VIS: VISION STATE-SPACE TOKEN MIXER

Anonymous authors
Paper under double-blind review

ABSTRACT

State-space models (SSMs) bring linear-complexity sequence modeling to vision, but only after the image is flattened into a 1D sequence under a handcrafted scan order, a causal bias that multi-directional scanning mitigates rather than removes. In contrast, we view SSMs from the perspective of token mixing: each token can aggregate information from other tokens by iteratively updating its own hidden state, as in an SSM recurrence. This perspective suggests that what must be defined is not a single global scan order, but a principled order in which each token aggregates its context. We introduce the *Vision State-Space Token Mixer* (ViS), which redefines the scan axis as spatial distance: every token's state absorbs information from concentric rings around it, yielding a scan-order-free token mixer with a global receptive field. Despite the apparent difficulty of parallelizing such position-wise recurrences, we derive an exact closed-form formulation as structured masked attention, enabling efficient implementation on modern attention-friendly hardware. Experiments across image classification, segmentation, object detection, and generation demonstrate ViS as a viable general-purpose vision backbone; ViS achieves up to $5.1\times$ higher throughput while maintaining comparable or better performance, and exhibits the strongest zero-shot resolution transfer among vision backbones through its coordinate-computed spatial prior.

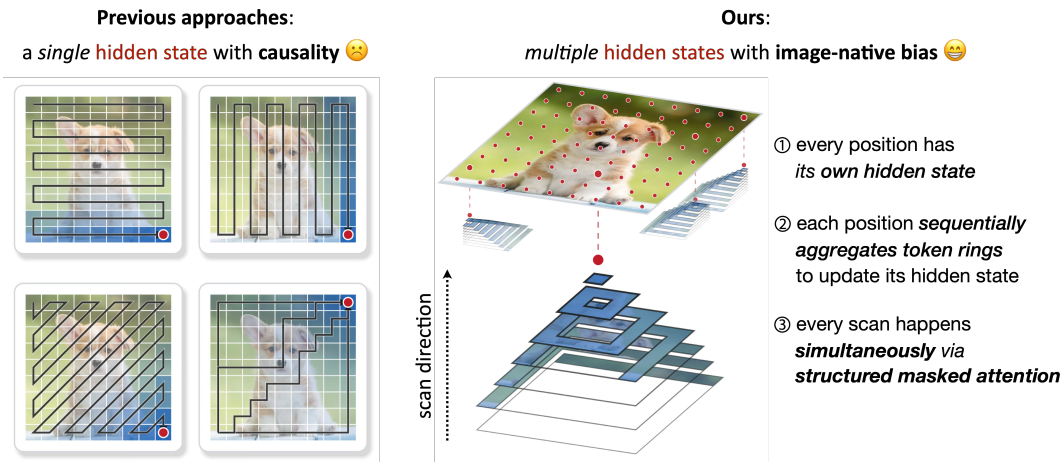

Figure 1: **From scan-order bias to image-native state spaces.** Previous vision SSMs flatten images into a single sequence and impose a scan-order bias. We introduce *Vision State-Space Token Mixer* (ViS), which maintains an independent hidden state at every spatial position and sequentially aggregates its nearby context. The resulting 2D recurrence admits an attention-like formulation that is highly efficient on modern hardware.

1 INTRODUCTION

The token mixer defines a vision backbone: it decides which pixels may influence which, at what cost, and under what prior. Attention (Vaswani et al., 2017) performs global, content-adaptive mixing at a quadratic cost in the number of tokens, based on the prior that comparing similarities across all pairs of tokens is effective. State-space models (SSMs), on the other hand, mix tokens sequentially by compressing information from previously visited tokens into a fixed-size state, which has turned

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.

out to recover much of the Transformer's quality at sub-quadratic cost in the vision domain (Liu et al., 2024; Zhu et al., 2024; Chen et al., 2024; Liang et al., 2024). Despite their effectiveness, extending SSMs to images has always faced a question that does not exist in 1D: *in what order should a recurrence visit a 2D grid?*

Every existing vision SSM answers this question with different scanning strategies: raster scans (Zhu et al., 2024), windowed scans (Huang et al., 2024), multi-scale scans (Shi et al., 2024), content-adaptive scans (Li et al., 2026), and multi-directional scans (Shaker et al., 2025; Xiao et al., 2025). While these designs provide empirical answers to how SSMs can effectively encode visual data, they all rely on heuristic choices rather than a principled foundation that can be systematically extended or improved, thereby limiting their potential to serve as a general-purpose token mixer.

In contrast, this paper aims to bring the core idea of SSMs to the vision domain through a more principled approach. Our observation is that the inductive prior of SSM lies not in a particular scan order, but in its mechanism for accumulating context into a compressed state through a consistent update rule. From the perspective of the token mixer, where every token should be able to aggregate global context, this suggests that every token should maintain a hidden state that systematically accumulates contextual information. Consequently, the recurrence should be instantiated for *every token*, rather than following a single predefined scan over the image.

We therefore introduce *Vision State-Space Token Mixer* (ViS), where each token runs its own recurrence in a parallel manner. Specifically, every token maintains its own hidden state, and that state evolves by absorbing the concentric neighboring rings around the token. Although performing such position-wise iterative updates across all spatial locations in parallel may appear overly complex, we find that they can be realized through a simple operation of structured masked attention within the State Space Duality (SSD) framework (Dao & Gu, 2024), under the scalar-times-identity parameterization of the state transition matrix $\mathbf{A}$. This allows our framework to leverage modern hardware optimized for general matrix multiplications (GEMMs) while preserving its geometric prior, retaining the benefits of an image-native SSM without the overhead of sequential, multi-directional scanning required by prior vision SSMs. As a result, this formulation enables ViS to achieve higher efficiency than Mamba-based vision models (Zhu et al., 2024; Shi et al., 2024; Huang et al., 2024), despite their use of production-level Mamba kernels (Gu & Dao, 2024; Dao & Gu, 2024). We further provide fused GPU kernels for our computation; code will be released.

Consequently, ViS achieves state-of-the-art or on-par performance with higher computational efficiency, demonstrating that its image-native geometric prior can be retained without sacrificing practical speed. We evaluate ViS as a general-purpose backbone across image classification, object detection, segmentation, and class-conditional image generation. Across these diverse tasks, ViS consistently maintains a favorable accuracy–efficiency trade-off, achieving comparable or better performance while boosting throughput by up to $5.1\times$. Its image-native design also exhibits stronger zero-shot resolution adaptivity, indicating that the spatial prior of ViS is inherently tied to image structure rather than a fixed scan order or resolution.

Our contributions can be summarized as follows:

- We introduce the *Vision State-Space Token Mixer* (ViS), a novel 2D state-space operator whose recurrence happens at every position and runs over spatial distance rather than sequence position, removing the scan-order bias at its root instead of ensembling over it (Sec. 3).
- We derive an exact closed form for the recurrence as structured masked attention under a low-rank mask, extending the SSD from causal sequences to spatial grids (Sec. 3.3).
- We develop the operator and its surrounding components to enable its effective integration into modern vision backbones (Sec. 3.4 – 3.5).
- We demonstrate that ViS supports a general-purpose backbone across classification, segmentation, detection, and generation, with resolution elasticity by construction (Sec. 4).

2 PRELIMINARIES

SSMs An SSM transforms a length-T sequence $\mathbf{x} = [x_0\ x_1 \cdots\ x_{T-1}]^\top$ to another length-T sequence $\mathbf{y} = [y_0\ y_1\ \cdots\ y_{T-1}]^\top$ through following recurrences:

(LTI SSM)
$$\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1} + \mathbf{B}x_t, \quad (1a)$$
$$y_t = \mathbf{C}^\top \mathbf{h}_t, \quad (1b)$$

(Selective SSM)
$$\mathbf{h}_t = \mathbf{A}_t\mathbf{h}_{t-1} + \mathbf{B}_t x_t, \quad (2a)$$
$$y_t = \mathbf{C}_t^\top \mathbf{h}_t, \quad (2b)$$

where $\mathbf{h} \in \mathbb{R}^N$ is a hidden state, $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a discretized state transition matrix, and $\{\mathbf{B}, \mathbf{C}\} \subset \mathbb{R}^{N\times 1}$ are projection weights in linear time invariant (LTI) SSM of Eq. (1). Mamba (Gu & Dao, 2024) adopts input-dependent SSM parameters that vary along the recurrence to achieve selectivity, resulting in the time-varying recurrent update in Eq. (2).

State-Space Duality Mamba-2 (Dao & Gu, 2024) observes that unrolling this recurrence yields an equivalent formulation as *structured masked attention*, when $\mathbf{A}_t$'s are constrained to be a scalar multiple of the identity, *i.e.*, $\mathbf{A}_t := a_t\mathbf{I}$ where $(a_0, a_1, \ldots, a_{T-1}) \in \{1\} \times \mathbb{R}^{T-1}$:

$$y_0 = \mathbf{C}_0^\top \mathbf{B}_0 x_0, \quad y_1 = \mathbf{C}_1^\top \mathbf{A}_1 \mathbf{B}_0 x_0 + \mathbf{C}_1^\top \mathbf{B}_1 x_1, \quad \cdots, \quad y_{T-1} = \mathbf{C}_{T-1}^\top \sum_{t=0}^{T-1} \mathbf{A}_{T-1:t+1} \mathbf{B}_t x_t.$$

$$\iff \mathbf{y} = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{T-1:1} & a_{T-1:2} & \cdots & 1 \end{bmatrix}}_{:=L} \circ \begin{bmatrix} — \mathbf{C}_0^\top — \\ — \mathbf{C}_1^\top — \\ \vdots \\ — \mathbf{C}_{T-1}^\top — \end{bmatrix} \begin{bmatrix} | & | & & | \\ \mathbf{B}_0 & \mathbf{B}_1 & \cdots & \mathbf{B}_{T-1} \\ | & | & & | \end{bmatrix} \mathbf{x} := M\mathbf{x}. \quad (3)$$

Here, the SSM recurrence is realized in a form of matrix multiplication: $\mathbf{y} = M\mathbf{x}$, which we refer to as its *quadratic* form, in contrast to its recurrent form in Eq. (2). The interchangeability between the two forms, which we call *State-Space Duality* (SSD), holds if the matrix L has a structure that enables subquadratic computation of the matrix multiplication by L. This connects to the notion of *structured masked attention*, where the attention weights are modulated by a structured mask L: $\mathbf{y} = \big(L \circ (QK^\top)\big)V$. Dao & Gu (2024) show that this attention can be computed subquadratically by reordering the tensor contractions from (Q, K, L, V) to (V, K, L, Q) via exploitation of the low-rank structure of L, and realize that this reordering is equivalent to how the two forms in SSD allow different ways of evaluating the same underlying SSM recurrence. In later sections, we show that our formulation of 2D SSM also coincides with the SSD framework, allowing our algorithm to leverage matrix multiplications that are well suited to modern GPUs.

3 Vision State-Space Token Mixer

In this section we present *Vision State-Space Token Mixer* (ViS), a 2D state-space operator in which every token runs its own recurrence over the concentric rings around it, so that spatial distance rather than a scan order serves as the recurrence axis. We first fix the notation for tokens, distances, and rings on a 2D grid (Sec. 3.1), and then define the per-token recurrence that consumes the rings from the farthest to the nearest (Sec. 3.2). Under the scalar-times-identity parameterization of the state transition matrix, we show that this recurrence admits an exact closed form as structured masked attention with a low-rank mask, which places ViS within the SSD framework and makes it computable with matrix multiplications (Sec. 3.3). Since the resulting operator is isotropic, we introduce two lightweight touches, a sector gain on the mask and a rotary embedding on the projections, that restore directional information without altering the recurrence or its closed form (Sec. 3.4). Finally, we describe how the operator is assembled into a block and a hierarchical backbone (Sec. 3.5).

3.1 Notations

Token index We consider a token mixer operating on an $H \times W$ token grid, where the output tokens $\{y_i\}_{i=1}^{HW}$ are defined as a function of the input tokens $\{x_j\}_{j=1}^{HW}$:

$$\mathrm{ViS}(\cdot) : \mathbb{R}^{HW} \to \mathbb{R}^{HW}, \qquad \mathbf{x} \mapsto \mathbf{y}. \quad (4)$$

Here, we assume scalar-valued tokens x_j and y_i to keep the notation coherent with Eq. (3). In practice, the tokens are D-dimensional vectors, which can be processed channel-wise using D independent mixers with shared SSM parameters, following Mamba-2 (Dao & Gu, 2024). We use i and j to index output and input tokens, respectively. Since each token is associated with a spatial position, we denote the position of the i-th token by $\boldsymbol{\rho}(i) = (\rho_1(i), \rho_2(i)) \in [0, H-1] \times [0, W-1]$.

Distance and ring We define the Chebyshev distance between two tokens, say, the k-th and l-th tokens, as $d(k, l) = \max\big(|\rho_1(k) - \rho_1(l)|, |\rho_2(k) - \rho_2(l)|\big)$, under which the set of tokens at distance t from k-th token forms a ring $\mathcal{R}_t(k) = \{l | d(k, l) = t\}$. Note that a ring may be truncated, as

illustrated in Fig. 1, when part of the ring extends beyond the boundaries of the given token grid. The maximum possible radius of a ring is $\max(H, W) - 1$, attained when the corresponding token is located at a corner of the grid. For brevity, we denote this maximum radius by T.

3.2 A RECURRENCE OVER RINGS

ViS runs HW hidden states across all spatial positions, which we denote as $\{\mathbf{h}_i\}_{i=1}^{HW} \subset \mathbb{R}^N$. The i-th token's hidden state is updated by the following recurrence over $t = 0, 1, \ldots, T$:

$$(\mathbf{h}_i)_t \;=\; \mathbf{A}_i\,(\mathbf{h}_i)_{t-1} \;+\; \frac{1}{|\mathcal{R}_{T-t}(i)|}\sum_{j\in\mathcal{R}_{T-t}(i)} \mathbf{B}_j\,x_j, \qquad (\mathbf{h}_i)_{-1} = 0, \qquad y_i \;=\; \mathbf{C}_i^\top (\mathbf{h}_i)_T. \quad (5)$$

For hidden states away from the corners, $\mathcal{R}_{T-t}(i)$ can be empty for the initial steps. In such cases, we skip the update by defining the summation term to be zero, *i.e.*, $\frac{1}{|\mathcal{R}_{T-t}(i)|}\sum_{j\in\mathcal{R}_{T-t}(i)} \mathbf{B}_j x_j = 0$ when $\mathcal{R}_{T-t}(i) = \varnothing$. The state absorbs the mean of each Chebyshev ring in turn, farthest first, and is read out once all rings are consumed. Three design choices matter here. First, $\mathbf{A}_i$, $\mathbf{B}_j$, and $\mathbf{C}_i$ are *pixel-local* rather than step-varying. This distinction stems from ViS explicitly separating the temporal and spatial axis: the hidden state evolves over temporal axis at each spatial location, *i.e.*, $(\mathbf{h}_i)_t$, whereas previous methods identify temporal axis *with* spatial coordinates, *i.e.*, $\mathbf{h}_t$ where there exists a mapping $t \mapsto i$ that determines the spatial location visited at each step. Second, the $1/|\mathcal{R}_{T-t}(i)|$ factor is a boundary-clipped ring mean: since the ring size varies substantially with the time step t, we normalize each ring to have even contribution independent of its size. Third, the hidden state consumes the *farthest ring first*, causing information from farther rings to decay more than those from closer rings, as the recurrence proceeds. This better reflects the spatial inductive bias of images, where the representation at the i-th position is typically more strongly correlated with nearby positions. We provide empirical ablations on this order in Appendix D.

3.3 CLOSED FORM: EQUIVALENCE TO STRUCTURED MASKED ATTENTION

If we constrain $\mathbf{A}_i$ to be a scalar multiple of the identity, unrolling Eq. (5) provides a matrix multiplication form, allowing the simultaneous computation of the recurrence across all spatial locations.

Proposition 3.1 (Equivalence to structured masked attention)**.** *Let $\mathbf{A}_i = a_i\mathbf{I} \in \mathbb{R}^{N\times N}$. Then the proposed SSM recurrence of Eq. (5) admits the following equivalent closed-form formulation as structured masked attention:*

$$\mathbf{y} = L \circ \begin{bmatrix} — \mathbf{C}_0^\top — \\ — \mathbf{C}_1^\top — \\ \vdots \\ — \mathbf{C}_{HW-1}^\top — \end{bmatrix} \begin{bmatrix} | & | & & | \\ \mathbf{B}_0 & \mathbf{B}_1 & \cdots & \mathbf{B}_{HW-1} \\ | & | & & | \end{bmatrix} \mathbf{x} := M\mathbf{x}, \quad (6)$$

$$\textit{where} \qquad L_{i,j} = \frac{a_i^{d(i,j)}}{|\mathcal{R}_{d(i,j)}(i)|}, \qquad M_{i,j} = \frac{a_i^{d(i,j)}\mathbf{C}_i^\top\mathbf{B}_j}{|\mathcal{R}_{d(i,j)}(i)|}. \quad (7)$$

The derived low-rank mask L also allows subquadratic contraction introduced in SSD:

Corollary 3.1 (Two contraction orders)**.** *Let $Q, K \in \mathbb{R}^{HW\times N}$ and $V \in \mathbb{R}^{HW}$ be arbitrary, and let L be the mask in Eq. (7). Then the structured masked attention $\mathbf{y} = \big(L \circ (QK^\top)\big)V$ admits two contraction orders that evaluate to the same $\mathbf{y}$: the order (Q, K, L, V), which forms $QK^\top$ first and costs $O\big((HW)^2N\big)$, and the order (V, K, L, Q), which contracts V with K first and costs $O\big(HW \cdot \max(H, W) \cdot N\big)$.*

We provide the proofs in Appendix B. Setting $(Q_i, K_j, V) = (\mathbf{C}_i, \mathbf{B}_j, \mathbf{x})$ recovers the two forms of ViS: the order (Q, K, L, V) is the closed form of Proposition 3.1, whereas the order (V, K, L, Q) is the recurrence in Eq. (5). We visualize the mask L in Fig. 2. Its structure reveals an innate spatial prior: the mask exhibits a regular pattern over (W, W)-sized blocks with point symmetry, with the highest values concentrated along the diagonal. This indicates that positions symmetric about the center of the spatial grid share the same mask, while each position i contributes most strongly to its corresponding output y_i. Note that conventional scanning methods induce causal masks (Dao & Gu, 2024), where each token attends only to a subset of the spatial domain determined by the scan order. In contrast, our formulation allows tokens to incorporate information from the entire spatial domain.

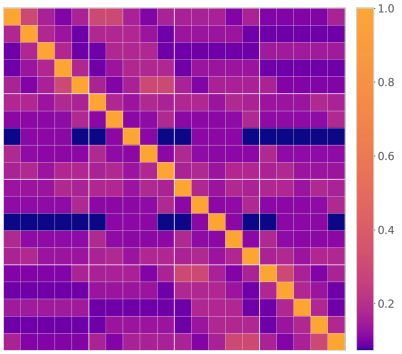

Figure 2: **The structure of the mask L.** We set $(H, W) = (4, 5)$ and $A_i = 0.9$ for visualization.

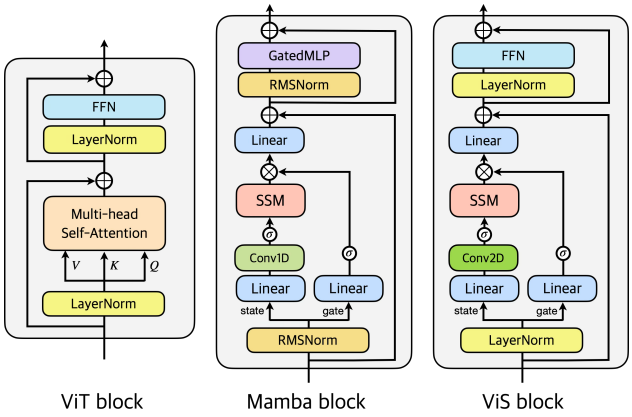

Figure 3: **ViS block architecture.** We adopt vision- and SSM-related components from ViT and Mamba block.

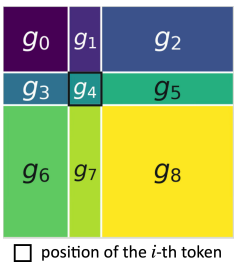

Figure 4: **Sector gain assignment $g_{s(i,j)}$.**

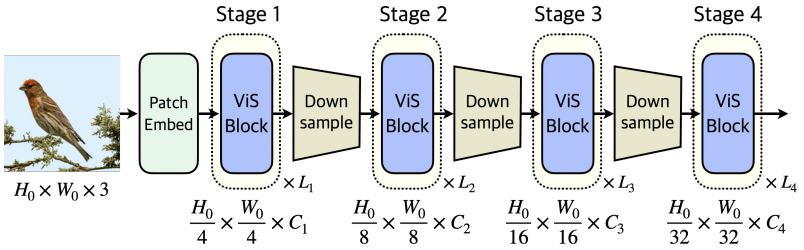

Figure 5: **Macro architecture of ViS.** We adopt 4-stage hierarchical architecture with a downsampling layer halving the token grid between stages.

3.4 Breaking Isotropy

The mask L in Eq. (7) depends on the pair (i, j) only through the distance $d(i, j)$, so all tokens on the same ring around the i-th token receive an identical weight from the mask L. The content term $\mathbf{C}_i^\top \mathbf{B}_j$ does not resolve this ambiguity between tokens from the same ring: $\mathbf{B}_j$ and $\mathbf{C}_i$ are projections of the token contents, and their inner product carries no explicit information about where the j-th token lies relative to the i-th. ViS is therefore *isotropic*: tokens at the same distance from the i-th token but in different directions are treated identically when contributing to y_i. This isotropy eliminates scan-order bias, but is arguably too strong a spatial prior for token mixers, where the directional offset between two tokens can be as informative as their distance. In conventional SSMs, the recurrence itself provides the inductive structure along the sequence axis, without requiring positional information to be explicitly added to the input tokens. Following the same principle, we seek to incorporate directional awareness directly into the 2D SSM operator, rather than introducing it through an external positional encoding. We therefore add two lightweight touches that restore directional information while leaving the recurrence in Eq. (5) and the closed form in Sec. 3.3 intact. The first rescales the geometric term L coarsely and independently of content, and the second makes the content term $\mathbf{C}_i^\top$ and $\mathbf{B}_j$ depend on the signed offset between two tokens. Both do not rely on the spatial grid size H and W, so a model trained at one resolution applies to any other without interpolation or retabulation (Sec. 4.1).

Sector gain We partition the j-th tokens around the i-th token into eight sectors according to the direction (Fig. 4), assign eight learnable scalars $\{g_{s(i,j)}\} = \{g_0, g_1, \ldots, g_8\} \setminus \{g_4\}$, and rescale the mask elementwise as

$$\tilde{L}_{i,j} = e^{\gamma_{i,j}} L_{i,j}, \qquad \gamma_{i,j} = \begin{cases} g_{s(i,j)} & \text{if } i \neq j, \\ 0 & \text{if } i = j, \end{cases} \tag{8}$$

In the recurrent form, Eq. (8) amounts to weighting each summand $\mathbf{B}_j x_j$ in the ring sum of Eq. (5) by $e^{\gamma_{i,j}}$ before it enters the state, so the recurrence and the closed form remain interchangeable. All

$g_{s(i,j)}$ are initialized to zero for each head, so the operator starts from the isotropic formulation as in Sec. 3.3, while each head may learn to emphasize or suppress a direction during training. The cost is eight scalars per head in each block, a negligible fraction of the model's parameters, and essentially no additional computation, since the gain enters only as an elementwise factor on the mask.

Rotary embedding on B and C The sector gain is coarse, with eight directions, and content-agnostic. Direction must therefore enter through the matrix M (Eq. (7)), which determines the contribution of the j-th input token to the i-th output token. Specifically, we inject the geometric relationship between these tokens by manipulating the terms $\mathbf{C}_i^\top$ and $\mathbf{B}_j$, resembling how RoPE (Su et al., 2024) is applied to query and key vectors to consider their relative position. Note that $\mathbf{B}_j$ and $\mathbf{C}_i$ respectively act as *key* and *query* of attention (Sec. 3.3), in that they are the outputs attained by input projection and the term $\mathbf{C}_i^\top \mathbf{B}_j$ accounts for the j-th token's contribution to the i-th token. We thus apply planar rotations to $\mathbf{B}_j$ and $\mathbf{C}_i$ depending on their token coordinates, injecting the relative position prior between the i-th and j-th tokens. Concretely, for a token at position i, let $\mathbf{U}_i \in \mathbb{R}^{N\times N}$ be a block-diagonal matrix whose 2×2 blocks apply planar rotations to pairs of state channels. We partition the blocks to coordinate the two spatial axes, such that each block is rotated according to either $\rho_1(i)$ or $\rho_2(i)$:

$$\mathbf{U}_i = \mathrm{diag}\Big(R\big(\rho_1(i)\theta_1\big),\dots,R\big(\rho_1(i)\theta_K\big),R\big(\rho_2(i)\theta_1\big),\dots,R\big(\rho_2(i)\theta_K\big)\Big),\ R(\phi)=\begin{bmatrix}\cos\phi & -\sin\phi\\ \sin\phi & \cos\phi\end{bmatrix}.$$

Here, $\{\theta_k\}_{k=1}^K$ follows the geometric frequency schedule of RoPE (Su et al., 2024). We then apply the rotation matrix to $\mathbf{B}_j$ and $\mathbf{C}_i$ as follows:

$$\tilde{\mathbf{B}}_j = \mathbf{U}_j\mathbf{B}_j, \qquad \tilde{\mathbf{C}}_i = \mathbf{U}_i\mathbf{C}_i, \qquad \tilde{\mathbf{C}}_i^\top\tilde{\mathbf{B}}_j = \mathbf{C}_i^\top\mathbf{U}_i^\top\mathbf{U}_j\mathbf{B}_j. \tag{9}$$

Note that the relative rotation $\mathbf{U}_i^\top\mathbf{U}_j$ depends only on the signed spatial offset between the two tokens, as $R(\phi)^\top = R(-\phi)$ and rotations in the same plane compose by adding their angles:

$$\mathbf{U}_i^\top\mathbf{U}_j = \mathrm{diag}\Bigg(R\Big(\big(\rho_1(j)-\rho_1(i)\big)\theta_1\Big),\dots,R\Big(\big(\rho_1(j)-\rho_1(i)\big)\theta_K\Big),\\ R\Big(\big(\rho_2(j)-\rho_2(i)\big)\theta_1\Big),\dots,R\Big(\big(\rho_2(j)-\rho_2(i)\big)\theta_K\Big)\Bigg), \tag{10}$$

which makes $\mathbf{U}_i^\top\mathbf{U}_j$ a block-diagonal rotation with angles $\big(\rho_1(j)-\rho_1(i)\big)\theta_k$ and $\big(\rho_2(j)-\rho_2(i)\big)\theta_k$. Hence, the $\tilde{\mathbf{C}}_i^\top\tilde{\mathbf{B}}_j$ term in Eq. (9) becomes a function of the two tokens' contents and of their signed offset, so tokens with identical content on the same ring are no longer interchangeable. Substituting $\tilde{\mathbf{B}}_j$ and $\tilde{\mathbf{C}}_i$ for $\mathbf{B}_j$ and $\mathbf{C}_i$ leaves Eq. (5) and the closed form in Sec. 3.3 unchanged, and the rotation adds only a cost linear in the number of tokens.

3.5 Architecture

Micro architecture We illustrate the ViS block architecture in Fig. 3, compared to the standard ViT (Dosovitskiy et al., 2021) and Mamba (Gu & Dao, 2024) block designs. The ViS block adopts vision-specific design choices from ViT and SSM-specific choices from Mamba. Specifically, input parameter projection, gating mechanism, and output projection layers are adopted from those of Mamba's, but with one modification: causal 1D convolution before the actual token mixer block is replaced with 2D convolutional block, as ours does not impose such causality among the hidden states across different spatial positions. For the normalization and FFN blocks following the token-mixing operation, we empirically adopt LayerNorm and FFN over RMSNorm (Zhang & Sennrich, 2019) and GatedMLP, respectively; we ablate the effect of these choices in Appendix D. When blocks are stacked, the output tokens $\{y_i\}$ of one block become the input tokens $\{x_j\}$ of the next, so each block applies the ring recurrence to tokens that already carry context from the previous one.

Macro architecture We follow the conventional four-stage hierarchical architecture adopted in previous studies (Liu et al., 2021; Wang et al., 2021), as illustrated in Fig. 5: an RGB image of shape $(H_0, W_0, 3)$ is first patchified into $\frac{H_0}{4}\times\frac{W_0}{4}$ tokens, and the spatial resolution is halved at the beginning of each subsequent stage. A task-adaptive head is attached to the final stage, with its specific design determined by the downstream task: for image classification, we apply global mean pooling followed by a linear classifier; for semantic segmentation, we use UPerNet (Xiao et al., 2018) with an auxiliary FCN head (Zhao et al., 2017); and for object detection and instance segmentation, we employ Mask R-CNN (He et al., 2017) with a Feature Pyramid Network (FPN) (Lin et al., 2017).

Table 1: **ImageNet-1k classification at 224^2.** Trained for 300 epochs. Throughput ('Tput') in images/s and peak memory ('mem.') in GB.

Method	# Par.	FLOPs	Top-1	Tput	Mem.
Transformer-, Convolution-based					
Swin-T (Liu et al., 2021)	28M	4.5G	81.3	4929	1.69
ConvNeXt-T (Liu et al., 2022)	29M	4.5G	82.1	4265	1.07
MambaOut-T (Yu & Wang, 2025)	27M	4.5G	82.9	4975	1.66
SSM-based					
Vim-S (Zhu et al., 2024)	26M	5.1G	80.5	2263	**0.73**
VMamba-T (Liu et al., 2024)	30M	4.9G	82.5	2820	2.34
LocalVMamba-T (Huang et al., 2024)	26M	5.7G	82.7	650	7.55
MSVMamba-T (Shi et al., 2024)	33M	4.6G	82.8	1565	2.76
Spatial-Mamba-T (Xiao et al., 2025)	27M	4.5G	83.5	2365	1.60
VSSD-T (Shi et al., 2025)	24M	4.5G	83.7	2250	1.78
DAMamba-T (Li et al., 2026)	26M	4.8G	83.8	2120	1.10
ViS-T (Ours)	28M	5.3G	**84.1**	**3114**	1.36
Transformer-, Convolution-based					
Swin-S (Liu et al., 2021)	50M	8.7G	83.0	3167	1.77
ConvNeXt-S (Liu et al., 2022)	50M	8.7G	83.1	2750	1.16
MambaOut-S (Yu & Wang, 2025)	48M	9.0G	84.1	3279	1.75
SSM-based					
VMamba-S (Liu et al., 2024)	50M	8.7G	83.6	1518	4.06
LocalVMamba-S (Huang et al., 2024)	50M	11.4G	83.7	373	7.64
Spatial-Mamba-S (Xiao et al., 2025)	43M	7.1G	84.6	1638	1.66
VSSD-S (Shi et al., 2025)	40M	7.4G	84.1	1426	1.85
DAMamba-S (Li et al., 2026)	45M	10.3G	**84.8**	1184	**1.36**
ViS-S (Ours)	53M	9.1G	84.7	**1885**	1.44
Transformer-, Convolution-based					
Swin-B (Liu et al., 2021)	88M	15.4G	83.5	2490	2.41
ConvNeXt-B (Liu et al., 2022)	89M	15.4G	83.8	2117	1.58
MambaOut-B (Yu & Wang, 2025)	85M	15.8G	84.2	2020	2.37
SSM-based					
Vim-B (Zhu et al., 2024)	98M	20.9G	83.2	365	4.15
VMamba-B (Liu et al., 2024)	89M	15.4G	83.9	1202	5.45
Spatial-Mamba-B (Xiao et al., 2025)	96M	15.8G	85.3	1132	2.57
VSSD-B (Shi et al., 2025)	89M	16.1G	84.7	993	2.63
DAMamba-B (Li et al., 2026)	86M	16.3G	85.2	930	**1.69**
ViS-B (Ours)	90M	16.6G	**85.4**	**1398**	2.15

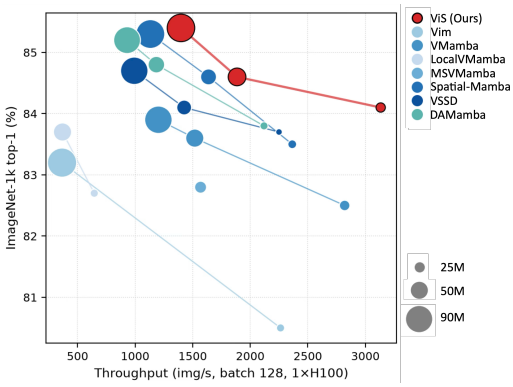

Figure 6: **Throughput-accuracy plot.** The GPU-friendly formulation of ViS enables higher speed while achieving competitive accuracy.

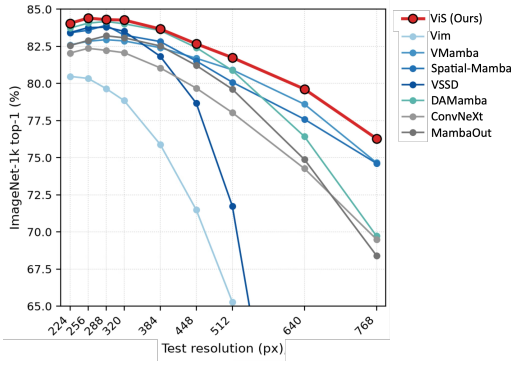

Figure 7: **Zero-shot resolution transfer of 224^2-trained weights.**

4 Experiments

In this section, we evaluate ViS as a general-purpose vision backbone on image classification, object detection, instance segmentation, semantic segmentation, and class-conditional image generation. We then examine its zero-shot transfer to unseen resolutions (Sec. 4.1), compare the complexity of its quadratic and recurrent forms (Sec. 4.2), and qualitatively analyze its learned representations (Sec. 4.3). Due to space constraints, we provide ablation studies in Appendix D.

ImageNet-1k classification We train ViS at three scales on ImageNet-1k (Deng et al., 2009) at 224^2 for 300 epochs following DAMamba (Li et al., 2026), and measure throughput (images/s) and peak memory (GB) on a single H100 GPU at a batch size of 128 (Table 1). ViS-T and ViS-B achieve the highest top-1 accuracy among all compared models at their scales, and ViS-S comes within 0.1%p of DAMamba-S. Among SSM-based backbones, which trade throughput for accuracy relative to Transformer- and convolution-based ones, ViS runs $1.1 \sim 5.1\times$ faster (Fig. 6) and uses $1.1 \sim 5.6\times$ less memory than all but DAMamba and Vim-S. We attribute both to its GEMM-friendly closed form (Sec. 3.3), which parallelizes simultaneous scans with a single matrix multiplication.

COCO object detection and instance segmentation We train Mask R-CNN (He et al., 2017) with FPN (Lin et al., 2017) over the four stages on COCO (Lin et al., 2014) under the $1\times$ schedule using MMDetection (Chen et al., 2019), initializing the backbone with the ImageNet-1k pretrained weights (Table 2). ViS-T and ViS-B outperform all compared backbones on every metric, and ViS-S achieves the highest AP^b_{50}, AP^m, and AP^m_{50} among the small backbones while matching the best AP^b. The margins are larger for instance segmentation than for detection, suggesting that the spatial prior of ViS particularly benefits dense, pixel-level prediction.

ADE20K semantic segmentation Following the standard protocol (Liu et al., 2021; 2024), we train UPerNet (Xiao et al., 2018) with an auxiliary FCN head (Zhao et al., 2017) on ADE20K for 160K iterations on 512^2 crops, initialized from the ImageNet-1k weights (Table 3). ViS-S achieves the

Table 2: **COCO object detection and instance segmentation.** Mask R-CNN attached, 1× schedule.

Backbone	# Par.	AP^b	AP^b_{50}	AP^m	AP^m_{50}
Swin-T (Liu et al., 2021)	48M	42.7	65.2	39.3	62.2
ConvNeXt-T (Liu et al., 2022)	48M	44.2	66.6	40.1	63.3
VMamba-T (Liu et al., 2024)	50M	47.3	69.3	42.7	66.4
LocalVMamba-T (Huang et al., 2024)	45M	46.7	68.7	42.2	65.7
MSVMamba-T (Shi et al., 2024)	53M	46.9	68.8	42.2	65.6
GroupMamba-T (Shaker et al., 2025)	40M	47.6	69.8	42.9	66.5
VSSD-T (Shi et al., 2025)	44M	46.9	69.4	42.6	66.4
Spatial-Mamba-T (Xiao et al., 2025)	46M	47.6	69.6	42.9	66.5
DAMamba-T (Li et al., 2026)	45M	48.5	70.3	43.4	67.2
ViS-T (ours)	48M	**48.6**	**70.9**	**44.0**	**67.9**
Swin-S (Liu et al., 2021)	69M	44.8	68.6	40.9	65.3
ConvNeXt-S (Liu et al., 2022)	70M	45.4	67.9	41.8	65.2
VMamba-S (Liu et al., 2024)	70M	48.7	70.0	43.7	67.3
LocalVMamba-S (Huang et al., 2024)	69M	48.4	69.9	43.2	66.7
VSSD-S (Shi et al., 2025)	59M	48.4	70.1	43.5	67.2
Spatial-Mamba-S (Xiao et al., 2025)	63M	49.2	70.8	44.0	67.9
DAMamba-S (Li et al., 2026)	65M	49.8	71.2	44.5	68.4
ViS-S (ours)	73M	49.8	**71.6**	**44.8**	**68.9**
Swin-B (Liu et al., 2021)	107M	46.9	69.2	42.3	66.0
ConvNeXt-B (Liu et al., 2022)	107M	47.0	69.4	42.7	66.3
VMamba-B (Liu et al., 2024)	108M	49.2	71.4	44.1	68.3
Spatial-Mamba-B (Xiao et al., 2025)	115M	50.4	71.8	45.1	69.1
DAMamba-B (Li et al., 2026)	105M	50.6	71.9	44.9	68.9
ViS-B (ours)	109M	**51.1**	**73.1**	**45.9**	**70.3**

Table 3: **Semantic segmentation on ADE20K.** UPerNet attached, 160K iterations.

Backbone	# Par.	SS	MS
Swin-T (Liu et al., 2021)	60M	44.4	45.8
ConvNeXt-T (Liu et al., 2022)	60M	46.0	46.7
MambaOut-T (Yu & Wang, 2025)	54M	47.4	48.6
VMamba-T (Liu et al., 2024)	62M	48.0	48.8
LocalVMamba-T (Huang et al., 2024)	57M	47.9	49.1
MSVMamba-T (Shi et al., 2024)	65M	47.6	48.5
VSSD-T (Shi et al., 2025)	53M	47.9	48.7
Spatial-Mamba-T (Xiao et al., 2025)	57M	48.6	49.4
DAMamba-T (Li et al., 2026)	55M	**50.3**	**51.2**
ViS-T (ours)	58M	50.1	50.5
Swin-S Liu et al. (2021)	81M	47.6	49.5
ConvNeXt-S Liu et al. (2022)	82M	48.7	49.6
MambaOut-S (Yu & Wang, 2025)	76M	49.5	50.6
VMamba-S (Liu et al., 2024)	82M	50.6	51.2
LocalVMamba-S (Huang et al., 2024)	81M	50.0	51.0
Spatial-Mamba-S (Xiao et al., 2025)	73M	50.6	51.4
DAMamba-S (Li et al., 2026)	75M	51.2	52.0
ViS-S (ours)	82M	**51.5**	**52.3**
Swin-B Liu et al. (2021)	121M	48.1	49.7
ConvNeXt-B Liu et al. (2022)	122M	49.1	49.9
MambaOut-B (Yu & Wang, 2025)	112M	49.6	51.0
VMamba-B Liu et al. (2024)	122M	51.0	51.6
Spatial-Mamba-B Xiao et al. (2025)	127M	51.8	52.6
DAMamba-B Li et al. (2026)	117M	51.9	52.3
ViS-B (ours)	122M	**52.9**	**53.1**

Table 4: **Class-conditional generation on ImageNet** 256^2 **images.** Trained for 400K steps.

Mixer	# Par.	FID↓	IS↑
DiT-S/2 (Peebles & Xie, 2023)	33.0M	67.04	20.54
w/ cfg=1.50	33.0M	43.91	35.24
DiS-S/2 (Ours)	31.6M	**62.19**	**23.18**
w/ cfg=1.50	31.6M	**37.31**	**44.01**
DiT-B/2 (Peebles & Xie, 2023)	130.5M	42.66	34.05
w/ cfg=1.50	130.5M	20.54	70.91
DiS-B/2 (Ours)	124.5M	**41.85**	**35.79**
w/ cfg=1.50	124.5M	**18.86**	**79.11**

Table 5: **Complexity of the recurrent and quadratic forms.** Measured on a single ViS block with a 56^2 token grid of 64 channels and a batch size of 128, using a single H100 GPU.

Form	FLOPs	Throughput (imgs/s)	Inf. time (ms)
Recurrent	0.193G	820	156
Quadratic	0.787G	11800	10.9
Quadratic (Fused kernel)	0.787G	19900	6.4

highest single- and multi-scale mIoU among the small backbones, and ViS-B the highest single-scale mIoU among the base backbones. ViS-T ranks second only to DAMamba-T and outperforms all other SSM backbones by a clear margin. As segmentation runs at inputs larger than the 224^2 used for pretraining, these results indicate that the spatial prior of ViS carries over to dense prediction.

Image generation We further test ViS as a token mixer for class-conditional image generation on ImageNet 256^2. We replace the transformer blocks of DiT (Peebles & Xie, 2023) with ViS blocks, termed *DiS*, and train both for 400K steps, using the VAE from Stable Diffusion (Rombach et al., 2022) (Table 4). DiS achieves lower FID and higher IS than DiT at both the S and B scales, with and without classifier-free guidance, while using fewer parameters. We attribute this gain in part to the spatial prior of ViS, which weights tokens by their distance on the grid by construction. Self-attention, in contrast, treats tokens as an unordered set and receives spatial structure only through positional embeddings.

4.1 Resolution robustness

Since vision backbones are often applied at resolutions higher than those used for pretraining, we evaluate publicly released 224^2-trained checkpoints of 30M-sized models at resolutions up to 768^2 without fine-tuning (Fig. 7). Beyond the slight gain just above the training resolution, accuracy degrades as the resolution grows, and stronger geometric priors slow this degradation among SSMs. VMamba (Liu et al., 2024), which scans in four directions, is more robust than the bidirectional Vim (Zhu et al., 2024), and models with state fusion or adaptive scans (Xiao et al., 2025; Li et al., 2026) also surpass the convolution-based ConvNeXt (Liu et al., 2022) and MambaOut (Yu & Wang, 2025). ViS achieves the highest accuracy at every tested resolution, with a margin that widens as the resolution grows. We attribute this to its spatial prior, which depends on the distance and direction between token pairs rather than on absolute positions in a fixed grid, and thus adapts to any grid size.

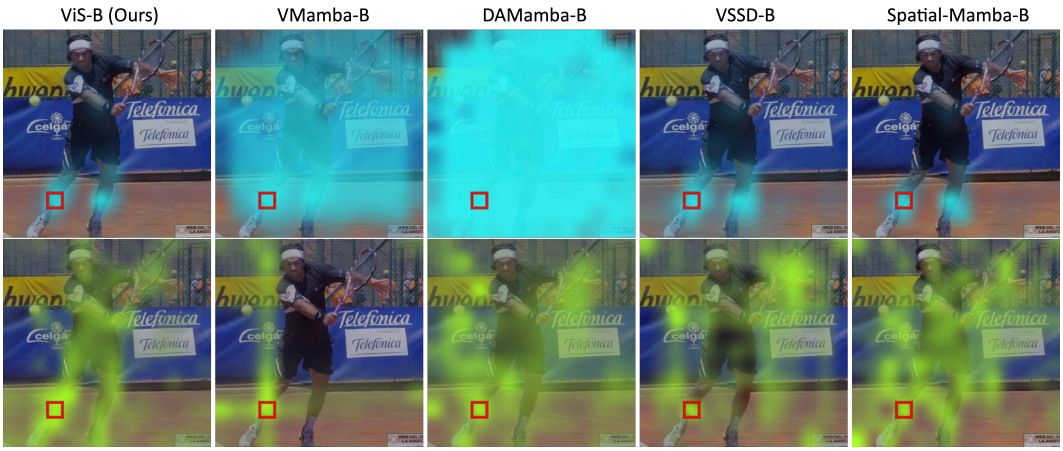

Figure 8: **Token similarity and mixing maps of base-sized vision SSMs.** For the query token marked by the red box, **(Top)** cosine similarity between each stage-3 output token and the query token; **(Bottom)** information of each token passed to the query token in the last mixer of stage 3.

4.2 Recurrent vs. quadratic form

Corollary 3.1 provides two ways to compute the same output of ViS: the recurrent form, which accumulates the ring sums of each token step by step, and the quadratic form, which evaluates the closed form of Proposition 3.1 with matrix multiplications. Table 5 compares the two on a single ViS block. The recurrent form requires about $4\times$ fewer FLOPs, consistent with its sub-quadratic cost, yet the quadratic form runs about $14\times$ faster. It is consistent with Mamba-2's observation (Dao & Gu, 2024): quadratic form consists of dense matrix multiplications that modern GPUs execute efficiently on tensor cores, whereas the recurrence is dominated by step-wise updates that cannot use them. Our fused kernel for the quadratic form further improves throughput by $1.7\times$, and ViS uses this kernel throughout our experiments.

4.3 Qualitative analysis

Fig. 8 compares ViS-B with four vision SSMs with respect to a token marked by the red box, which we refer to as *query* from the token-mixing perspective. The top row shows which output tokens have representations similar to the query token by cosine similarity. The bottom row shows where the query token aggregates information from, and brighter locations contribute more. The contribution of the j-th token is computed via the magnitude of its term in the output y_i, summed over channels. For ViS, both maps focus on the query-relevant context: similar tokens lie on the legs, and contributions follow the player's body, racquet, and ball. VMamba and DAMamba, in contrast, spread their similarity over most of the image. VSSD and Spatial-Mamba keep the similarity aligned with query semantics, but their contributions, like those of VMamba, scatter over the background or form stripes along the image axes. These maps suggest that ViS lets each token aggregate context from semantically related regions, rather than from positions dictated by a scan order or a single global state.

5 Conclusion

We have introduced ViS, a vision token mixer in which every token runs its own state-space recurrence over the concentric rings around it, and shown that this recurrence has an exact closed form as structured masked attention. The lesson we draw is that the inductive bias of an SSM lies in how a state accumulates context, not in the order in which a sequence is visited. Once the recurrence is instantiated at every token and run over spatial distance, the scan order, which vision SSMs have chosen, ensembled, and learned through many designs, is no longer a design decision at all. This change comes at no cost in practicality: the operator stays within the SSD framework, runs as dense matrix multiplications on modern hardware, and is competitive with or outperforms scan-based SSMs across recognition, dense prediction, and generation, while retaining the highest accuracy at unseen resolutions among the models we compared. Two limitations remain: the sub-quadratic recurrent form is still slower in practice than the quadratic form (Sec. 4.2), so its asymptotic advantage has yet to be realized in a kernel, and its complexity relies on $HW \cdot \max(H, W)$, which makes it underperform on images of extreme aspect ratio.

AI USE STATEMENT

In this work, we used generative AI tools for assisting in writing of proofs and supporting qualitative and thematic data analysis. We have not used generative AI tools for developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, designing or providing feedback on research methodology or experiments, implementing methods, interpreting results, and generating synthetic datasets, proposing or refining hypotheses, assisting with translation, cleaning and reformatting dataset are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, creating or editing software code, drafting parts of a research paper, editing a research paper to improve readability, and proposing a title or keywords for a research paper. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by three authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

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.

Guo Chen, Yifei Huang, Jilan Xu, Baoqi Pei, Zhe Chen, Zhiqi Li, Jiahao Wang, Kunchang Li, Tong Lu, and Limin Wang. Video mamba suite: State space model as a versatile alternative for video understanding. *arXiv preprint arXiv:2403.09626*, 2024.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*. PMLR, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: a large-scale hierarchical image database. In *CVPR*, 2009.

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 *ICLR*, 2021.

Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. *Advances in Neural Information Processing Systems*, 35:23439–23451, 2022.

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

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021a.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021b.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022a.

Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Ré. How to train your hippo: State space models with generalized orthogonal basis projections. *arXiv preprint arXiv:2206.12037*, 2022b.

Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems*, 35:22982–22994, 2022.

Ramin Hasani, Mathias Lechner, Tsun-Hsuan Wang, Makram Chahine, Alexander Amini, and Daniela Rus. Liquid structural state-space models. In *The Eleventh International Conference on Learning Representations*, 2023.

Ali Hatamizadeh and Jan Kautz. Mambavision: A hybrid mamba-transformer vision backbone. In *2025 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 25261–25270. IEEE, 2025.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. In *European conference on computer vision*, pp. 12–22. Springer, 2024.

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

Aakash Sunil Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, volume 2026, pp. 86173–86202, 2026.

Tanzhe Li, Caoshuo Li, Jiayi Lyu, Hongjuan Pei, Baochang Zhang, Taisong Jin, and Rongrong Ji. Damamba: Vision state space model with dynamic adaptive scan. *Advances in Neural Information Processing Systems*, 38:39777–39799, 2026.

Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *Advances in neural information processing systems*, 37:32653–32677, 2024.

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 *ECCV*, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37:103031–103063, 2024.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals with state spaces. *Advances in neural information processing systems*, 35:2846–2861, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pp. 9355–9366. PMLR, 2021.

Abdelrahman Shaker, Syed Talal Wasim, Salman Khan, Juergen Gall, and Fahad Shahbaz Khan. Groupmamba: Efficient group-based visual state space model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14912–14922. IEEE, 2025.

Yuheng Shi, Minjing Dong, and Chang Xu. Multi-scale vmamba: Hierarchy in hierarchy visual state space model. *Advances in Neural Information Processing Systems*, 37:25687–25708, 2024.

Yuheng Shi, Mingjia Li, Minjing Dong, and Chang Xu. Vssd: Vision mamba with non-causal state space duality. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10819–10829. IEEE, 2025.

Jimmy TH Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Aaron Voelker, Ivana Kajić, and Chris Eliasmith. Legendre memory units: Continuous-time representation in recurrent neural networks. In *Advances in Neural Information Processing Systems*, pp. 15544–15553, 2019.

Hongjun Wang, Wonmin Byeon, Jiarui Xu, Jinwei Gu, Ka Chun Cheung, Xiaolong Wang, Kai Han, Jan Kautz, and Sifei Liu. Parallel sequence modeling via generalized spatial propagation network. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4473–4483. IEEE, 2025.

Hongjun Wang, Collin McCarthy, David Wehr, Hanrong Ye, Xinhao Li, Ka Chun Cheung, Wonmin Byeon, Jinwei Gu, Ke Chen, Kai Han, et al. Gspn-2: Efficient parallel sequence modeling. *Advances in Neural Information Processing Systems*, 38:58884–58911, 2026.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *ICCV*, 2021.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational visual media*, 8(3):415–424, 2022.

Chaodong Xiao, Minghan Li, Deyu Meng, Lei Zhang, et al. Spatial-mamba: Effective visual state space models via structure-aware state fusion. In *International Conference on Learning Representations*, volume 2025, pp. 73777–73795, 2025.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European conference on computer vision*, pp. 432–448. Springer, 2018.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *Advances in neural information processing systems*, 37: 115491–115522, 2024.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*, volume 2025, pp. 29687–29707, 2025.

Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4484–4496. IEEE, 2025.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, July 2017.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *International Conference on Machine Learning*, pp. 62429–62442. PMLR, 2024.

VIS: VISION STATE-SPACE TOKEN MIXER

– SUPPLEMENTARY MATERIALS –

We describe the details omitted in the manuscript due to the space constraint.

A RELATED WORK

State-space models. State-space model (SSM) is a sequence-to-sequence model, which transforms the input sequence by evolving a hidden state under a transition matrix and reading the output sequence from it. One of the pioneering studies is HiPPO (Gu et al., 2020; 2022b), which compresses the history of a 1D sequence into the coefficients of an orthogonal polynomial projection and derives a transition matrix that updates this compressed memory online, in line with the Legendre memory units of Voelker et al. (2019). LSSL (Gu et al., 2021b) turns this formulation into a trainable sequence layer, and S4 (Gu et al., 2021a) makes it practical for long sequences through a structured parameterization of the transition matrix, computed as a global convolution. Subsequent work shows that simpler parameterizations suffice: DSS (Gupta et al., 2022) and S4D (Gu et al., 2022a) restrict the transition matrix to be diagonal, and S5 (Smith et al., 2023) adopts a multi-input multi-output state computed with a parallel scan. These models are linear time-invariant, and Liquid-S4 (Hasani et al., 2023) is among the first to let the transition depend on the input, through a liquid time-constant formulation. Mamba (Gu & Dao, 2024) makes the SSM parameters fully input-dependent with a hardware-aware selective scan, and has since become the dominant SSM design across modalities. In parallel, linear attention (Katharopoulos et al., 2020) and its gated and delta-rule variants, such as GLA (Yang et al., 2023), DeltaNet (Schlag et al., 2021; Yang et al., 2024), and Gated DeltaNet (Yang et al., 2025), maintain a matrix-valued state updated token by token, and can be viewed as SSMs with particular transition structures.

Vision state-space models. Vim (Zhu et al., 2024) first brings the selective recurrence of Mamba (Gu & Dao, 2024) to images by flattening the patch grid into a sequence and running a forward and a backward scan. Subsequent backbones keep the 1D recurrence and resolve the 1D/2D mismatch by *ensembling scan orders*: VMamba (Liu et al., 2024) scans four raster routes and merges them, GroupMamba (Shaker et al., 2025) assigns one route to each of four channel groups, LocalVMamba (Huang et al., 2024) scans within local windows and searches the direction per layer, MSVMamba (Shi et al., 2024) repeats the scans at several downsampled scales, and DAMamba (Li et al., 2026) predicts a data-dependent scan order. Spatial-Mamba (Xiao et al., 2025) and MambaVision (Hatamizadeh & Kautz, 2025) instead keep a single scan and repair its locality afterwards, by fusing neighboring hidden states with dilated convolutions or by appending self-attention blocks. These designs set strong accuracy baselines at sub-quadratic cost, and each added route indeed helps. They are, however, all limited in the sense that the recurrence still visits the grid in some order, so two adjacent tokens can be far apart along the scan, and each additional order multiplies the mixer cost while the ensemble remains an arbitrary basis of orders. A second line of work defines the recurrence in two dimensions from the outset. S4ND (Nguyen et al., 2022) forms a separable global convolution from independent one-dimensional S4 kernels along each axis, the 2D SSM layer of Baron et al. (2024) unrolls a Roesser-type recurrence with horizontal and vertical states into a convolution kernel, and GSPN (Wang et al., 2025; 2026) propagates row by row with row-stochastic tridiagonal weights from four sides of the image, reducing the effective sequence length to $\sqrt{HW}$. Although these recurrences operate on the grid rather than on a flattened sequence, they still sweep it in a fixed direction, from one corner or one side to the opposite one, creating an unnatural casuality to images. In contrast, ViS anchors the recurrence at *every token*: each token owns a hidden state that absorbs its concentric rings, so no order is chosen and no ensemble is needed, while the parameters remain input-dependent as in Mamba.

SSD and structured masked attention. Mamba-2 (Dao & Gu, 2024) shows that a selective SSM with scalar-times-identity transitions computes structured masked attention, $\mathbf{y} = (L \circ CB^\top)\mathbf{x}$ with a causal semiseparable mask L, and that the low rank of L permits a sub-quadratic contraction order. Mamba-3 (Lahoti et al., 2026) refines the recurrence with a trapezoidal discretization, a complex-valued state that it shows to be equivalent to a data-dependent rotary embedding, and a

multi-input multi-output formulation. Both remain one-dimensional: the mask encodes causality and recency along the sequence, and the rotary angles accumulate along the scan. VSSD (Shi et al., 2025) is among the first attempts to instantiate 2D SSMs into the SSD framework, deriving an equivalent formulation of hidden states under which different scanning routes yield consistent outcomes. In short, VSSD lets each token write into the state with a weight of its own instead of decaying the state at every step, so the order in which tokens are visited no longer matters, and all tokens end up reading from one shared global state. The price is that the mask disappears altogether: the operator reduces to a gated linear attention with no spatial decay, so distance between tokens plays no role. ViS, on the other hand, keeps the SSD form and changes only the mask. Each token retains its own hidden state, the causal mask becomes a radial one that decays with Chebyshev distance and is computed from token coordinates. This is what lets ViS inherit the Mamba-2 contraction orders (Corollary 3.1) while carrying a spatial prior that neither SSD nor VSSD provides.

B PROOFS

B.1 PROOF OF PROPOSITION 3.1

Proposition 3.1 (Equivalence to structured masked attention)**.** *Let* $\mathbf{A}_i = a_i\mathbf{I} \in \mathbb{R}^{N\times N}$. *Then the proposed SSM recurrence Eq. (5) admits the following equivalent closed-form formulation as structured masked attention:*

$$\mathbf{y} = L \circ \begin{bmatrix} — \mathbf{C}_0^\top — \\ — \mathbf{C}_1^\top — \\ \vdots \\ — \mathbf{C}_{HW-1}^\top — \end{bmatrix} \begin{bmatrix} | & | & & | \\ \mathbf{B}_0 & \mathbf{B}_1 & \cdots & \mathbf{B}_{HW-1} \\ | & | & & | \end{bmatrix} \mathbf{x} := M\mathbf{x}, \tag{6}$$

$$\textit{where} \qquad L_{i,j} = \frac{a_i^{d(i,j)}}{|\mathcal{R}_{d(i,j)}(i)|}, \qquad M_{i,j} = \frac{a_i^{d(i,j)}\mathbf{C}_i^\top \mathbf{B}_j}{|\mathcal{R}_{d(i,j)}(i)|}. \tag{7}$$

Proof. For brevity, let $(s_{ij})_t := \frac{1}{\|\mathcal{R}_{T-t(i)}\|}\sum_{j\in\mathcal{R}_{T-t}(i)} \mathbf{B}_j\, x_j$, then the recurrence Eq. (5) becomes

$$\begin{aligned} y_i &= \mathbf{C}_i^\top (\mathbf{h}_i)_T \\ &= \mathbf{C}_i^\top \Big(\mathbf{A}_i(\mathbf{h}_i)_{T-1} + (s_{ij})_T\Big) \\ &= \mathbf{C}_i^\top \Big(\mathbf{A}_i^2(\mathbf{h}_i)_{T-2} + \mathbf{A}_i(s_{ij})_{T-1} + (s_{ij})_T\Big) \\ &\quad\ \vdots \\ &= \mathbf{C}_i^\top \Big(\mathbf{A}_i^T(s_{ij})_0 + \mathbf{A}_i^{T-1}(s_{ij})_1 + \cdots + \mathbf{A}_i(s_{ij})_{T-1} + (s_{ij})_T\Big) \\ &= \mathbf{C}_i^\top \sum_{t=0}^{T} \mathbf{A}_i^{T-t}(s_{ij})_t. \end{aligned} \tag{11}$$

From here, we can unfold $(s_{ij})_t$ and reorganize the terms in Eq. (11):

$$\begin{aligned} y_i &= \mathbf{C}_i^\top \sum_{t=0}^{T} \mathbf{A}_i^{T-t}(s_{ij})_t \\ &= \mathbf{C}_i^\top \sum_{t=0}^{T} \Big(\mathbf{A}_i^{T-t} \cdot \frac{1}{|\mathcal{R}_{T-t}(i)|} \sum_{j\in\mathcal{R}_{T-t}(i)} \mathbf{B}_j \cdot x_j\Big) \qquad (12) \\ &= \mathbf{C}_i^\top \sum_{\forall j} \Big(\mathbf{A}_i^{d(i,j)} \cdot \frac{1}{|\mathcal{R}_{d(i,j)}(i)|}\mathbf{B}_j\Big) \cdot x_j \qquad (13) \end{aligned}$$

Note that the transition from Eq. (12) to (13) is obtained by regrouping the terms according to the contribution of the j-th token to the i-th token, instead of grouping them by the T rings. Under the

scalar-times-identity parameterization of $\mathbf{A}_i$, Eq. (13) naturally coincides with the notion introduced in Eq. (3):

$$\mathbf{y} = L \circ \begin{bmatrix} — \mathbf{C}_0^\top — \\ — \mathbf{C}_1^\top — \\ \vdots \\ — \mathbf{C}_{HW-1}^\top — \end{bmatrix} \begin{bmatrix} | & | & & | \\ \mathbf{B}_0 & \mathbf{B}_1 & \cdots & \mathbf{B}_{HW-1} \\ | & | & & | \end{bmatrix} \mathbf{x} := M\mathbf{x}, \tag{14}$$

where $L_{i,j} = \frac{a_i^{d(i,j)}}{|\mathcal{R}_{d(i,j)}(i)|}$ and $M_{i,j} = \frac{a_i^{d(i,j)} \mathbf{C}_i^\top \mathbf{B}_j}{|\mathcal{R}_{d(i,j)}(i)|}$. □

B.2 Proof of Corollary 3.1

Corollary 3.1 (Two contraction orders)**.** *Let $Q, K \in \mathbb{R}^{HW \times N}$ and $V \in \mathbb{R}^{HW}$ be arbitrary, and let L be the mask in Eq. (7). Then the structured masked attention $\mathbf{y} = \big(L \circ (QK^\top)\big)V$ admits two contraction orders that evaluate to the same $\mathbf{y}$: the order (Q, K, L, V), which forms $QK^\top$ first and costs $O\big((HW)^2 N\big)$, and the order (V, K, L, Q), which contracts V with K first and costs $O\big(HW \cdot \max(H, W) \cdot N\big)$.*

Proof. Let $\mathbf{q}_i, \mathbf{k}_j \in \mathbb{R}^N$ be the i-th row of Q and the j-th row of K written as column vectors, and $v_j \in \mathbb{R}$ be the j-th entry of V. Since $(QK^\top)_{i,j} = \mathbf{q}_i^\top \mathbf{k}_j$, the i-th entry of $\mathbf{y} = \big(L \circ (QK^\top)\big)V$ becomes:

$$y_i = \sum_{j=1}^{HW} L_{i,j} \left(\mathbf{q}_i^\top \mathbf{k}_j\right) v_j. \tag{15}$$

Evaluating Eq. (15) as written is the order (Q, K, L, V): computing $QK^\top \in \mathbb{R}^{HW \times HW}$ costs $O\big((HW)^2 N\big)$, while the elementwise product with L and the product with V each cost $O\big((HW)^2\big)$.

For the order (V, K, L, Q), we use that every summand in Eq. (15) is a scalar multiple of $\mathbf{q}_i$, which does not depend on j. Note that $\{\mathbf{q}_i^\top \mathbf{k}_j, L_{i,j}, v_j\} \subset \mathbb{R}$ are scalars, and $L_{i,j}(\mathbf{q}_i^\top \mathbf{k}_j) v_j = \mathbf{q}_i^\top \left(L_{i,j}\, v_j \mathbf{k}_j\right)$ holds. Then, by linearity of the inner product:

$$y_i = \mathbf{q}_i^\top \mathbf{w}_i, \qquad \mathbf{w}_i := \sum_{j=1}^{HW} L_{i,j}\, \mathbf{z}_j, \qquad \mathbf{z}_j := v_j \mathbf{k}_j \in \mathbb{R}^N. \tag{16}$$

Computing $\{\mathbf{z}_j\}_{j=1}^{HW}$ costs $O(HW \cdot N)$, and computing $\{y_i\}_{i=1}^{HW}$ from $\{\mathbf{w}_i\}_{i=1}^{HW}$ costs $O(HW \cdot N)$. It remains to show that the middle contraction $\{\mathbf{w}_i\}_{i=1}^{HW}$ can be computed in $O\big(HW \cdot \max(H, W) \cdot N\big)$, which is where the structure of L enters.

A key step toward the switched contraction is to group tokens that are equidistant from the i-th token and reorganize the sum in Eq. (16) accordingly. Note that for every i and every $t \in \{0, \ldots, T\}$ with $\mathcal{R}_t(i) \neq \varnothing$, the value $\lambda_i(t) := L_{i,j}$ is the same for all $j \in \mathcal{R}_t(i)$; for $\mathcal{R}_t(i) = \varnothing$ we set $\lambda_i(t) := 0$. Since $d(i, j) \leq T$ for every pair, the rings $\{\mathcal{R}_t(i)\}_{t=0}^{T}$ partition the token grid, and we may regroup the sum in Eq. (16) by rings:

$$\mathbf{w}_i = \sum_{t=0}^{T} \lambda_i(t)\, \boldsymbol{\rho}_t(i), \qquad \boldsymbol{\rho}_t(i) := \sum_{j \in \mathcal{R}_t(i)} \mathbf{z}_j, \tag{17}$$

with $\boldsymbol{\rho}_t(i) = \mathbf{0}$ for an empty ring. Each ring sum $\boldsymbol{\rho}_t(i)$ can be obtained in $O(N)$ time after an $O(HW \cdot N)$ precomputation. Let $\mathcal{S}_t(i) := \{\, j \mid d(i, j) \leq t \,\}$ be the set of tokens within Chebyshev distance t of the i-th token, with $\mathcal{S}_{-1}(i) := \varnothing$. By the definition of $d(i, j)$, $\mathcal{S}_t(i)$ is the set of grid positions in the axis-aligned rectangle $[\max(0, \rho_1(i) - t),\, \min(H - 1, \rho_1(i) + t)] \times [\max(0, \rho_2(i) - t),\, \min(W - 1, \rho_2(i) + t)]$. Since $\mathcal{S}_{t-1}(i) \subseteq \mathcal{S}_t(i)$ and $\mathcal{R}_t(i) = \mathcal{S}_t(i) \setminus \mathcal{S}_{t-1}(i)$, we have

$$\boldsymbol{\rho}_t(i) = \sum_{j \in \mathcal{S}_t(i)} \mathbf{z}_j - \sum_{j \in \mathcal{S}_{t-1}(i)} \mathbf{z}_j. \tag{18}$$

Now, we can define a 2D prefix sum $\mathbf{P}(u, v)$ for $(u, v) \in [-1, H-1] \times [-1, W-1]$ as follows:

$$\mathbf{P}(u,v) := \begin{cases} 0 & u = -1 \\ 0 & v = -1 \\ \sum_{j:\, \rho_1(j) \le u,\ \rho_2(j) \le v} \mathbf{z}_j & \text{otherwise} \end{cases}, \tag{19}$$

so that all HW values of $\mathbf{P}$ are computed in $O(HW \cdot N)$ by the following recursion:

$$\mathbf{P}(u,v) = \mathbf{z}_{(u,v)} + \mathbf{P}(u-1,v) + \mathbf{P}(u,v-1) - \mathbf{P}(u-1,v-1), \tag{20}$$

where $\mathbf{z}_{(u,v)}$ denotes $\mathbf{z}_j$ for the token at position (u, v). By inclusion–exclusion, the sum of $\mathbf{z}_j$ over any rectangle $[u_0, u_1] \times [v_0, v_1]$ equals $\mathbf{P}(u_1, v_1) - \mathbf{P}(u_0 - 1, v_1) - \mathbf{P}(u_1, v_0 - 1) + \mathbf{P}(u_0 - 1, v_0 - 1)$, which is four lookups and $O(N)$ arithmetic. Applying this to the two rectangles in Eq. (18) yields $\boldsymbol{\rho}_t(i)$ in $O(N)$ time.

Combining the above, computing the vector $\mathbf{w}_i$ in Eq. (17) is a sum of $T + 1 = \max(H, W)$ terms, each computed in $O(N)$, so all $\{\mathbf{w}_i\}_{i=1}^{HW}$ cost $O\big(HW \cdot \max(H, W) \cdot N\big)$ in addition to the $O(HW \cdot N)$ precomputation of $\mathbf{P}$. Together with the $O(HW \cdot N)$ cost of forming $\{\mathbf{z}_j\}$ and of the final inner products, the order (V, K, L, Q) costs $O\big(HW \cdot \max(H, W) \cdot N\big)$ in total, which is $O\big((HW)^{3/2} N\big)$ for grids of bounded aspect ratio. □

C Model configurations

All ViS variants share the four-stage hierarchical architecture in Fig. 5 and differ in the depth and width of their stages. Table 6 lists the configuration of each variant, where N_k is the dimension of the hidden state at stage k. 'FFN expansion' denotes the ratio of the FFN hidden dimension to the channel dimension of the block. Following the DAMamba (Li et al., 2026) training recipe, all variants are trained with MESA (Du et al., 2022), which distills from an exponential moving average (EMA) of the model weights, using an EMA decay rate of 0.9995. All runs are trained for 300 epochs at a global batch size of 4096.

Table 6: **Configurations of ViS variants.** L_k, C_k and N_k denote the number of blocks, channel dimension, and state size of the k-th stage, respectively.

	ViS-T	ViS-S	ViS-B
# Params	28M	53M	90M
Depths (L_1, L_2, L_3, L_4)	(1, 3, 11, 4)	(2, 8, 16, 10)	(2, 6, 20, 8)
Dims (C_1, C_2, C_3, C_4)	(64, 128, 320, 512)	(64, 128, 320, 512)	(96, 192, 448, 640)
State size (N_1, N_2, N_3, N_4)	(16, 16, 32, 32)	(16, 16, 32, 32)	(16, 16, 32, 32)
Drop path rate	0.2	0.4	0.6
FFN expansion	4.0	4.0	4.0
MESA weight	1.0	1.5	2.0

D Ablation studies

We ablate the components of ViS with ViS-T on ImageNet-1k at 224^2 (Table 7). Unless noted, each ablation is a single-seed 300-epoch run at a global batch size of 4096, and we report the best top-1 validation accuracy during training. Each variant changes one component of the default model, shown in the first row of each sub-table.

Breaking isotropy Removing either of the two components of Sec. 3.4 lowers accuracy, by 0.4%p for the sector gain and 0.3%p for the rotary embedding on $\mathbf{B}$ and $\mathbf{C}$ (Table 7(a)). We additionally test injecting a positional embedding on the input tokens without applying the two proposed components, and it reaches only the accuracy of the model without the rotary embedding.

Ring order Consuming the rings from the farthest to the nearest outperforms the reverse order by 0.5%p (Table 7(b)). This agrees with the locality argument of Sec. 3.2: under the farthest-first order, information from farther rings decays more before the state is read out.

Block design LayerNorm slightly outperforms RMSNorm (Table 7(d)). For FFN, we applied ConvFFN (Wang et al., 2022), an enhanced FFN module adapted for vision tasks, following prior studies (Li et al., 2026; Shi et al., 2024). As the original GatedMLP has no spatial prior, we adopt Gated Dconv Feed-Forward Network (GDFN) from Restormer (Zamir et al., 2022), which is an established FFN structure equipped with both gating mechanism and spatial prior. As ConvFFN turns out to surpass GDFN by 0.4%p (Table 7(e)), we adopt LayerNorm and ConvFFN in the ViS block (Sec. 3.5).

Mixer type To separate the effect of the token mixer from that of the macro design, we keep the ViS-T backbone and training recipe fixed and replace only the token mixer in every block (Table 7(c)). We compare the ring recurrence of ViS with the bidirectional scan of Vim (Zhu et al., 2024), the dynamic adaptive scan of DAMamba (Li et al., 2026), and self-attention. ViS outperforms Vim, DAMamba, and self-attention by 0.4%p, 0.8%p, and 2.1%p, respectively.

Table 7: **Ablations on ViS-T.** ImageNet-1k top-1 accuracy (%) at 224^2. The first row of each sub-table is the default configuration.

(a) Breaking isotropy

Method	Top-1
ViS-T	**84.1**
w/o sector gain	83.7
w/o RoPE on $\mathbf{B}, \mathbf{C}$	83.8
w/ pos. embedding	83.8

(b) Ring order

Order	Top-1
far $\rightarrow$ near	**84.1**
near $\rightarrow$ far	83.6

(c) Mixer type

Method	Top-1
ViS	**84.1**
Vim	83.7
DAMamba	83.3
Transformer	82.0

(d) Normalization

Method	Top-1
LayerNorm	**84.1**
RMSNorm	84.0

(e) FFN type

Method	Top-1
ConvFFN	**84.1**
GDFN	83.7

E Resolution robustness

We present the values that are used to plot Fig. 7, in Table 8. The accuracies at 224^2 of prior studies are based on the reproduced validation accuracy from their released checkpoints, and therefore they slightly deviate from their reported performance.

Table 8: **Zero-shot resolution transfer of 224^2-trained checkpoints.**

Method	# Par.	224^2	256^2	288^2	320^2	384^2	448^2	512^2	640^2	768^2
ConvNeXt-T (Liu et al., 2022)	29M	82.0	82.4	82.2	82.1	81.1	79.7	78.0	74.3	69.5
MambaOut-T (Yu & Wang, 2025)	27M	82.6	82.9	83.2	83.1	82.5	81.2	79.6	74.9	68.4
Vim-S (Zhu et al., 2024)	26M	80.5	80.3	79.7	78.8	75.9	71.5	65.3	49.6	33.8
VMamba-T (Liu et al., 2024)	30M	82.6	82.8	82.9	82.9	82.4	81.7	80.9	78.6	74.7
Spatial-Mamba-T (Xiao et al., 2025)	27M	83.4	83.6	83.9	83.3	82.8	81.5	80.1	77.6	74.6
VSSD-T (Shi et al., 2025)	24M	83.4	83.8	83.8	83.5	81.8	78.7	71.7	43.3	18.8
DAMamba-T (Li et al., 2026)	26M	83.7	84.1	84.2	84.0	83.6	82.4	80.9	76.4	69.7
ViS-T (Ours)	28M	**84.1**	**84.4**	**84.3**	**84.3**	**83.7**	**82.7**	**81.7**	**79.6**	**76.3**

F Additional qualitative results: token similarity map

Fig. 9 visualizes the cosine similarity between the output token at a query position, marked by a white box, using the output tokens of the third stage of ViS-B. Each column shows one image, and the two rows use two different query positions within it. The similarity is not constrained to the neighborhood of the query, but extends to distant tokens that share its semantics. Within the same image, a different query selects a different region, such as the needle instead of the numerals or the peppers instead of the pizza rim.

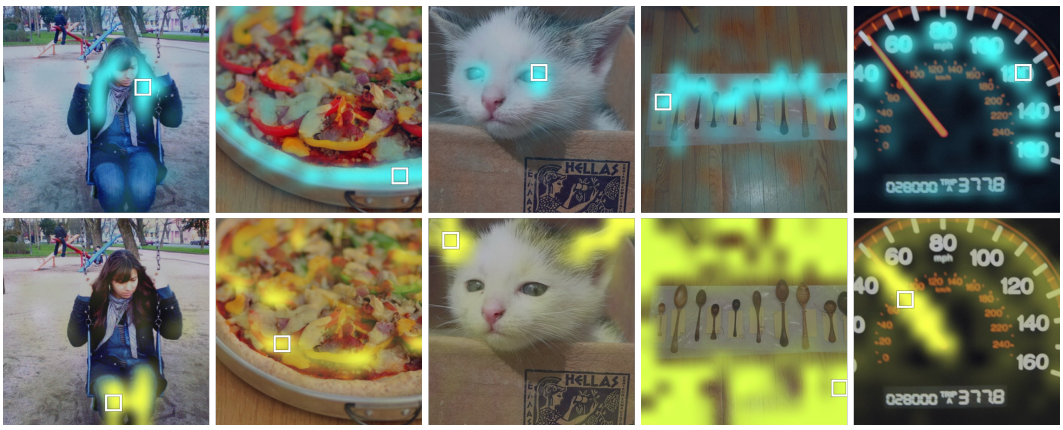

Figure 9: **Token similarity map of ViS.** Similarity of stage-3 output tokens to the query token marked by the white box, for two query positions per image (top and bottom rows). Tokens covering semantically similar regions show higher similarity, even when they are far from the query.