Appendix

G. Ablation Study

To determine the optimal window size for MambaVision models, we study its impact on the performance of MambaVision-T in different tasks such as image classification, object detection and instance segmentation. Given Q, K, V as the query, key and value tensors respectively, self-attention is computed according to

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^{\intercal}}{\sqrt{d_h}})V. \quad (9)$$

d_h represents the number of attention heads. If the input size is larger than the window size, the attention is computed in the local windows. Specifically, we study two different architectures with window sizes of 7 and 14 in their stage 3 of the model. We also measure image throughput for the task of image classification with a batch size of 128. As presented in Table S.1, our analysis reveals that increasing the window size to 14 offers a favorable trade-off between performance and computational cost. While maintaining nearly identical throughput (6298 img/s vs. 6318 img/s), the larger window size achieves consistent improvements across all vision benchmarks: ImageNet top-1 accuracy increases to 82.3%, COCO mask AP improves to 41.8%. These gains, though modest, come with minimal computational overhead on modern hardware such as the NVIDIA A100 GPU. Based on this empirical evidence, we selected 14 and 7 as our default window sizes, as this combination provides better vision understanding capabilities while preserving the model's efficiency. The negligible 0.3% decrease in throughput is well justified by the improved performance in various vision tasks.

Model	Window Size	Throughput (img/s)	ImageNet top-1	COCO AP^{box}	COCO AP^{mask}
MambaVision-T	7,7	6318	82.2	46.4	41.7
MambaVision-T	14,7	6298	82.3	46.4	41.8

Table S.1 – Ablation study on window size for MambaVision model's performance. Experiments on COCO dataset [19] are performed using Mask-RCNN [13] head and ×1 LR schedule. Throughput is measured for image classification on a single NVIDIA A100 GPU with batch size 128.

H. Architecture Details

In Table S.2, we present the comprehensive architectural specifications of MambaVision variants. The backbone follows a hierarchical design with 4 stages, each employing convolutional down-sampling operations that progressively reduce spatial resolution by a factor of two. A key innovation in our architecture appears in Stages 3 and 4, where we introduce a hybrid design that synergistically combines Mamba-based sequence modeling with self-attention mechanisms. This hybrid approach leverages Mamba's efficient sequence processing capabilities while benefiting from the global context modeling strengths of self-attention layers. Each variant (T, S, B, and L) maintains this fundamental structure while scaling the channel dimensions and layer counts to achieve different complexity-performance trade-offs.

I. Training Details

Image classification experiments are conducted on the ImageNet-1K dataset [4]. All models have been trained for 300 epochs using 32 A100 GPUs, with LAMB optimizer, batch size of 4096, and learning rate of 4e-3. The self-attention formulation in stages 3 and 4 of all MambaVision variants use a window size of 14 and 7, respectively. To evaluate the performance of downstream tasks, we used our pre-trained models as backbones for object detection, instance segmentation, and semantic segmentation tasks using the MS COCO dataset [19] and ADE20K dataset [39], respectively. For all downstream tasks, we used an AdamW optimizer and batch size of 16. Specifically, for object detection and instance segmentation, we used the Cascade Mask-RCNN [13] head with hyperparameters such as ×3 LR schedule. For semantic segmentation, we used a UperNet network [34] segmentation head.

J. Interpretability

To demonstrate the interpretability of MambaVision models, we visualize the attention patterns learned by our model across diverse object categories. Figure S.1 presents a comprehensive analysis of attention mechanisms through paired examples.

Our paired visualization analysis reveals several key insights about MambaVision's visual processing capabilities:

- **Consistent Pattern Recognition**: Each triplet (input-heatmap-overlay) demonstrates how the model maintains consistent attention patterns across different instances of similar object categories.
- **Contextual Understanding**: The paired examples within each row often represent contrasting scenarios (e.g., man-made objects vs. natural subjects), showing the model's adaptability across domains.
- **Fine-grained Detail**: The attention heat maps precisely highlight discriminative features, from the texture of animal fur to the structural elements of vehicles and containers.
- **Robust Localization**: Across all example pairs, the overlaid visualizations demonstrate accurate object boundary detection, regardless of the subject's position or background complexity.

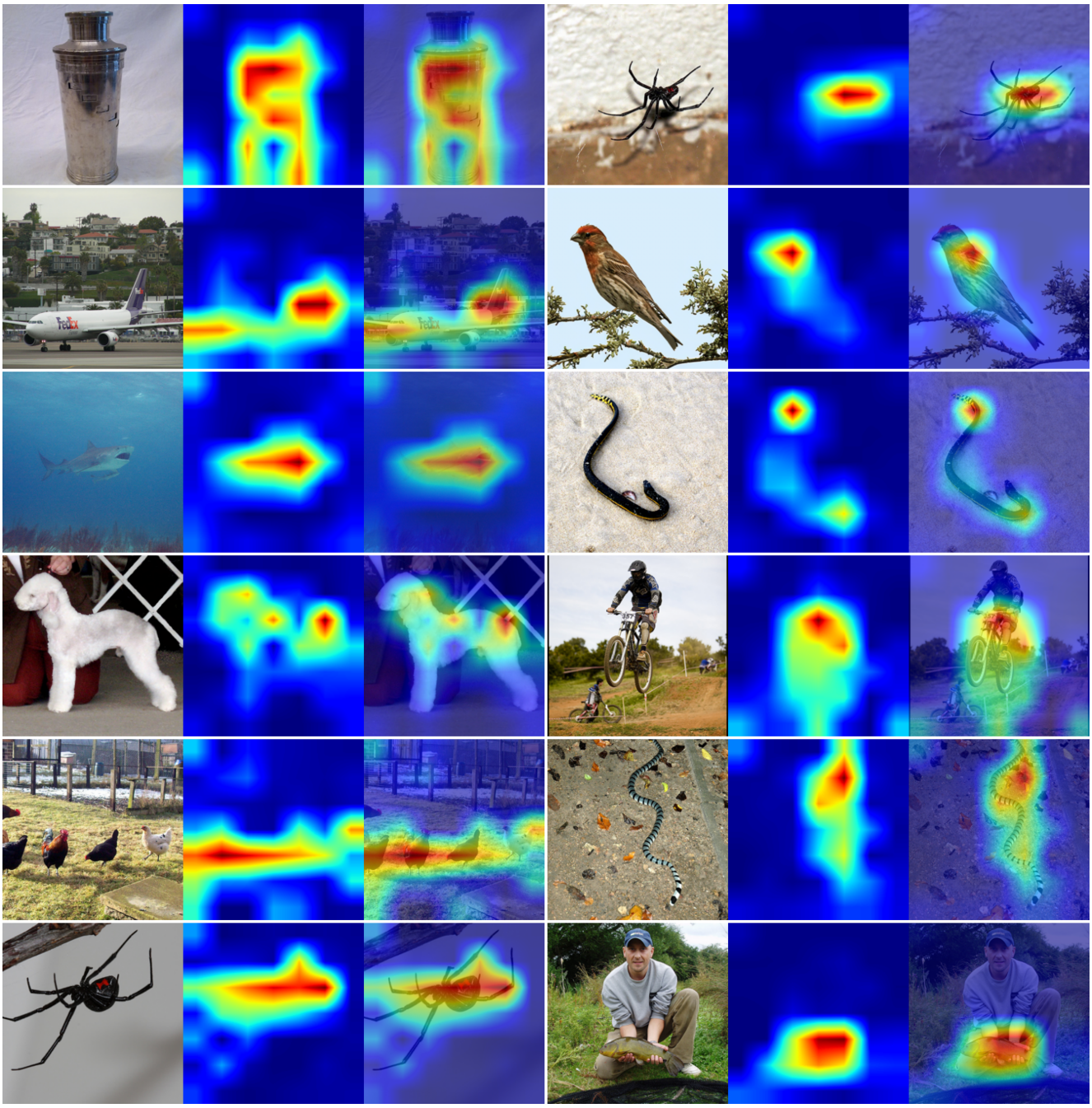

Figure S.1 – Visualization of MambaVision's attention patterns. Each row contains two example cases, with each case showing a triplet of: (left) original input image, (middle) attention heat map, and (right) attention overlay on the input image. The examples showcase diverse scenarios: containers and spiders (row 1), aircraft and birds (row 2), marine life and snakes (row 3), groomed dogs and extreme sports (row 4), poultry and snakes (row 5), and arachnids and outdoor activities (row 6). The attention maps reveal how MambaVision effectively localizes key semantic regions and object boundaries across this wide range of categories.

	Output Size (Downs. Rate)	MambaVision-T	MambaVision-S	MambaVision-B	MambaVision-L
Stem	112×112 (2×)	[Conv-BN-ReLU C:32, S:2] × 1	[Conv-BN-ReLU C:64, S:2] × 1	[Conv-BN-ReLU C:64, S:2] × 1	[Conv-BN-ReLU C:64, S:2] × 1
		[Conv-BN-ReLU C:80] × 1	[Conv-BN-ReLU C:96] × 1	[Conv-BN-ReLU C:128] × 1	[Conv-BN-ReLU C:196] × 1
Stage 1	56×56 (4×)	Conv, C:160, S:2	Conv, C:192, S:2	Conv, C:256, S:2	Conv, C:392, S:2
		[ResBlock C:160] ×1,	[ResBlock C:192] × 3,	[ResBlock C:256] × 3,	[ResBlock C:392] × 3,
Stage 2	28×28 (8×)	Conv, C:320, S:2	Conv, C:384, S:2	Conv, C:512, S:2	Conv, C:768, S:2
		[ResBlock C:320] × 3,	[ResBlock C:384] × 3,	[ResBlock C:512] × 3,	[ResBlock C:768] × 3,
Stage 3	14×14 (16×)	Conv, C:640, S:2	Conv, C:768, S:2	Conv, C:1024, S:2	Conv, C:1568, S:2
		[MV C:640] × 4, [SA C:640, head:8] × 4 ,	[MV C:768] × 4, [SA C:768, head:8] × 3,	[MV C:1024] × 4, [SA C:1024, head:8] × 4,	[MV C:1568] × 4, [SA C:1568, head:16] × 3,
Stage 4	7×7 (32×)	Conv, C:1280, S:2	Conv, C:1536, S:2	Conv, C:2048, S:2	Conv, C:3136, S:2
		[MV C:1280] × 4, [SA C:1280, head:16] × 4,	[MV C:1536] × 4, [SA C:1536, head:16] × 2,	[MV C:2048] × 4, [SA C:2048, head:16] × 2,	[MV C:3136] × 4, [SA C:3136, head:32] × 2,

Table S.2 – Architecture configurations of MambaVision models. SA and MV refer to self-attention and MambaVision mixer blocks respectively. BN denote Batch Normalization.