VISIONALPHABET: COMPRESSING SPATIAL CONTEXT WITH LEARNED LAPLACE POLES

Anonymous authors
Paper under double-blind review

ABSTRACT

VisionAlphabet introduces a distinct approach to compact visual backbones, whereby spatial context is represented and propagated through learned Laplace poles. Each pole provides a stable complex-valued state that carries spatial context: damping controls how far contributions persist across space, while phase modulates how contributions from different locations combine. Four directional scans propagate spatial evidence across the image; the terminal scan's gain-normalized pole energies are spatially averaged to form the image-level classifier descriptor. Under a shared 100-epoch ImageNet-1K protocol without distillation, the 3.25M- and 5.08M-parameter models attain 72.35% and 75.02% Top-1 accuracy, respectively, while using 10–12% fewer parameters than their corresponding compact baselines. The larger variant also outperforms the compared baselines with 10% of the ImageNet-1K training data and, when pretrained on full ImageNet-1K, transfers to COCO detection and instance segmentation. Controlled interventions on a frozen checkpoint show that recognition changes when pole-state transport is disrupted or phase is removed, supporting the interpretation of pole states as functional spatial memory.

1 INTRODUCTION

Visual recognition depends on local evidence—edges, textures, and object parts—but reliable prediction requires integrating this evidence across positions and spatial scales. A visual backbone must therefore preserve useful local structure while progressively combining spatially distributed evidence into compact representations. Convolutional backbones achieve this through stacked local operators and hierarchical downsampling (He et al., 2016; Woo et al., 2023). Vision Transformers instead model interactions between image patches with self-attention (Dosovitskiy et al., 2021), while hierarchical variants recover multi-scale feature maps through local windows and patch merging (Liu et al., 2021). More recently, visual state-space models propagate context through recurrent spatial scans with linear scaling in the number of image locations (Zhu et al., 2024; Liu et al., 2024). These families differ in how they aggregate and transport local evidence over space.

We introduce VisionAlphabet, which represents spatial context through learned Laplace poles. Each pole defines a spatial memory mode: damping controls how far evidence persists, while phase determines whether evidence collected from different locations reinforces or cancels. Local image features update these memories, and four directional scans propagate them across the image. As spatial resolution decreases, a direct feature path preserves local evidence alongside the compressed spatial memories, and a nonlinear refinement combines these two sources of information. The final classifier reads normalized directional energies, retaining the effects of phase-dependent interference formed during propagation.

Beyond predictive performance, we ask whether a trained VisionAlphabet relies on persistent spatial memories and phase-dependent interference. Frozen-checkpoint interventions on ImageNet-1K probe this dependence by periodically resetting spatial states, neutralizing selected terminal-state coordinates, and disabling learned phase rotation in the terminal scan. These interventions disrupt recognition and reveal greater held-out impact for contribution-selected coordinates than for size-matched random selections.

This paper makes three contributions. First, we introduce VisionAlphabet, a parameter-compact backbone that propagates spatial context through four-direction complex Laplace-pole memories and

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.

retains pole-specific evidence in a gain-normalized energy readout. Second, we connect this design to its behavior by characterizing product-pole energies under stated stationarity assumptions and testing memory, phase, and pole selection with frozen-checkpoint interventions. Third, we establish its empirical utility under matched protocols. K128 reaches 75.02% ImageNet-1K Top-1 with 5.08M parameters; its lead persists with limited supervision and on Food-101, and its pretrained features attain 39.60 ± 0.18 box AP and 36.64 ± 0.05 mask AP on COCO with a common head.

2 Related Work

Compact visual backbones. Compact backbones improve accuracy–efficiency trade-offs through depthwise convolutions and inverted-residual blocks, compound scaling, and latency-driven architectural design (Sandler et al., 2018; Tan & Le, 2019; Li et al., 2022). Recent small-scale designs include ConvNeXtV2-Atto, ParC-Net, TinyNeXt, and EMOv2 (Woo et al., 2023; Zhang et al., 2022; Zeng et al., 2025; Zhang et al., 2025). Among state-space and hybrid backbones, MambaVision uses convolutional early stages followed by Mamba-based mixer and self-attention blocks (Hatamizadeh & Kautz, 2025). EfficientViM mixes features within compressed hidden states and fuses prediction logits derived from multiple stages with those from its final output features (Lee et al., 2025). TinyViM's Laplace mixer separates low- and high-frequency features for Mamba and depthwise-convolution processing, respectively (Ma et al., 2025).

Structured and multidimensional state spaces. S4 develops structured state-space layers, S5 uses a multi-input multi-output formulation, and LRU employs complex-diagonal linear recurrences (Gu et al., 2022; Smith et al., 2023; Orvieto et al., 2023); Mamba makes the dynamics input-dependent (Gu & Dao, 2024). ALPHABET represents temporal history with stable complex Laplace poles whose real and imaginary parts control decay and oscillation, and reads modal energies and lag moments from the resulting states (Ko et al., 2026). S4ND extends structured state spaces to multidimensional signals such as images and videos, producing global continuous convolutional kernels over multiple axes (Nguyen et al., 2022). Baron et al. (2024) instead introduce a 2-D state-space layer with coupled horizontal and vertical states to provide spatial inductive bias in vision backbones. In parallel, Vim scans bidirectionally; VMamba uses four-route 2D selective scans; LocalMamba searches windowed scan choices; and PlainMamba combines continuous 2D scanning with direction-aware state updates (Zhu et al., 2024; Liu et al., 2024; Huang et al., 2024; Yang et al., 2024). TCP-SSM makes visual recurrence dynamics explicit through token-conditioned stable real and complex-conjugate poles (Shoouri et al., 2026). VisionAlphabet differs by transporting learned Laplace-pole states through four directional product scans and forming the classifier descriptor directly from their normalized directional energies.

Complex-valued representations. Complex-valued neural networks use coupled real and imaginary components to represent magnitude and phase (Hirose, 2012). Complex-valued convolution has been explored for phase-sensitive visual features (Guberman, 2016), while Deep Complex Networks develops complex convolution, normalization, and initialization for deep learning (Trabelsi et al., 2018). Widely-linear maps additionally combine a complex input with its conjugate, taking the form $y = Az + B\bar{z}$ (Picinbono & Chevalier, 1995). In VisionAlphabet, complex states carry the learned spatial pole responses; their interactions precede the real-valued energy readout.

3 VisionAlphabet

Laplace interpretation of state-space dynamics. The Laplace transform represents a signal using complex exponential modes parameterized by $s = \sigma + i\omega$, where σ controls exponential decay and ω controls oscillatory phase. State-space models propagate information through recurrent latent states whose dynamics can likewise be interpreted through characteristic complex modes. This establishes a natural connection between the two views: a stable complex *pole* determines both how rapidly past information decays and how its phase evolves over successive state updates. VisionAlphabet makes this connection explicit by parameterizing spatial state transitions with learned complex poles, allowing its hidden states to act as Laplace-like memory modes over spatial history.

A state-space model retains input history through a recurrent latent state, whose transition can be parameterized by a stable Laplace mode $s = \sigma + i\omega$. Under constant-input discretization with step

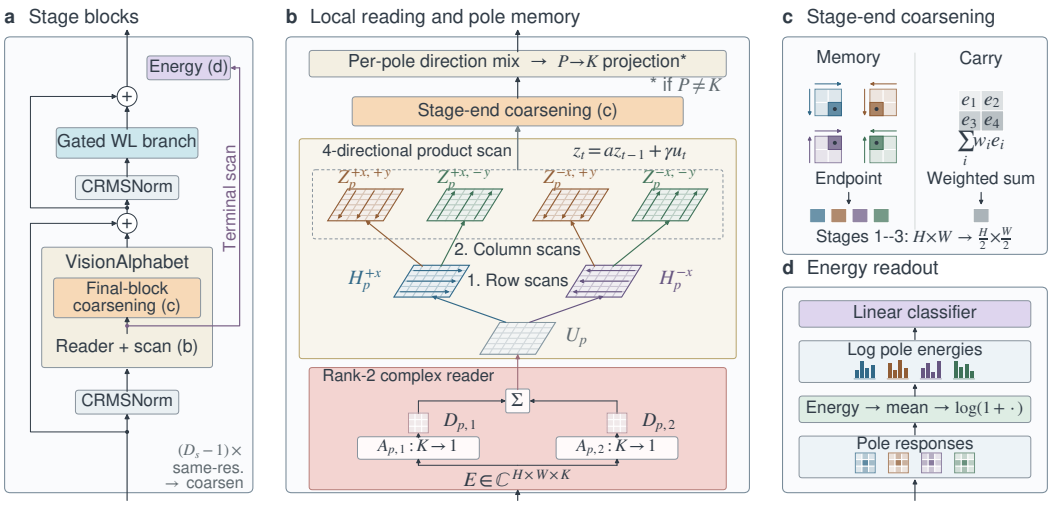

Figure 1: **VisionAlphabet architecture.** **(a)** Block structure and terminal readout. **(b)** Local reader and spatial memory in a stage-ending unit. **(c)** Memory and carry coarsening. **(d)** Directional-energy readout.

size h,

$$z_t = az_{t-1} + \gamma u_t, \qquad a = e^{-sh}, \qquad \gamma = \frac{1 - e^{-sh}}{s}.$$

Here, a propagates the accumulated state, while γ integrates the current input. After n transitions, a past contribution is weighted by $a^n = e^{-\sigma hn}e^{-i\omega hn}$, so that σ controls exponential decay and ω controls phase evolution. VisionAlphabet extends this principle to spatial state transitions, learning distinct damping and phase parameters along the horizontal and vertical axes so that each pole represents a spatial memory mode with its own persistence and interference pattern.

Complex feature encoding and pole excitation. The input RGB image is first mapped to an initial complex feature field $E^{(0)}$ using two stride-2 Conv–LayerNorm–GELU blocks, followed by residual channel mixing, RMS normalization, and an analysis projection. At block ℓ, the incoming feature field is denoted by $E^{(\ell)} \in \mathbb{C}^{H_\ell \times W_\ell \times K_\ell}$, and a rank-factorized local reader maps it to P_ℓ pole-specific excitation fields, collectively forming $U^{(\ell)} \in \mathbb{C}^{H_\ell \times W_\ell \times P_\ell}$. For each pole p, the corresponding effective 3×3 complex kernel is constructed as a rank-R combination of learned channel and spatial factors, with $R = 2$ in our implementation. Writing k for an input channel and $\Delta \in \{-1, 0, 1\}^2$ for a spatial offset, this is

$$\mathcal{W}^{\text{raw}}_{p,k,\Delta} = \sum_{r=1}^{R} D_{p,r,\Delta} A_{p,r,k}. \tag{1}$$

This factorization allows each pole to extract a distinct combination of local spatial structure and feature-channel information while avoiding a dense 3×3 complex kernel. The resulting excitation $U_p^{(\ell)}$ serves as the direct input to the corresponding spatial pole recurrence. We omit the block index ℓ below when it is unambiguous.

Four-direction product pole memory. The pole-specific excitation U_p is propagated along both spatial axes using learned damping and phase parameters. For axis $j \in \{x, y\}$ and pole p, we define

$$s_{j,p} = \frac{d_{j,p} - i\theta_{j,p}}{h_j}, \qquad a_{j,p} = e^{-s_{j,p}h_j}, \qquad \gamma_{j,p} = \frac{1 - e^{-s_{j,p}h_j}}{s_{j,p}},$$

where $h_x = 1/W$ and $h_y = 1/H$. Under the convention $s = \sigma + i\omega$ above, this corresponds to $\sigma_{j,p} = d_{j,p}/h_j$ and $\omega_{j,p} = -\theta_{j,p}/h_j$. Since $|a_{j,p}| = e^{-d_{j,p}} < 1$, the damping $d_{j,p}$ controls spatial persistence, while the phase $\theta_{j,p}$ controls interference across spatial lags.

Here, u and v index the vertical and horizontal spatial coordinates, respectively, and $U_{u,v,p}$ denotes the excitation of pole p at location (u, v). The horizontal and vertical recurrences are

$$H^{s_x}_{u,v,p} = a^{s_x}_x H^{s_x}_{u,v-s_x,p} + \gamma^{s_x}_x U_{u,v,p},$$
$$Z^{s_x,s_y}_{u,v,p} = a^{s_y}_y Z^{s_x,s_y}_{u-s_y,v,p} + \gamma^{s_y}_y H^{s_x}_{u,v,p}, \qquad s_x, s_y \in \{+1, -1\}.$$

The superscripts s_x and s_y on the scan coefficients index the forward and reverse directions rather than exponentiation. The recurrence $z_t = az_{t-1} + \gamma u_t$ is the generic one-dimensional state update. In the spatial model, it is applied first along the horizontal axis to produce $H^{s_x}_{u,v,p}$, and then along the vertical axis to produce $Z^{s_x,s_y}_{u,v,p}$. Thus, $Z^{s_x,s_y}_{u,v,p}$ is the two-dimensional realization of the generic state z_t for pole p and scan direction (s_x, s_y).

Reverse scans use conjugate coefficients, preserving the damping magnitude while reversing the phase rotation associated with spatial direction. This yields four directional memory fields corresponding to $(+,+)$, $(-,+)$, $(+,-)$, and $(-,-)$. These separable product scans preserve the two spatial axes while allowing each pole to accumulate context according to its learned decay and phase dynamics.

Directional aggregation and stage transition. Following the four-direction scans, each directional memory state is normalized using a transported reference gain,

$$v_t = |a|^2 v_{t-1} + |\gamma|^2 v^{\text{in}}_t, \qquad \widetilde{z}_t = \frac{z_t}{\sqrt{\max(\text{sg}(v_t), \epsilon)}}.$$

Here, v^{in}_t denotes the incoming reference gain, v_t tracks the reference response scale induced by the pole dynamics, and $\text{sg}(\cdot)$ denotes stop-gradient. The horizontal scan uses unit input gain, while the vertical scan receives the corresponding horizontal reference gain. Applying this normalization to each spatial directional state yields the normalized memory fields $\widetilde{Z}^{s_x,s_y}_{u,v,p}$.

At a stage transition, each scan direction selects its downstream endpoint within the corresponding 2×2 cell before directional fusion, while the carry branch aggregates the input features from the same cell. For each pole p, the four normalized directional states are fused by a learned $4 \to 4 \to 1$ widely-linear network with complex SiLU, yielding a complex memory response m_p. In parallel, the carry branch forms C from the corresponding input features. The resulting memory responses $m = \{m_p\}$ and carry features C are projected to a common feature space and merged as

$$X = \Pi_M m + \Pi_C C,$$

where Π_M and Π_C are learned projections for the memory and carry branches, respectively.

The merged feature is then refined by a gated widely-linear residual block. Let

$$\widehat{X} = \text{CRMSNorm}(X), \qquad \text{WL}_r(z) = A_r z + B_r \bar{z}, \quad r \in \{v, o\}.$$

The refinement is

$$\begin{aligned} V &= \text{WL}_v(\widehat{X}), \\ G &= \text{SiLU}\Big(W_g[\Re\widehat{X}; \Im\widehat{X}]\Big), \\ Y &= X + \text{WL}_o(V \odot G), \end{aligned} \tag{2}$$

where WL_v and WL_o denote the widely-linear value and output projections, respectively, W_g is a learned real projection producing the gate G, and $\odot$ denotes element-wise modulation. Same-resolution blocks retain the normalized directional states at each spatial location and use the incoming feature field directly as the carry.

Directional-energy readout. For classification, we form a directional-energy descriptor only from the normalized pole-memory states of the terminal scan. Let T denote this scan, with spatial size $H_T \times W_T$ and P_T poles. For pole p and direction pair $(s_x, s_y) \in \{+1, -1\}^2$, we define

$$q^{(T)}_{s_x,s_y,p} = \log\left(1 + \frac{1}{H_T W_T} \sum_{u=1}^{H_T} \sum_{v=1}^{W_T} \left|\widetilde{Z}^{(T),s_x,s_y}_{u,v,p}\right|^2\right). \tag{3}$$

Concatenating these values over four directions and P_T poles gives the $4P_T$-dimensional terminal descriptor. Non-affine BatchNorm followed by a learned affine classifier maps this descriptor to class logits. Intermediate stages propagate their feature states to the terminal scan; their energy descriptors are not included in the classifier input.

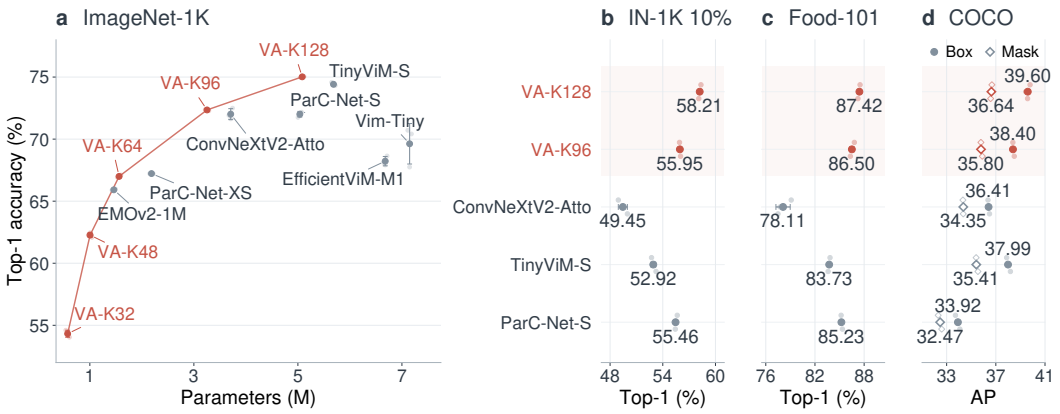

Figure 2: **VisionAlphabet performance across classification and dense prediction. (a)** ImageNet-1K Top-1 accuracy versus parameter count after 100 epochs. **(b)** ImageNet-1K 10% Top-1 accuracy. **(c)** Food-101 Top-1 accuracy. **(d)** COCO box and mask AP with a common Mask R-CNN/FPN head. Large markers show reported means, whiskers show sample SD where available, and faint markers show individual seed results.

4 Spectral Interpretation and Separability of Pole Energy

For a fixed, second-order stationary scalar excitation field on $\mathbb{Z}^2$ with finite spectral measure F, the reference-gain-normalized population energy takes the product-Poisson form

$$\mathcal{Q}_F(a_x, a_y) = \int_{\mathbb{T}^2} P_{a_x}(\omega_x) P_{a_y}(\omega_y)\, F(d\omega_x, d\omega_y), \qquad P_a(\omega) = \frac{1 - |a|^2}{|e^{i\omega} - a|^2}, \tag{4}$$

where $\mathbb{T} = \mathbb{R}/(2\pi\mathbb{Z})$ and $a_x, a_y \in \mathbb{D} = \{a \in \mathbb{C} : |a| < 1\}$ are stable scan coefficients. Writing $a_j = r_j e^{i\phi_j}$, the phase ϕ_j determines the spectral center of the kernel, while the radius $r_j = e^{-d_j}$ controls its concentration.

The characteristicness of Laplace-pole energy measurements was established for fixed, fully observed, equal-step, second-order stationary temporal feature processes in ALPHABET (Ko et al., 2026): the continuum of Poisson-localized modal measurements identifies the underlying spectrum, and almost every admissible mode separates any fixed finite collection of distinct spectra. Our spatial measurement is the separable two-dimensional counterpart of this construction. Applying the same analytic and Fourier–Stieltjes argument along the two axes, equality

$$\mathcal{Q}_F(a_x, a_y) = \mathcal{Q}_G(a_x, a_y)$$

on a positive-measure subset of $\mathbb{D}^2$ implies $F = G$. Consequently, for any finite collection of pairwise distinct spatial spectra, almost every stable product-pole pair yields pairwise distinct population energies.

This result motivates retaining direction–pole energies as separate readout coordinates. The monotone $\log(1 + \cdot)$ transform preserves distinctions between population energies while compressing their dynamic range, and, following the affine-realizability argument of Ko et al. (2026), distinct population energy prototypes can be separated by an affine head. Phase-dependent interference acts on the complex pole states before this energy pooling.

5 Experimental Results

5.1 ImageNet-1K observations

Figure 2(a) reports three-seed ImageNet-1K (Deng et al., 2009; Russakovsky et al., 2015) results under the shared 100-epoch protocol across compact vision baselines (Zeng et al., 2025; Zhang et al., 2025; 2022; Woo et al., 2023; Ma et al., 2025; Lee et al., 2025; Zhu et al., 2024). All models are

Table 1: Top-1 accuracy (%) under the scratch classification protocols. Parameter counts include the 1,000-class classifier. All entries report the mean ± sample SD over three runs. Bold marks the highest reported accuracy in each column.

Backbone	Params (M)	IN-1K 100%	IN-1K 10%	Food-101
VisionAlphabet-K128	5.08	$\mathbf{75.02 \pm 0.05}$	$\mathbf{58.21 \pm 0.13}$	$\mathbf{87.42 \pm 0.15}$
VisionAlphabet-K96	3.25	72.35 ± 0.06	55.95 ± 0.07	86.50 ± 0.26
ConvNeXtV2-Atto	3.71	72.01 ± 0.44	49.45 ± 0.52	78.11 ± 0.87
TinyViM-S	5.68	74.42 ± 0.14	52.92 ± 0.27	83.73 ± 0.12
ParC-Net-S	5.04	72.01 ± 0.23	55.46 ± 0.18	85.23 ± 0.11

trained from scratch at 224×224 with the same AdamW/cosine schedule and RandAugment, Mixup, and random erasing. For the VisionAlphabet variants, K denotes the number of complex feature coordinates propagated between blocks.

At fixed depth, scaling the complex feature and pole widths improves mean Top-1 by 20.70 points from K32 to K128. The larger variants also advance the accuracy–parameter frontier of the tested models: K96 reaches 72.35% with 3.25M parameters, exceeding ConvNeXtV2-Atto while using 12.3% fewer parameters. K128 attains the panel's highest mean accuracy, 75.02%, with 5.08M parameters—0.60 points above TinyViM-S despite 10.6% fewer parameters, and 3.01 points above similarly sized ParC-Net-S.

5.2 Additional classification settings

We evaluate the two primary VisionAlphabet variants, K96 and K128, against three strong and structurally distinct baselines from the main ImageNet-1K panel. ConvNeXtV2-Atto represents a modern hierarchical ConvNet; TinyViM-S is the strongest visual state-space baseline in the panel; and ParC-Net-S represents position-aware global convolution. We retain this five-model panel in two complementary classification settings. ImageNet-1K 10% uses the public 10% subset released with SimCLR (Chen et al., 2020): 128,116 training images covering all 1,000 classes, with 128–129 images per class. We pin the released membership list and use the same subset for every architecture and training seed; only model initialization, augmentation order, and optimization randomness vary. The original 50,000-image ImageNet-1K validation set is retained in full. Food-101 (Bossard et al., 2014) uses its official split of 75,750 training and 25,250 test images. All models are trained from scratch at 224×224 resolution for 100 epochs.

Table 1 shows that, under the ImageNet-1K 10% protocol, both VisionAlphabet variants lead the five-model panel: K128 attains 58.21% Top-1 and K96 55.95%, exceeding TinyViM-S and ConvNeXtV2-Atto by 5.29 and 6.50 points, respectively, while using fewer parameters. The advantage extends to Food-101, where K96 and K128 exceed the strongest baseline, ParC-Net-S, by 1.27 and 2.19 points, respectively.

5.3 Object detection and instance segmentation

Image classification tests global recognition, but does not show whether the learned representation preserves the spatial detail needed to localize objects and predict object masks. We therefore evaluate ImageNet-pretrained VisionAlphabet features on detection and instance segmentation using a common Mask R-CNN/FPN downstream architecture.

We fine-tune the ImageNet-1K seed-501, epoch-100 checkpoints on COCO2017 (Lin et al., 2014). The existing VisionAlphabet interface exposes the real and imaginary coordinates of four spatial excitation features E_l at strides 4, 8, 16, and 32, omitting the terminal classification readout. Mask R-CNN (He et al., 2017) uses a 256-channel FPN and common proposal, box, and mask heads for 12 epochs.

COCO. We report COCO mAP averaged over categories and IoU thresholds $0.50 : 0.05 : 0.95$ (the standard COCO AP metric). Across downstream seeds 501, 509, and 521, VisionAlphabet-K128 attains 39.60 ± 0.18 box mAP and 36.64 ± 0.05 mask mAP. K128 also leads the compared baselines on AP_{75} and AP_S (Table 2). Under the same downstream protocol, VisionAlphabet-K96 attains

38.40 ± 0.10 box mAP and 35.80 ± 0.10 mask mAP. ParC-Net-S uses FP32 global blocks and train-only BN recalibration after epoch one, with BN running statistics fixed thereafter; the detection head, optimizer, effective batch size, and 12-epoch schedule remain shared (Appendix C).

Table 2: COCO detection and instance segmentation with the common Mask R-CNN/FPN head. Values are mean $\pm$ sample SD over three runs. mAP averages categories and IoU thresholds $0.50 : 0.05 : 0.95$. All scores use the 0–100 scale.

Backbone	Box			Mask		
	mAP	AP_{75}	AP_S	mAP	AP_{75}	AP_S
VisionAlphabet-K128	$\mathbf{39.60 \pm 0.18}$	$\mathbf{42.74 \pm 0.41}$	$\mathbf{23.07 \pm 0.94}$	$\mathbf{36.64 \pm 0.05}$	$\mathbf{39.22 \pm 0.22}$	$\mathbf{20.40 \pm 0.73}$
VisionAlphabet-K96	38.40 ± 0.10	41.24 ± 0.23	21.97 ± 0.48	35.80 ± 0.10	38.04 ± 0.45	19.54 ± 1.36
ConvNeXtV2-Atto	36.41 ± 0.07	38.91 ± 0.23	20.02 ± 0.27	34.35 ± 0.03	36.22 ± 0.08	17.93 ± 0.29
TinyViM-S	37.99 ± 0.17	40.54 ± 0.11	21.55 ± 0.34	35.41 ± 0.12	37.35 ± 0.16	19.20 ± 0.22
ParC-Net-S	33.92 ± 0.18	35.75 ± 0.12	20.07 ± 0.20	32.47 ± 0.15	34.40 ± 0.18	18.21 ± 0.04

6 What Does the Learned Memory Do?

We probe three aspects of learned memory: dependence on spatial-state transport, dependence on phase and damping, and input- and decision-dependent pole use. The ImageNet-1K analyses use the same frozen VisionAlphabet-K128 seed-501 checkpoint at epoch 100.

6.1 Spatial Memory Dependence

The state-reset study uses all 50,000 validation images (50 per class).

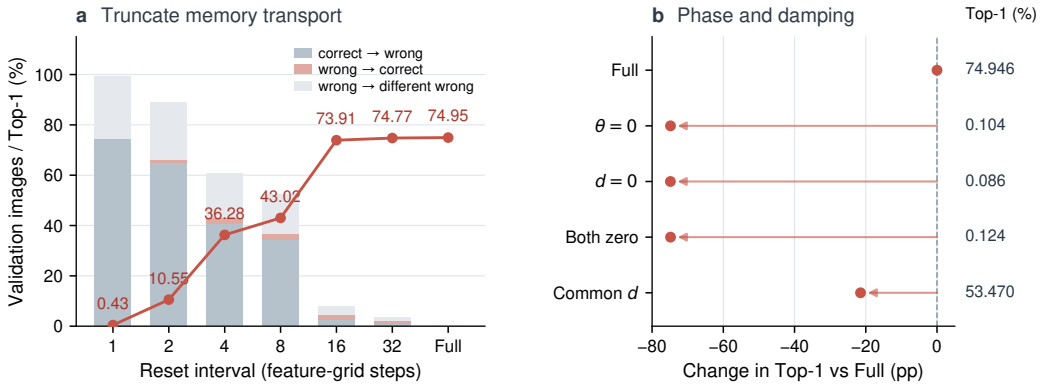

Figure 3: **Spatial-memory interventions on ImageNet-1K. (a)** Top-1 accuracy and prediction changes as scan memory is reset at shorter intervals. **(b)** Top-1 changes after all-stage phase and damping interventions. Error bars are paired class-bootstrap 95% intervals.

Resets set the transport multiplier a to zero at segment boundaries in each scan direction, discarding the incoming state while retaining the current excitation. Intervals 32 and 16 retain 74.770% and 73.910% Top-1, close to the 74.946% Full reference, whereas interval 8 drops to 43.022% and resetting at every step leaves 0.430%. The sharp drop between intervals 16 and 8, dominated by correct-to-wrong prediction changes, shows that the trained model relies on spatial evidence accumulated across scan steps.

6.2 Phase and Damping Dependence

All-stage interventions. Removing phase, damping, or both across all stages collapses Top-1 to roughly 0.1% (Figure 3(b)). Here $d = 0$ means unit-modulus transport rather than a memory reset.

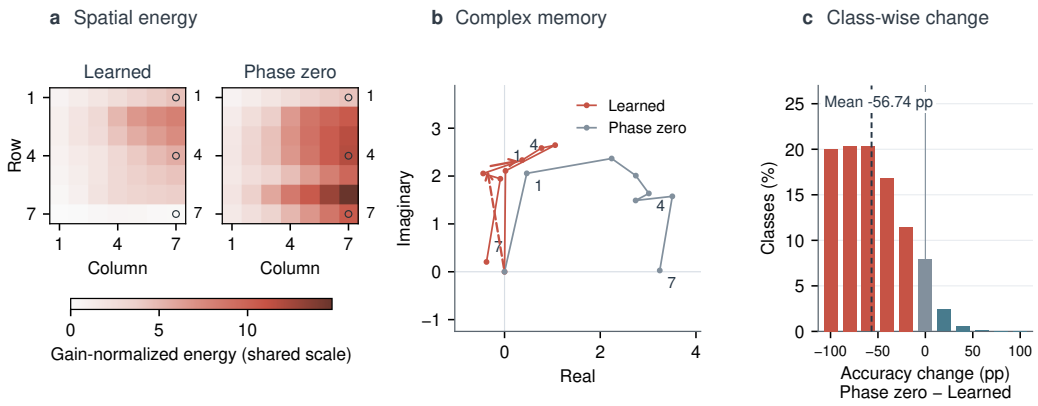

Figure 4: **Terminal phase in spatial memory and recognition.** **(a)** Pole-energy maps with learned versus zero terminal-axis phase. **(b)** Corresponding complex-state trajectories. **(c)** Class-wise accuracy changes on 5,000 validation images (five per class); the dashed line marks the mean. The K128 checkpoint, upstream excitations, and damping remain fixed; only the terminal scan and reference gain are recomputed.

Replacing pole-specific damping by the median learned damping across poles within each block and axis retains 53.470%, showing that the learned diversity of memory timescales also matters.

Terminal-only phase intervention. To examine phase dependence while holding upstream computation fixed, we set both terminal-axis scan phase parameters to zero and recompute the terminal scan and its reference gain. In a forward scan, the phase in $a = e^{-d+i\theta}$ rotates the carried state before it is added to the next excitation. After two positions, $z_1 = \gamma(\theta)(u_1 + e^{-d+i\theta}u_0)$, whose energy contains the cross term $2|\gamma(\theta)|^2 e^{-d}\Re(e^{i\theta}u_0\bar{u}_1)$. Changing θ can therefore alter the terminal energy even when the upstream excitations are fixed. The head pools gain-normalized squared magnitudes of these states (Eq. 3). On five fixed validation images per class, this intervention reduces Top-1 from 75.28% to 18.54% and changes 4,003 of 5,000 predictions (Figure 4(c)). Figure 4(a,b) illustrates the resulting spatial energy maps and complex-state trajectories, supporting dependence on terminal phase in this trained model.

6.3 Input- and decision-dependent pole use

A pole can carry information without contributing equally to every prediction. Using the same frozen checkpoint, we decompose each image's winner–runner-up margin into a class-pair bias and signed per-pole contributions. We sum each pole's four directional terms and rank poles by their absolute contribution. We use 25 images per ground-truth class to select poles and the remaining 25 only to evaluate them.

Pole use changes with context. The top-ranked poles from actual contributions have a median split-half top-16 Jaccard overlap of 0.60. This falls to 0.33 when image descriptors are shuffled while class pairs and classifier weights stay fixed (Figure 5a). Which class challenges the winner also matters: among eligible repeated winner–runner-up pairs, pair-specific selection captures more held-out contribution than selection based only on the winner or on a different runner-up (Figure 5b).

Selected poles affect predictions. We select 16 poles on discovery images, replace their four terminal coordinates with training-BN means on disjoint evaluation images, and recompute all 1,000 head logits. Top-1 falls from 74.49% to 62.07%; a size-matched random mask leaves it at 73.76% on average across 20 runs (Figure 5c). Thus, the selected terminal pole evidence matters to classification.

7 Corruption Robustness

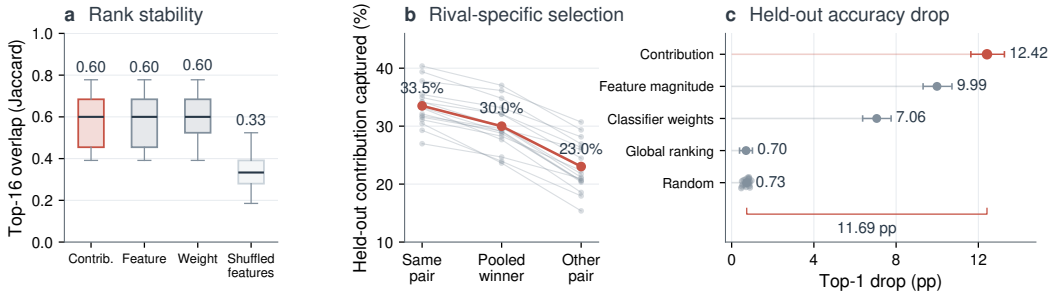

Figure 5: **Input- and decision-dependent pole use. (a)** Stability of top-ranked poles across disjoint image halves. **(b)** Held-out contribution captured when selection preserves or changes the competing class. **(c)** Top-1 drop after neutralizing selected poles on held-out images.

We evaluate whether recognition is maintained under common input corruptions using ImageNet-C (Hendrycks & Dietterich, 2019). Figure 6 reports mean Top-1 accuracy across 15 corruption types and five severity levels. Each model is evaluated with a single ImageNet-1K checkpoint.

In our comparison, VisionAlphabet-K128 achieves 45.83% mean corruption accuracy, closely matching TinyViM-S (45.86%) and exceeding ConvNeXtV2-Atto and ParC-Net-S. Vim-Tiny attains the highest mean accuracy in this comparison (46.65%).

Performance varies by corruption type. The model exceeds TinyViM-S under pixelation by 7.35 points on average, with a lead at every severity. Pixelation removes fine detail but retains coarse layout, consistent with the use of spatially accumulated pole energies. Glass blur and elastic deformation instead disrupt local order and shape; it trails the leading baselines under both. Strong fog and contrast also hurt, suggesting that clear local cues remain important.

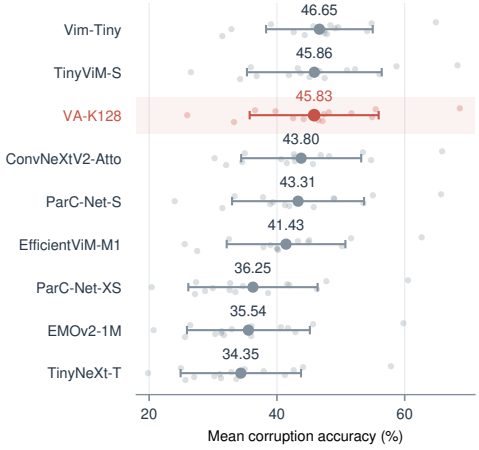

Figure 6: **ImageNet-C accuracy.** Large points: mean over 15 types; faint points: type means over five severities; whiskers: ± 1 SD across types.

8 Conclusion

We introduced VisionAlphabet, a parameter-compact visual backbone that represents spatial context with learned complex-pole memories. A factorized reader excites pole states, directional scans transport evidence, and a gain-normalized energy readout supports classification. Damping controls spatial persistence, while phase governs interference between locations.

Under the shared 100-epoch ImageNet-1K protocol, K96 reaches 72.35% Top-1 with 3.25M parameters and K128 reaches 75.02% with 5.08M, each outperforming a larger compact baseline. K128 also leads the five-model comparisons on ImageNet-1K 10% and Food-101. Its ImageNet-pretrained features transfer to COCO under a common Mask R-CNN/FPN head. On ImageNet-C, its mean accuracy closely matches TinyViM-S, although robustness varies by corruption type.

Frozen-checkpoint interventions connect the mechanism to the trained model's decisions. Resetting state transport, disabling learned terminal-scan phase rotation, and neutralizing selected poles change held-out predictions; contribution-selected poles have larger effects than size-matched random selections. Together with the recognition and dense-transfer results, these interventions support learned complex-pole memory as a parameter-efficient and interpretable spatial primitive.

AI USE STATEMENT

In this work, we used generative AI tools to improve the readability and organization of the manuscript, identify relevant literature, and provide feedback on paper structure and wording. All AI-assisted suggestions were reviewed by the authors; cited literature and experimental claims were manually checked against the original sources and results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Model configurations and training settings are provided in Appendix A, while operator and implementation details together with the spatial separability argument are given in Appendix B. Appendix C reports the COCO transfer protocol and ParC-Net-S stabilization settings. The main text specifies the frozen checkpoint and evaluation sets used for the mechanistic interventions, and Appendix D provides per-corruption ImageNet-C results.

REFERENCES

Ethan Baron, Itamar Zimerman, and Lior Wolf. A 2-dimensional state space layer for spatial inductive bias. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=BGkqypmGvm.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, pp. 446–461. Springer, 2014.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607, 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

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.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.00396.

Nitzan Guberman. On complex valued convolutional neural networks, 2016. URL https://arxiv.org/abs/1602.09046.

Ali Hatamizadeh and Jan Kautz. MambaVision: A hybrid mamba–transformer vision backbone. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25261–25270, 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*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision*, 2017.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1903.12261.

Akira Hirose. *Complex-Valued Neural Networks*, volume 400 of *Studies in Computational Intelligence*. Springer Berlin, Heidelberg, 2 edition, 2012. doi: 10.1007/978-3-642-27632-3.

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. URL https://arxiv.org/abs/2403.09338.

Daehwa Ko, JaeHyeon Kim, Oh Seong Kwon, and Jay Hoon Jung. ALPHABET: A laplace-pole history aggregator with banked exponential transport. *arXiv preprint arXiv:2608.24051*, 2026. URL https://arxiv.org/abs/2608.24051.

Sanghyeok Lee, Joonmyung Choi, and Hyunwoo J. Kim. EfficientViM: Efficient vision Mamba with hidden state mixer based state space duality. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14923–14933, 2025.

Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. EfficientFormer: Vision transformers at MobileNet speed. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, 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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/baa2da9ae4bfed26520bb61d259a3653-Abstract.html.

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*, pp. 10012–10022, 2021.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

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

Xiaowen Ma, Zhenliang Ni, and Xinghao Chen. TinyViM: Frequency decoupling for tiny hybrid vision Mamba. In *IEEE/CVF International Conference on Computer Vision*, pp. 23519–23529, 2025.

Eric Martin and Chris Cundy. Parallelizing linear recurrent neural nets over sequence length. In *International Conference on Learning Representations*, 2018.

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. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 2846–2861. Curran Associates, Inc., 2022. doi: 10.52202/068431-0206. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/13388efc819c09564c66ab2dc8463809-Paper-Conference.pdf.

Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pp. 26670–26698, 2023. URL https://proceedings.mlr.press/v202/orvieto23a.html.

Bernard Picinbono and Pascal Chevalier. Widely linear estimation with complex data. *IEEE Transactions on Signal Processing*, 43(8):2030–2033, 1995. doi: 10.1109/78.403373.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Sara Shoouri, Morteza Tavakoli Taba, and Hun-Seok Kim. TCP-SSM: Efficient vision state space models with token-conditioned poles. *arXiv preprint arXiv:2605.11563*, 2026. URL https://arxiv.org/abs/2605.11563.

Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114, 2019.

Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J. Pal. Deep complex networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1T2hmZAb.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2301.00808.

Chenhongyi Yang, Zehui Chen, Miguel Espinosa, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J. Crowley. PlainMamba: Improving non-hierarchical Mamba in visual recognition. In *British Machine Vision Conference*, 2024. URL https://arxiv.org/abs/2403.17695.

Fanhong Zeng, Huanan Li, Juntao Guan, Rui Fan, Tong Wu, Xilong Wang, and Rui Lai. An efficient hybrid vision transformer for TinyML applications. In *IEEE/CVF International Conference on Computer Vision*, pp. 19914–19924, 2025.

Haokui Zhang, Wenze Hu, and Xiaoyu Wang. ParC-Net: Position aware circular convolution with merits from ConvNets and transformer. In *European Conference on Computer Vision*, pp. 613–630, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/papers/136860600.pdf.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Jiangning Zhang, Teng Hu, Haoyang He, Zhucun Xue, Yabiao Wang, Chengjie Wang, Yong Liu, Xiangtai Li, and Dacheng Tao. EMOv2: Pushing 5m vision model frontier. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(11):10560–10576, 2025. doi: 10.1109/TPAMI.2025.3596776.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI Conference on Artificial Intelligence*, pp. 13001–13008, 2020.

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 *International Conference on Machine Learning*, pp. 62429–62442, 2024. URL https://proceedings.mlr.press/v235/zhu24f.html.

A Model Configurations, Training, and Hardware Efficiency

Table 3: ImageNet-1K model configurations. Width and pole entries denote complex coordinates. Parameter counts include the classifier.

Model	$K_1, \ldots, K_4$	$P_1, \ldots, P_4$	Depth	Params (M)
VisionAlphabet-K32	$(32, 32, 32, 32)$	$(48, 48, 48, 48)$	$(2, 2, 6, 2)$	0.573
VisionAlphabet-K48	$(48, 48, 48, 48)$	$(64, 64, 64, 64)$	$(2, 2, 6, 2)$	1.005
VisionAlphabet-K64	$(64, 64, 64, 64)$	$(80, 80, 80, 80)$	$(2, 2, 6, 2)$	1.563
VisionAlphabet-K96	$(96, 96, 96, 96)$	$(128, 128, 128, 128)$	$(2, 2, 6, 2)$	3.253
VisionAlphabet-K128	$(128, 128, 128, 128)$	$(160, 160, 160, 128)$	$(2, 2, 6, 2)$	5.083

All ImageNet-1K models use AdamW (Loshchilov & Hutter, 2019) with an effective batch size of 256, learning rate 0.003, weight decay 0.05, five warmup epochs, and a per-update cosine schedule (Loshchilov & Hutter, 2017). Augmentation comprises RandAugment (Cubuk et al., 2020) (`rand-m9-mstd0.5-inc1`), Mixup with $\alpha = 0.8$ (Zhang et al., 2018), random erasing with probability 0.25 (Zhong et al., 2020), and label smoothing 0.1 (Szegedy et al., 2016); CutMix is disabled. Model computation uses BF16 and classification loss is evaluated in FP32. The reporting endpoint is final epoch-100 validation accuracy rather than the best intermediate epoch.

Measured hardware efficiency. Table 4 measures all 13 ImageNet-1K classifiers on one RTX 4090 (driver 580.178.04, PyTorch 2.9.1+CUDA 12.8) at 224×224. Inference loads an epoch-100 checkpoint and uses BF16 autocast and channels-last inputs. VisionAlphabet uses shape-aware autotuning of its Triton scan kernels. We measured eager, default-compile, and `reduce-overhead` execution for every model and batch size and report the fastest measured mode. All batch-1 entries and all batch-16 entries except Vim-Tiny use `reduce-overhead`; Vim-Tiny uses eager at batch 16. The table gives synchronized steady-state medians after five warmup passes over ten timed passes; compilation and image decoding are excluded. The batch-16 peak is `torch.cuda.max_memory_allocated`, including the model and its activations. Inputs are synthetic and resident on the GPU. FLOPs are counted for one 224×224 image.

Table 4: Single-RTX 4090 hardware efficiency at 224×224. Forward FLOPs use the same source-audited counting rule for all models, including custom operators. Inference uses trained checkpoints.

			Measured inference		
Model	Params (M)	FLOPs (G)	B1 (ms)	B16 (img/s)	B16 peak (MiB)
VisionAlphabet-K32	0.57	1.61	0.90	8,772	149
VisionAlphabet-K48	1.00	3.12	0.94	7,046	177
VisionAlphabet-K64	1.56	5.13	0.97	5,445	163
VisionAlphabet-K96	3.25	11.73	1.10	3,589	271
VisionAlphabet-K128	5.08	19.51	1.23	2,665	318
EMOv2-1M	1.46	0.60	0.63	13,338	168
ParC-Net-XS	2.19	1.53	0.69	8,929	325
ConvNeXtV2-Atto	3.71	1.14	0.42	20,925	134
EfficientViM-M1	6.68	0.48	0.49	23,024	156
TinyNeXt-T	1.03	0.52	0.42	24,134	134
Vim-Tiny	7.15	3.13	8.55	1,776	309
TinyViM-S	5.68	1.92	2.90	5,210	158
ParC-Net-S	5.04	2.63	0.89	5,161	336

The forward-FLOP convention counts two real operations per multiply–accumulate and one per scalar add, subtract, multiply, divide, or nonlinear evaluation; comparisons and data movement count as zero. Registered convolution and matrix operators are counted directly, including the real-block convolutions used for widely-linear maps. Consequently, our FLOP values need not match MAC-style counts reported elsewhere, which count a multiply–accumulate as one. Source-level formulas additionally cover VisionAlphabet's D4 complex recurrences, gain and path-fusion kernels; TinyViM-S and Vim-Tiny's selective scans; Vim-Tiny's fused causal convolution and normalization; and the remaining models' normalization, activation, pooling, and interpolation. For one D4 pole at one spatial position, two horizontal and four vertical recurrences each require two complex products

and a complex addition, giving $6(2 \cdot 6 + 2) = 84$ real operations before gain and path fusion. A real selective scan with state width N is counted from its ΔA and ΔBu transforms, state update, and Cx readout, with skip, gate, and causal-convolution terms added when present. This is a reproducible *mathematical* operation count, not a count of physical GPU instructions: parallel scan implementations may execute extra work, and a transcendental evaluation is assigned one operation by convention. Latency, throughput, and peak allocated memory include all executed kernels and are the primary practical-efficiency evidence. For K128 specifically, the fastest batch-1 mode uses 114 MiB at 1.23 ms, whereas default compilation uses 52 MiB at 2.99 ms; at batch 16 these settings use 318/260 MiB and take 6.00/6.52 ms, respectively. Thus the fastest implementation is not also the lowest-memory implementation. The optimized K128 batch-1 path is 15.2 times faster than its eager implementation (18.79 ms), but remains slower than the optimized ConvNeXtV2-Atto path (0.42 ms).

B Operator Details

B.1 Associative scan composition

For $T_i(z) = a_i z + b_i$, affine scan elements compose as $(a_2, b_2) \circ (a_1, b_1) = (a_2 a_1,\ a_2 b_1 + b_2)$. This associative operation permits parallel scan evaluation (Martin & Cundy, 2018).

B.2 Reader, directional mixing, and Gated WL details

Fix one stage with input $E \in \mathbb{C}^{H \times W \times K}$ and P pole coordinates. The learned damping is bounded as

$$d_{j,p} = d_{\min} + (d_{\max} - d_{\min})\,\mathrm{sigmoid}(\eta_{j,p}), \qquad 0 < d_{\min} < d_{\max}. \tag{5}$$

The transition parameters are learned but input independent, and γ is derived from the same pole rather than learned as a separate gate.

Equation 1 shows the reader's channel–spatial factorization; the implemented reader additionally normalizes its synthesized kernel. We distinguish the synthesized kernel $\mathcal{W}^{\mathrm{raw}}_{p,k} = \sum_r D_{p,r} A_{p,r,k}$ from its normalized form $\mathcal{W}_{p,k}$ in this appendix. Let $\Delta \in \{-1, 0, 1\}^2$ index spatial offsets. Kernel normalization and zero-padded convolution (using the cross-correlation convention) are

$$\begin{aligned} \mathcal{W}_{p,k,\Delta} &= \frac{\mathcal{W}^{\mathrm{raw}}_{p,k,\Delta}}{\sqrt{\max(\sum_{k',\Delta'} |\mathcal{W}^{\mathrm{raw}}_{p,k',\Delta'}|^2, \epsilon)}}, \\ \bar{U}_p(y,x) &= \sum_{k,\Delta} \mathcal{W}_{p,k,\Delta} \widetilde{E}_k(y + \Delta_y, x + \Delta_x), \end{aligned} \tag{6}$$

with $\widetilde{E} = 0$ outside the grid. Here, $\widetilde{E} = E$ for stage-ending readers, whereas same-resolution readers use the complex-RMS-normalized input. The factorization bounds the rank of each pole's $K \times 9$ channel–spatial kernel, not the entire convolution. Stage-ending readers match output RMS to the input at each location $\mathbf{r} = (y, x)$:

$$\beta(\mathbf{r}) = \sqrt{\frac{K^{-1} \sum_k |E_k(\mathbf{r})|^2 + \epsilon}{P^{-1} \sum_p |\bar{U}_p(\mathbf{r})|^2 + \epsilon}}, \qquad U_p(\mathbf{r}) = \beta(\mathbf{r}) \bar{U}_p(\mathbf{r}). \tag{7}$$

Same-resolution readers instead normalize E using the complex RMS operator in Eq. 15, with its own learned scales, then apply the normalized-kernel convolution without output RMS matching, so $U = \bar{U}$.

Let $\mathcal{D} = ((+1,+1), (-1,+1), (+1,-1), (-1,-1))$ be the ordered direction list for stacking paths and flattening the terminal descriptor (with ascending pole index within each direction). For the horizontal scan, $(a_x^{(+1)}, \gamma_x^{(+1)}) = (a_x, \gamma_x)$ and $(a_x^{(-1)}, \gamma_x^{(-1)}) = (\bar{a}_x, \bar{\gamma}_x)$. The negative pass also reverses the column order. The vertical pass likewise uses $(a_y^{(+1)}, \gamma_y^{(+1)}) = (a_y, \gamma_y)$ in increasing row order and $(a_y^{(-1)}, \gamma_y^{(-1)}) = (\bar{a}_y, \bar{\gamma}_y)$ in decreasing row order. With u and v denoting row and column indices,

$$H^{s_x}_{u,v,p} = a^{(s_x)}_{x,p} H^{s_x}_{u,v-s_x,p} + \gamma^{(s_x)}_{x,p} U_{u,v,p}, \tag{8}$$

$$Z^{s_x,s_y}_{u,v,p} = a^{(s_y)}_{y,p} Z^{s_x,s_y}_{u-s_y,v,p} + \gamma^{(s_y)}_{y,p} H^{s_x}_{u,v,p}, \quad (s_x, s_y) \in \mathcal{D}. \tag{9}$$

Every recurrence starts from zero at its incoming boundary.

The scan carries a reference gain $v_t = |a|^2 v_{t-1} + |\gamma|^2 v_t^{\text{in}}$ from zero initialization. Horizontal input gain is one and vertical input gain is the corresponding horizontal gain. For direction $d = (s_x, s_y)$, pointwise normalization is

$$\widetilde{Z}_d(u, v, p) = \frac{Z_d(u, v, p)}{\sqrt{\max(\mathrm{sg}(v_d(u, v, p)), \epsilon)}}. \tag{10}$$

Here sg stops gradients through the reference gain.

For even H, W, direction-aware coarsening selects the downstream endpoint in each 2×2 cell:

$$M_{i,j,p}^{s_x,s_y} = \widetilde{Z}_{2i+o_y,\,2j+o_x,\,p}^{s_x,s_y}, \qquad o_x = \mathbf{1}[s_x = +1], \quad o_y = \mathbf{1}[s_y = +1]. \tag{11}$$

For each pole, the directional-fusion module receives $z_{i,j,p} = (M_{i,j,p}^{s_x,s_y})_{(s_x,s_y)\in\mathcal{D}} \in \mathbb{C}^4$ and computes

$$m_{i,j,p} = \mathrm{WL}_{p,2}(\phi_{\mathbb{C}}(\mathrm{WL}_{p,1}(z_{i,j,p}))), \qquad \phi_{\mathbb{C}}(z) = \mathrm{SiLU}(\Re z) + i\,\mathrm{SiLU}(\Im z), \tag{12}$$

using biased $4 \to 4 \to 1$ widely-linear maps $\mathrm{WL}(z) = Bz + \widetilde{B}\bar{z} + b$, with learned complex coefficients and spatially shared parameters. A complex projection $\Pi_M : \mathbb{C}^P \to \mathbb{C}^{K_{\text{out}}}$ maps the fused pole vector to the next feature width.

The carry aggregates the four incoming features in the same cell with learned real weights shared by real and imaginary components:

$$C_{i,j,k} = \sum_{a,b\in\{0,1\}} w_{k,a,b} E_{2i+a,2j+b,k}, \qquad w_{k,a,b}\big|_{\text{init}} = 1/4. \tag{13}$$

When the width changes, a complex map $\Pi_C : \mathbb{C}^K \to \mathbb{C}^{K_{\text{out}}}$ projects the carry. The merged field is

$$X_{i,j} = \Pi_M(m_{i,j}) + \Pi_C(C_{i,j}) \in \mathbb{C}^{K_{\text{out}}}, \tag{14}$$

with an absent width-changing projection interpreted as the identity.

For $K = K_{\text{out}}$, complex RMS normalization is

$$\widehat{X}_k = \rho_k X_k \left(K^{-1} \sum_{\ell=1}^{K} |X_\ell|^2 + \epsilon \right)^{-1/2}, \qquad \rho \in \mathbb{R}^K. \tag{15}$$

The Gated WL block in Eq. 2 uses hidden width $H_{\text{WL}} = 1.5K_{\text{out}}$. Value, output, and gate projections are bias-free. At matching widths, the carry projection is the identity; the memory projection may also be the identity according to the configuration.

Same-resolution blocks omit corner selection and fuse the normalized memories at each original grid location. With m denoting this fused pole vector, their update is $\mathrm{GatedWL}(E + \Pi_M(m))$ at output width K. For stage depth L_s, the first $L_s - 1$ blocks preserve resolution; the last block coarsens for stages 1–3 or supplies terminal energies for stage 4.

Spatial pole-energy separability argument. For $a = re^{i\varphi}$ with $0 < r < 1$, the Poisson kernel has the Fourier expansion

$$P_{re^{i\varphi}}(\omega) = \sum_{k\in\mathbb{Z}} r^{|k|} e^{ik(\omega-\varphi)}. \tag{16}$$

Let $\Delta = F - G$. The difference $\mathcal{Q}_\Delta(a_x, a_y)$ is real analytic in the pole parameters. If it vanishes on a positive-measure subset of $\mathbb{D}^2$, real-analytic uniqueness implies that it vanishes throughout the connected parameter domain. Expanding both kernels then gives the double Fourier–Stieltjes coefficients of Δ, each multiplied by $r_x^{|k|} r_y^{|\ell|}$. These multipliers are nonzero for all $(k, \ell) \in \mathbb{Z}^2$, so every coefficient of Δ vanishes and hence $F = G$.

For pairwise distinct $F_1, \ldots, F_C$, each difference $\mathcal{Q}_{F_i} - \mathcal{Q}_{F_j}$ is nonzero and real analytic. Its zero set has measure zero; a finite union over class pairs gives the generic-separation claim. The monotone $\log(1 + \cdot)$ map preserves distinct energies. Finally, any distinct population descriptor prototypes $\mu_1, \ldots, \mu_C$ are classified exactly at the prototypes by affine logits $L_c(q) = 2\mu_c^\top q - \|\mu_c\|_2^2$, since $L_i(\mu_i) - L_j(\mu_i) = \|\mu_i - \mu_j\|_2^2 > 0$.

C COCO TRANSFER PROTOCOL

All dense runs initialize from the corresponding ImageNet-1K seed-501, epoch-100 checkpoint. Downstream seeds 501, 509, and 521 vary fine-tuning randomness while keeping the pretrained checkpoint fixed. The original interfaces use backbone channel schedules $(256, 256, 256, 256)$ for VisionAlphabet-K128, $(192, 192, 192, 192)$ for VisionAlphabet-K96, $(40, 80, 160, 320)$ for ConvNeXtV2-Atto, and $(48, 64, 168, 224)$ for TinyViM-S. Backbones and task heads are fine-tuned jointly.

COCO2017. We use 118,287 training and 5,000 validation images. The common torchvision Mask R-CNN/FPN has 256-channel pyramid outputs, anchor sizes 32–512, aspect ratios $\{0.5, 1, 2\}$, box ROIAlign size 7, and mask ROIAlign size 14. The input short side is 800 with maximum long side 1333; divisibility padding is retained. Training uses AdamW at 10^{-4}, weight decay 0.05, 1,000-update warmup, and learning-rate drops by 0.1 at epochs 8 and 11. Twelve epochs give 88,716 optimizer updates at effective batch 16 (physical batch 2). Box and mask mAP use the official COCO evaluation machinery on all validation images, averaging AP over categories and the ten IoU thresholds from 0.50 to 0.95 in increments of 0.05. Models share the head implementation; their input FPN projections differ in size to accept each backbone's channels.

ParC-Net-S training setting. ParC-Net-S retains the common head, optimizer, effective batch, and 12-epoch schedule, while its nine global circular-convolution blocks run in FP32 (BF16 elsewhere). After epoch one, BN running statistics are recalibrated using training images and then fixed; BN affine parameters remain trainable. The same setting is used for all three downstream seeds. Their box/mask APs are 33.71/32.31 (501), 33.96/32.51 (509), and 34.08/32.60 (521), with aggregate results in Table 2.

D DETAILED IMAGENET-C RESULTS

Table 5 reports accuracy by corruption type and AlexNet-normalized mCE for the nine models in Figure 6, following the ImageNet-C benchmark (Hendrycks & Dietterich, 2019).

Table 5: Official ImageNet-C by corruption type. Top-1 accuracy (percent) is averaged over five severities; the final rows give mean accuracy across 15 types and AlexNet-normalized mCE (lower is better). Each severity contains 50,000 images. Bold and underline denote the best and second-best displayed models in each row, respectively.

Corruption	VA-K128	ParC-Net-XS	ParC-Net-S	ConvNeXtV2-Atto	EMOv2-1M	TinyNeXt-T	TinyViM-S	EfficientViM-M1	Vim-Tiny
Gaussian Noise	<u>47.35</u>	34.68	43.61	42.77	32.93	32.87	46.67	41.00	**49.10**
Shot Noise	<u>46.48</u>	33.50	41.94	41.55	31.71	31.25	45.10	40.04	**47.84**
Impulse Noise	<u>45.37</u>	29.96	39.39	40.65	31.35	31.07	43.72	38.97	**47.44**
Defocus Blur	<u>36.59</u>	27.35	33.36	34.98	26.44	24.97	35.91	32.50	**39.07**
Glass Blur	25.97	20.40	24.01	<u>30.24</u>	20.70	19.84	26.54	25.57	**32.88**
Motion Blur	42.54	32.59	38.95	**42.81**	31.29	30.20	41.88	37.90	<u>42.62</u>
Zoom Blur	<u>33.29</u>	27.17	31.46	32.05	25.62	25.68	**34.24**	27.52	31.45
Snow	<u>47.15</u>	38.61	45.76	45.64	35.98	33.65	**47.48**	40.20	44.36
Frost	**51.66**	41.94	48.97	48.17	41.73	37.93	<u>50.98</u>	44.90	49.58
Fog	48.13	41.61	<u>48.90</u>	46.85	40.61	42.63	**52.24**	43.29	45.72
Brightness	**68.63**	60.52	65.69	65.86	59.80	57.84	<u>68.27</u>	62.66	64.88
Contrast	<u>55.46</u>	47.74	55.02	53.45	45.66	44.09	**58.71**	51.60	54.82
Elastic Transform	39.77	32.68	37.83	42.66	36.02	34.95	43.26	<u>45.06</u>	**48.37**
Pixelate	<u>44.18</u>	28.75	41.25	34.57	30.35	27.05	36.83	40.02	**47.43**
Jpeg Compression	<u>54.91</u>	46.26	53.53	54.74	42.90	41.24	**56.03**	50.17	54.13
Mean (15 types)	45.83	36.25	43.31	43.80	35.54	34.35	<u>45.86</u>	41.43	**46.65**
mCE (↓)	69.19	81.43	72.39	71.71	82.31	83.92	<u>69.13</u>	74.69	**68.16**