NORM OR DIRECTION? DECODING VISION MAMBAS FOR HIGH-RESOLUTION VISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision Mamba models replace quadratic self-attention with linear-complexity selective state space models (SSMs). MambaOut, however, shows that a Gated CNN block can match or exceed VMamba on image classification, questioning whether SSMs are necessary for vision. We ask whether the two models nonetheless encode visual information differently. Using centered kernel alignment (CKA), we find that VMamba's final stage diverges from both MambaOut and its own preceding blocks, and we trace this divergence to two components. The token mixer sets how far information spreads. VMamba's SS2D combines local channels with channels whose memory spans the entire image, some of which remain global at 768×768 without retraining. The classifier head sets which part of a token the prediction reads. MambaOut pools before normalizing and concentrates class evidence in high-norm foreground tokens, whereas VMamba normalizes before pooling, places high-norm tokens in the background, and preserves class evidence in token directions. Swapping only the head order moves VMamba's high-norm tokens to the foreground. Consistent with direction-based evidence aggregating more stably as tokens grow, VMamba spreads logit support more broadly, degrades less under resolution transfer across three model scales, and outperforms MambaOut on segmentation after full fine-tuning. We suggest token mixing range, token magnitude and direction, and the normalization–pooling order as design axes for high-resolution and dense prediction backbones.

1 INTRODUCTION

Mamba (Gu & Dao, 2024) 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 (Vaswani et al., 2017). Recent Vision Mamba models (Liu et al., 2024; Yang et al., 2024; Huang et al., 2024; Shaker et al., 2025; Hatamizadeh & Kautz, 2025; Pei et al., 2025; Zhu et al., 2024) adapt this principle to visual recognition. Among them, VMamba (Liu et al., 2024) builds a hierarchical backbone with a 2D selective scan module (SS2D) and achieves competitive performance on image classification, object detection, and semantic segmentation.

MambaOut (Yu & Wang, 2025), however, questions whether selective scan is necessary for vision: replacing Mamba blocks with Gated CNN blocks (Dauphin et al., 2017; Yu et al., 2024) is sufficient to match or outperform VMamba on image classification. MambaOut attributes this to the nature of the task, arguing that classification is neither a long-sequence nor an autoregressive task. Yet VMamba retains an advantage on dense prediction benchmarks, where spatial sequences are longer. This raises a question: do VMamba and MambaOut organize class-discriminative information in the same way, or do they rely on different encoding strategies? We address this question by comparing the internal representations of the two models. A centered kernel alignment (CKA) (Cortes et al., 2012; Kornblith et al., 2019) analysis first localizes the largest difference to VMamba's final stage. To explain it, we examine two components that play separable roles in shaping the final representation: the token mixer, which determines how far information spreads across the image, and the classifier head, which determines whether the prediction reads token magnitude or token direction. We then test whether these differences carry over to settings beyond standard-resolution classification: resolution transfer, high-resolution fine-tuning, and semantic segmentation. Our contributions are as follows:

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

- Using CKA, we show that VMamba's final stage forms representations distinct from both MambaOut and its own preceding blocks, and that these representations change substantially under dense supervision.
- By analyzing the memory length of each scan, we show that SS2D combines local channels with channels whose memory spans the entire image, some of which stay global at 768×768.
- We show that MambaOut concentrates class evidence in high-norm foreground tokens, whereas VMamba preserves it in token directions, and that swapping only the head's normalization–pooling order moves VMamba's high-norm tokens to the foreground.
- We relate both differences to downstream behavior. VMamba spreads logit support more broadly, degrades less under resolution transfer across three model scales, and adapts more readily under dense supervision, consistent with direction-based evidence aggregating more stably as tokens grow.

2 Related Work

Generic visual backbones. Convolutional neural networks (CNNs) (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; He et al., 2016; Xie et al., 2017) have long served as the dominant backbone for visual recognition. Vision Transformers (ViT) (Dosovitskiy et al., 2021; Touvron et al., 2021; Yuan et al., 2021; Han et al., 2021) later surpassed CNNs in classification, and hierarchical designs (Liu et al., 2021; Yang et al., 2021; Wang et al., 2021; Dong et al., 2022; Ding et al., 2022; Zhang et al., 2023) extended them to multi-scale recognition. However, the quadratic complexity of dot-product attention (Vaswani et al., 2017) makes long sequences costly (Tay et al., 2022), especially for high-resolution inputs. Linear attention (Katharopoulos et al., 2020; Choromanski et al., 2021; Qin et al., 2022) reduces the complexity to $O(N)$ but generally underperforms softmax attention.

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. S4 (Gu et al., 2022) parameterizes linear time-invariant (LTI) systems with static, input-independent kernels for 1D sequences, and S4ND (Nguyen et al., 2022) extends this formulation to images while keeping the kernels static. Mamba (Gu & Dao, 2024) makes the scan input-dependent, improving adaptability without sacrificing linear complexity. Vision adaptations of Mamba apply the scan to flattened 2D feature maps (Liu et al., 2024; Yang et al., 2024; Huang et al., 2024; Shaker et al., 2025; Pei et al., 2025), often with multi-directional or windowed scanning to capture spatial structure. VMamba (Liu et al., 2024) scans in four directions with SS2D. MambaOut (Yu & Wang, 2025) questions the necessity of SSMs for image classification but leaves their role in dense prediction open. We study how VMamba and MambaOut differ in their internal representations and how these differences relate to high-resolution and dense prediction.

3 Background

State Space Models (SSMs). Originating from the Kalman filter (Kalman, 1960), structured SSMs such as S4 (Gu et al., 2022) are LTI systems that map an input signal $u(t) \in \mathbb{R}$ to an output $y(t) \in \mathbb{R}$ through a hidden state $h(t) \in \mathbb{R}^N$:

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

where $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times 1}$, $C \in \mathbb{R}^{1\times N}$, and $D \in \mathbb{R}^{1\times 1}$. Mamba (Gu & Dao, 2024) introduces a selective SSM (S6) discretized with the zero-order hold (ZOH):

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

where the timescale Δ controls the discretization interval. In S6, B, C, and Δ depend on the input, enabling content-aware sequence modeling at linear cost.

2D selective scan (SS2D). Images have no natural one-dimensional order. SS2D (Liu et al., 2024) flattens a 2D feature map into four 1D sequences (top-to-bottom, bottom-to-top, left-to-right, and right-to-left), processes each with an S6 block, and merges the outputs back into the 2D layout. Through these complementary directions, each token can aggregate context from across the image.

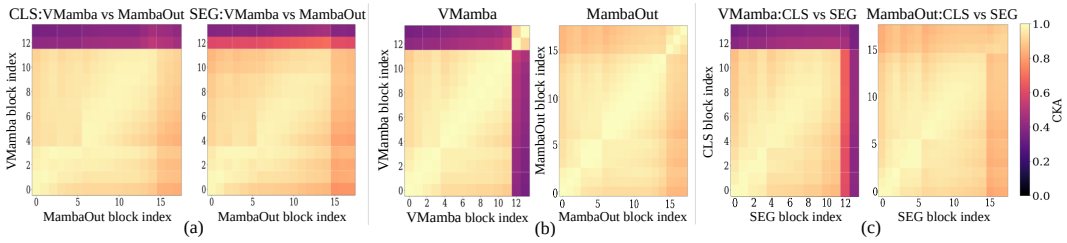

Figure 1: CKA similarity between VMamba-T and MambaOut-T. (a) Cross-model CKA for classification-pretrained (CLS) and segmentation fine-tuned (SEG) backbones. (b) Within-model CKA across all blocks. (c) Within-model CKA between CLS and SEG representations.

VMamba and MambaOut. Both models adopt a four-stage hierarchical backbone similar to ResNet (He et al., 2016), reducing spatial resolution and increasing channel dimension across stages. VMamba's Visual State Space (VSS) block follows the Transformer block (Vaswani et al., 2017), with SS2D replacing self-attention as the token mixer, followed by an FFN, and residual connections around each. MambaOut instead uses a Gated CNN block (Dauphin et al., 2017; Yu et al., 2024), in which a 7×7 depthwise convolution serves as the token mixer within a gating mechanism, following ConvNeXt (Liu et al., 2022). Both models aggregate the final feature map by global average pooling (GAP) (Lin et al., 2014; Szegedy et al., 2015) before the classifier, but they differ in where LayerNorm is applied: VMamba normalizes each token before pooling, whereas MambaOut normalizes the pooled feature. Section 4.4 shows this seemingly minor difference shapes how class information is encoded.

4 Representation Analysis

This section compares how VMamba and MambaOut build their representations, using the Tiny models for most analyses. Appendix F shows the same trends for Small and Base. Section 4.1 identifies where the two models diverge, and Sections 4.2–4.4 examine two components with separable roles. The token mixer sets how far information spreads, and the classifier head sets whether the prediction reads token magnitude or direction.

4.1 Block-level representation similarity

To locate where the two models diverge, we compute CKA (Cortes et al., 2012; Kornblith et al., 2019) between all pairs of blocks. CKA measures the similarity of two sets of representations on the same inputs, ranging from 1 (identical up to an orthogonal transformation) to 0 (no similarity). Appendix B provides implementation details. Figure 1(a) shows that the two models produce similar representations in the early stages. VMamba's final stage, however, differs not only from MambaOut but also from its own preceding blocks (Figure 1(b)) and from its segmentation fine-tuned counterpart (Figure 1(c)). MambaOut remains consistent in all three comparisons. VMamba's final stage therefore forms a distinct representation that is also sensitive to the type of supervision. This representation could depend on how the backbone mixes spatial information and how the classifier reads final stage tokens. We examine each in turn.

4.2 How far does VMamba's token mixer spread information?

We begin with the token mixer, the component in which the two models differ by design.

Effective receptive fields. We measure the effective receptive field (ERF) (Luo et al., 2016) at the output of each token mixer (Appendix C.1). As shown in Figure 2, MambaOut's ERF grows gradually with depth, as expected from stacked 7×7 convolutions, whereas VMamba's already covers a broad region in early blocks. We therefore examine each channel's range to see how this broad ERF arises.

Memory length of a scan. To measure how far each scan carries information, we examine how long a token's contribution survives in its state. SS2D scans a sequence of $L = H \times W$ tokens,

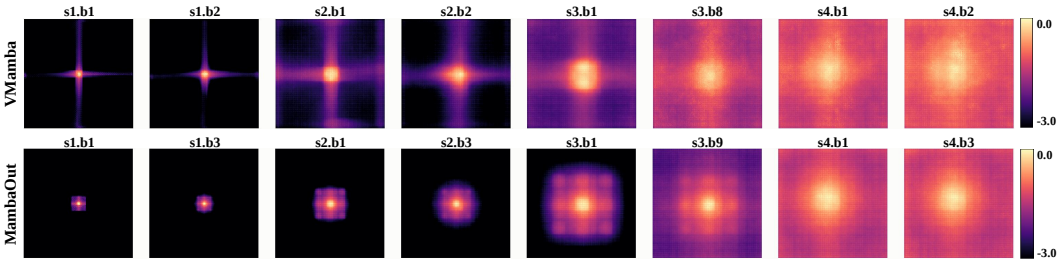

Figure 2: Effective receptive fields at the token mixer output of the first and last block of each stage.

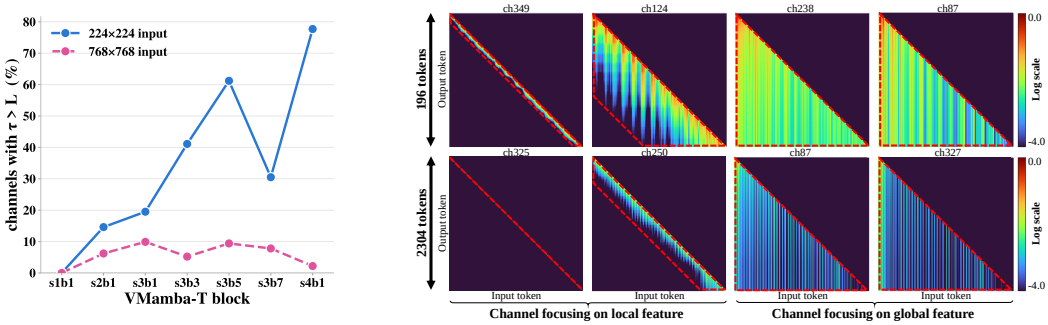

Figure 3: (Left) Fraction of global channels ($\tau > L$ in at least one scan direction) per block of VMamba-T at 224×224 (blue, training resolution) and 768×768 (pink, same weights without fine-tuning). (Right) Unrolled SSM weights of block `s3.b3` along the $\rightarrow$ scan for two local ($\tau < L$) and two global ($\tau > L$) channels at 224×224 (top) and 768×768 (bottom). Red dashed lines outline how far the weights spread from the diagonal.

and each direction–channel pair runs its own recurrence. In Eq. equation 2, the state is multiplied by $\bar{A} = \exp(\Delta a)$ at every step with a learned $a < 0$, so the fraction of a token's contribution that survives d tokens later is approximately

$$\exp(-|a|\,\bar{\Delta}\,d) = \exp\Big(-\frac{d}{\tau}\Big), \qquad \tau = \frac{1}{|a|\,\bar{\Delta}}, \tag{3}$$

where $\bar{\Delta}$ is the mean step size. Since $|a|\bar{\Delta}$ determines the per-token decay rate, its inverse τ characterizes how many tokens a scan can remember, and we call it the *memory length*. Because L is the length of the whole flattened image, comparing τ with L tells whether a scan can carry information across the entire image. Following the global/local classification of LongMamba (Ye et al., 2025), we call a scan *global* if $\tau > L$, so that a token's contribution remains substantial over the full sequence, and *local* otherwise. Since the four directional outputs are summed per channel, a channel is global if any of its scans is. Appendix C gives the details.

Global channels. Figure 3 (left) reports the fraction of global channels in each block. Global channels already appear in early blocks of stage 2, and their fraction varies substantially across blocks rather than increasing with depth. Blocks in stage 3 range from 4.9% to 61.2% despite sharing the same L. When the same weights are applied to 768×768 inputs, the sequence length grows by about 11 times, yet 0.9–9.9% of the channels remain global. The attention maps in Figure 3 (right), obtained by unrolling the left-to-right scan of each channel into a weighted sum of earlier tokens (Ali et al., 2025; Ben-Kish et al., 2025), show how global channels differ from local ones in the same block. Local channels concentrate weights near the diagonal, whereas global channels spread them over the whole causal range, where a few strongly written tokens appear as vertical stripes. Both patterns persist at 768×768 (bottom row).

VMamba's token mixer can thus propagate information across the whole image through a subset of channels, which a local convolution does not provide within a single layer. Since local and global channels coexist within a block, each token also combines context at several ranges at once, whereas

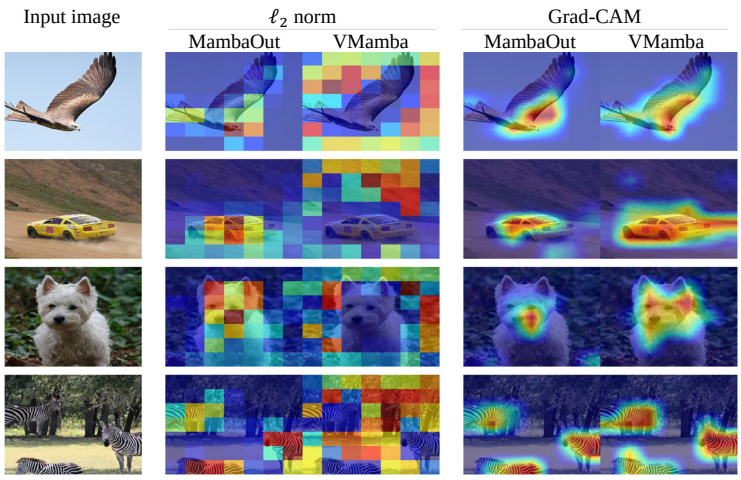

Figure 4: Token ℓ_2 norm and Grad-CAM maps of the final block outputs of VMamba and MambaOut.

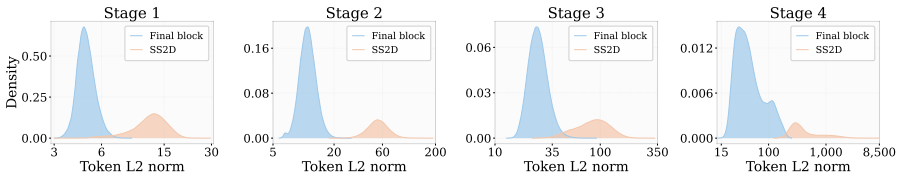

Figure 5: Token norm distributions across stages and block components.

a single convolution has one fixed range. This may benefit high-resolution and dense prediction, where each position can draw on context both near and far. However, how the final stage tokens are read out for prediction also matters. Since CKA identifies the largest divergence at this stage, we examine it next.

4.3 Token Magnitude and Direction

To understand how the final stage tokens carry class information, we decompose each token into its magnitude and direction, starting with magnitude.

Where do high-norm tokens appear? Figure 4 compares the ℓ_2 norm map of the final block with Grad-CAM (Selvaraju et al., 2017) for the predicted class. In MambaOut, high-norm tokens lie on the foreground object, where Grad-CAM also concentrates, so token magnitude and classification attribution are spatially aligned. In VMamba, high-norm tokens often appear in the background, while Grad-CAM remains on the object. Similar background high-norm tokens have been reported in ViTs (Touvron et al., 2022; Ilharco et al., 2021; Oquab et al., 2024) and Vim (Zhu et al., 2024), where they form a separate outlier mode in a bimodal norm distribution and act as global information repositories (Darcet et al., 2024; Wang et al., 2025). VMamba's norm distribution is instead smooth and unimodal (Figure 5), suggesting a different origin. We therefore test directly whether VMamba's high-norm tokens carry class-discriminative information.

High-norm test. If high-norm tokens are the main carriers of class information, the image label should be predictable from them alone (Darcet et al., 2024; Wang et al., 2025). We select the top 5%, 10%, and 20% of tokens by norm and classify them with the pretrained head and with a linear probe trained by ridge regression. Appendices D.1 and D.2 describe the two protocols. As shown in Figure 6(a,b), MambaOut is most accurate with the fewest tokens under both protocols, indicating that its class evidence is concentrated in a small high-magnitude subset. VMamba improves as more tokens are included under the pretrained head and varies little under the probe. Its highest-norm tokens are therefore not its most informative ones.

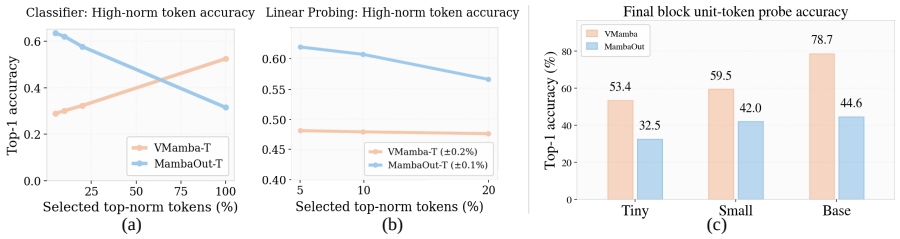

Figure 6: (a) Accuracy of the pretrained head on high-norm tokens, (b) linear probing on high-norm tokens, and (c) single-token linear probing on ℓ_2-normalized tokens for Tiny, Small, and Base models.

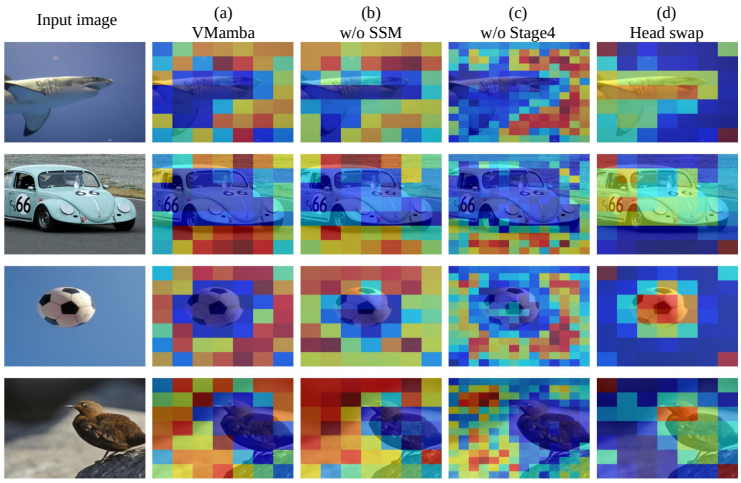

Figure 7: Token ℓ_2 norm maps of the final block outputs. (a) Original model. (b) SS2D removed and retrained. (c) Stage 4 removed and retrained. (d) Head order swapped to GAP→LN and retrained.

Unit-vector test. If magnitude does not carry VMamba's class information, direction may. We train the same linear probe on final block tokens normalized to unit ℓ_2 norm, so that each prediction depends only on the direction of a single token. As shown in Figure 6(c), VMamba reaches 53.4%, 59.5%, and 78.7% for Tiny, Small, and Base, against 32.5%, 42.0%, and 44.6% for MambaOut. VMamba tokens thus carry class information in their directions, whereas MambaOut relies more on token magnitudes. We next ask what gives rise to this difference.

4.4 What determines the roles of magnitude and direction?

We start from the background high-norm pattern, which is the most visible symptom of the difference. A natural candidate for its cause is SS2D, the component that most distinguishes VMamba from MambaOut. However, the pattern persists when SS2D is removed and the model is retrained (Figure 7(b)). When stage 4 is removed instead, the pattern reappears in stage 3, which now feeds the classifier head (Figure 7(c)). The pattern is thus tied neither to the token mixer nor to a particular stage, but to whichever stage feeds the head. This points to the head itself.

Normalization–pooling order. Given the final stage tokens $\{x_i\}_{i=1}^{N}$, the two heads differ in the order of normalization and pooling:

$$z_{\text{VMamba}} = h\Big(\tfrac{1}{N}\textstyle\sum_i \mathrm{LN}(x_i)\Big), \qquad z_{\text{MambaOut}} = h\Big(\mathrm{LN}\big(\tfrac{1}{N}\textstyle\sum_i x_i\big)\Big), \tag{4}$$

where $h(\cdot)$ denotes the classifier. VMamba normalizes each token before pooling, which removes its scale, so every token contributes with a comparable norm and only its direction matters. High norms therefore need not coincide with class evidence. MambaOut pools first, so each token contributes in proportion to its norm, which gives training an incentive to place class evidence in a few high-norm foreground tokens. This matches the tests in Section 4.3.

Table 1: ObjectNet zero-shot evaluation and ImageNet-1k classification.

ObjectNet Zero-shot Top-1 Accuracy (%)				
Scale	Model	224	384	768
Tiny	VMamba	45.4	45.0	**30.6**
	MambaOut	**46.1**	**45.6**	30.5
Small	VMamba	48.7	49.0	**39.2**
	MambaOut	**49.1**	49.0	33.4
Base	VMamba	**49.5**	**49.9**	**40.3**
	MambaOut	49.1	48.7	34.4

ImageNet-1k Top-1 Accuracy (%)				
Dataset	Res.	Setting	VMamba-T	MambaOut-T
ImageNet-1k	224	Pretrained	81.3	**81.5**
ImageNet-1k	768	Fine-tuning	**83.4**	83.3

Head swap. To test this account, we retrain VMamba with only the head order swapped to GAP→LN. High-norm tokens then move to the foreground (Figure 7(d)), showing that the normalization–pooling order is sufficient to change where high-norm tokens appear. In summary, the token mixer sets how far information spreads, and the head sets which component of a token carries class evidence. This leads to two expectations. First, since VMamba spreads class evidence over the directions of many tokens and keeps some global channels at higher resolution, its aggregation should remain stable as the number of tokens grows. Second, since its final stage is sensitive to supervision (Section 4.1) and does not rely on a few high-norm tokens, dense supervision should reorganize it more readily. Section 5 tests both expectations.

5 High-Resolution and Dense Prediction

We now test whether the differences identified in Section 4 translate into performance differences once the task departs from standard 224×224 ImageNet classification (Deng et al., 2009). Section 5.1 increases the number of tokens while keeping the task fixed, and Section 5.2 changes the supervision from image-level to dense labels.

5.1 High-resolution classification

This section tests the first expectation: whether VMamba aggregates evidence more stably as the number of tokens grows.

Resolution transfer. To isolate the effect of sequence length, we apply the 224×224-pretrained weights directly at 384×384 and 768×768 without fine-tuning. This increases the number of tokens by about 3 and 11 times while keeping the weights fixed. We evaluate on ObjectNet (Barbu et al., 2019), a zero-shot benchmark with controlled variation in viewpoint, rotation, and background, following Taori et al. (2020); Radford et al. (2021), at three model scales.

Table 1 (top) shows the results. At 224×224, MambaOut is more accurate for Tiny and Small, and VMamba for Base. Both models degrade at 768×768, but VMamba degrades less at every scale, losing 14.8, 9.5, and 9.2 points for Tiny, Small, and Base, compared with 15.6, 15.7, and 14.7 for MambaOut. The gap is small for Tiny but reaches about 6 points for Small and Base. This trend is consistent with the global channels that remain at 768×768 (Section 4.2), although this experiment alone does not separate the contribution of the token mixer from that of the head.

High-resolution fine-tuning. We also fine-tune the 224×224-pretrained models on ImageNet at 768×768 for 30 epochs. Table 1 (bottom) shows that MambaOut leads at 224×224 (81.5% vs. 81.3%), whereas after fine-tuning the two models are on par, with VMamba slightly ahead (83.4% vs. 83.3%). The margin is small, but the ranking shifts in the same direction as under resolution transfer.

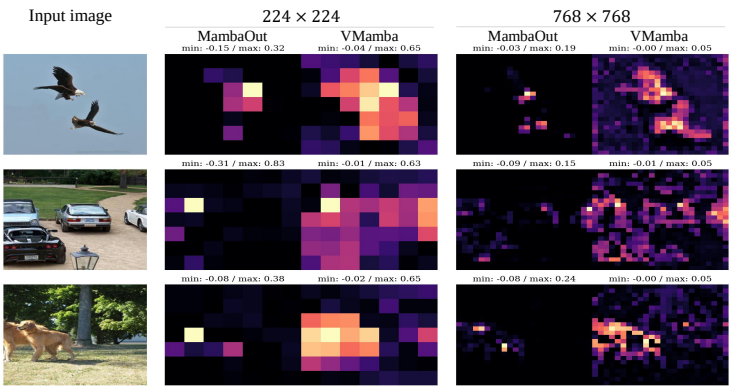

Figure 8: Token replacement attribution at 224×224 and 768×768. Each final block token is replaced with the image-wise mean token, and the resulting drop in the target class logit is shown (min/max above each map).

Table 2: Semantic segmentation (mIoU). MambaOut is stronger with a frozen backbone, whereas VMamba is stronger after full fine-tuning and is less affected by removing token magnitudes.

Dataset	Res.	Setting	VMamba-T	MambaOut-T
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

Token replacement attribution. To examine how the final stage supports the prediction at each resolution, we measure the contribution of individual tokens by perturbation, following perturbation-based explanation methods (Zeiler & Fergus, 2014; Fong & Vedaldi, 2017). Given the final stage feature map $X = \{x_i\}_{i=1}^{N}$, we replace each token x_i with the image-wise mean $\bar{x} = \frac{1}{N}\sum_{j=1}^{N} x_j$ and define the token replacement score as

$$e_i = f_c(X) - f_c\big(X^{(i \leftarrow \bar{x})}\big), \tag{5}$$

where $f_c(\cdot)$ is the target class logit. A larger e_i indicates stronger positive support for the prediction. Appendix E provides implementation details and shows that replacing each token with a random vector of the same norm yields consistent patterns.

Figure 8 shows that, at both resolutions, VMamba's support spans broader object regions, whereas MambaOut concentrates it in fewer locations. This matches the representation analysis: VMamba has channels that mix information beyond a token's neighborhood, and its class evidence lies in the directions of many tokens rather than in a few high-norm ones. At 224×224, a few strong object tokens may suffice for image-level prediction, consistent with MambaOut's advantage at that resolution. At 768×768, evidence held in a few tokens may become more sensitive to spatial sampling, whereas broader support may provide a more stable basis for aggregation.

5.2 Semantic Segmentation

This section tests the second expectation: whether dense supervision reorganizes VMamba's representation more readily. Segmentation supervises every spatial location, so background regions become part of the prediction target, and the misalignment between VMamba's high-norm tokens and its class attribution matters less. The classification head is discarded and the decoder reads multi-stage features directly, but fine-tuning starts from the representation that the head shaped during pretraining.

Frozen backbone vs. Full fine-tuning. With a frozen backbone, MambaOut outperforms VMamba on ADE20K (Zhou et al., 2017) (40.7 vs. 40.2 mIoU, Table 2), consistent with its magnitude-aligned foreground cues, which a new decoder can exploit directly. Under full fine-tuning, the ranking reverses

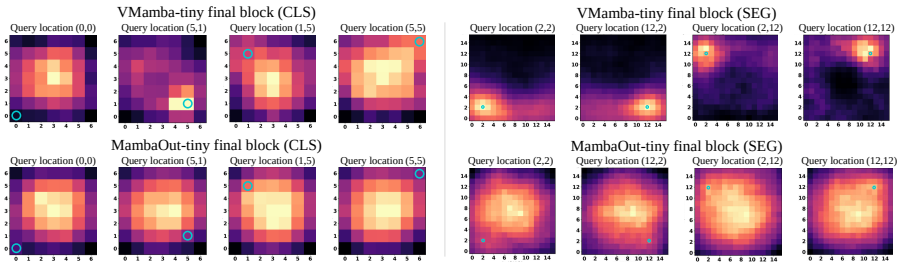

Figure 9: CKA-based spatial localization of classification-pretrained and segmentation fine-tuned models (green circle: query location).

on ADE20K (47.9 vs. 47.1), with a larger margin on Pascal VOC 2012 (Everingham et al., 2010) (81.6 vs. 79.2). This reversal coincides with the substantial change in VMamba's final stage under dense supervision (Figure 1(c)). We next examine which properties of the fine-tuned representation support this advantage.

Unit-token decoder test. To test whether dense prediction can rely on direction alone, we replace each stage feature passed to the decoder with its ℓ_2-normalized direction while preserving the overall feature scale. VMamba drops only from 47.9 to 47.5 mIoU, whereas MambaOut falls from 47.1 to 41.5 (Table 2). VMamba's dense prediction thus remains largely intact without raw token magnitudes. This does not imply that VMamba ignores magnitude, but its token directions are more sufficient for dense prediction than those of MambaOut.

Spatial localization by CKA. We further examine whether dense supervision changes the spatial structure of the representations, using CKA-based spatial localization (Raghu et al., 2021) (Appendix B). Both classification-pretrained models localize weakly (Figure 9). After segmentation fine-tuning, VMamba becomes markedly more spatially selective, whereas MambaOut changes little, mirroring the CKA shift in Figure 1(c). These results suggest two reasons for VMamba's advantage under fine-tuning. Dense supervision reshapes its representation into a spatially selective one, and its token directions retain the information the decoder needs. MambaOut is easier to exploit with a frozen backbone, where its foreground magnitude cues are directly available, but benefits less once the backbone is updated.

6 Discussion

Starting from the observation that VMamba's final stage diverges from MambaOut, we examined two components with separable roles. The token mixer combines local and global channels within a block, and some global channels remain global at resolutions well beyond those seen during training. The classifier head normalizes tokens before pooling, so class evidence is carried by token directions rather than by a few high-norm tokens. Removing SS2D leaves the background high-norm pattern intact, whereas swapping only the head order moves high-norm tokens to the foreground. These properties align with the downstream results. Magnitude-concentrated evidence suffices at standard resolution, whereas direction-based, spatially broad evidence is associated with more robust resolution transfer, comparable or slightly better high-resolution classification, and stronger segmentation after full fine-tuning. As the token count grows, each position can still draw on context both near and far, and evidence spread across many token directions may remain easier to aggregate than evidence held in a few tokens. The question for high-resolution backbones is therefore not only whether they use an SSM, but how far they mix information and where they store class evidence. Token mixing range, token magnitude and direction, and the normalization–pooling order are useful design axes, and angular or token-wise contrastive objectives are promising for dense and high-resolution tasks.

Limitations. The links we draw between memory length, token encoding, and high-resolution performance are correlational. The head swap changes where high-norm tokens appear, but its downstream effect and the causal role of global channels remain to be tested.

AI DISCLOSURE

We used generative AI tools to polish the English and correct grammatical errors. We did not use generative AI tools to develop conceptual frameworks, formulate mathematical claims or hypotheses, design or provide feedback on methodology and experiments, implement methods, or interpret results. Generating synthetic data, writing proofs, cleaning datasets, and qualitative data analysis are not applicable to this work. We reviewed all AI-assisted text and take responsibility for the final content of this work.

REFERENCES

Ameen Ali Ali, Itamar Zimerman, and Lior Wolf. The hidden attention of mamba models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1516–1534, 2025.

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.

Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. Decimamba: Exploring the length extrapolation potential of mamba. In *International Conference on Learning Representations*, volume 2025, pp. 101148–101170, 2025.

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.

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

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

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

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.

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.

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.

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.

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.

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.

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.

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

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

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

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

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.

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.

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

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

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

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

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

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

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.

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.

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.

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

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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

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.

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.

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

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.

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

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.

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.

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.

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)*, pp. 418–434, 2018.

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.

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.

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.

Zhifan Ye, Kejing Xia, Yonggan Fu, Xin Dong, Jihoon Hong, Xiangchi Yuan, Shizhe Diao, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. Longmamba: Enhancing mamba's long-context capabilities via training-free receptive field enlargement. In *International Conference on Learning Representations*, volume 2025, pp. 68486–68504, 2025.

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.

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.

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.

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

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.

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.

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.

Additional details and results for each section are provided below.

A Experimental Setup

All experiments use the ImageNet-1k (Deng et al., 2009) pretrained weights of VMamba-T and MambaOut-T and are conducted on four NVIDIA RTX PRO 6000 GPUs. Unless otherwise stated, all figures and analyses are produced on the ImageNet-1k validation set. For high-resolution classification, we fine-tune the 224×224-pretrained models on ImageNet-1k at 768×768 for 30 epochs, reducing the original learning rate by a factor of 10 for stable training. For semantic segmentation, we initialize from the ImageNet-1k pretrained backbones and fine-tune UperNet (Xiao et al., 2018) on ADE20K (Zhou et al., 2017) for 160K iterations with a batch size of 16 and the AdamW optimizer (Loshchilov & Hutter, 2019), following the standard protocol of VMamba (Liu et al., 2024) and MambaOut (Yu & Wang, 2025). On Pascal VOC 2012 (Everingham et al., 2010), we train with a batch size of 16 for 20K iterations.

B CKA Implementation Details

In Sections 4.1 and 5.2, we measure representational similarity with centered kernel alignment (CKA) (Cortes et al., 2012; Kornblith et al., 2019), which allows representations to be compared across layers and models. Let $X \in \mathbb{R}^{m \times d_1}$ and $Y \in \mathbb{R}^{m \times d_2}$ denote the activations of two layers on the same m samples, with linear Gram matrices $K = XX^\top$ and $L = YY^\top$. Using the centering matrix $H = I_m - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$, we compute the centered Gram matrices $K_c = HKH$ and $L_c = HLH$. CKA is then defined as

$$\mathrm{CKA}(K, L) = \frac{\mathrm{HSIC}(K_c, L_c)}{\sqrt{\mathrm{HSIC}(K_c, K_c)\,\mathrm{HSIC}(L_c, L_c)}}, \tag{6}$$

where the Hilbert–Schmidt independence criterion (HSIC) (Gretton et al., 2007; Kornblith et al., 2019) is

$$\mathrm{HSIC}(K_c, L_c) = \frac{1}{(m-1)^2}\mathrm{tr}(K_c L_c). \tag{7}$$

This measure is invariant to orthogonal transformations and isotropic scaling of the representations. We use the minibatch estimator of Nguyen et al. (2021), sampling 2,560 validation images per run with a batch size of 128, and report the average over 10 runs. This configuration yields results comparable to the larger-sample settings of prior work (Raghu et al., 2021).

Block-level comparison. Following Raghu et al. (2021), we compute CKA between all pairs of blocks, using block outputs extracted with forward hooks and reshaped to $\mathbb{R}^{m \times d}$. The same procedure gives the within-model comparison between classification-pretrained and segmentation fine-tuned backbones.

Spatial localization. Also following Raghu et al. (2021), we measure how strongly each token represents its own input location. For a query token, we compute CKA between its representation and the patch embeddings at every spatial position, which gives one similarity map per query. A peak at the query location indicates a spatially localized representation, whereas a flat map indicates a globally distributed one.

C Measuring the Reach of the Token Mixer

This section describes the two measurements used in Section 4.2, namely the effective receptive field of each token mixer and the memory length of each selective scan.

C.1 Effective receptive fields

We compute the gradient-based effective receptive field of Luo et al. (2016), which VMamba (Liu et al., 2024) also uses to compare vision backbones. It is the absolute gradient of the centre token with respect to the input, summed over channels and averaged over 200 validation images. Unlike

the standard protocol, which uses the network output, we take the gradient at the output of the token-mixing operation so that the two models are compared at the same point. This is the SS2D scan output before `out_norm` for VMamba-T and the gated product $\mathrm{act}(g) \odot [\, i,\ \mathrm{conv}(c)\,]$ for MambaOut-T.

C.2 Memory length

SS2D unfolds an $H \times W$ feature map into four sequences of length $L = H \times W$, scans each with S6 (Gu & Dao, 2024), and sums the four outputs. For direction k and channel c, the scan follows the S6 recurrence as implemented in VMamba,

$$h_t = \exp(\Delta_t a)\, h_{t-1} + \Delta_t B_t\, u_t, \qquad y_t = C_t\, h_t + D\, u_t, \tag{8}$$

where $\Delta_t > 0$ is the input-dependent step size and $a < 0$ is a learnable parameter. The VMamba models we analyze have a state dimension of $N = 1$, so all terms are scalars. Since a and Δ_t are specific to each (k, c) pair, we treat each pair as one scan, which gives $4D$ scans per block for an SS2D width D. As in prior analyses of Mamba (Ali et al., 2025; Ye et al., 2025), unrolling Eq. equation 8 shows that the fraction of a token's contribution at position s that survives until position t is

$$\exp\Big(a \sum_{r=s+1}^{t} \Delta_r\Big), \tag{9}$$

which depends on the step sizes accumulated between the two positions (Ben-Kish et al., 2025). To summarize each scan by a single number, we replace every Δ_r with its mean $\bar{\Delta}$, so that the survival fraction depends only on the distance $d = t - s$,

$$\exp(-|a|\, \bar{\Delta}\, d) = \exp\Big(-\frac{d}{\tau}\Big), \qquad \tau = \frac{1}{|a|\, \bar{\Delta}}. \tag{10}$$

Since $|a|\bar{\Delta}$ determines the per-token decay rate, its inverse τ is the characteristic number of tokens a scan can remember, and after τ tokens a contribution has decayed to $e^{-1} \approx 37\%$. We read a from the learned parameters and average Δ_t over all tokens of 5,000 validation images.

C.3 Global and local scans

Since $L = H \times W$ is the length of the whole flattened feature map, a scan with $\tau > L$ can carry information across the entire image. We therefore call a scan *global* if $\tau > L$ and *local* otherwise. This follows the global/local classification of LongMamba (Ye et al., 2025), which regards a channel of a language model as global when its cumulative decay over the training sequence exceeds a threshold tuned on downstream tasks. In our notation this cumulative decay is $\exp(-L/\tau)$, and $\tau > L$ corresponds to a threshold of e^{-1}, the value that defines τ. We further adapt the criterion to vision in three ways. We use the scan length of each stage as L instead of the training length, we need no averaging over the state dimension since $N = 1$, and we regard a channel as global if any of its four scans is global, since SS2D sums the four directional outputs per channel.

At 768×768, the learned a is unchanged, but Δ_t depends on the input. We therefore re-measure $\bar{\Delta}$ with the same protocol and compare τ with the new L, which grows from 196 to 2,304 tokens in stage 3. No fine-tuning is performed.

C.4 Channel-wise attention maps

Following Ali et al. (2025), we unroll Eq. equation 8 into a weighted sum of earlier inputs,

$$y_t = \sum_{s \le t} \alpha_{t,s}\, u_s + D\, u_t, \qquad \alpha_{t,s} = C_t \exp\Big(a \sum_{r=s+1}^{t} \Delta_r\Big) \Delta_s B_s. \tag{11}$$

The weight of token s in output t is thus the product of how strongly t reads the state, how much of s survives until t, and how strongly s is written. Following Ben-Kish et al. (2025), we normalize $|\alpha_{t,s}|$ per row, which cancels C_t,

$$\tilde{\alpha}_{t,s} = \frac{\exp\Big(a \sum_{r=s+1}^{t} \Delta_r\Big) \Delta_s |B_s|}{\sum_{s' \le t} \exp\Big(a \sum_{r=s'+1}^{t} \Delta_r\Big) \Delta_{s'} |B_{s'}|}, \qquad s \le t. \tag{12}$$

As in Ben-Kish et al. (2025) and Ye et al. (2025), we display the maps on a log scale, clipping $\log_{10} \tilde{\alpha}$ to $[-4, 0]$ and masking entries with $s > t$.

In our setting, each map corresponds to one direction–channel scan of SS2D over the flattened feature map, and Figure 3 (right) shows local and global examples selected by their memory length. For a global scan, information written at an early position remains in the state over much of the sequence. Thus, tokens that are strongly written into the state can contribute to many later outputs, producing vertical stripes in the attention map.

D Evaluation Details

D.1 High-norm test

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

D.2 Linear probing

In Section 4.3, we evaluate the selected tokens with the regularized least-squares probe of Dosovitskiy et al. (2021); Raghu et al. (2021). Given features $X \in \mathbb{R}^{n \times d}$ and targets $Y \in \{-1, +1\}^{n \times C}$ for $C = 1{,}000$ classes, the probe weights are obtained in closed form as

$$W^* = \arg\min_W \ \|XW - Y\|_F^2 + \lambda \|W\|_F^2 = \big(X^\top X + \lambda I\big)^{-1} X^\top Y, \tag{13}$$

without updating the backbone. Tokens are selected by norm as above, which gives $\lceil Tk/100 \rceil$ tokens per image. The probe is trained on 10 images per class sampled uniformly from the ImageNet-1k training set with $\lambda = 0.1$ and evaluated on the full validation set by the average per-token accuracy.

For the unit-vector test in Section 4.3, the same probe is trained and evaluated on all T tokens after normalizing each to unit ℓ_2 norm, for the Tiny, Small, and Base models. We use a probe rather than the pretrained head because both heads apply LayerNorm to a single token, and LayerNorm is invariant to positive scaling, so normalizing a token before the head would leave its prediction unchanged.

D.3 Unit-token decoder test

In Section 5.2, each token x_i of every stage feature map passed to the UperNet decoder (Xiao et al., 2018) is replaced by

$$\tilde{x}_i = \sqrt{C}\, \frac{x_i}{\|x_i\|_2}, \tag{14}$$

where the factor $\sqrt{C}$ keeps the feature scale comparable to that of standard-normalized inputs. The substitution is applied to all four stage outputs at once and only at inference time, with the fine-tuned backbone and decoder weights unchanged.

E Token Replacement Attribution

In Section 5.1, we measure the contribution of each final block token by replacement. Let $X \in \mathbb{R}^{T \times d}$ be the token matrix at the output of the final block and $f_c(X)$ the logit of the target class c, defined as the top-1 prediction of each model. For each position i, we replace the token with the image-wise mean $\bar{x} = \frac{1}{T} \sum_j x_j$ and record

$$e_i = f_c(X) - f_c\big(X^{i \leftarrow \bar{x}}\big), \tag{15}$$

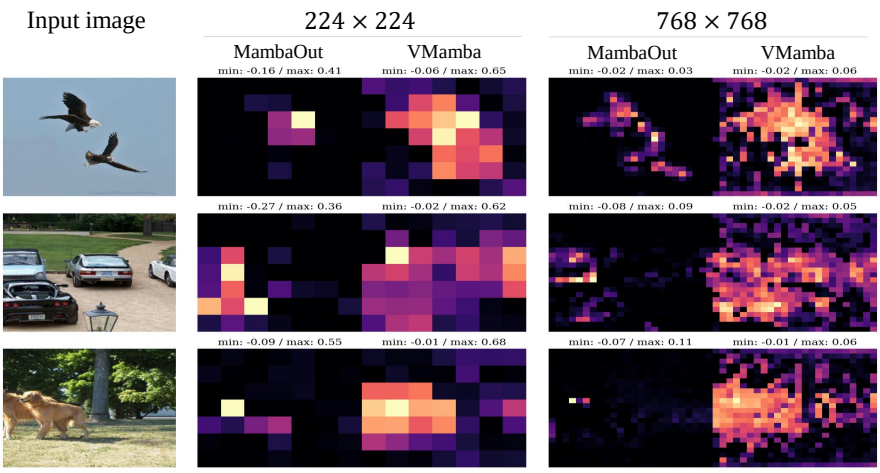

Figure 10: Token replacement attribution with norm-preserving random replacement at 224×224 and 768×768. Each final block token is replaced with a random vector of the same ℓ_2 norm, and the resulting drop in the target class logit is shown (min/max above each map).

so that a larger e_i indicates stronger positive support from token i. We evaluate at 224×224 with the pretrained models and at 768×768 with the high-resolution fine-tuned models.

Norm-preserving random replacement. The mean token differs from x_i in both norm and direction. As a control, we instead replace x_i with a random vector of the same ℓ_2 norm, which changes only its direction. The resulting attribution maps in Figure 10 show the same qualitative difference between the two models.

F Results at Larger Model Scales

Most analyses in the main text use the Tiny models. We repeat them for the Small and Base models and observe the same trends at every scale. VMamba's final stage diverges from MambaOut and from its own preceding blocks (Figure 11). Its high-norm tokens remain in the background while Grad-CAM stays on the object, whereas MambaOut's high-norm tokens remain aligned with Grad-CAM (Figure 12). Global channels appear across blocks at both scales (Figure 13), and local and global channels coexist within a block with the same attention patterns (Figure 14). In the high-norm test, MambaOut is most accurate with the fewest tokens, whereas VMamba's accuracy decreases less or increases as more tokens are included (Figure 15).

G Additional Qualitative Results

Figures 16, 17, and 18 extend Figures 4, 7, and 8 of the main text with further examples.

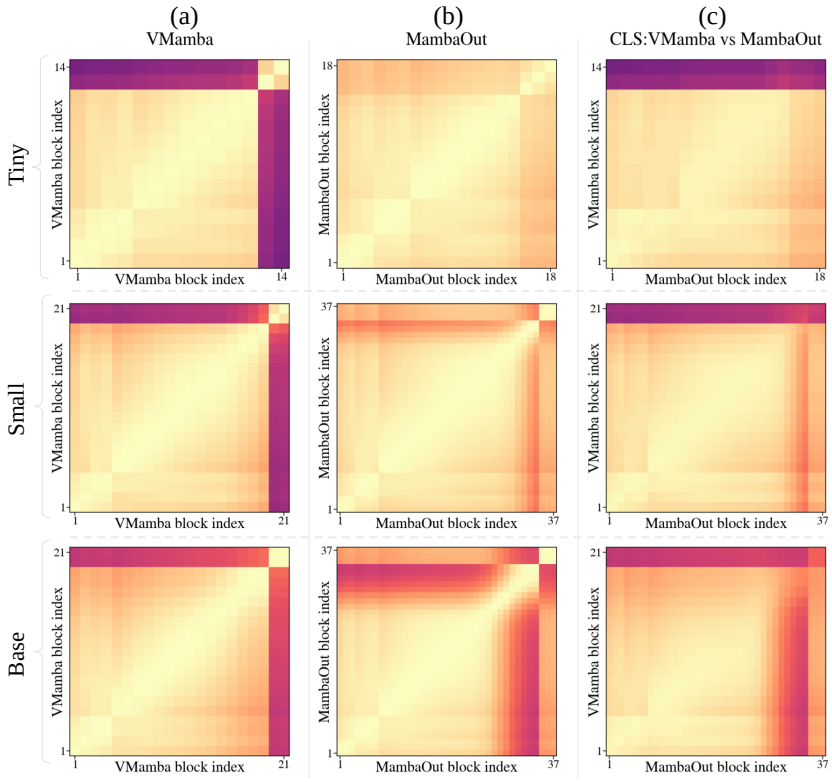

Figure 11: CKA similarity of Figure 1 for the Tiny, Small, and Base models.

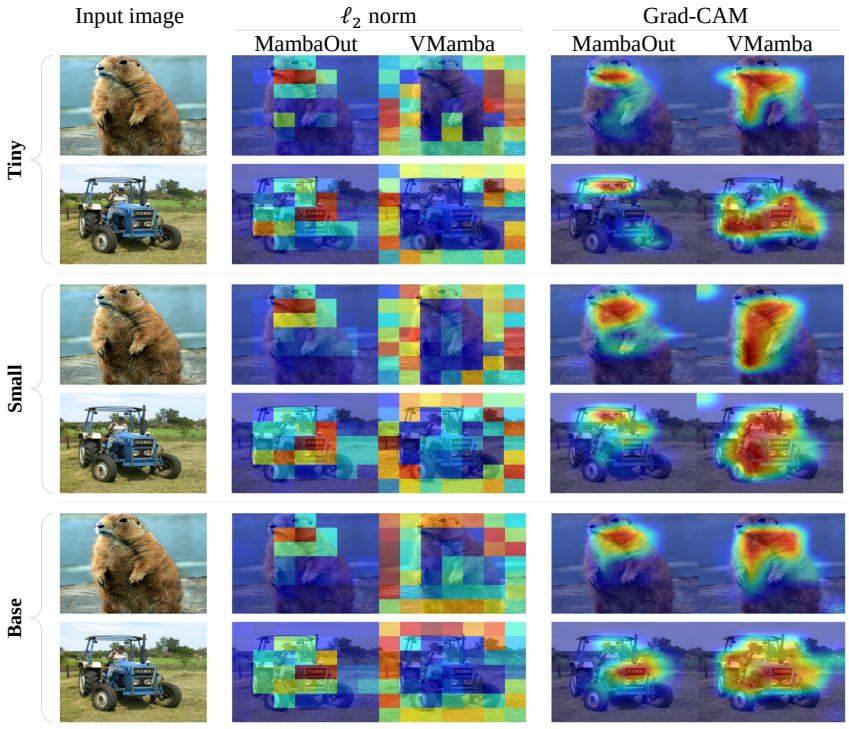

Figure 12: Token ℓ_2 norm and Grad-CAM maps of Figure 4 for the Tiny, Small, and Base models.

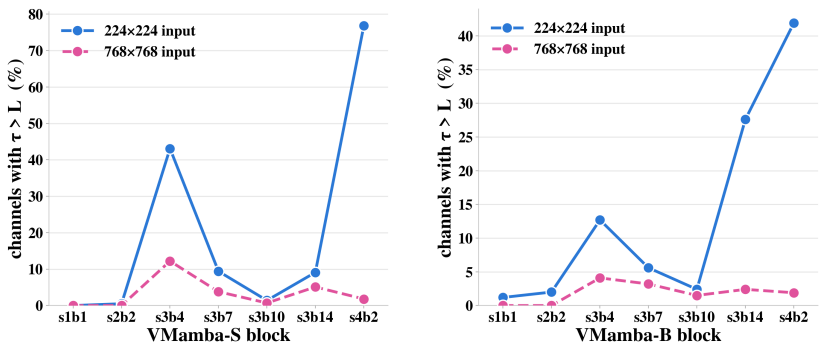

Figure 13: Fraction of global channels per block, as in Figure 3 (left), for the Small and Base models.

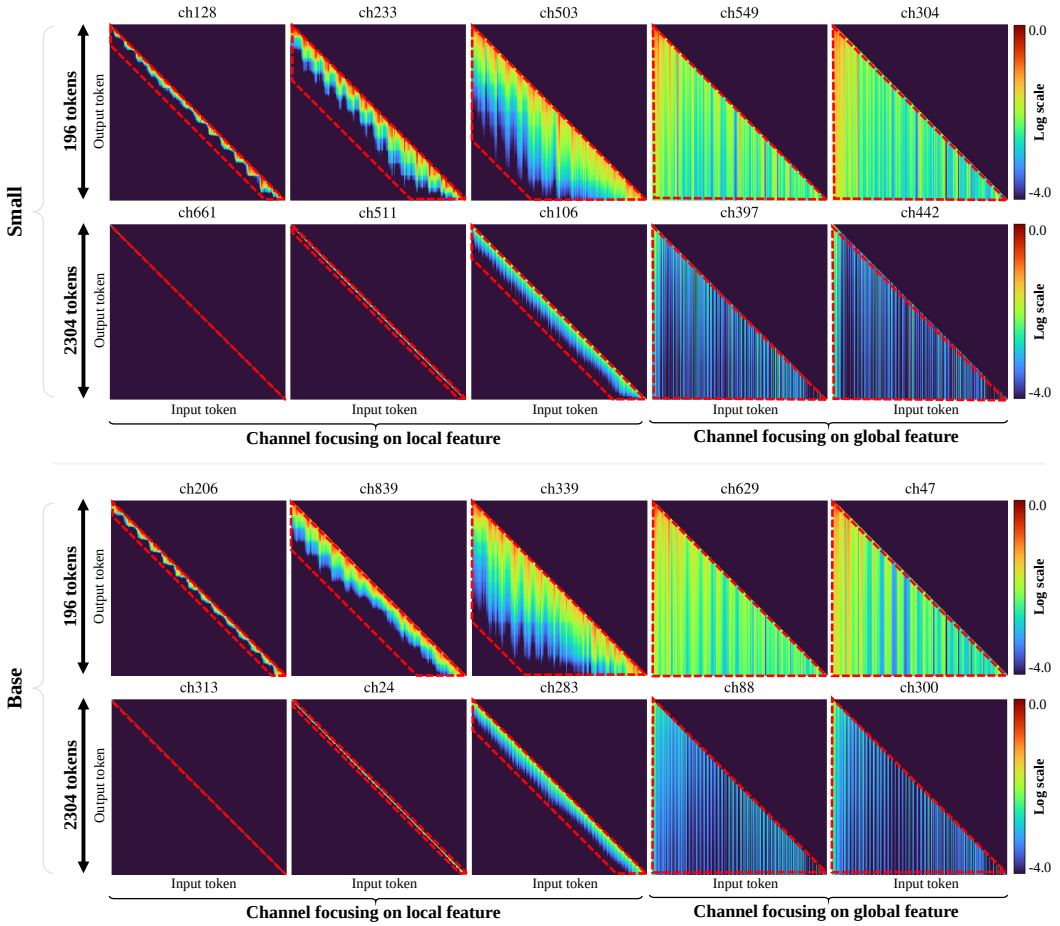

Figure 14: Channel-wise attention maps, as in Figure 3 (right), for the Small and Base models.

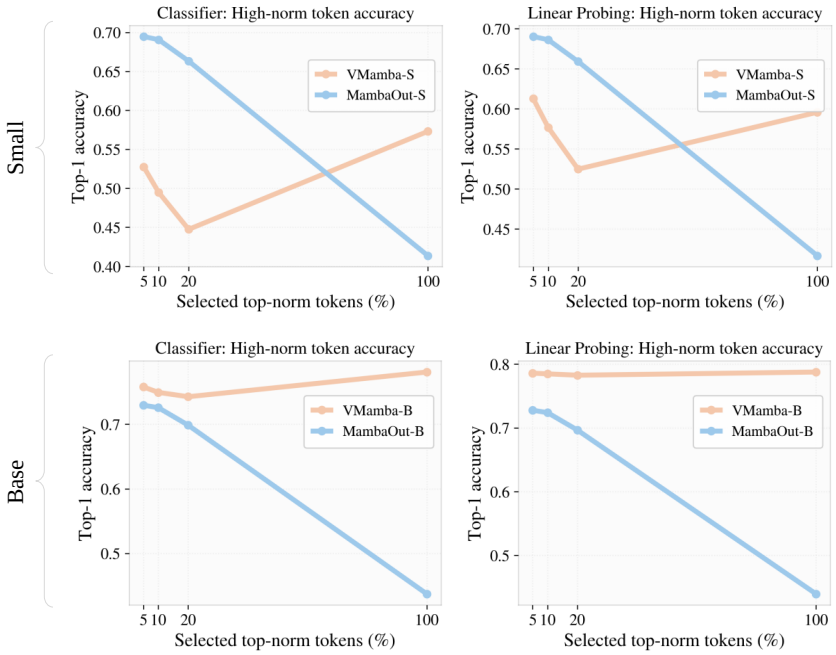

Figure 15: High-norm test and high-norm linear probing of Figure 6(a,b) for the Small and Base models.

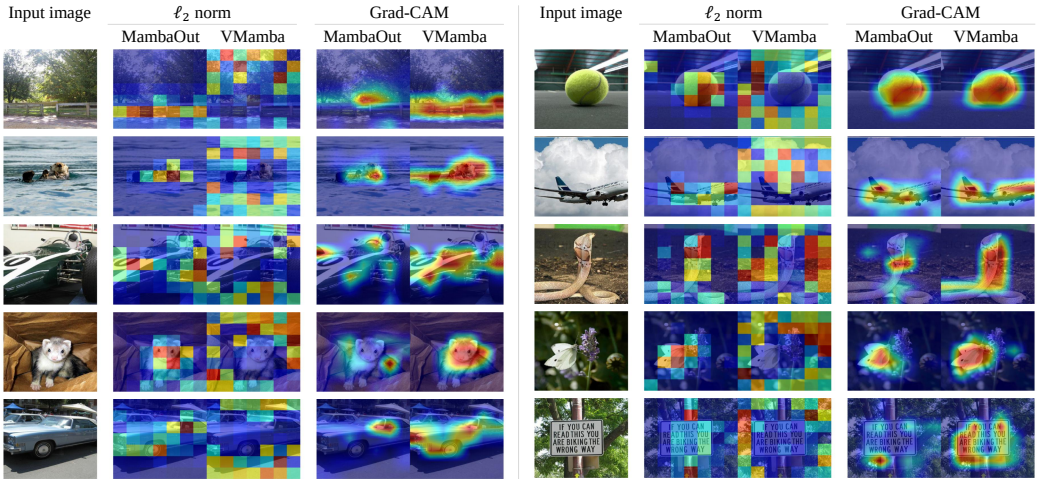

Figure 16: Token ℓ_2 norm and Grad-CAM maps of the final block outputs of VMamba and MambaOut.

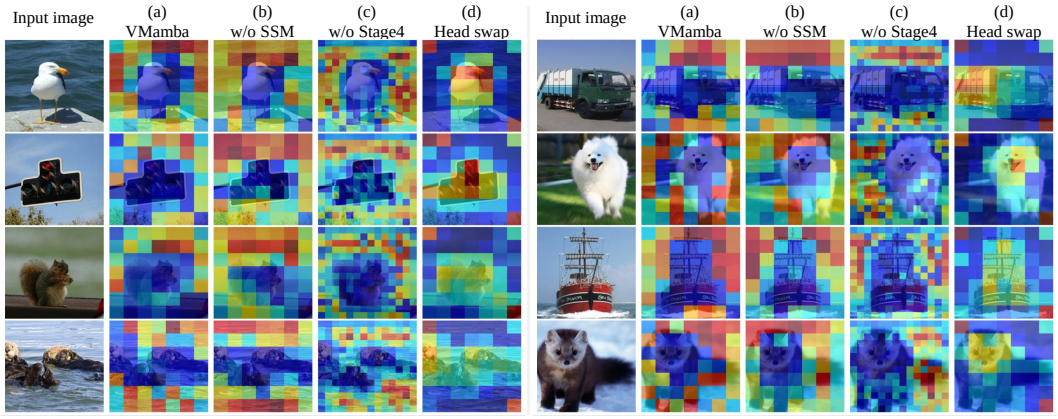

Figure 17: Token ℓ_2 norm maps of the final block outputs for the original VMamba and the three retrained variants of Figure 7.

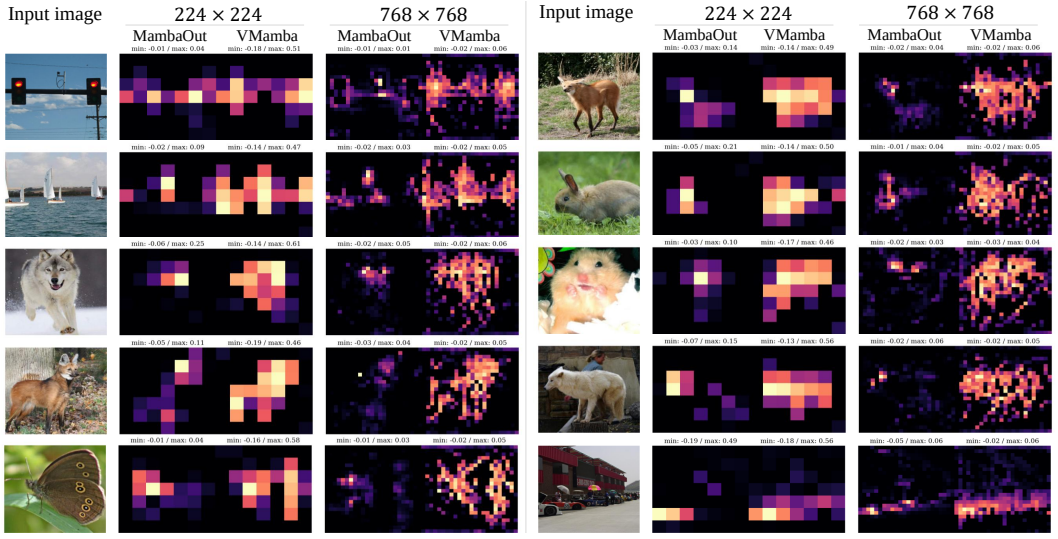

Figure 18: Token replacement attribution at 224×224 and 768×768 for VMamba and MambaOut.