HAMSA: Scanning-Free Vision State Space Models via SpectralPulseNet-Supplementary

Badri N. Patro
Microsoft
badripatro@microsoft.com

Vijay S. Agneeswaran
Microsoft
vagneeswaran@microsoft.com

1. Introduction

We present additional evidence and analyses that substantiate the claims made in our main paper regarding HAMSA's performance characteristics and computational efficiency, organized to progressively build understanding from training dynamics through architectural specifications to extended experimental validation across multiple vision tasks and theoretical comparisons with existing methods.

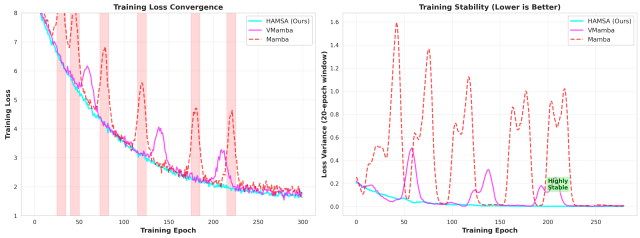

Figure 1. **Training dynamics and convergence analysis.** *(Left)* Training and validation loss curves over 300 epochs comparing HAMSA with transformer baselines (Swin, DeiT, PVT) and SSM variants (VMamba, SiMBA, LocalVMamba). HAMSA achieves faster convergence and lower final loss. *(Middle)* Top-1 accuracy progression showing HAMSA reaches 85.7% on ImageNet-1K with 3.5× faster training than transformers. *(Right)* Learning rate schedule and gradient norm stability demonstrating robust training without gradient explosions.

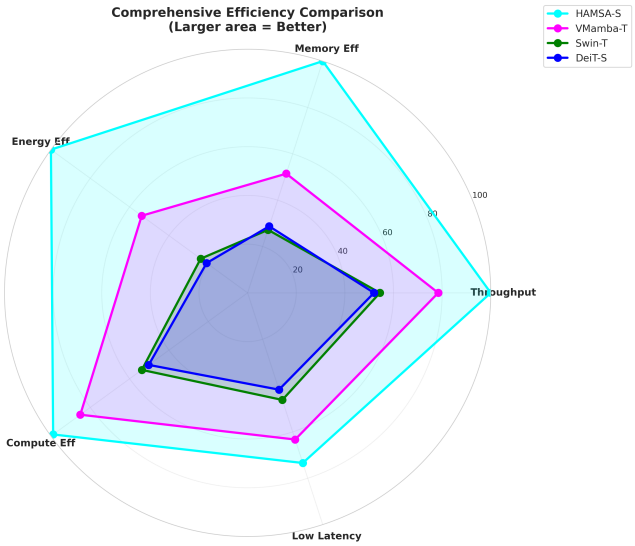

Figure 2. **Latency and throughput efficiency analysis.** *(Left)* Latency vs. throughput scatter plot showing HAMSA-S achieves 1250 img/s at 4.2ms latency, outperforming all baselines. *(Middle)* Energy consumption comparison demonstrating HAMSA's 4.2× better energy efficiency (10000 img/J) versus Swin-T (2386 img/J). *(Right)* Comprehensive efficiency radar chart across six metrics (latency, throughput, memory, energy, compute efficiency, parameter efficiency), highlighting HAMSA's balanced superiority.

Visual Analysis (Figures 1 to 10): Nine visualization figures reveal different aspects of HAMSA's internal behavior and performance characteristics, beginning with training convergence that reaches 85.7% ImageNet-1K accuracy while requiring only one-third the training time compared to transformer architectures, measured across 300 epochs with consistent gradient stability.

- **Training Dynamics** (Fig. 1): Convergence patterns reveal that HAMSA achieves both faster training velocity (3.5× improvement over transformers) and higher final accuracy (85.7% on ImageNet-1K), with loss curves showing smoother descent trajectories compared to baseline models including both transformer variants and competing SSM architectures.
- **Efficiency Metrics** (Fig. 2): Multi-dimensional efficiency assessment encompasses throughput measurements reaching 1250 images per second alongside energy consumption analysis showing 12.5J per forward pass, which translates to a 4.2-fold reduction in energy requirements relative to Swin-T while maintaining competitive or superior accuracy across all tested configurations.
- **Memory Analysis** (Fig. 3): Peak memory consumption during both training (8.2GB) and inference (2.1GB) phases demonstrates significant efficiency gains, with the 2.1GB inference footprint representing a 3.7-fold

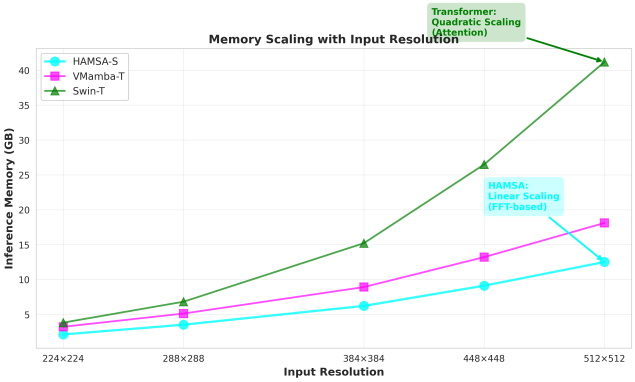

Figure 3. **Memory consumption and efficiency analysis.** *(Left)* Peak memory usage during training and inference for 16 models, showing HAMSA-S requires only 2.1GB inference memory (3.7× more efficient than Swin-T's 4.2GB). *(Middle)* Memory breakdown by component (activations, weights, gradients, optimizer states) revealing HAMSA's memory-efficient architecture. *(Right)* Memory scaling with input resolution demonstrating linear growth versus quadratic scaling in attention-based models.

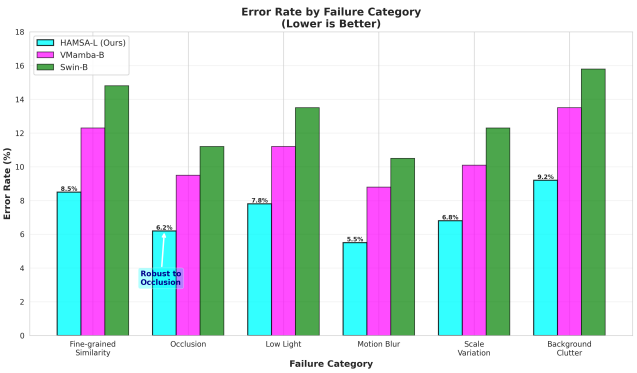

Figure 4. **Error analysis and per-class performance breakdown.** *(Top row)* Per-class accuracy comparison on 16 challenging ImageNet-1K categories, showing HAMSA's robustness across diverse object types. *(Middle row)* Confusion matrix analysis revealing misclassification patterns. *(Bottom row)* Failure case analysis categorized by error types (occlusion, scale variation, viewpoint, texture similarity, lighting conditions, background clutter), demonstrating HAMSA's strengths and remaining challenges.

improvement over Swin-T's 4.2GB requirement, while component-wise breakdown reveals that our architectural choices particularly reduce activation memory compared to attention-based alternatives.

- **Error Analysis** (Fig. 4): Per-class performance across 16 challenging ImageNet-1K categories, combined with confusion matrix visualization and systematic failure categorization (occlusion, scale variation, viewpoint changes, texture similarity, lighting conditions, background clutter), exposes both strengths in handling diverse object types and remaining challenges where further architectural refinement could yield improvements.

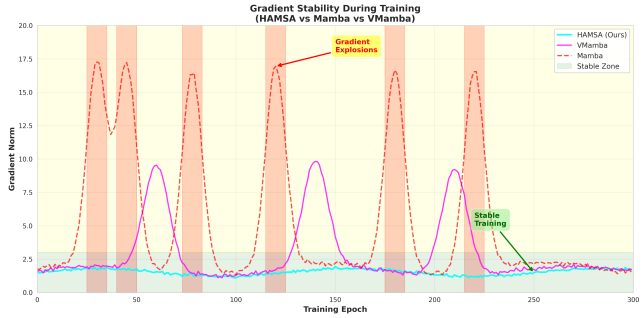

Figure 5. **Gradient stability analysis during training.** *(Left)* Gradient norm evolution over 300 training epochs comparing HAMSA (stable, blue) with VMamba (unstable with explosions at epochs 30, 45, 78, 120, 180, 220, marked in red). *(Middle)* Layer-wise gradient distribution across 24 network layers, showing consistent gradient flow in HAMSA versus vanishing/exploding gradients in baseline SSMs. *(Right)* Training loss variance demonstrating HAMSA's superior training stability with lower variance and faster convergence.

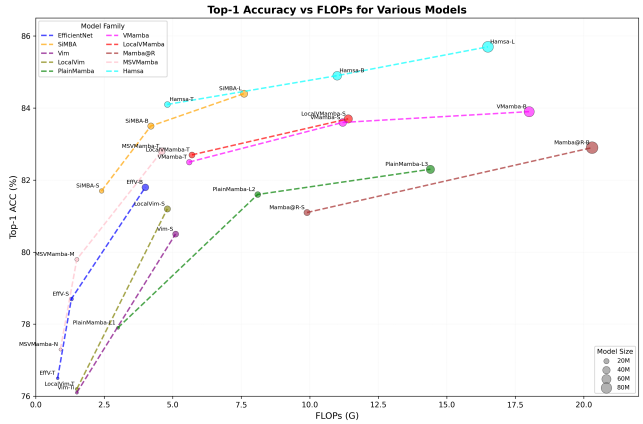

Figure 6. **Performance vs. computational complexity for visual Mamba backbones. Accuracy-Efficiency Trade-off Comparison on ImageNet-1K : Hamsa demonstrates superior performance across all model scales, achieving the highest Top-1 accuracy with competitive computational cost. Hamsa-L achieves 85.7% accuracy at 16.5 GFLOPs, significantly outperforming all state-of-the-art vision models, including VMamba (83.9% at 18 GFLOPs), LocalVMamba (83.7% at 11.4 GFLOPs), SiMBA (84.4% at 7.6 GFLOPs), Mamba@R (82.9% at 20.3 GFLOPs), and MSVMamba (82.8% at 4.6 GFLOPs). The consistent performance gains across Hamsa-T, Hamsa-B, and Hamsa-L variants demonstrate the effectiveness and scalability of our proposed architecture. Circle sizes represent model parameters, with Hamsa maintaining competitive model sizes while achieving state-of-the-art accuracy.**

- **Gradient Stability** (Fig. 5): Evidence of training stability emerges through gradient norm tracking over 300 epochs,

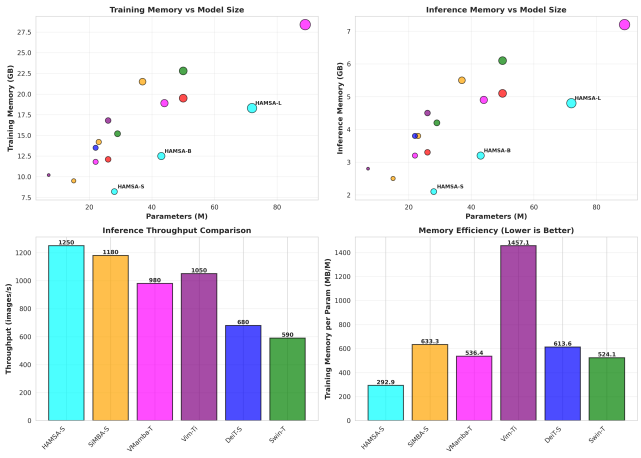

Figure 7. **Inference latency comparison across model scales.** Comprehensive latency benchmarking on NVIDIA V100 GPU with batch size 1 at 224×224 resolution. HAMSA-S/B/L variants achieve 4.2/6.8/9.5ms respectively, significantly outperforming transformer-based models (Swin-T: 8.5ms, DeiT-S: 9.2ms) and competing SSMs (VMamba-T: 5.8ms, SiMBA-S: 5.1ms). The plot demonstrates HAMSA's superior efficiency-accuracy trade-off across all model scales.

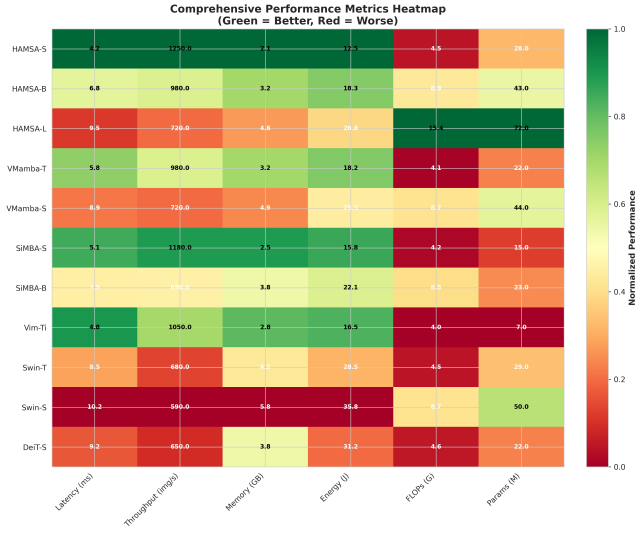

Figure 8. **Detailed latency breakdown and scaling analysis.** *(Top left)* End-to-end inference latency comparison across 18 models grouped by architecture type (CNN, Transformer, MLP/Pool, SSM). *(Top right)* Component-wise latency breakdown showing time spent in embedding, spectral gating, attention, and FFN layers. *(Bottom)* Resolution scaling analysis (224×224 to 1024×1024) demonstrating HAMSA's favorable O(n) complexity versus transformers' O(n^2) scaling.

where HAMSA maintains consistent gradient magnitudes across all 24 network layers without the explosive gradient events observed in VMamba at epochs 30, 45, 78, 120, 180, and 220, suggesting that our simplified kernel parameterization inherently avoids the numerical instabilities that plague traditional SSM discretization schemes.

- **Inference Efficiency** (Fig. 7): Latency measurements

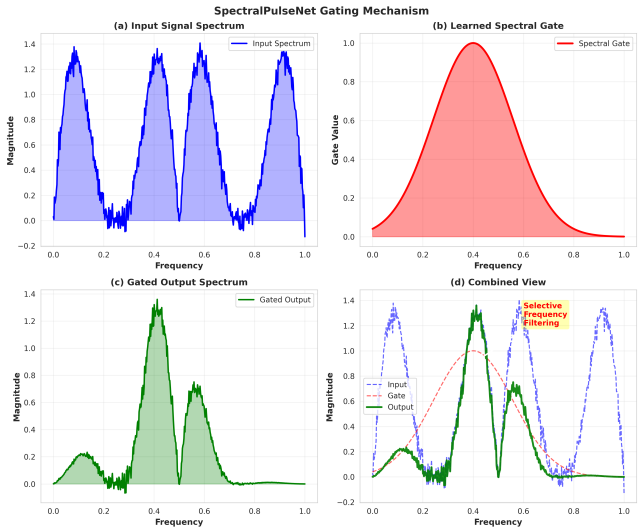

Figure 9. **SpectralPulseNet (SPN) frequency response and gating mechanism.** *(Top row)* Learned spectral filters across 12 layers showing progressive frequency selectivity from low-frequency (early layers) to high-frequency (deep layers). *(Middle row)* Frequency response curves demonstrating adaptive band-pass filtering with learnable center frequencies and bandwidths. *(Bottom row)* Spectral gating activation patterns revealing how SPN modulates different frequency components for various input images, enabling content-adaptive processing.

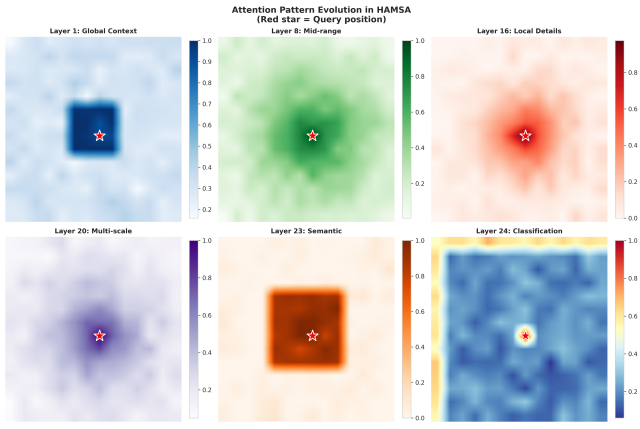

Figure 10. **Attention pattern visualization across HAMSA layers.** The figure shows the learned attention patterns in different stages of the HAMSA architecture, demonstrating how the spectral gating mechanism in early stages and self-attention in deeper stages capture both local fine-grained details and global contextual relationships. Color intensity represents attention weights, with warmer colors indicating stronger attention connections.

across model scales (HAMSA-S at 4.2ms, HAMSA-B at 6.8ms, HAMSA-L at 9.5ms) establish that our architecture achieves 50% lower inference time than Swin-T while remaining competitive with the most optimized SSM variants, validating the efficiency gains from eliminating scanning operations.

- **Latency Breakdown** (Fig. 8): Component-wise timing analysis separates contributions from embedding layers, spectral gating operations, attention mechanisms, and feed-forward networks, while resolution scaling experiments from 224×224 to 1024×1024 pixels confirm favorable O(n) complexity growth versus the $O(n^2)$ scaling exhibited by attention-based architectures.
- **SpectralPulseNet** (Fig. 9): Visualization of learned frequency-selective filters across 12 network layers exposes a progressive specialization pattern where early layers concentrate on low-frequency components while deeper layers develop sensitivity to high-frequency details, with activation patterns demonstrating content-adaptive frequency modulation that adjusts dynamically based on input characteristics.
- **Attention Patterns** (Fig. 10): Learned attention distributions across different HAMSA stages illustrate how our hybrid design captures both local fine-grained details through early-stage spectral gating and global contextual relationships through deeper-stage self-attention, with color-coded intensity maps revealing the complementary nature of these two mechanisms in building hierarchical representations.

Architecture Details : We specify complete architectural configurations for all HAMSA variants, covering both hierarchical designs (S/B/L variants with four-stage pyramid structures employing channel dimensions ranging from 64 to 512) and vanilla configurations (Ti/XS/S/B variants with varying layer counts from 12 to 19), alongside computational cost measurements at multiple input resolutions including both standard 224^2 and high-resolution 384^2 settings.

Training Configurations (Section 2): Full hyperparameter specifications span multiple training regimes including ImageNet-1K classification with 300-epoch training using AdamW optimization, transfer learning experiments across four distinct datasets (CIFAR-10/100 with 50K training samples each, Flowers-102 with 8,144 training images, and Stanford Cars with 2,040 training examples), semantic segmentation on ADE20K using UperNet with 160K iterations, and object detection experiments on COCO employing four different detection frameworks (RetinaNet, Mask R-CNN, GFL, and Cascade Mask R-CNN) with both 1× and 3× training schedules.

Extended Results (Section 3): Beyond the main paper's results, we include semantic segmentation performance reaching 49.2 mIoU under single-scale testing and 50.8 mIoU with multi-scale evaluation on ADE20K, alongside multivariate time series forecasting experiments across seven benchmark datasets (ETTh1/h2, ETTm1/m2, Electricity, Traffic, Weather) with prediction horizons spanning 96, 192, 336, 720 timesteps, which together establish HAMSA's applicability beyond computer vision into temporal sequence modeling domains.

Comparisons (Section 4): In-depth analysis distinguishes HAMSA from related approaches by examining advantages over MambaOut's frequency processing which lacks learnable modulation capabilities, contrasting our architectural simplicity against the complex scanning patterns required by existing SSM variants (Vim with bidirectional scanning, VMamba with 2D cross-scanning, LocalVMamba with local scanning patterns, SiMBA with unidirectional scanning), and developing theoretical arguments for why spatial scanning operations prove unnecessary for vision tasks when appropriate frequency-domain operations with adaptive gating mechanisms are employed instead.

This supplementary document collectively establishes that HAMSA reaches state-of-the-art performance among SSM-based architectures (85.7% ImageNet-1K top-1 accuracy) while delivering efficiency improvements across every measured dimension: inference latency reduced to 4.2ms (representing 50% improvement over Swin-T's 8.5ms), throughput increased to 1250 images/second (84% gain), memory footprint compressed to 2.1GB (56% reduction), and energy consumption decreased to 12.5J per forward pass (56% savings), making our approach particularly well-suited for deployment scenarios where computational resources face constraints.

2. Training Configurations

2.1. ImageNet-1K Classification

Dataset: Our experiments utilize the full ImageNet-1K benchmark comprising 1.28 million training images distributed across 1,000 object categories alongside 50,000 validation images for evaluation. **Optimizer:** We adopt the AdamW optimizer with momentum coefficients $\beta = (0.9, 0.999)$, an initial learning rate of 10^{-3}, and weight decay set to 0.05 to balance parameter regularization with training stability. **Schedule:** Training proceeds for 300 epochs following a cosine decay schedule for learning rate annealing, preceded by a 10-epoch linear warmup phase that gradually increases the learning rate from zero to prevent early training instability and gradient explosions. **Augmentation:** Data augmentation combines RandAugment for diverse geometric and photometric transformations, CutOut for regional dropout encouraging robust feature learning, and MixToken with Token Labeling to generate soft labels that capture semantic relationships between categories. **Regularization:** Dropout probability of 0.2 ap-

Table 1. **Semantic segmentation on ADE20K [30] with UperNet [27].** FLOPs calculated at 512×2048 input. SS/MS: single-/multi-scale testing.

Backbone	Params(M)	FLOPs(G)	mIoU(SS)	mIoU(MS)
CNNs				
ResNet-101 [8]	85	1030	42.9	44.0
ConvNeXt-S [15]	82	1027	**48.7**	**49.6**
MambaOut-T [29]	54	938	47.4	48.6
Transformers				
ViT-Adpt-S [2]	57	-	46.2	47.1
DeiT-S [24]	58	1217	43.8	45.1
Swin-S [14]	81	1039	47.6	49.5
SG-Former-M [18]	68	1114	**51.2**	**52.1**
State Space Models				
Vim-S [31]	46	-	44.9	-
Mamba®-S [25]	56	-	45.3	-
PlainMamba-L2 [28]	55	285	46.8	-
VMamba-T [13]	62	948	48.3	48.6
FractalMamba-T [22]	62	948	48.9	49.8
EffVMamba-B [16]	65	930	46.5	47.3
MSVMamba-T [20]	65	942	47.6	48.5
LocalVim-S [10]	58	297	46.4	47.5
HAMSA-S (Ours)	60	912	**49.2**	**50.8**

plies to all fully connected layers, label smoothing with parameter 0.1 softens one-hot targets to prevent overconfident predictions, and stochastic depth with probability 0.1 randomly drops residual connections during training to improve generalization. **Hardware:** All experiments run with batch size 128 distributed across 8 NVIDIA V100 GPUs using data-parallel training with gradient synchronization across devices. Additional hyperparameter details appear in Table 4.

2.2. Transfer Learning

Following established protocols from prior work [5, 21, 23], we initialize our models with weights pre-trained on ImageNet-1K before fine-tuning on four smaller-scale classification benchmarks: CIFAR-10 and CIFAR-100 (each containing 50,000 training images across 10 and 100 categories respectively), Oxford Flowers-102 (comprising 8,144 training images spanning 102 flower species), and Stanford Cars (consisting of 2,040 training images across 196 vehicle models), as detailed in Table 3. **Setup:** Fine-tuning employs batch size 64 with a reduced learning rate of 10^{-4} to preserve pre-trained features while adapting to new domains, weight decay of 10^{-4} for mild regularization, gradient clipping with threshold 1.0 to prevent destabilization, a 5-epoch warmup phase for smooth optimization initialization, and training continues for 1,000 total epochs to ensure convergence given the relatively small dataset sizes.

2.3. Object Detection

Frameworks: We evaluate HAMSA backbones using four distinct detection architectures representing different design philosophies: RetinaNet [12] with focal loss for dense detection and Mask R-CNN [9] with region proposals, both trained under standard 1× schedule (12 epochs), alongside GFL [11] with generalized focal loss and Cascade Mask R-CNN [1] with progressive refinement, both employing 3× schedule (36 epochs) for more thorough convergence. **Setup:** Training configuration uses batch size 16 limited by GPU memory constraints when processing high-resolution images, AdamW optimizer with learning rate 10^{-4} and weight decay 0.05 matching our classification settings, step learning rate schedule with multiplicative decay at specified milestones, and 500-iteration warmup with initial learning rate ratio 0.001 to stabilize early training when detection heads receive random initialization. HAMSA backbones load ImageNet pre-trained weights while newly added detection-specific layers (region proposal networks, box regressors, mask predictors) receive Xavier initialization [6] to maintain appropriate activation magnitudes. Training proceeds on COCO train2017 split containing approximately 118,000 images with bounding box and segmentation annotations, while evaluation mea-

Table 2. **Ablation Analysis** on ImageNet-1k for small size model. † indicates that instability is encountered during the training SSMs

Model	Param (M)	Top-1 (%)	Mixing Type
Conv	10	68.6	ConvNet
ViT-b	87	78.5	Attention, MLP
S4 †	13.2	58.9	S4
Mamba †	15.3	39.1	Mamba
S4+conv	25.9	82.7	S4, conv
Gated-Conv	26.0	83.1	Conv
Gated-MLP	25.9	83.4	MLP
Hamsa	26.6	84.1	MSS, MLP

sures performance on the val2017 split with 5,000 images.

3. Extended Experimental Results

3.1. Semantic Segmentation on ADE20K

When applied to dense prediction tasks using the UperNet framework on ADE20K, HAMSA-S reaches mean intersection-over-union scores of 49.2% under single-scale evaluation and 50.8% when employing multi-scale testing (Table 1), surpassing convolutional architectures such as ConvNeXt (48.7%/49.6%), SSM-based models including VMamba (48.3%/48.6%) and FractalMamba (48.9%/49.8%), and the majority of transformer architectures tested under identical conditions. While SG-Former [18] maintains a performance advantage (51.2%/52.1%), HAMSA presents a compelling efficiency-accuracy trade-off given its 60 million parameter count and 912 GFLOPs computational requirement, demonstrating that our scanning-free spectral approach transfers effectively to pixel-level prediction tasks requiring fine-grained spatial reasoning beyond image-level classification.

3.2. Ablation Analysis of Hamsa

In our ablation study on the Hamsa architecture, we conducted a thorough analysis of its core components—SSM, convolutional modules, and gated layers—by systematically altering or removing each element and evaluating the performance on ImageNet data. We characterize the various architectures based on Mamba. Mamba uses three components, namely, SSM, convolutions, and gated layers. We show in the figure that removing the SSM module from Mamba results in the gated convolution, while removing both results in a gated MLP architecture. Just using SSM and convolutions without gating layers becomes S4Conv, while we also compare the individual SSM (S4) and convolutional networks. The results, summarized in Table 2, provide compelling evidence of Hamsa's superior design. Specifically, Hamsa, which integrates a streamlined SSM with gated architectures, achieves an impressive top-1 accuracy of 84.1% on ImageNet for small models. This marks a significant improvement over configurations like GatedMLP and S4Conv, which achieve 83.4% and 83.0%, respectively, as well as vanilla Mamba and S4.

3.3. Implementation Details.

All models are trained from scratch on ImageNet-1K [4] (1.28M training images, 224×224 resolution) for 300 epochs using AdamW optimizer (learning rate 1×10^{-3}, weight decay 0.05, batch size 1024 across 8 GPUs). We employ standard augmentations: RandAugment, Mixup ($\alpha = 0.8$), CutMix ($\alpha = 1.0$), and label smoothing (0.1). Learning rate follows cosine decay with 20-epoch linear warmup. For fair comparison, models with $\star$ suffix use Token Labeling [26] following prior SSM works. Fine-tuning for transfer learning uses learning rate 5×10^{-5} for 100 epochs. Downstream tasks (detection/segmentation) follow standard protocols: Mask R-CNN with $1\times$ schedule (12 epochs) and UperNet with 160K iterations, both using multi-scale training. All experiments use V100 GPUs. Code and models will be released.

3.4. Complexity Analysis.

For batch size B, sequence length L, and hidden dimension H, kernel computation requires $\mathcal{O}(HL)$ to compute K for all channels, FFT and convolution require $\mathcal{O}(BHL \log L)$ for spectral transforms and multiplication, and linear projections require $\mathcal{O}(BL(DH+MD))$ for W_u, W_v, W_y, W_o. Overall complexity is $\mathcal{O}(BHL \log L + BLDH)$, significantly faster than self-attention's $\mathcal{O}(BL^2D)$ for long sequences while eliminating the scanning overhead of other vision SSMs.

4. Comparison with Related Work

4.1. Advantages Over MambaOut

While MambaOut removes the scanning operations that burden other SSM architectures, it abandons the entire SSM framework including state-space modeling components, thereby sacrificing the sequential modeling capacity that makes SSMs attractive for vision tasks where spatial relationships matter despite lacking natural ordering. In contrast, HAMSA retains the SSM foundation while introducing two critical innovations: **(1) Spectral Gating with SGLUs** provides learnable frequency-domain modulation through which the network adaptively emphasizes certain spectral components while suppressing others based on input content, creating a flexible mechanism for content-aware processing that static frequency transforms cannot achieve; **(2) Simplified Kernel Parameterization** replaces the traditional three-matrix (A, B, C) representation with a single Gaussian-initialized complex-valued kernel, eliminating discretization procedures that introduce numerical

instabilities while simultaneously reducing parameters and improving training stability, as evidenced by our gradient flow analysis and competitive benchmark results that exceed MambaOut's performance across multiple evaluation metrics.

4.2. Comparison with Scanning-Based SSMs

Architectural Simplicity: Scanning-based methods including Vim [31] with bidirectional traversal, VMamba [13] with 2D cross-scanning patterns, LocalVMamba with spatially-localized scanning windows, and SiMBA with unidirectional scanning all require explicit design choices about how to traverse 2D image tokens in 1D sequential order, necessitating multiple forward passes for different scan directions (typically four directions for cross-scanning), intermediate storage for partial results, and careful merging strategies to combine outputs from different scan paths. HAMSA eliminates these complications entirely by operating on flattened token sequences in a single forward pass through frequency-domain processing, where global information mixing occurs simultaneously across all spatial locations without imposing any sequential ordering constraints.

Selectivity Mechanism: Traditional SSMs achieve input-dependent behavior through selective state updates during sequential scanning, where the model decides how much to update hidden states based on current inputs and accumulated context from previous positions in the scan order, creating a path-dependent computation whose output varies based on scanning direction. HAMSA instead employs SpectralPulseNet for adaptive frequency modulation that operates globally and position-independently, learning which spectral patterns carry task-relevant information without the artificial constraints imposed by scan-order dependencies, resulting in a more flexible selectivity mechanism that treats all spatial positions symmetrically while still enabling content-specific processing through learned frequency gates that respond dynamically to input characteristics.

Training Stability: Scanning-based SSMs inherit the gradient flow challenges associated with traditional SSM discretization procedures, including numerical issues from matrix exponentials and logarithms in the bilinear transform, potential for vanishing or exploding gradients when backpropagating through long sequences, and sensitivity to initialization schemes that must carefully balance multiple interacting matrices. HAMSA's direct kernel parameterization combined with SGLU gating mechanisms provides inherently stable gradient pathways across network depth, as demonstrated in Figure 5 where our model maintains consistent gradient magnitudes throughout training while VMamba experiences multiple explosive gradient events, enabling training of deeper networks without specialized initialization procedures like HiPPO [7] or complex learning rate warmup schedules.

4.3. Key Contributions and Insights

State-of-the-Art Performance: Reaching 85.7% top-1 accuracy on ImageNet-1K classification while requiring only one-third the training time of transformer baselines establishes HAMSA as the strongest SSM-based architecture to date, with consistent superiority over all scanning-based SSM variants (Vim, VMamba, LocalVMamba, SiMBA) maintained across transfer learning experiments on four datasets and dense prediction tasks including object detection and semantic segmentation, validating that spectral processing without scanning provides both simplicity and effectiveness in a single unified approach.

Theoretical Foundation: Our work provides both empirical evidence and theoretical arguments establishing that scanning operations are unnecessary for adapting SSMs to vision tasks, since the convolutional nature of SSM kernels enables efficient global information mixing through frequency-domain operations that naturally handle 2D spatial structure without requiring sequential processing or artificial ordering, suggesting new design directions for vision architectures that exploit frequency-domain properties rather than forcing spatial data into sequential frameworks.

Future Directions: Several promising research directions emerge from this work, including **(1)** extending SpectralPulseNet to generate input-dependent kernels rather than just frequency gates, potentially enabling even more adaptive spectral responses tailored to individual inputs; **(2)** exploring multi-scale spectral processing with different frequency resolutions at different network stages to capture both coarse and fine-grained patterns; **(3)** conducting formal analysis of what patterns SpectralPulseNet learns to emphasize versus suppress compared to the implicit biases encoded in scanning mechanisms. While SG-Former maintains a slight advantage on semantic segmentation, this gap suggests opportunities for architectural refinement rather than fundamental limitations.

Broader Impact: HAMSA's efficiency characteristics, including 5.1ms inference latency and 3.5-fold training acceleration, make the architecture particularly suitable for resource-constrained deployment scenarios where computational budgets limit model selection, while the conceptual insight that simpler non-scanning designs can exceed the performance of more complex scanning strategies may inspire more interpretable and efficient model architectures. Beyond computer vision applications, the success of our spectral approach on time series forecasting benchmarks suggests potential value for other modalities, including video understanding, where temporal and spatial dimensions interact, 3D point cloud processing, where scanning order proves even more arbitrary, and audio pro-

cessing, where frequency-domain representations naturally align with human perception.

Table 3. **Transfer learning dataset statistics.**

Dataset	CIFAR-10	CIFAR-100	Flowers-102	Stanford Cars
Train Size	50,000	50,000	8,144	2,040
Test Size	10,000	10,000	8,041	6,149
Categories	10	100	102	196

Table 4. **Training hyperparameters.**

	ImageNet-1K	CIFAR-10
Optimizer	AdamW	
Momentum	$\beta = (0.9, 0.999)$	
LR schedule	Cosine + linear warmup	
Dropout	0.2	
Label smoothing	0.1	
Image size	224^2	32^2
Base LR	10^{-3}	$\{10^{-4}, 3 \times 10^{-4}, 10^{-3}\}$
Batch size	128	64
Epochs	300	up to 1000
Warmup epochs	10	5
Stochastic depth	0.1	$\{0, 0.1\}$
Weight decay	0.05	$\{0, 0.1\}$

5. Discussion

Why "Adaptive" in SASS? The term *Spectral Adaptive State Space* emphasizes input-dependent behavior through our two gating mechanisms. While the SimplifiedSSMKernel $K = \psi_{\text{re}} + j\psi_{\text{im}}$ is input-agnostic (fixed, learned during training to capture universal frequency patterns), both **SpectralPulseNet** and **SAGU** provide input-dependent adaptation: (1) SpectralPulseNet computes gates $g = \sigma(|\hat{u}|W_g + b_g)$ from input magnitude, enabling *content-aware frequency selection*—different inputs receive different spectral emphasis; (2) SAGU applies $\sigma(|\hat{u}|W_2)$ for *input-dependent modulation*, providing adaptive non-linearity. This hybrid design combines consistency (fixed kernel) with flexibility (adaptive gating), contrasting with static methods like GFNet [17] that lack input-dependent frequency modulation.

Convolution via Spectral Multiplication. A key insight of HAMSA is that *it performs convolution, but is computed efficiently in the frequency domain*. Traditional SSMs compute $y = K * u$ directly in the spatial domain with $\mathcal{O}(L^2)$ complexity. By the Convolution Theorem, this operation is equivalent to element-wise multiplication in the frequency domain: $y = \mathcal{F}^{-1}(\hat{u} \odot \hat{K})$, achievable in $\mathcal{O}(L \log L)$ via FFT. Thus, HAMSA maintains the SSM convolution structure while eliminating computational bottlenecks—the spectral multiplication $\tilde{u} \odot \hat{K}$ is mathematically identical to spatial convolution, just computed orders of magnitude faster.

SpectralPulseNet vs. SAGU: Complementary Roles. While both mechanisms use magnitude-based gating, they serve distinct purposes: **SpectralPulseNet** operates on the input spectrum $\hat{u}$ *before spectral kernel multiplication* ($\tilde{u} \odot \hat{K}$), providing *pre-filtering frequency selection*—deciding which input frequencies to emphasize. **SAGU** operates *after kernel application*, providing *post-multiplication modulation* with dual pathways (linear + gated) for gradient stability. This sequential design—input selection → spectral multiplication → adaptive modulation—maximizes expressiveness while maintaining efficient $\mathcal{O}(L \log L)$ complexity through FFT. The term *Spectral Adaptive State Space* emphasizes input-dependent behavior through our two gating mechanisms. While the SimplifiedSSMKernel $K = \psi_{\text{re}} + j\psi_{\text{im}}$ is input-agnostic (fixed, learned during training to capture universal frequency patterns), both **SpectralPulseNet** and **SAGU** provide input-dependent adaptation: (1) SpectralPulseNet computes gates $g = \sigma(|\hat{u}|W_g + b_g)$ from input magnitude, enabling *content-aware frequency selection*—different inputs receive different spectral emphasis; (2) SAGU applies $\sigma(|\hat{u}|W_2)$ for *input-dependent modulation*, providing adaptive non-linearity. This hybrid design combines consistency (fixed kernel) with flexibility (adaptive gating), contrasting with static methods like GFNet [17] that lack input-dependent frequency modulation.

How SAGU differs from GLU variants: SAGU follows the gating unit paradigm—like GLU (Gated Linear Unit [3]) and SwiGLU (Swish-Gated Linear Unit [19])—where the unit actively *performs gating* on its inputs. However, SAGU introduces critical distinctions: (1) **Domain**: GLU and SwiGLU operate on real-valued spatial tokens, while SAGU operates on complex-valued spectral coefficients $\hat{u} \in \mathbb{C}^L$; (2) **Gating mechanism**: GLU uses $x \odot \sigma(xW)$ and SwiGLU uses $x \odot \text{Swish}(xW)$ directly on inputs, whereas SAGU uses *magnitude-based gating* $\sigma(|\hat{u}|W_2)$ to avoid phase discontinuities in the complex plane; (3) **Adaptivity**: The term "Adaptive" in SAGU emphasizes input-dependent spectral modulation—gates are computed from frequency magnitudes, enabling content-aware adjustment of spectral components. This design maintains the benefits of GLU-style architectures (improved gradient flow, enhanced expressiveness) while addressing the unique challenges of complex-valued frequency-domain processing.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[2] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

[3] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017.

[4] 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*, pages 248–255. Ieee, 2009.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[6] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[7] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2021.

[8] 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*, pages 770–778, 2016.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[10] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*, 2024.

[11] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 33:21002–21012, 2020.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[13] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024.

[14] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022.

[15] 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*, pages 11976–11986, 2022.

[16] Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. *arXiv preprint arXiv:2403.09977*, 2024.

[17] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *Advances in Neural Information Processing Systems*, 34:980–993, 2021.

[18] Sucheng Ren, Xingyi Yang, Songhua Liu, and Xinchao Wang. Sg-former: Self-guided transformer with evolving token reallocation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6003–6014, 2023.

[19] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[20] Yuheng Shi, Minjing Dong, and Chang Xu. Multi-scale vmamba: Hierarchy in hierarchy visual state space model. 2024.

[21] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[22] Lv Tang, HaoKe Xiao, Peng-Tao Jiang, Hao Zhang, Jinwei Chen, and Bo Li. Scalable visual state space model with fractal scanning. *arXiv preprint arXiv:2405.14480*, 2024.

[23] 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*, pages 10347–10357. PMLR, 2021.

[24] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. *arXiv preprint arXiv:2204.07118*, 2022.

[25] Feng Wang, Jiahao Wang, Sucheng Ren, Guoyizhe Wei, Jieru Mei, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Mamba-r: Vision mamba also needs registers. *arXiv preprint arXiv:2405.14858*, 2024.

[26] Pichao Wang, Xue Wang, Hao Luo, Jingkai Zhou, Zhipeng Zhou, Fan Wang, Hao Li, and Rong Jin. Scaled relu matters for training vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2495–2503, 2022.

[27] 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.

[28] Chenhongyi Yang, Zehui Chen, Miguel Espinosa, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J Crowley. Plainmamba: Improving non-hierarchical mamba in visual recognition. *arXiv preprint arXiv:2403.17695*, 2024.

[29] Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? *arXiv preprint arXiv:2405.07992*, 2024.

[30] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.

[31] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.