visual mamba, 2024. URL https://arxiv.org/abs/2403.09977.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K. K., et al. RWKV: reinventing rnns for the transformer era. In *EMNLP*, pp. 14048–14077, 2023.

Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., and Hsieh, C.-J. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Rao, Y., Liu, Z., Zhao, W., Zhou, J., and Lu, J. Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks, 2023. URL https://arxiv.org/abs/2207.01580.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

Sengupta, S., Harris, M., Zhang, Y., and Owens, J. D. Scan primitives for gpu computing. In *Proceedings of the 22nd ACM SIGGRAPH/EUROGRAPHICS Symposium on Graphics Hardware*, GH '07, pp. 97–106, Goslar, DEU, 2007. Eurographics Association. ISBN 9781595936257.

Shi, M., Zhou, Y., Yu, R., Li, Z., Liang, Z., Zhao, X., Peng, X., Vedantam, S. R., Zhao, W., Wang, K., and You, Y. Faster vision mamba is rebuilt in minutes via merged

token re-training, 2025. URL https://arxiv.org/abs/2412.12496.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Smith, J. T., Warrington, A., and Linderman, S. Simplified state space layers for sequence modeling. In *ICLR*, 2022.

Tang, Y., Han, K., Wang, Y., Xu, C., Guo, J., Xu, C., and Tao, D. Patch slimming for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12165–12174, 2022.

Tang, Y., Dong, P., Tang, Z., Chu, X., and Liang, J. Vmrnn: Integrating vision mamba and lstm for efficient and accurate spatiotemporal forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5663–5673, 2024.

Teng, Y., Wu, Y., Shi, H., Ning, X., Dai, G., Wang, Y., Li, Z., and Liu, X. Dim: Diffusion mamba for efficient high-resolution image synthesis. *arXiv preprint arXiv:2405.14224*, 2024.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *ICML*, pp. 10347–10357, 2021.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Wang, Z., Zheng, J.-Q., Zhang, Y., Cui, G., and Li, L. Mamba-unet: Unet-like pure visual mamba for medical image segmentation. *arXiv preprint arXiv:2402.05079*, 2024.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Huggingface's transformers: State-of-the-art natural language processing, 2020. URL https://arxiv.org/abs/1910.03771.

Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I. S., and Xie, S. Convnext v2: Co-designing and scaling convnets with masked autoencoders, 2023. URL https://arxiv.org/abs/2301.00808.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., and Sun, J. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018.

Xing, Z., Ye, T., Yang, Y., Liu, G., and Zhu, L. Segmamba: Long-range sequential modeling mamba for 3d medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 578–588. Springer, 2024.

Yang, C., Chen, Z., Espinosa, M., Ericsson, L., Wang, Z., Liu, J., and Crowley, E. J. Plainmamba: Improving non-hierarchical mamba in visual recognition. In *35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25-28, 2024*. BMVA, 2024. URL https://papers.bmvc2024.org/0755.pdf.

Yang, J., Shi, R., and Ni, B. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 191–195, 2021.

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., and Ni, B. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Yin, H., Vahdat, A., Alvarez, J., Mallya, A., Kautz, J., and Molchanov, P. Adavit: Adaptive tokens for efficient vision transformer, 2022. URL https://arxiv.org/abs/2112.07658.

Yu, R., Li, A., Chen, C.-F., Lai, J.-H., Morariu, V. I., Han, X., Gao, M., Lin, C.-Y., and Davis, L. S. Nisp: Pruning networks using neuron importance score propagation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9194–9203, 2018.

Yue, Y. and Li, Z. Medmamba: Vision mamba for medical image classification, 2024. URL https://arxiv.org/abs/2403.03849.

Zhan, Z., Kong, Z., Gong, Y., Wu, Y., Meng, Z., Zheng, H., Shen, X., Ioannidis, S., Niu, W., Zhao, P., et al. Exploring token pruning in vision state space models. *arXiv preprint arXiv:2409.18962*, 2024.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., and Wang, X. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *ICML*, 2024.

Zhu, M., Tang, Y., and Han, K. Vision transformer pruning. *arXiv preprint arXiv:2104.08500*, 2021.

Zhuang, Z., Tan, M., Zhuang, B., Liu, J., Guo, Y., Wu, Q., Huang, J., and Zhu, J. Discrimination-aware channel pruning for deep neural networks. *Advances in neural information processing systems*, 31, 2018.

A. Related Work

A.1. State Space Models

State Space Models (SSMs) (Gu et al., 2020; 2021; Smith et al., 2022; Fu et al., 2022; Gu & Dao, 2023) have gained notable traction in recent for their efficient scaling properties, offering linear computational complexity with respect to sequence length, which provides an advantage over the quadratic complexity of transformers. This efficiency, combined with their ability to capture global context, has made SSMs an appealing choice for handling long-range dependencies. To further minimize the resource demands of SSMs, S4 (Gu et al., 2021) applied a structured approach using a diagonal matrix configuration enhanced with a low-rank update, which reduced computational overheads. Building on this, subsequent works such as S5 (Smith et al., 2022) and H3 (Fu et al., 2022) introduced innovations including parallel scanning methods and optimized hardware utilization, further advancing the efficiency of SSM-based architectures. With Mamba (Gu & Dao, 2023), the introduction of the S6 block mark a key advancement by incorporating data-dependent parameters, breaking from the Linear Time Invariant (LTI) constraints of earlier models and enabling SSMs to outperform transformers on large-scale datasets.

In the field of vision tasks, S4ND (Nguyen et al., 2022) is one of the first models to adapt SSMs for visual data processing, representing it as 1D, 2D, and 3D signals. ViM (Zhu et al., 2024) and VMamba (Liu et al., 2024) further integrated SSMs into vision backbones, introducing the ViM and VSS blocks, respectively, which employ multiple scanning directions to handle the non-sequential characteristics of image data. This adaptation enables these models to achieve competitive performance with ViTs and CNNs. The success of ViM and VMamba has since inspired a range of Mamba-based methods that address diverse vision tasks, including medical image segmentation (Ma et al., 2024; Wang et al., 2024; Xing et al., 2024), video understanding (Li et al., 2025; Tang et al., 2024), and image generation (Hu et al., 2024; Teng et al., 2024). Collectively, these contributions highlight the potential of SSM-based approaches to drive advancements in computer vision applications.

A.2. Pruning

Neural network pruning has been a pivotal technique in the evolution of deep learning, aiming to enhance model efficiency by reducing computational and memory demands. Early foundational work (LeCun et al., 1989) introduced the concept of Optimal Brain Damage, which systematically removes less significant weights based on their second-order derivatives, effectively reducing model complexity without substantial loss in performance. This approach lays the groundwork for subsequent pruning methods, including weight pruning and activation pruning.

Weight pruning (Han et al., 2015; Frankle & Carbin, 2018; Li et al., 2016; He et al., 2018) involves eliminating less essential weights from the model, effectively reducing the number of parameters and computational load, making it well-suited for deployment in resource-constrained environments. In contrast, activation pruning (Zhuang et al., 2018; Yu et al., 2018; Lin et al., 2020; He et al., 2017) focuses on the intermediate outputs of the network, pruning redundant activations during inference to reduce computational costs without modifying the learned parameters.

The rise of ViTs (Dosovitskiy et al., 2020; Liu et al., 2022; Touvron et al., 2021) introduced significant computational challenges, leading to the development of token pruning and token merging techniques to shorten sequence lengths in the attention layers. Token pruning (Yin et al., 2022; Rao et al., 2023; 2021; Tang et al., 2022) selectively removes less informative tokens, reducing the computational burden of self-attention layers while preserving accuracy. Alternatively, token merging (Bolya & Hoffman, 2023; Bolya et al., 2023) combines similar tokens, effectively decreasing token count and enhancing throughput, thereby preserving ViT performance while reducing the token dimension.

Pruning techniques in the context of State Space Models (SSMs) have limited explored. A pruning-aware hidden state alignment method (Zhan et al., 2024) is introduced to stabilize the neighborhood of remaining tokens and improve performance. They also propose a token importance evaluation method tailored for SSMs to guide token pruning, though this method was applied only to ViM (Zhu et al., 2024). R-MeeTo (Shi et al., 2025) proposes a merged-token re-training strategy that periodically combines similar token pairs in ViM and then fine-tunes the model to restore accuracy. While this achieves good trade-offs, it requires retraining and is not applicable in purely post-training deployment pipelines. EfficientVMamba (Pei et al., 2024) integrates an atrous-based selective scan approach with efficient skip sampling. This approach introduces a concept similar to pruning, but it is applied before training.

B. Preliminaries

In this section, we provide an overview of the State Space Model (SSM) (Kálmán, 1960) and introduce two recent methods that leverage SSM in innovative ways: the selective state space model (Mamba) (Gu & Dao, 2023) and VMamba (Liu et al., 2024).

B.1. State Space Model

The State Space Model is a mathematical framework for modeling the evolution of a system over time. It represents the relationship between the system's state and observations at each time step through a set of equations. The most general form of an SSM is a continuous-time linear dynamical system, as shown in Equation (1).

$$\begin{aligned} h'(t) &= A(t)h(t) + B(t)u(t) \\ y(t) &= C(t)h(t) + D(t)u(t) \end{aligned} \tag{1}$$

Here, $h(t) \in \mathbb{R}^n$ denotes the state variable at time $t \in \mathbb{R}$, often referred to as the hidden state in recent machine learning literature. The input is represented by $u(t) \in \mathbb{R}^m$, and the output by $y(t) \in \mathbb{R}^p$. The system matrices $A(t) \in \mathbb{R}^{n\times n}$, $B(t) \in \mathbb{R}^{n\times m}$, $C(t) \in \mathbb{R}^{p\times n}$, and $D(t) \in \mathbb{R}^{p\times m}$ govern the system dynamics at each time step. For simplicity, we consider $u(t)$ and $y(t)$ as scalars, setting $m = p = 1$.

When the system matrices $A(t)$, $B(t)$, $C(t)$, and $D(t)$ remain constant over time, the continuous-time linear dynamical system simplifies to a linear time-invariant (LTI) system, represented in Equation (2). This LTI system can be transformed into a discrete-time linear dynamical system, defined by Equation (3), using discretization techniques. A common method in the SSM literature is zero-order hold (ZOH) discretization (Franklin et al., 2002), shown in Equation (4).

$$\begin{aligned} h'(t) &= Ah(t) + Bu(t) \\ y(t) &= Ch(t) + Du(t) \end{aligned} \tag{2}$$

$$\begin{aligned} h_t &= \bar{A}h_{t-1} + \bar{B}u_t \\ y_t &= Ch_t + Du_t \end{aligned} \tag{3}$$

$$\begin{aligned} \bar{A} &= \exp(\Delta A) \\ \bar{B} &= (\Delta A)^{-1}\exp(\Delta A - I)\Delta B \end{aligned} \tag{4}$$

B.2. Selective State Space Model (Mamba)

Mamba (Gu & Dao, 2023) extends the discrete-time linear dynamical system by introducing a timescale parameter, Δ, which transforms the continuous variables A and B into their discrete counterparts, $\bar{A}$ and $\bar{B}$. Beyond discretization, Mamba relaxes the time-invariance constraint on the system matrices by introducing a *selection* mechanism. This mechanism allows certain parameters—specifically Δ, B, and C—to vary over time as functions, s, of the input u. The formulations are defined in Equation (5).

$$\begin{aligned} s_B(u) &= Linear_N(u) \\ s_C(u) &= Linear_N(u) \\ s_\Delta(u) &= Broadcast_D(Linear_1(u)) \\ \Delta &= \tau_\Delta(Parameter + s_\Delta(u)) \end{aligned} \tag{5}$$

The $Linear_d$ is a parameterized linear projection to dimension d, and $\tau_\Delta = softplus$. Since the selection mechanism loses equivalence to the convolution form in equation (4), Mamba further incorporates a work-efficient parallel algorithm, called *associate scan* (Sengupta et al., 2007), into its GPU kernel implementation to enable parallel computation of the system.

B.3. VMamba

The original Mamba block was designed for 1-dimensional input and output, making it unsuitable for computer vision tasks that require 2-dimensional processing. To address this limitation, VMamba (Liu et al., 2024) introduced a new module

called 2D-Selective-Scan (SS2D), which adapts Mamba for 2D input and output. The SS2D module consists of three steps: cross-scan, selective scan (Mamba block), and cross-merge.

In the cross-scan step, the input feature map is unfolded into four 1D sequences, each capturing information from a distinct spatial direction (Figure 2). These sequences, processed in parallel by the Selective Scan module, encode diverse spatial perspectives critical for 2D feature processing. The cross-merge step then recombines the processed sequences into a 2D feature map, enabling a global receptive field. VMamba stacks multiple SS2D blocks within each layer to construct the complete model.

C. Experimental Setup

C.1. Datasets

QuarterMap is evaluated on two standard benchmarks: ImageNet-1K (Deng et al., 2009) for image classification and ADE20K (Zhou et al., 2017) for semantic segmentation. For both datasets, only the validation sets are used.

C.2. Models

Experiments utilize VMamba backbone models (Liu et al., 2024), pre-trained on ImageNet-1K. For semantic segmentation, the UperNet framework (Xiao et al., 2018) is used with the VMamba backbone, trained on ADE20K. The VMamba backbone models are available in three configurations: *tiny*, *small*, and *base*, which vary primarily in the number of layers and the dimensions of sequence length L and channel dimension D within the SS2D block.

Each backbone model consists of four layers. In the *tiny* configuration, these layers are arranged as [2, 2, 8, 2], while the *small* and *base* versions use a configuration of [2, 2, 15, 2]. For each configuration, the dimensions L and D are consistent within each layer but vary across layers. Specifically, the channel dimension D doubles, and the sequence length L decreases by a factor of 4 as depth increases.

C.3. Additional information

The evaluation metric for the image classification task is top-1 accuracy, while for the semantic segmentation task, we utilize all pixel accuracy (aAcc) and mean intersection over union (mIoU) (Everingham et al., 2008) to assess performance. The batch size for image classification is set to 128, and for semantic segmentation, it is limited to 1 due to the variable input sizes in the validation set. All experiments are conducted on a single NVIDIA A100-SXM4 GPU with 40GB of memory.

D. Additional Experiments

D.1. Classification Performance on VMamba Varients

Table 4. Performance comparison of VMamba models with and without QuarterMap (QM) on ImageNet-1K. The results demonstrate that applying QM improves throughput while maintaining comparable accuracy. The additional overhead in ToMe stems from merge and unmerge operations, whereas QM's overhead arises from the proposed pruning and upsampling stages.

Model	Method	K	Acc@1 (%)	Throughput	Speedup	Scan Kernel	Add. overheadd
VMamba-B	Baseline	-	83.88	590	1×	1.4ms	-
	ToMe	3	83.52 (-0.36)	424	0.72×	0.7ms	9.7ms
	ToMe	2	83.07 (-0.81)	389	0.66×		
	QM (ours)	3	83.02 (-0.86)	654	1.11×	0.6ms	0.2ms
	QM (ours)	2	82.58 (-1.30)	682	1.16×		
VMamba-S	Baseline	-	83.64	811	1×	1.1ms	-
	ToMe	3	83.09 (-0.55)	575	0.71x	0.5ms	7.3ms
	ToMe	2	82.79 (-0.85)	534	0.66x		
	QM (ours)	3	82.42 (-1.22)	890	1.10×	0.5ms	0.2ms
	QM (ours)	2	81.56 (-2.08)	921	1.14×		
VMamba-T	Baseline	-	82.60	1548	1×	0.5ms	-
	ToMe	3	82.15 (-0.45)	1218	0.79x	0.3ms	3.8ms
	ToMe	2	81.64 (-0.96)	1138	0.74x		
	QM (ours)	3	81.50 (-1.10)	1628	1.05×	0.2ms	0.1ms
	QM (ours)	2	80.01 (-2.59)	1671	1.08×		

We provide full classification results for VMamba *Tiny*, *Small*, and *Base* on ImageNet-1K with and without QuarterMap. As shown in Table 4, applying QuarterMap with $k = 3$ leads to modest accuracy drops of 1.1%, 1.22%, and 0.86% for the Tiny, Small, and Base models, respectively. These losses are offset by consistent throughput improvements, with the *Base* model achieving a $1.11\times$ speedup, and up to $1.16\times$ when using $k = 2$. The main gains stem from reduced sequence lengths passed through the Mamba block, directly lowering the computational cost of SSM recurrence and selective attention.

Although VMamba-S achieves slightly higher accuracy and throughput than VMamba-B with QuarterMap, this comparison does not account for the common deployment pipeline in real-world applications. In practice, larger models are often chosen for their superior generalization and transferability, especially in downstream tasks. QuarterMap is designed to fit this workflow by enabling efficient post-training compression of high-capacity models, without retraining or access to the original training data, thus preserving transfer performance while meeting run-time constraints.

This post-training strategy is particularly relevant in scenarios where retraining is costly or infeasible, such as on-device deployment, privacy-sensitive settings, or edge environments. QuarterMap supports this use case by offering a lightweight, architecture-aware pruning method that requires no model reconfiguration or finetuning, making it well suited for efficient deployment of pre-trained VMamba models.

D.2. Semantic Segmentation on ADE20K

Table 5. Performance comparison of VMamba-Upernet models with baseline and QuarterMap (QM) methods with $k = 3$ on ADE20K semantic segmentation.

Backbone	Method	Acc (%)	mIoU (%)
Base	Baseline	83.7	50.96
	QM (ours)	82.94 (-0.76)	49.21 (-1.75)
Small	Baseline	83.47	50.6
	QM (ours)	82.5 (-0.97)	48.81 (-1.79)
Tiny	Baseline	82.44	47.93
	QM (ours)	81.22 (-1.22)	44.99 (-2.94)

We evaluate QuarterMap on semantic segmentation using VMamba-Upernet (Liu et al., 2024; Xiao et al., 2018) backbones on ADE20K (Zhou et al., 2017). As shown in Table 5, QuarterMap incurs only a 0.76% decrease in all-pixel accuracy (aAcc) and a 1.75% drop in mean Intersection over Union (mIoU) in *Base* model. For the *Small* and *Tiny* variants, aAcc decreases by 0.97% and 1.22%, and mIoU by 1.79% and 2.94%, respectively.

D.3. BloodMNIST Class-Wise Accuracy

Table 6. Class-wise accuracy on BloodMNIST before and after applying QuarterMap. QuarterMap preserves fine-grained classification performance across all classes.

Class	Baseline Acc. (%)	QM Acc. (%)
Basophil	97.95	97.95
Eosinophil	99.68	99.68
Erythroblast	96.78	96.78
Immature granulocytes	96.55	96.72
Lymphocyte	99.59	99.59
Monocyte	94.72	94.72
Neutrophil	96.25	96.40
Platelet	100.00	100.00

To further evaluate QuarterMap's stability in fine-grained settings, we report class-wise performance on BloodMNIST in MedMNIST (Yang et al., 2021; 2023). As shown in Table 6, accuracy remains nearly identical across all categories, confirming that QuarterMap preserves semantic precision in medical imaging tasks. This finding demonstrate that QuarterMap generalizes effectively to VMamba-derived architectures and can be confidently applied in domain-specific deployments where accuracy preservation is critical.

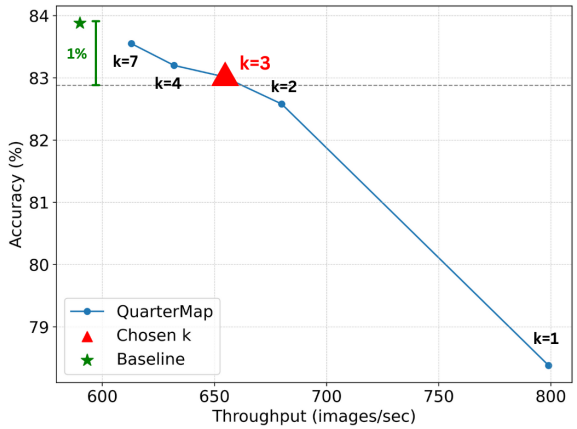

Figure 5. Pareto-optimal analysis from different block selection interval k.

D.4. Extended Restuls on Other Architectures

Table 7. Accuracy comparison of QuarterMap on CNN, Transformer, and Mamba models for ImageNet-1K classification. QM significantly impacts CNNs, causes a larger accuracy drop in Transformers, and is specifically designed to work on VMamba models.

Model	Type	Baseline Acc. (%)	QM (Ours) Acc. (%)
ConvNeXt-B	Conv	84.89	45.71
ConvNeXt-T	Conv	82.94	48.26
EfficientNetv2-L	Conv	75.75	64.71
EfficientNetv2-M	Conv	72.92	64.05
DeiT-B	Transformer	81.80	79.50
DeiT-S	Transformer	79.83	74.31
Swin-B	Transformer	85.17	82.91
Swin-S	Transformer	83.21	81.22
Swin-T	Transformer	81.18	78.56
ViM-Base	Mamba (1D)	80.40	71.00
ViM-Small	Mamba (1D)	80.50	73.20
PlainMamba-L2	Mamba (4D variant)	81.60	79.30
PlainMamba-L1	Mamba (4D variant)	77.70	73.30
VMamba-B	Mamba (4D)	83.88	83.02
VMamba-S	Mamba (4D)	83.64	82.42
VMamba-T	Mamba (4D)	82.60	81.50

To complement the analysis in the main text, we provide extended results of applying QuarterMap to additional CNNs, Transformers, and SSM-based architectures. As shown in Table 7, the variants of each model are reported. The results illustrate QuarterMap's relative effectiveness across architecture families, with VMamba achieving the most favorable trade-off between accuracy and throughput. In contrast, QuarterMap significantly degrades CNN performance and moderately affects ViTs and 1D-scanning SSMs like ViM and PlainMamba (Yang et al., 2024).

E. Ablation Study

E.1. Block selection

We investigate the impact of the block selection interval (k) for applying QuarterMap to the VMamba-B model on the ImageNet-1K classification task. As shown in Figure 5, selecting smaller values for (k) yields greater throughput gains but comes at the cost of significant accuracy degradation. To guide this trade-off, we highlight the 1% accuracy difference in the figure, demonstrating that $k = 3$ represents a reasonable choice based on the Pareto-optimal curve.

Additionally, we assess the impact of applying QuarterMap to different layers of the VMamba-B model, uniformly pruning all blocks within a layer. As shown in Table 8, applying QuarterMap to the first layer results in the most significant accuracy

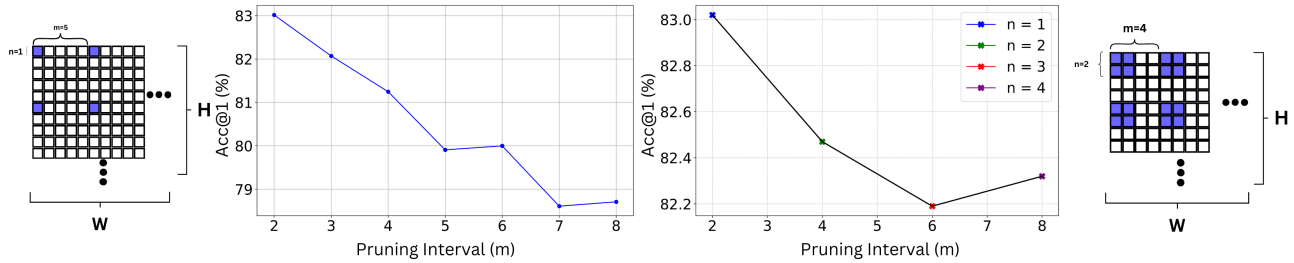

Figure 6. Ablation studies on feature map pruning in QuarterMap on ImageNet-1K classification.

drop, highlighting the critical role of early blocks in encoding fundamental low-level features essential for downstream tasks. This finding aligns with prior computer vision studies emphasizing the sensitivity of initial layers to pruning.

In contrast, applying QuarterMap to the third and deepest layer yields the largest throughput improvement due to its higher computational load and greater resilience to pruning. These results underscore the trade-off between accuracy retention and computational efficiency, emphasizing the importance of strategic layer selection when applying QuarterMap.

E.2. Feature map pruning methods

We conduct two ablation studies to investigate different methods for pruning feature maps when applying QuarterMap to the ImageNet-1K classification task. These experiments utilize the VMamba-B model, applying QuarterMap with $k = 3$ starting from the second layer. The ablation studies focus on varying the pruning interval m and and the number of consecutively retained tokens n.

In the first study, we retain one pixel out of every m pixels in both spatial dimensions of the feature map ($n = 1$). For instance, when $m = 4$, the original $H \times W$ feature map is reduced to $\lceil H/4 \rceil \times \lceil W/4 \rceil$, as in top-left of Figure 6. Results indicate that model accuracy decreases as m reflecting the trade-off between pruning granularity and accuracy. Smaller intervals preserve more spatial information, leading to better performance, whereas larger intervals result in greater information loss. Interestingly, comparable accuracy is observed across certain intervals (*e.g.*, $m = 5, 6$ and $m = 7, 8$) likely due to similar numbers of retained pixels despite differences in interval size.

In the second study, we select n continuous pixels out of every m pixels in both spatial dimensions, as depicted in the top-right of Figure 6. The results, shown in the bottom-right of Figure 6, indicate that none of the tested configurations outperformed the baseline (blue point). This observation supports our hypothesis that maintaining a critical density of spatial information is crucial for preserving model accuracy.

Table 8. Ablation study on applying QuarterMap to different model layers. The throughput is measured in images per second (img/s).

Model Layer	Acc@1 (%)	Throughput	Speedup
Baseline	83.88	590	1×
1	78.75	645	1.09×
2	83.00	618	1.05×
3	80.48	748	1.27×
4	83.79	597	1.01×

Table 9. Comparison of different upsampling methods in terms of accuracy and throughput measured in images per second.

Method	Acc@1 (%)	Throughput
Baseline	83.88	590
Nearest	83.02	654
Bilinear	82.97	205
Bicubic	82.04	161

E.3. Upsampling

We assess the impact of different upsampling methods, specifically nearest neighbor, bilinear, and bicubic, on accuracy (Acc@1) and throughput, utilizing the VMamba-B model on the ImageNet-1K classification task. As shown in Table 9, both bilinear and bicubic upsampling lead to a notable reduction in throughput. This is attributed to their computationally intensive interpolation processes, which involve calculations across multiple neighboring pixels, thereby imposing a substantial computational burden. In contrast, nearest neighbor upsampling achieves higher throughput through simpler calculations, making it a more efficient choice for implementing QuarterMap.