arXiv:2607.18625v1 [cs.CV] 21 Jul 2026

Norm or Direction? Decoding Vision Mambas for High-Resolution Vision

Jin Yu
Center for Humanoid Research
Korea Institute of Science and Technology
jinyu@kist.re.kr

Juyoun Park*
Center for Humanoid Research
Korea Institute of Science and Technology
juyounpark@kist.re.kr

Abstract

Vision Mamba models replace quadratic self-attention with linear complexity selective state space models (SSMs), emerging as efficient visual backbones. However, MambaOut demonstrates that a Gated CNN block can match or exceed VMamba on image classification, questioning the necessity of SSMs for vision. This raises a fundamental question: do VMamba and MambaOut encode visual information differently at the representation level? To investigate, we apply cross model centered kernel alignment (CKA) analysis and find that VMamba's final stage blocks form representations distinctly different from both MambaOut and its own preceding blocks. We therefore focus on the final block features, decomposing each spatial token into magnitude and direction. MambaOut concentrates class-discriminative information in high-norm foreground tokens that align with Grad-CAM attribution. VMamba, by contrast, produces high-norm tokens predominantly in background regions, misaligned with Grad-CAM, yet preserves discriminative signals primarily in token directions. These observations reveal that the two models rely on different encoding strategies. We connect this difference to high-resolution classification and semantic segmentation. VMamba distributes logit support broadly across object regions, whereas MambaOut relies on sparse dominant tokens, a strategy that becomes less stable as token counts grow. Under full fine-tuning for segmentation, VMamba consistently outperforms MambaOut. These results suggest that VMamba's advantage in dense prediction stems not merely from the SSM mechanism or sequence length, but from how semantic evidence is organized across token magnitude, direction. Ultimately, we conclude that token magnitude and directional structure serve as critical axes for improving visual backbones, particularly under dense supervision.

1 Introduction

Mamba [1] is an RNN-like sequence model that processes sequences with linear complexity through a selective scan mechanism, offering an efficient alternative to quadratic self-attention [2]. Recent Vision Mamba models [3, 4, 5, 6, 7, 8, 9] adapt this principle to visual recognition. Among them, VMamba [3] builds a hierarchical backbone via a 2D selective scan module (SS2D), achieving competitive performance across image classification, object detection, and semantic segmentation.

However, MambaOut [10] questions whether selective scan is strictly necessary for vision: replacing Mamba blocks with Gated CNN blocks [11, 12] is sufficient to match or outperform VMamba on image classification. MambaOut attributes this to task properties, arguing that classification is neither a long-sequence nor an autoregressive task. Yet VMamba retains advantages on dense prediction benchmarks, where longer spatial sequences arise. This raises a fundamental question: do

*corresponding author

Preprint.

VMamba and MambaOut organize class-discriminative information in the same way, or do they rely on fundamentally different encoding strategies?

To investigate, we apply cross model centered kernel alignment (CKA) [13, 14] and find that VMamba's final stage blocks form representations that are distinct from both MambaOut and their own preceding blocks, exhibiting cross model divergence and greater sensitivity to supervision change. We trace this to a difference in how each model encodes class-discriminative information. Decomposing spatial tokens into magnitude and direction, we find that MambaOut concentrates class information in high-norm foreground tokens that align with Grad-CAM [15] attribution, whereas VMamba produces high-norm tokens predominantly in background regions and preserves discriminative signals primarily in token directions. We then connect this representational difference to task performance. Token replacement attribution reveals that VMamba distributes logit support broadly across object regions, while MambaOut concentrates it in fewer dominant tokens. This distributional difference is consistent with VMamba's growing advantage as token count increases at higher resolutions. Under dense supervision, VMamba's direction-based encoding reorganizes more readily into spatially selective representations, contributing to its stronger performance under full fine-tuning for segmentation. The unit-token decoder test further confirms that VMamba's dense prediction remains largely intact when token magnitudes are removed, while MambaOut degrades sharply.

Taken together, these findings suggest that VMamba's advantage in dense prediction stems not merely from long-range dependency modeling, but from how discriminative information is distributed across token directions. Our results identify token magnitude and direction as key axes for designing high-resolution visual backbones, and point to foreground-aligned magnitude regularization and angular regularization of token directions as concrete directions for improving performance in dense and high-resolution settings.

2 Related Work

Generic visual backbones. Convolutional neural networks (CNNs) [16, 17, 18, 19] have long served as the dominant backbone for visual recognition. Vision Transformers (ViT) [20, 21, 22, 23] later demonstrated superiority over CNNs in classification. To better support multi-scale visual recognition, subsequent work introduced hierarchical designs [24, 25, 26, 27, 28, 29]. However, the quadratic complexity of dot-product attention [2] poses fundamental challenges in processing long sequences [30], which becomes particularly problematic for high-resolution visual inputs. Linear attention [31] offers inherent $O(N)$ complexity and has since been widely explored [32, 33, 34, 35]. Despite their efficiency, linear attention methods consistently underperform softmax attention, limiting their practical adoption.

State Space Models (SSMs) for visual recognition. SSMs provide an alternative path to linear complexity modeling by capturing long-range dependencies through structured state transitions. Early approaches such as S4 [36] parameterize linear time-invariant (LTI) systems with static, input-independent kernels and were originally designed for 1D sequence modeling. S4ND [37] extends this formulation to images via an outer-product construction while retaining the same static limitation. Mamba [1] addresses this limitation by introducing input-dependent selective scanning, improving adaptability without sacrificing linear complexity. Recent efforts adapt Mamba to vision by applying sequence modeling to flattened 2D feature maps [3, 4, 5, 6, 8], often incorporating multi-directional or windowed scanning to better capture spatial structure. Among these, VMamba [3] proposes a 2D selective scan (SS2D) over four directions. In contrast, MambaOut [10] questions the necessity of SSMs for image classification, while leaving open their role in dense prediction. Motivated by this, we study how VMamba and MambaOut differ in their internal representations and how these differences relate to performance on high-resolution dense prediction tasks.

3 Background

State Space Models (SSMs). Originating from the Kalman filter [38], structured SSMs such as S4 [36] are linear time-invariant (LTI) systems that map the input signal $u(t) \in \mathbb{R}$ to the output response $y(t) \in \mathbb{R}$ via the hidden state $h(t) \in \mathbb{R}^N$, expressed as linear Ordinary Differential equations (ODEs):

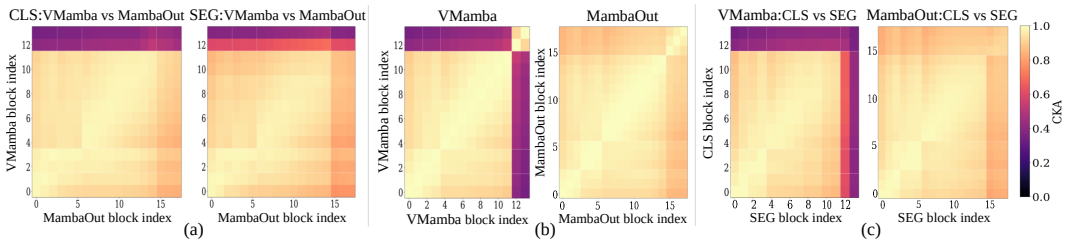

Figure 1: CKA similarity heatmaps between VMamba-T and MambaOut-T. (a) Cross model CKA for classification-pretrained (CLS) and segmentation fine-tuned (SEG) backbone, and (b) within model CKA across all blocks of VMamba and MambaOut. (c) Within model CKA between CLS and SEG representations.

$$h'(t) = Ah(t) + Bu(t),\ y(t) = Ch(t) + Du(t) \tag{1}$$

where $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times 1}$, $C \in \mathbb{R}^{1\times N}$, and $D \in \mathbb{R}^{1\times 1}$ are weighting parameters. Mamba [1] introduces a selective state space model (S6) with an explicit discretization based on the Zero-Order Hold (ZOH) method:

$$\begin{aligned} &\bar{A} = exp(\Delta A), \bar{B} = (\Delta A)^{-1}(exp(\Delta A) - I) \cdot \Delta B \\ &h_t = \bar{A}h_{t-1} + \bar{B}u_t,\ y_t = Ch_t + Du_t \end{aligned} \tag{2}$$

Δ is a timescale parameter that controls the discretization interval. In S6, B, C, and Δ are input-dependent, enabling content-aware sequence modeling while maintaining linear computational complexity.

Adapting S6 to vision requires bridging 1D sequential scanning and 2D spatial structure. A common approach flattens the 2D feature map into a 1D sequence prior to applying S6 [9, 3, 4, 5, 6]. VMamba [3] follows this approach, scanning the flattened sequence along four complementary directions via the proposed SS2D module.

2D Selective Scan (SS2D). Images do not have a natural one-dimensional order. SS2D addresses this by converting a 2D feature map into multiple 1D sequences. Given an input feature map, SS2D scans it in four directions: top-to-bottom, bottom-to-top, left-to-right, and right-to-left. Each sequence is processed by an S6 block [1], and the resulting sequences are reshaped and merged back into the original 2D spatial layout. Through these complementary scan directions, each spatial token can aggregate long-range context from different parts of the image.

VMamba and MambaOut. VMamba [3] and MambaOut [10] adopt a hierarchical four-stage backbone similar to ResNet [18], where feature resolution is gradually reduced across stages while the channel dimension increases. In both models, the final feature map is aggregated via global average pooling (GAP) [39, 40] before being passed to a classifier head. VMamba's Visual State Space (VSS) block follows the architecture of vanilla Transformer block [2], where the SS2D module replaces self-attention as the token mixer, followed by an FFN, with residual connections around each. MambaOut instead uses a Gated CNN block [11, 12], where a 7×7 depthwise convolution serves as the token mixer within a gating mechanism, following ConvNeXt [41]. MambaOut argues that SSM is unnecessary for standard image classification because classification is neither a long-sequence nor an autoregressive task. However, it leaves open the possibility that SSM remains useful for dense prediction, where the token sequence is much longer. This motivates our representation analysis: if the two models behave differently across tasks, their internal representations may also differ in a systematic way.

4 Representation Analysis

We compare how VMamba and MambaOut construct visual representations using the Tiny models (VMamba-T and MambaOut-T) throughout all experiments. We begin by identifying where

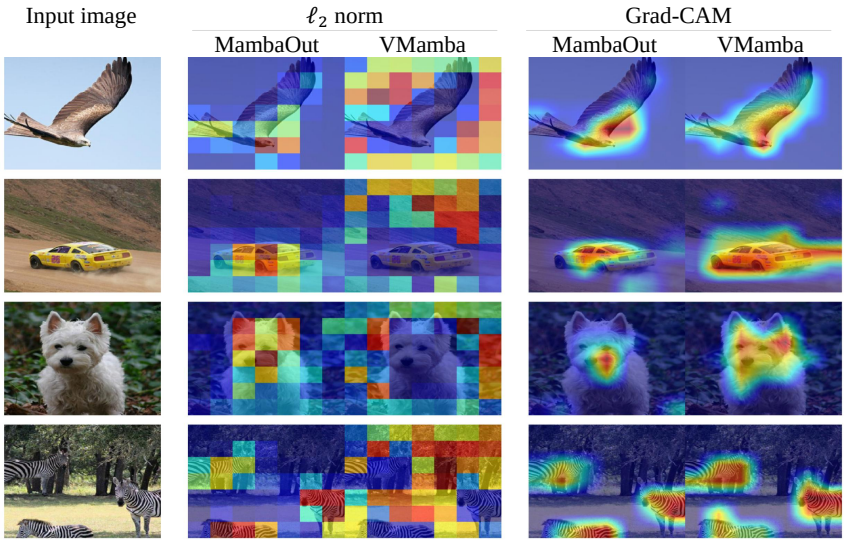

Figure 2: Visual comparison of token ℓ_2 norm and Grad-CAM heatmaps from the final block outputs of VMamba and MambaOut.

representational differences arise across stages, then focus on how the final stage features encode class-discriminative information.

4.1 Block-level Representation Similarity

Centered Kernel Alignment (CKA). We apply CKA [13, 14] to compare block-level representations across all stages of VMamba and MambaOut. CKA measures the similarity between two sets of representations evaluated on the same inputs, with a score of 1 indicating identical representations up to orthogonal transformation and 0 indicating no similarity. We construct three types of heatmap: cross model comparisons between VMamba and MambaOut, within model comparisons across all block pairs in each model, and within model comparisons between classification pretrained and segmentation fine-tuned backbones. Implementation details are provided in Appendix C.

Observation. As shown in Figure 1(a), earlier stages produce high CKA scores across the two models, indicating similar representations. However, VMamba's final stage blocks show low CKA scores not only with MambaOut but also with their own preceding blocks (Figure 1(b)), whereas MambaOut maintains consistently similar representations across layers. Figure 1(c) further shows that VMamba's classification and segmentation representations exhibit low CKA scores at the final stage, whereas MambaOut maintains high CKA scores throughout. These results indicate that VMamba's final stage forms qualitatively different representations, exhibiting both cross model divergence and sensitivity to supervision changes. This motivates a closer analysis of what they encode.

4.2 Where do high activations appear?

ℓ_2 norm and Grad-CAM. To investigate what distinguishes VMamba's final stage representations, we examine the spatial distribution of token activations. A high token norm indicates that the corresponding spatial location has a large activation magnitude. We compare this norm map with Grad-CAM [15], using the predicted class logit as the target. Grad-CAM identifies which spatial regions contribute the most to the classification decision.

Observation. Figure 2 shows a clear difference between the two models. In MambaOut, high-norm tokens are concentrated on foreground objects, and the Grad-CAM heatmap is also foreground-focused. Thus, token magnitude and classification attribution are spatially aligned. VMamba behaves differently. Its high-norm tokens often appear in background regions, while Grad-CAM remains concentrated on the foreground object. This spatial mismatch raises a direct question: do VMamba's high-norm background tokens actually carry class-discriminative information? Previous

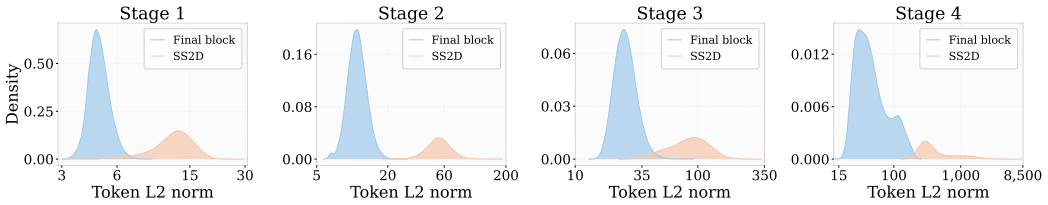

Figure 3: Token norm distributions across stages and block components.

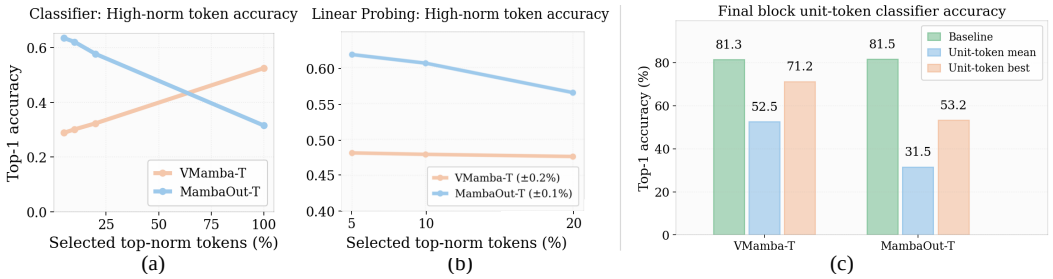

Figure 4: (a) High-norm token accuracy, (b) high-norm linear probing, and (c) unit-vector evaluation for VMamba and MambaOut.

works [42, 43] on ViT [44, 45, 46] and Vision Mamba [9] have reported a similar background concentration of high-norm tokens, characterizing them as global information repositories with a clearly bimodal norm distribution. VMamba, however, presents a subtly different picture. Rather than a discrete outlier population, its norm distribution is smooth and unimodal (Figure 3). This pattern suggests that VMamba's background tokens may not arise from the same outlier mechanism reported in prior work, and we investigate whether they nonetheless encode class-discriminative content in the following subsection.

4.3 Do high-norm tokens carry class information?

The previous observation raises a direct question: are VMamba's high-norm background tokens actually informative for classification? Prior work [42, 43] tests similar outlier tokens by using them for image classification. If high-norm tokens are the primary carriers of class-discriminative information, a classifier should be able to predict the image label using only those tokens. We evaluate this under two protocols: feeding high-norm tokens directly to the pretrained classification head, and training a linear probe via ridge regression, with implementation details in Appendix F.1 and D.

High norm test. Figure 4(a) reports the average accuracy obtained from the top 5%, 10% and 20% high-norm tokens. In MambaOut, accuracy is highest when using only the most high-norm tokens and decreases as more tokens are included. This indicates that class-discriminative evidence is concentrated in a small number of high-norm tokens, suggesting that MambaOut uses token magnitude as a strong cue for class-discriminative information. VMamba shows the opposite trend. Accuracy improves as more tokens are included. This means that the high-norm tokens are not necessarily the most informative ones. This suggests that VMamba does not use high-norm tokens as its primary class-information carriers.

High norm linear probing. Figure 4(b) shows a similar pattern under linear probing. MambaOut performs best with fewer high-norm tokens, again suggesting that class information is concentrated in a small high magnitude subset. VMamba shows relatively small variation across token ratios. This suggests that its class information is less concentrated in high-norm tokens and more distributed across spatial positions.

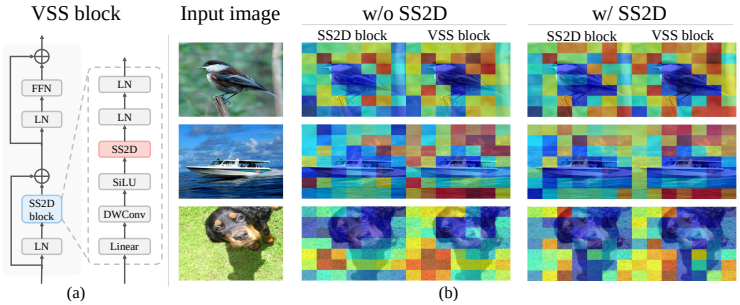

Figure 5: (a) Architecture of the VMamba's VSS block, where the SS2D block applies SS2D as its core scanning operation. (b) Token ℓ_2 norm heatmaps of two intermediate outputs (from the SS2D block and VSS block) under two conditions (with and without SS2D). The SS2D block without SS2D refers to the block with only the SS2D module removed.

4.4 VMamba preserves information in token directions

If VMamba does not primarily encode class information in token magnitude, where is the information preserved? A token vector can be decomposed into magnitude and direction, and we focus on the directional component. Specifically, we normalize each token to unit norm and evaluate individual normalized tokens using the pretrained classification head. This removes magnitude information and leaves only direction. If classification accuracy remains high after normalization, then class-discriminative information must be preserved in the angular component of the representation. Implementation details are provided in Appendix F.2.

Unit-vector test. Figure 4(c) shows that MambaOut suffers a substantial drop after token-wise normalization. Its average single-token accuracy becomes 31.5%, indicating that much of its discriminative signal depends on magnitude. VMamba remains much stronger after normalization. Its best single-token accuracy reaches 71.2%, and its average accuracy is 52.5%, which is 21% higher than MambaOut. These results suggest that VMamba and MambaOut use different encoding strategies. MambaOut relies more heavily on token magnitude. VMamba, by contrast, preserves class-discriminative information more strongly in token directions. This suggests a representation where information is preserved in token directions and distributed across spatial tokens, enabling both aggregation across tokens and adaptability under dense supervision. We test these implications in Section 5. Before that, we investigate what gives rise to this representation.

4.5 What gives rise to background high-norm?

Having established that VMamba's high-norm tokens are concentrated in background regions, we ask what gives rise to this pattern. A natural hypothesis is that this pattern is caused by the SS2D, since SS2D is the main selective scan mechanism in VMamba. However, our observations suggest that background high-norm is not solely an SSM-specific effect. As shown in Figure 5(b), the background high-norm pattern still appears when SS2D is removed from VMamba and the model is retrained. These results suggest that the phenomenon is not solely attributable to the selective scan mechanism.

These results characterize VMamba's representation. High-norm tokens are not necessarily aligned with semantic foreground evidence, while *class-discriminative information remains preserved in token directions* across spatial locations. This is consistent with the observations in Section 4.1, where VMamba differs from MambaOut and shows larger changes under different supervision. We next examine how this representation behaves when the task requires longer token sequences or dense spatial predictions.

5 High-Resolution Dense Prediction

We now relate the representation analysis to high-resolution recognition. Our goal is to test whether the representation differences identified in Section 4 translate into performance differences when the task departs from standard 224×224 ImageNet classification [47]. We focus on two implications from

Table 1: Classification comparison. MambaOut performs slightly better at standard resolution, whereas VMamba slightly surpasses MambaOut after high-resolution fine-tuning.

Task	Dataset	Res.	Setting	VMamba-Tiny	MambaOut-Tiny
Classification	ImageNet-1k	224	Pretrained	81.3	**81.5**
	ObjectNet	224	Zero-shot	45.4	**46.1**
	ImageNet-1k	768	Fine-tuning	**83.4**	83.3

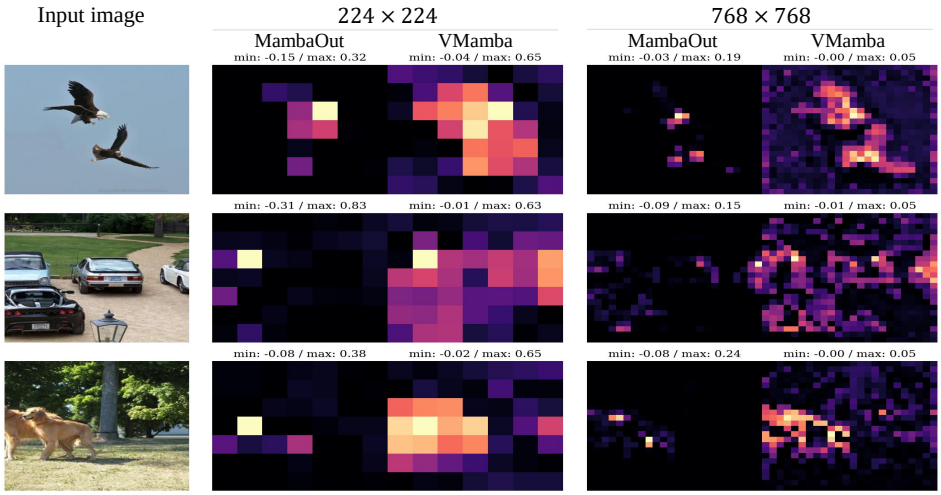

Figure 6: Token replacement attribution at 224×224 and 768×768 resolution for VMamba and MambaOut. For each final block spatial token, we replace only that token with the image-wise mean token and measure the drop in the target class logit.

Section 4.4: stable evidence aggregation as the number of tokens increases, and adaptability under dense supervision. We consider three settings designed to disentangle distinct factors: generalization robustness, sequence length, and dense spatial supervision.

5.1 High-resolution classification

We ask whether VMamba's advantage in dense prediction stems from sequence length or whether it reflects a more fundamental representational difference. To separate these two factors, we evaluate both models at 768×768 resolution, where the number of spatial tokens is approximately 11 times larger than at 224×224, while the task remains classification. Since segmentation already constitutes a long-sequence dense prediction setting analyzed separately below, we apply high-resolution fine-tuning only to the classification task. We fine-tune the 224×224-pretrained VMamba and MambaOut models on ImageNet at 768×768 resolution for 30 epochs. We also evaluate on ObjectNet [48], a robustness benchmark in which test images are collected under controlled variation in viewpoint, rotation, and background, following the evaluation protocol of [49, 50]. Since ObjectNet has no training set, evaluation is zero-shot. We include this as a robustness check, testing whether MambaOut's advantage at 224×224 persists under changes in viewpoint, rotation, and background.

Table 1 shows that MambaOut outperforms VMamba at 224×224, with 81.5% vs. 81.3% top-1 accuracy on ImageNet-1k. The same trend holds on ObjectNet zero-shot evaluation, where MambaOut achieves 46.1% compared with VMamba's 45.4%, suggesting that its standard-resolution advantage is not confined to the ImageNet validation distribution. After high-resolution fine-tuning at 768×768, however, VMamba obtains 83.4%, slightly surpassing MambaOut at 83.3%. At this resolution, the number of spatial tokens is approximately 11 times larger. To better understand this change, we examine how the target logit is supported by individual final stage spatial tokens at both resolutions.

Token replacement attribution. We further analyze evidence aggregation using a token-level perturbation analysis. Inspired by perturbation-based visual explanation methods [51, 52], we adapt token-level perturbation to measure evidence aggregation at the final stage. Given the feature map

Table 2: Segmentation comparison. MambaOut is stronger with a frozen backbone, whereas VMamba is stronger after full fine-tuning.

Task	Dataset	Res.	Setting	VMamba-Tiny	MambaOut-Tiny
Segmentation	ADE20K	512	Backbone frozen	40.2	**40.7**
	ADE20K	512	Full fine-tuning	**47.9**	47.1
	ADE20K	512	Unit-token input	**47.5**	41.5
	Pascal VOC 2012	512	Full fine-tuning	**81.6**	79.2

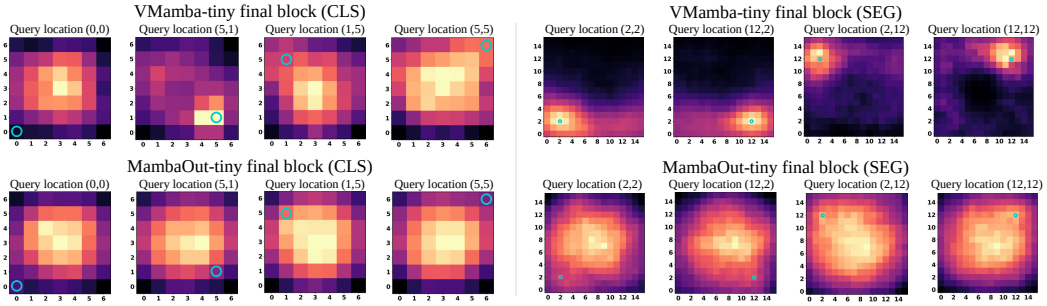

Figure 7: CKA-based spatial localization comparison between classification-pretrained and segmentation fine-tuned VMamba and MambaOut. (Green circle: query location)

$X = \{x_i\}_{i=1}^{N}$, we replace each token x_i with the image-wise mean $\bar{x} = \frac{1}{N}\sum_{j=1}^{N} x_j$ and measure the resulting drop in the target class logit. We define the token replacement score as

$$e_i = f_c(X) - f_c\big(X^{(i \leftarrow \bar{x})}\big), \tag{3}$$

where $f_c(\cdot)$ denotes the target class logit. A larger e_i indicates that token i provides stronger positive support for the prediction. Implementation details are provided in Appendix E.

In Figure 6, brighter regions indicate larger positive logit drops. The min/max values shown above each heatmap denote the minimum and maximum token replacement scores within that heatmap. This figure shows that VMamba and MambaOut exhibit different evidence aggregation patterns. At both 224×224 and 768×768, VMamba shows logit-supporting evidence over broader object regions, whereas MambaOut concentrates evidence in fewer object locations. This suggests that MambaOut relies on sharper but sparser evidence peaks, while VMamba uses a broader spatial support. This observation provides a representation-level correlate of the high-resolution trend. It suggests that distributing information across token directions enables more effective aggregation as the number of tokens increases. At 224×224, concentrating evidence in a few object tokens can be sufficient for image-level classification, consistent with MambaOut's advantage. At 768×768, however, the final feature map contains more spatial tokens. In this setting, relying on a small set of dominant evidence tokens may make aggregation more sensitive to spatial sampling and resolution changes, whereas VMamba's broader evidence support may provide a more stable basis for global classification.

5.2 Semantic segmentation

Semantic segmentation differs from classification in a fundamental way. Classification predicts one label for the whole image, so only image-level evidence is directly supervised. Segmentation predicts a label for every spatial location. Therefore, background regions are not merely context; they are part of the prediction target. This distinction changes how we interpret VMamba's background high-norm tokens. In classification, they appear misaligned with class-discriminative attribution, with Grad-CAM focusing on foreground objects and high-norm tokens in the background. In segmentation, however, every spatial location is supervised, making this misalignment less consequential. We further investigate whether dense supervision alters the spatial structure of the learned representations.

Frozen backbone. When the backbone is frozen, MambaOut outperforms VMamba on ADE20K, achieving 40.7 mIoU compared with VMamba's 40.2 mIoU (first row of Table 2). This is consistent

with MambaOut's magnitude-aligned foreground evidence: a newly trained segmentation head can exploit these strong spatial magnitude cues without updating the backbone.

Full fine-tuning. When the entire backbone is fine-tuned, VMamba outperforms MambaOut on ADE20K, achieving 47.9 mIoU compared with 47.1, and shows a larger margin on Pascal VOC 2012 [53], where VMamba reaches 81.6 mIoU against MambaOut's 79.2 (second and fourth row of Table 2). This pattern is also reflected in the CKA analysis (Figure 1). VMamba's representations change substantially at the final stage under dense supervision, suggesting that backbone updates induce a representational structure better suited to segmentation. This aligns with the observed performance reversal. It also raises a natural question: does VMamba's advantage under fine-tuning primarily arise from its token directions, or does it also depend on magnitude information?

Unit-token decoder test. To test whether dense prediction can be supported by token direction alone, we replace each stage feature passed to the decoder with its ℓ_2-normalized direction at a preserved feature scale. Under this modification, VMamba drops only slightly on ADE20K [54], from 47.9 to 47.5 mIoU. In contrast, MambaOut declines sharply from 47.1 to 41.5 mIoU (third row of Table 2). This suggests that VMamba's dense prediction representation remains largely usable when raw token magnitude is removed, whereas MambaOut tends to rely more on magnitude information. This does not imply that VMamba ignores magnitude, but it indicates that its token directions are more sufficient for dense prediction than those of MambaOut.

Spatial localization by CKA. We examine whether dense supervision changes the spatial structure of the learned representations using a CKA-based spatial localization analysis [55], evaluating both classification-pretrained and segmentation fine-tuned backbones on ImageNet-1K validation images. Details are in Appendix C. As shown in Figure 7, classification-pretrained models show weak spatial localization, whereas segmentation fine-tuned VMamba exhibits markedly clearer spatial selectivity. MambaOut shows little change after fine-tuning, consistent with its magnitude-concentrated encoding being less amenable to spatial reorganization. These results indicate that VMamba's direction-based encoding is more readily reshaped by dense supervision into spatially selective representations, consistent with the within model CKA shift observed in Section 4.1.

The CKA analysis and unit-token decoder test suggest two complementary reasons for VMamba's fine-tuning advantage. Under dense supervision, VMamba develops clearer spatial selectivity, and its token directions retain useful semantic and spatial information even when magnitude is removed. Together, these results indicate that directional encoding yields a more adaptable representation once the backbone is updated under dense supervision. Overall, the results suggest that VMamba's advantage is not explained solely by the presence of SSM. Its representation appears better suited to adaptation under dense spatial supervision. MambaOut appears easier to exploit with a frozen backbone, possibly because its magnitude-aligned foreground evidence provides accessible spatial cues for the segmentation head. *VMamba, in contrast, becomes stronger under full fine-tuning, where dense supervision can reshape spatial selectivity and exploit direction-preserved semantic evidence.*

6 Discussion

VMamba and MambaOut encode class-discriminative information through distinctly different strategies. MambaOut concentrates on a small set of high-norm foreground tokens, while VMamba distributes it across spatial locations through token directions. Both patterns appear to originate from VMamba's block structure, which produces background high-norm tokens associated with direction-based encoding. This unified accounts for the cross model representational divergence at the final stage and VMamba's greater sensitivity to supervision change. The practical consequences follow directly. Magnitude-aligned encoding supports efficient aggregation at standard resolution, where a few dominant tokens suffice for image-level prediction. Direction-based encoding becomes advantageous when token count grows or supervision is spatially dense, as directional information aggregates more stably and reorganizes more readily under fine-tuning.

These results identify token magnitude and direction as key design axes for high-resolution visual backbones. Foreground-aligned magnitude regularization may strengthen standard-resolution classification, while angular regularization or token-wise contrastive objectives may improve dense and high-resolution performance.

Limitations. The observed differences are attributed to VMamba's block structure, but the precise contributing components remain unclear, as the SSM cannot be isolated from the surrounding architecture without targeted ablations.

References

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

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[3] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37, 2024.

[4] Chenhongyi Yang, Zehui Chen, Miguel Espinosa Minano, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J Crowley. Plainmamba: Improving non-hierarchical mamba in visual recognition. In *35th British Machine Vision Conference*, 2024.

[5] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. In *European conference on computer vision*, 2024.

[6] Abdelrahman Shaker, Syed Talal Wasim, Salman Khan, Juergen Gall, and Fahad Shahbaz Khan. Groupmamba: Efficient group-based visual state space model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[7] Ali Hatamizadeh and Jan Kautz. Mambavision: A hybrid mamba-transformer vision backbone. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[8] Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. In *Proceedings of the AAAI conference on artificial intelligence*, 2025.

[9] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: efficient visual representation learning with bidirectional state space model. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[10] Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[11] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, 2017.

[12] Weihao Yu, Pan Zhou, Shuicheng Yan, and Xinchao Wang. Inceptionnext: When inception meets convnext. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, 2024.

[13] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13, 2012.

[14] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, 2019.

[15] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[19] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, 2021.

[22] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[23] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in neural information processing systems*, 34, 2021.

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

[25] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[26] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[27] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[28] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *European conference on computer vision*, 2022.

[29] Xiaosong Zhang, Yunjie Tian, Lingxi Xie, Wei Huang, Qi Dai, Qixiang Ye, and Qi Tian. Hivit: A simpler and more efficient design of hierarchical vision transformer. In *The Eleventh International Conference on Learning Representations*, 2023.

[30] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55, 2022.

[31] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[32] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

[33] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. In *International Conference on Learning Representations*, 2022.

[34] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, 2021.

[35] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, et al. Rwkv: Reinventing rnns for the transformer era. In *Findings of the association for computational linguistics: EMNLP 2023*, 2023.

[36] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

[37] Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals with state spaces. *Advances in neural information processing systems*, 35, 2022.

[38] RE Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82, 1960.

[39] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. 2014.

[40] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[41] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[42] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024.

[43] Feng Wang, Jiahao Wang, Sucheng Ren, Guoyizhe Wei, Jieru Mei, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Mamba-reg: Vision mamba also needs registers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[44] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *European conference on computer vision*, 2022.

[45] Gabriel Ilharco, Mitchell Wortsman, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, et al. Openclip. 2021.

[46] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 2024.

[47] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 2009.

[48] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 2019.

[49] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33, 2020.

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 2021.

[51] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 2014.

[52] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[53] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88, 2010.

[54] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[55] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34, 2021.

[56] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

[57] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[58] Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. *Advances in neural information processing systems*, 20, 2007.

[59] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2021.

Appendix

Additional details and results from the different sections are included below.

A Experimental Setting

All experiments use ImageNet-1K [47] pretrained weights of VMamba-T and MambaOut-T and are conducted on 4 NVIDIA RTX PRO 6000 GPUs. Unless otherwise stated, all figures and analyses are produced using the ImageNet-1K validation set. We fine-tune the 224×224-pretrained models on ImageNet-1K at 768×768 resolution for 30 epochs, reducing the original learning rate by a factor of $1/10$ for stable fine-tuning. For semantic segmentation, we initialize from the ImageNet-1K pretrained backbones and fine-tune with UperNet [56] on ADE20K [54] for 160K iterations with batch size 16 and AdamW optimizer [57], following the standard protocol used by both VMamba [3] and MambaOut [10]. For Pascal VOC 2012 [53], we train with a batch size of 16 for 20K iterations.

B Additional Qualitative Results

Figures 8, 9, and 10 provide additional qualitative results extending Figures 2, 5, and 6 in the main paper, respectively.

C CKA Implementation Details

In Sections 4.1 and 5.2, we measure representational similarity using Centered Kernel Alignment (CKA) [13, 14], which enables consistent comparison of representations across layers and models. Let $X \in \mathbb{R}^{m \times d_1}$ and $Y \in \mathbb{R}^{m \times d_2}$ denote activation matrices from two layers evaluated on the same set of m samples. The corresponding linear Gram matrices are $K = XX^\top$ and $L = YY^\top$.

Using the centering matrix $H = I_m - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$, we compute centered Gram matrices $K_c = HKH$ and $L_c = HLH$. CKA is then defined as

$$\mathrm{CKA}(K, L) = \frac{\mathrm{HSIC}(K_c, L_c)}{\sqrt{\mathrm{HSIC}(K_c, K_c)\,\mathrm{HSIC}(L_c, L_c)}}, \tag{4}$$

where the Hilbert-Schmidt Independence Criterion (HSIC) [58, 14] is computed as

$$\mathrm{HSIC}(K_c, L_c) = \frac{1}{(m-1)^2}\mathrm{tr}(K_c L_c). \tag{5}$$

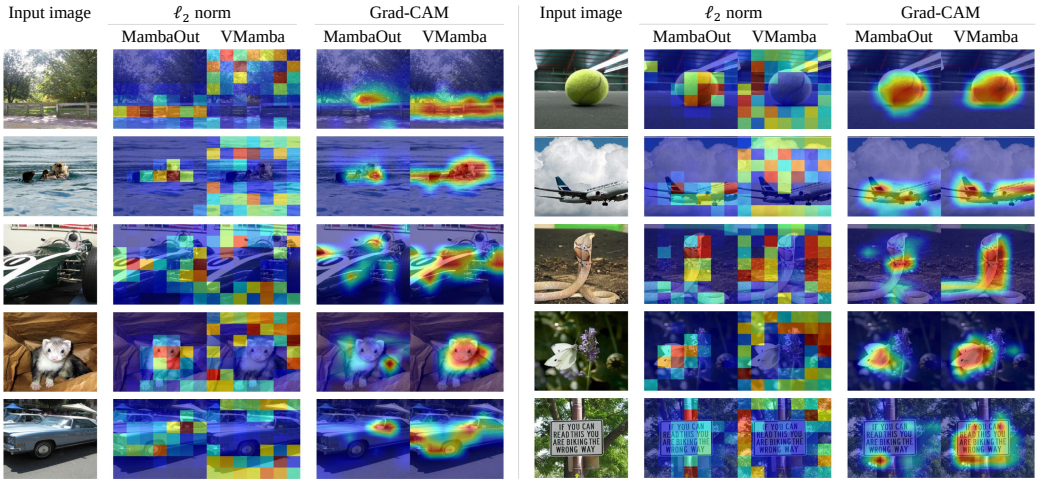

Figure 8: Visual comparison of token ℓ_2 norm and Grad-CAM heatmaps from the final block outputs of VMamba and MambaOut.

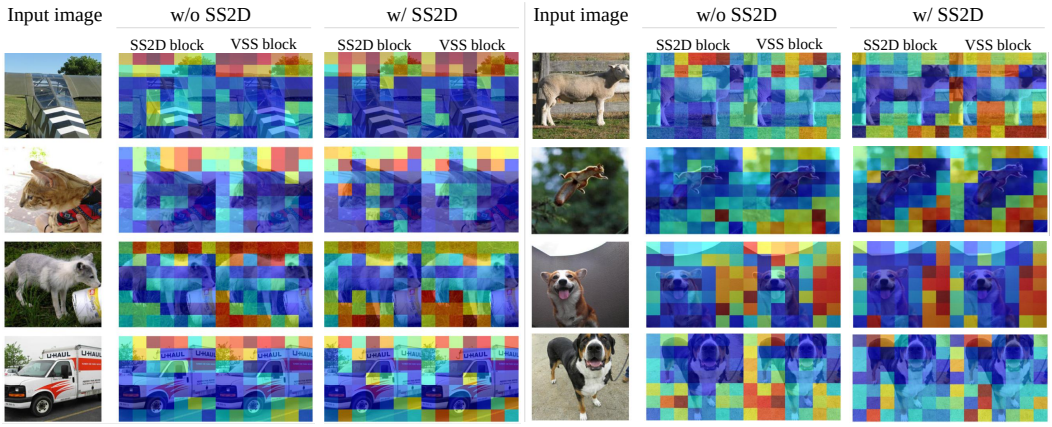

Figure 9: Token ℓ_2 norm heatmaps at two intermediate outputs (SS2D block and VSS block) under two conditions (with and without SS2D).

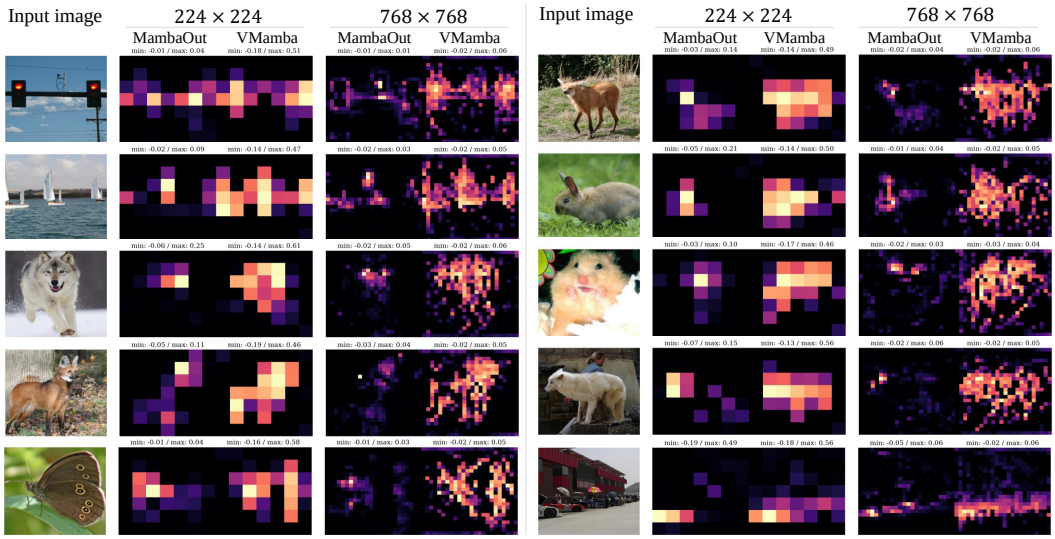

Figure 10: Token replacement attribution at 224×224 and 768×768 resolution for VMamba and MambaOut. For each final block spatial token, we replace only that token with the image-wise mean token and measure the drop in the target class logit.

This formulation is invariant to orthogonal transformations and isotropic scaling of representations, enabling meaningful comparison across layers. We compute CKA using the minibatch estimator of [59]. We sample 2,560 images from the ImageNet-1K validation set per run (batch size 128) and report averages over 10 independent repetitions. This configuration yields results comparable to larger sample settings used in prior work [55].

Cross model comparison. Following [55], we compute CKA between all pairs of blocks in VMamba and MambaOut, producing a cross-model similarity matrix that captures representational correspondence across architectures. For each block, we use the block outputs. Activations are extracted via forward hooks and reshaped to $\mathbb{R}^{m \times d}$ prior to computing Gram matrices. In addition, we compute within model CKA between classification pretrained and segmentation fine-tuned models to quantify representation shifts under different supervision.

Spatial localization. Following [55], we evaluate spatial localization by measuring the correspondence between token representations and their associated input spatial positions. For each token, we compute its CKA similarity with patch embeddings at all spatial locations, forming a spatial similarity

map. A peaked response at the token's corresponding location indicates strong spatial localization, whereas a more uniform response suggests globally distributed or non-localized representations.

D Linear Probing Details

In Section 4.3, we evaluate the class-discriminability of intermediate representations via linear probing, following the regularized least-squares protocol of [20, 55]. Let $X \in \mathbb{R}^{n\times d}$ denote the feature matrix aggregated over n samples and $Y \in \{-1,+1\}^{n\times C}$ the corresponding target matrix, where $C = 1{,}000$ is the number of ImageNet classes. The probe weights $W \in \mathbb{R}^{d\times C}$ are obtained by solving

$$W^* = \arg\min_W \ \|XW - Y\|_F^2 + \lambda\|W\|_F^2 = \big(X^\top X + \lambda I\big)^{-1} X^\top Y, \tag{6}$$

and the solution is recovered in closed form without updating the backbone.

Top-norm token selection. We restrict the probe to spatial tokens with high activation magnitude. For each image, we rank all T tokens from the final block by their ℓ_2 norm and retain the top-$k\%$, yielding $\lceil T \cdot k/100 \rceil$ tokens per image. This focuses the probe on strongly activated regions while discarding low-activation background tokens. We use $k \in \{5, 10, 20\}$ and operate on raw activations. Performance is reported as the average per-token classification accuracy over the selected tokens.

Training setup. We train probes using 10 images per class (10-shot), sampled uniformly from the ImageNet-1K training set, with regularization parameter $\lambda = 0.1$. Evaluation is performed on the full ImageNet-1K validation set.

E Token Replacement Attribution

In Section 5.1, we quantify the spatial importance of tokens in the final block using a token replacement attribution method.

Attribution definition. Let $X \in \mathbb{R}^{T\times d}$ denote the spatial token matrix at the output of the final block, and let $f_c(X)$ denote the logit for target class c. For each token position $i \in \{1, \dots, T\}$, we define the attribution score as

$$e_i = f_c(X) - f_c\big(X^{i\leftarrow \bar{x}}\big), \tag{7}$$

where $X^{i\leftarrow\bar{x}}$ replaces the i-th token with the image-wise mean token $\bar{x} = \frac{1}{T}\sum_{j=1}^{T} x_j$. A larger e_i indicates greater contribution of token i to the target class.

Implementation details. The target class c is defined as the top-1 predicted class of each model, and attribution is computed with respect to the model's own prediction. We evaluate inputs at resolutions of 224×224 and 768×768.

F Evaluation Details

F.1 High norm test.

In Section 4.3, for each image, we rank all T spatial tokens from the final block by their ℓ_2 norm and select the top-$k\%$ highest-norm tokens with $k \in \{5, 10, 20\}$, operating on raw activations. Each selected token is then passed individually through the pretrained classification head, and top-1 accuracy is averaged over all selected tokens across the ImageNet-1K validation set.

F.2 Unit vector test.

In Sections 4.4 and 5.2, both evaluations remove token magnitude by normalizing to unit ℓ_2 norm, leaving only the directional component. For classification (Section 4.4), every token in the final block is normalized and passed individually to the pretrained head, and top-1 accuracy is averaged over

all T tokens. For segmentation (Section 5.2), each token x_i in every stage feature map fed to the UperNet decoder [56] is replaced by

$$\tilde{x}_i = \sqrt{C}\,\frac{x_i}{\|x_i\|_2}, \tag{8}$$

where the $\sqrt{C}$ factor keeps the feature scale comparable to standard-normalized inputs. This substitution is applied to all four stage outputs simultaneously, with backbone weights remaining unchanged.