TCP-SSM: EFFICIENT VISION STATE SPACE MODELS WITH TOKEN-CONDITIONED POLES

Anonymous authors
Paper under double-blind review

ABSTRACT

State Space Models (SSMs) have emerged as a compelling alternative to attention models for long-range vision tasks, offering input-dependent recurrence with linear complexity. However, most efficient SSM variants reduce computation cost by modifying scan routes, resolutions, or traversal patterns, while largely leaving the recurrent dynamics implicit. Consequently, the model's state-dependent memory behavior is difficult to control, particularly in compact backbones where long scan paths can exceed the effective memory horizon. We propose Token-Conditioned Poles SSM (TCP-SSM), a structured selective SSM framework that improves efficiency while making recurrence dynamics explicit and interpretable through stable poles. TCP-SSM explicitly controls SSM's state-dependent memory behavior using 1) real poles that model monotone or sign-alternating decay, and 2) complex-conjugate poles that capture damped oscillatory responses. Using bounded radius and angle modulation, TCP-SSM learns token-dependent poles, allowing each scan step to adapt its memory behavior to the current visual token while preserving pole stability. For practical scalability with reduced complexity, we integrate a lightweight low-rank input pathway, yielding an efficient scan operator that preserves linear-time scan complexity. Across image classification, semantic segmentation, and object detection, TCP-SSM reduces SSM computation complexity up to 44% in Vision Mamba-style models while maintaining or surpassing baseline accuracy.

1 INTRODUCTION

Efficient long-range modeling remains a central challenge in modern computer vision. Convolutional Neural Networks Krizhevsky et al. (2012); Simonyan & Zisserman (2014); He et al. (2016); Huang et al. (2019); Liu et al. (2022); Chollet (2017); Szegedy et al. (2016) remain appealing due to their structured inductive biases and efficiency, but their localized kernels require long-range dependencies to be accumulated progressively through depth. Vision Transformers Liu et al. (2021); Bao et al. (2021); Zamir et al. (2022); Dosovitskiy et al. (2020); Wang et al. (2021); Touvron et al. (2021b) address this limitation with self-attention, enabling direct global token interactions and strong long-range modeling, although their quadratic complexity scales poorly at high spatial resolutions. More recently, State Space Models (SSMs) Gu et al. (2021a); Smith et al. (2022); Fu et al. (2022); Gu & Dao (2023); Gu et al. (2021b); Gupta et al. (2022); Liu et al. (2024); Zhu et al. (2024), especially Mamba-style selective scans, have emerged as compelling alternatives by combining linear-time sequence modeling with input-dependent selective dynamics. This balance between global context modeling and efficiency makes them well-suited for large-receptive-field vision tasks, including image classification, object detection, and segmentation.

Despite this promise, designing efficient visual SSMs remains challenging since visual data are inherently two-dimensional, whereas recurrence operates over a one-dimensional token ordering. Current Vision Mamba architectures address this mismatch with multi-directional cross-scanning strategies Liu et al. (2024); Zhu et al. (2024) that better preserve spatial structure after flattening. However, in compact backbones with limited hidden-state capacity, these trajectories can exceed the layer's effective memory horizon, causing distant information to attenuate before contributing meaningfully to the representation. Prior work addresses this through scan engineering, including skip sampling and multi-scale traversal schemes that reduce long-range forgetting Pei et al. (2025); Yang et al. (2024); Huang et al. (2024); Shi et al. (2024). Although effective, these methods modify how spatial tokens are visited rather than how the recurrent dynamics store, retain, or propagate

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.

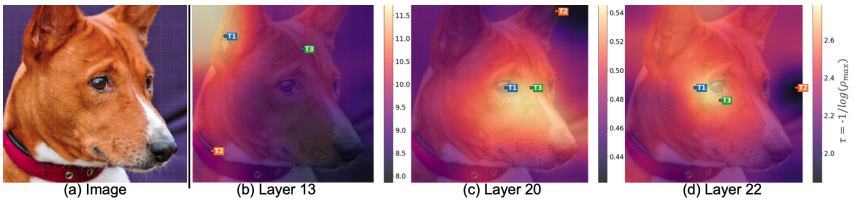


Figure 1: Illustration of the dominant pole-derived memory horizon at layers 13, 20, and 22 (of 24) for an example image. Brighter regions denote larger τ (slower decay, longer memory). T1, T2, and T3 highlight the longest memory, fastest decay, and strongest radius-weighted oscillation ($\rho_{\max}\theta/\pi$). Across depth, the pole-conditioned recurrence evolves from coarse object-level memory toward discriminative regions such as the face and eyes.

information. As a result, the scan's memory behavior remains largely implicit, making it difficult to characterize the learned timescales, signal decay rates, and adaptation to local image content.

We take a different perspective and argue that efficient visual SSMs should be improved not only through scan design, but also through the recurrence structure itself. From a control theory perspective, the memory and frequency response of recurrent systems are governed by *poles* in the complex plane, where the pole radius controls information decay and the pole angle determines localized oscillatory behavior Maestre et al. (2017); Beck et al. (2022). Rather than relying on an unconstrained selective recurrence to implicitly discover suitable timescales, we introduce efficient SSMs with Token-Conditioned Poles (TCP-SSM) with an explicit and stable parameterization that *adapts in a controlled manner*. TCP-SSM equips the scan operator with a stable base denominator built from real and complex pole components, where *real poles capture monotone memory decay* and *complex poles model frequency-selective oscillatory memory behavior*. The pole parameters vary with the input within bounded ranges, enabling spatially adaptive decay and oscillatory modes while preserving a compact and interpretable dynamical structure.

To ensure efficiency, we pair the structured pole parameterization with a lightweight low-rank input pathway, preserving SSM content selectivity without substantial hidden-state overhead. We further reduce computation through grouped pole sharing, allowing different channel groups to learn distinct dynamical profiles while avoiding fully independent poles for every feature dimension. This design retains the linear-time complexity of selective scanning while enhancing long-range expressiveness. In addition to improving efficiency, the pole-based formulation provides a direct view of the learned memory dynamics, which are often entangled within opaque state updates in conventional selective scans. Pole radii reveal how quickly past information is forgotten, pole angles characterize oscillatory modes and frequency preferences, and token-conditioned modulation shows how these behaviors vary across spatial contexts, offering a concrete interpretation of when the model preserves or suppresses information, as shown in Figure 1. We evaluate the proposed framework on Vision Mamba (Vim) backbones and EfficientVMamba (EVM) Pei et al. (2025). Across image classification, segmentation and object detection, our method improves the accuracy-efficiency trade-off, reducing SSM computation by up to roughly 44% in Vim and 70% in EVM-Base while maintaining comparable/better accuracy.

Our contributions can be summarized as follows:

- We introduce a new structured selective SSM formulation with token-conditioned poles in which stable real and complex poles define the base recurrence dynamics, while token-conditioned modulation adapts memory and frequency response to the input.
- We propose grouped pole sharing with a lightweight low-rank input pathway, preserving linear-time scanning while reducing long-range modeling complexity in visual SSMs.
- Across classification, segmentation, and object detection, explicit TCP-SSM improves the accuracy-efficiency trade-off of visual SSMs, reducing SSM computation by 44% in Vim and 70% in EVM-Base while maintaining comparable/improved accuracy.

2 Related Work

2.1 Efficient Vision Backbones

Computationally efficient vision backbones have evolved from lightweight convolutional networks to mobile-oriented transformers and hybrid architectures. Early designs such as ShuffleNetV2 Ma et al. (2018) improve practical efficiency by optimizing memory access costs, while MobileNetV3 Koonce (2021) advances accuracy-latency trade-offs through hardware-aware neural architecture

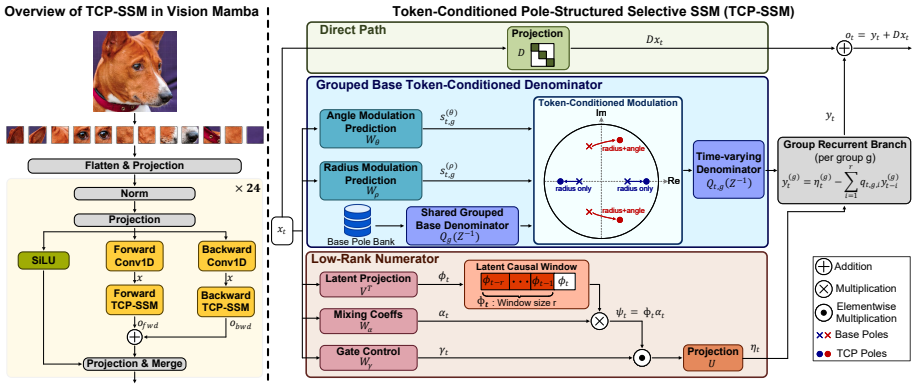


Figure 2: Overview of TCP-SSM Model. Left: TCP-SSM is inserted into the Vision Mamba model. Right: For token x_t, a shared grouped pole bank defines a stable base denominator $Q_g(z^{-1})$, while token-conditioned radius and angle modulation produce the local denominator $Q_{t,g}(z^{-1})$. A lightweight low-rank numerator forms the driving signal η_t, and the grouped recurrent branch outputs y_t. The final output o_t combines recurrent memory with a learned diagonal direct path.

search and platform-specific refinement. GhostNetV2 Tang et al. (2022) further improves inexpensive convolutional feature generation with hardware-friendly long-range spatial interaction. Hardware-aware search frameworks such as Once-for-All Cai et al. (2019) and FBNetV3 Dai et al. (2021) extend this direction by learning adaptable model families or jointly optimizing architectures and training recipes under deployment constraints. These ideas have also shaped transformer and hybrid backbones, where MobileViT Mehta & Rastegari (2021) introduces global token mixing into mobile models, MobileViTv2 Mehta & Rastegari (2022) reduces latency with separable linear-complexity attention, and AutoFormer Chen et al. (2021), EfficientFormerV2 Li et al. (2023), EfficientViT Liu et al. (2023), and FasterViT Hatamizadeh et al. (2023) improve mobile transformer efficiency through search, memory-efficient attention, or hierarchical design. Recent hybrids such as SwiftFormer Shaker et al. (2023), RepViT Wang et al. (2024a), and MobileNetV4 Qin et al. (2024) further explore additive attention, transformer-inspired convolutions, and unified search spaces. Our work is complementary to this line of research, but instead of introducing another lightweight backbone or attention mechanism, we improve the recurrence dynamics of compact vision state space models.

2.2 State Space Models for Visual Representation

State space models have become competitive alternatives to attention for visual representation learning, with S4 Gu et al. (2021a), S5 Smith et al. (2022), Mamba Gu & Dao (2023), and Mamba-2 Dao & Gu (2024) establishing efficient long-range modeling through structured state space parameterizations, input-dependent selective dynamics, and state space duality. In vision, S4ND Nguyen et al. (2022) extends SSMs to images and videos, while Vision Mamba Zhu et al. (2024) and VMamba Liu et al. (2024) demonstrate strong generic backbones through bidirectional modeling and 2D cross-scan operators. This paradigm has expanded through Mamba-ND Li et al. (2024) for multidimensional data, MambaVision Hatamizadeh & Kautz (2025) for hybrid Mamba-Transformer backbones, Mamba-R Wang et al. (2025) for register-token scaling, and V2M Wang et al. (2024b) and 2DMamba Zhang et al. (2025a) for native two-dimensional state updates. VM-UNet Ruan et al. (2024), SegMamba Xing et al. (2025), and PointMamba Liang et al. (2024) further show their versatility across visual domains. To improve 2D efficiency, LocalMamba Huang et al. (2024), PlainMamba Yang et al. (2024), and EfficientVMamba Pei et al. (2025) optimize spatial token serialization through windowed, continuous, or atrous scanning, while DefMamba Liu et al. (2025) and Multi-Scale VMamba Shi et al. (2024) introduce adaptive and multi-scale traversals. Other works redesign the state space block itself, including VSSD Shi et al. (2025) with non-causal state space duality, EfficientViM Lee et al. (2025) with compressed hidden states, and GroupMamba Shaker et al. (2025) with grouped visual state space modeling. Our work shares the goal of efficient visual SSM design but directly structures recurrence through stable poles and token-conditioned modulation, enabling efficient and interpretable long-range modeling.

3 Method

This section introduces the proposed TCP-SSM: *Efficient State Space Model with Token-Conditioned Poles*. The overview of TCP-SSM is demonstrated in Figure 2. We first establish notation, review state space and control theory preliminaries, and then present our algorithm.

Notation: Let $F \in \mathbb{R}^{B\times H\times W\times E}$ be the input feature map with batch size B, spatial dimensions of $H \times W$, and E channels. A scan rule flattens F into a token sequence $X \in \mathbb{R}^{B\times M\times E}$, where $M = HW$. For clarity, we describe the method for a single scan route. In practice, it is applied independently along each scan direction of a Vision Mamba-style backbone, with outputs fused exactly as in the original architecture. We partition the E channels into G equal groups of size $E_g = E/G$. For a token $x_t \in \mathbb{R}^E$ at position t, its g-th group is $x_t^{(g)} \in \mathbb{R}^{E_g}$. The recurrent output prior to the direct path is $y_t \in \mathbb{R}^E$, with its group block denoted $y_t^{(g)} \in \mathbb{R}^{E_g}$. The final processed output is $o_t \in \mathbb{R}^E$. We define r as the denominator polynomial order and r_f as the low-rank numerator dimension. We reserve the scalar variables u_t and v_t for classical dynamical systems and use x_t, y_t, and o_t for multidimensional feature representations in our model.

3.1 PRELIMINARIES

State Space Models: Classical continuous-time SSMs map a scalar input $u(\tau) \in \mathbb{R}$ to an output $v(\tau) \in \mathbb{R}$ via a latent state $h(\tau) \in \mathbb{R}^N$:

$$\dot{h}(\tau) = A_c h(\tau) + b_c u(\tau), \qquad v(\tau) = c^\top h(\tau) + d\, u(\tau), \tag{1}$$

where $A_c \in \mathbb{R}^{N\times N}$ is the continuous transition matrix, $b_c \in \mathbb{R}^N$ injects the input into the latent state, $c \in \mathbb{R}^N$ maps the state to the output, and $d \in \mathbb{R}$ is a direct feedthrough term. Here N denotes the latent dynamical system size. Discretizing this system with a step size $\Delta > 0$ under a zero-order hold (ZOH) Wang et al. (2008) assumption yields the discrete-time update:

$$h_t = A h_{t-1} + b\, u_t, \qquad v_t = c^\top h_t + d\, u_t. \tag{2}$$

When A_c is invertible, then $A = \exp(\Delta A_c)$ and $b = A_c^{-1}\big(\exp(\Delta A_c) - I\big)b_c$, where I is the identity matrix.

In visual backbones, the input at position t is a feature token $x_t \in \mathbb{R}^E$. Thus, the scalar state space model is replicated across all E channels. For a given channel e, the scalar input $x_{t,e}$ drives a distinct latent state $h_{t,e} \in \mathbb{R}^N$ according to $h_{t,e} = A_e h_{t-1,e} + b_e x_{t,e}$ and $v_{t,e} = c_e^\top h_{t,e} + d_e x_{t,e}$. Selective SSMs enhance this baseline by making the discretization and state interactions token-dependent, dynamically adapting memory at each spatial position.

Control-Theoretic View: A control-theoretic perspective in the z-domain reveals the memory mechanism more explicitly. Consider the discrete-time linear time-invariant system (LTI) from Eq. 2. Assuming a zero initial state, the transfer function relating the input $U(z)$ and output $V(z)$ after the z-domain transform Rantzer (2011) is defined as

$$H(z) \triangleq \frac{V(z)}{U(z)} = d + c^\top (I - Az^{-1})^{-1} b. \tag{3}$$

This function can equivalently be expressed in rational form as $H(z) = \frac{P(z^{-1})}{Q(z^{-1})}$, where P and Q are the numerator and denominator polynomials. A denominator of order r expands to $Q(z^{-1}) = 1 + \sum_{i=1}^{r} q_i z^{-i}$. Assuming a unit numerator, the corresponding difference equation $v_t = u_t - \sum_{i=1}^{r} q_i\, v_{t-i}$ demonstrates that the denominator uniquely dictates how past information persists or decays. These dynamics are governed strictly by the polynomial roots (i.e., the system poles), which coincide with the eigenvalues of A for a minimal realization without pole-zero cancellations (see Section B). System stability requires all poles to reside strictly inside the unit disk Van Gestel et al. (2002). Translating these insights into a practical architecture relies on two elementary stable factors in $Q(z^{-1})$. A first-order real factor $Q_r(z^{-1}) = 1 - az^{-1}$ (where $|a| < 1$) enforces monotone ($a > 0$) or alternating ($a < 0$) exponential decay proportional to a^k, where k is the discrete lag index. A second-order complex factor is defined as

$$Q_c(z^{-1}) = 1 - 2\rho\cos(\theta)\, z^{-1} + \rho^2 z^{-2}, \tag{4}$$

where $0 < \rho < 1$ and $0 < \theta < \pi$. The corresponding conjugate poles $\rho e^{\pm j\theta}$ generate a decaying oscillation. The radius ρ sets the exponential envelope and the angle θ determines the angular frequency, providing an interpretable basis for parameterizing recurrence denominators.

3.2 TCP-SSM: TOKEN-CONDITIONED POLE-STRUCTURED SELECTIVE SSM

The proposed formulation relies on three foundational components: 1) a structured denominator of stable real and complex-conjugate poles shared across channel groups, 2) input-dependent modulation of these poles, and 3) a lightweight low-rank numerator for the driving signal. This design preserves the linear-time efficiency of selective scanning while rendering the dynamics more interpretable, structured, and analytically tractable. All proofs are deferred to the supplementary material.

Stable Grouped Base Denominator: Each of G channel groups shares a *base denominator polynomial*, defining its pole bank prior to token-conditioned modulation. Given L real and K complex pole families (yielding a total order of $r = L + 2K$), the base denominator for group g is:

$$Q_g(z^{-1}) = \prod_{\ell=1}^{L} \left(1 - a_{g,\ell} z^{-1}\right) \prod_{k=1}^{K} \left(1 - 2\rho_{g,k}\cos(\theta_{g,k}) z^{-1} + \rho_{g,k}^2 z^{-2}\right). \tag{5}$$

To guarantee stability, all poles must strictly reside inside the unit disk ($a_{g,\ell} \in (-1, 1)$, $\rho_{g,k} \in (0, 1)$, and $\theta_{g,k} \in (0, \pi)$). We enforce this by mapping unconstrained learned variables into valid ranges using a constant boundary margin $0 < \varepsilon < 1$. Complex poles are bounded via sigmoid function (σ) applied to trainable pole parameters $\hat{\rho}_{g,k}$ and $\hat{\theta}_{g,k}$ such that:

$$\rho_{g,k} = (1 - \varepsilon)\sigma(\hat{\rho}_{g,k}), \qquad \theta_{g,k} = \pi\sigma(\hat{\theta}_{g,k}). \tag{6}$$

For real poles, we use a decoupled signed-magnitude parameterization that separates the response sign $\bar{s}_{g,\ell}$ from the magnitude $\bar{\rho}^R_{g,\ell}$, enabling the subsequent selective construction. Given trainable parameters $\hat{\rho}^R_{g,\ell}$ and $\hat{s}_{g,\ell}$, we define the real pole in a way that guarantees $|a_{g,\ell}| < 1$:

$$a_{g,\ell} = \bar{s}_{g,\ell}\, \bar{\rho}^R_{g,\ell}, \qquad \bar{\rho}^R_{g,\ell} = (1 - \varepsilon)\sigma(\hat{\rho}^R_{g,\ell}) \in (0, 1), \qquad \bar{s}_{g,\ell} = \tanh(\hat{s}_{g,\ell}) \in (-1, 1). \tag{7}$$

Proposition 1. *If $|a_{g,\ell}| < 1$ for all real factors and $0 < \rho_{g,k} < 1$, $0 < \theta_{g,k} < \pi$ for all complex factors, then all poles associated with Q_g lie strictly inside the unit disk. Hence, each grouped base denominator is Schur stable Borcea & Brändén (2009) (see Section C).*

Token-conditioned Grouped Denominator: We introduce time-varying dynamics by replacing the base grouped denominator Q_g with a token-conditioned denominator $Q_{t,g}$. The current token x_t modulates the shared base poles to achieve input-dependent selectivity. For each group g, we obtain a strictly positive radius scale $s^{(\rho)}_{t,g}$ and a bounded angle scale $s^{(\theta)}_{t,g}$ via:

$$s^{(\rho)}_{t,g} = \frac{\delta_{\min} + \mathrm{softplus}\big([W_\rho x_t + b_\rho]_{\kappa(g)}\big)}{\delta_0}, \qquad s^{(\theta)}_{t,g} = 1 + \lambda_\theta \tanh\big([W_\theta x_t + b_\theta]_{\kappa(g)}\big). \tag{8}$$

Here $\delta_{\min} > 0$, $\delta_0 > 0$, and $\lambda_\theta \in [0, 1)$ are fixed hyperparameters, where $\delta_{\min}$ and δ_0 set the offset and normalization of the radius scale and λ_θ bounds the angle modulation around one. The trainable parameters are $W_\rho, W_\theta \in \mathbb{R}^{C\times E}$ and $b_\rho, b_\theta \in \mathbb{R}^C$, where $C = G$ gives group-specific modulation and $C = 1$ gives shared modulation. The index map satisfies $\kappa(g) = g$ in the group-specific case and $\kappa(g) = 1$ in the shared case, so the obtained scales are broadcast across all groups. We adopt the shared configuration to improve the trade-off between performance and efficiency (see Table 5). The $s^{(\rho)}_{t,g} > 0$ modulates the effective memory timescale by contracting or stretching pole magnitudes. The $s^{(\theta)}_{t,g}$, centered at 1, governs oscillatory behavior and its bounded $\tanh$ formulation prevents abrupt frequency shifts. These scales dynamically update the pole families as follows:

1. Real Poles: We update the signed-magnitude form as $a_{t,g,\ell} = \bar{s}_{g,\ell} \exp(s^{(\rho)}_{t,g} \log \bar{\rho}^R_{g,\ell})$. This scales the magnitude while preserving the sign $\bar{s}_{g,\ell}$, ensuring the response type (monotone vs. sign-alternating) remains fixed. Since $s^{(\rho)}_{t,g} > 0$ and $\bar{\rho}^R_{g,\ell} \in (0, 1)$, the modulated magnitude strictly remains in $(0, 1)$, ensuring the pole stays safely inside the unit disk.

2. Complex Poles: We similarly adjust the decay envelope and oscillation frequency via:

$$\rho_{t,g,k} = \exp\Big(s^{(\rho)}_{t,g} \log \rho_{g,k}\Big), \qquad \theta_{t,g,k} = \mathrm{clip}\Big(s^{(\theta)}_{t,g}\theta_{g,k}, 0, \pi\Big). \tag{9}$$

This construction natively preserves $\rho_{t,g,k} \in (0, 1)$ and enforces valid angular intervals, ensuring the poles remain in unit disk. Together with the real-pole update above, these modulations convert the stable base pole bank defined by Q_g into a token-conditioned pole bank for scan step t. Specifically, the base poles $a_{g,\ell}$ and $\rho_{g,k}e^{\pm j\theta_{g,k}}$ are transformed into token-conditioned poles as $a_{t,g,\ell}$ and $\rho_{t,g,k}e^{\pm j\theta_{t,g,k}}$ using the modulation scales ($s^{(\rho/\ \theta)}_{t,g}$) predicted from x_t. These modulated poles define the token-conditioned denominator $Q_{t,g}$ used by the recurrent scan. When the modulation is identity, with $s^{(\rho/\ \theta)}_{t,g} = 1$, the construction recovers the base denominator Q_g.

Substituting these modulated poles back into the grouped base form yields the fully expanded token-conditioned denominator:

$$Q_{t,g}(z^{-1}) = \prod_{\ell=1}^{L} \left(1 - a_{t,g,\ell} z^{-1}\right) \prod_{k=1}^{K} \left(1 - 2\rho_{t,g,k}\cos(\theta_{t,g,k}) z^{-1} + \rho_{t,g,k}^2 z^{-2}\right). \tag{10}$$

Proposition 2. *Assume that for all t, g, ℓ, k, we have $|a_{t,g,\ell}| \leq 1 - \varepsilon_\rho$, $0 < \rho_{t,g,k} \leq 1 - \varepsilon_\rho$, and $\theta_{t,g,k} \in [0, \pi]$. Then every frozen local denominator $Q_{t,g}$ has all poles strictly inside the unit disk. Hence, each local grouped denominator is Schur stable (see Section C).*

Low-rank Numerator: While the denominator defines the system's poles, a complete transfer function requires a numerator to dictate how past inputs excite these dynamics. Freezing the time-varying coefficients at step t yields a local transfer function $H_t(z) = D + \frac{P_{t,sc}(z^{-1})}{Q_t(z^{-1})}$, which explicitly isolates the zero-lag direct path D from the strictly causal numerator $P_{t,sc}$. We preserve D and construct the causal branch using our token-conditioned denominator alongside a lightweight low-rank numerator.

For a multidimensional feature, a general strictly causal numerator at step t takes the form $P_{t,sc}(z^{-1}) = \sum_{i=1}^{r} B_{t,i} z^{-i}$ with $B_{t,i} \in \mathbb{R}^{E \times E}$, inducing a time-domain driving signal $\eta_t = \sum_{i=1}^{r} B_{t,i} x_{t-i}$. Since constructing the driving signal with dense matrices $B_{t,i}$ is computationally expensive, we approximate this numerator via a low-rank factorization.

Tokens are first projected into a compact latent space $\phi_t = V^\top x_t \in \mathbb{R}^{r_f}$ ($V \in \mathbb{R}^{E \times r_f}$, $r_f < E$), and the past r steps are assembled into a zero-padded causal window $\Phi_t = [\phi_{t-1}, \ldots, \phi_{t-r}] \in \mathbb{R}^{r_f \times r}$. From the current token x_t, we then predict token-dependent numerator coefficients and gating weights:

$$\alpha_t = W_\alpha x_t \in \mathbb{R}^r, \qquad \gamma_t = \sigma(W_\gamma x_t) \in \mathbb{R}^{r_f}, \tag{11}$$

where trainable parameters $W_\alpha \in \mathbb{R}^{r \times E}$ and $W_\gamma \in \mathbb{R}^{r_f \times E}$.

The mixing coefficients α_t blend the causal window into a low-rank latent mixture using $\psi_t = \Phi_t \alpha_t = \sum_{i=1}^{r} \alpha_{t,i} \phi_{t-i} \in \mathbb{R}^{r_f}$. Gating and up-projecting this mixture forms the final multidimensional driving signal such that:

$$\eta_t = U(\gamma_t \odot \psi_t) \in \mathbb{R}^E, \qquad U \in \mathbb{R}^{E \times r_f}, \tag{12}$$

where $\odot$ denotes elementwise multiplication. Equivalently, the induced low-rank numerator can be written as

$$P^{\mathrm{LR}}_{t,\mathrm{sc}}(z^{-1}) = U \,\mathrm{Diag}(\gamma_t) \left(\sum_{i=1}^{r} \alpha_{t,i} z^{-i} \right) V^\top = \sum_{i=1}^{r} \alpha_{t,i} \, U \,\mathrm{Diag}(\gamma_t) V^\top z^{-i}. \tag{13}$$

Hence, the dense matrices $B_{t,i}$ are represented implicitly through the rank-r_f factorization such that $B_{t,i} = \alpha_{t,i} \, U \,\mathrm{Diag}(\gamma_t) V^\top$.

Combining Numerator and Denominator: Finally, we partition the driving signal into G groups, $\eta_t = [\eta_t^{(1)}; \ldots; \eta_t^{(G)}]$, ensuring each group receives the portion corresponding to its denominator $Q_{t,g}$. Given the expanded token-conditioned denominator $Q_{t,g}(z^{-1}) = 1 + \sum_{i=1}^{r} q_{t,g,i} z^{-i}$, the grouped recurrent branch is computed directly as:

$$y_t^{(g)} = \eta_t^{(g)} - \sum_{i=1}^{r} q_{t,g,i} y_{t-i}^{(g)}. \tag{14}$$

Concatenating these groups yields the full recurrent state $y_t \in \mathbb{R}^E$.

Lastly, incorporating a learned diagonal direct path $D \in \mathbb{R}^{E \times E}$ (as in Vision Mamba), yields the output as $o_t = y_t + D x_t$. This preserves local token information and stabilizes optimization when long-range recurrence is unnecessary. Ultimately, this explicitly isolates architectural roles: the denominator governs memory via stable poles, the low-rank numerator dictates state excitation, and the direct path preserves instantaneous context.

Computational efficiency: Given a single scan route, our proposed method has a dominant per-token-channel inference cost of $\mathcal{O}(2r + 3r_f)$. In contrast, a standard selective SSM incurs an $\mathcal{O}(7N)$ cost in addition to the step-size projection. Thus, by keeping r and r_f small, the proposed operator achieves computational efficiency while maintaining linear scaling with respect to sequence length M and channel dimension E (see Section D).

TCP-SSM Model Training: We train our token-conditioned pole model via feature-level distillation using the pre-trained original SSM model as a teacher. Both share identical backbones, differing only that the student adopts our proposed TCP-SSM operators. The trainable parameters include the base denominator pole parameters $\hat{\rho}_{g,k}$, $\hat{\theta}_{g,k}$, $\hat{\rho}^R_{g,\ell}$, and $\hat{s}_{g,\ell}$, the token-modulation heads W_ρ, b_ρ, W_θ, and b_θ, the low-rank numerator parameters U, V, W_α, and W_γ, and the diagonal direct path D.

Table 1: Comparison on ImageNet-1K. SSM FLOPs are reported for models with an SSM branch.

Method	Mixer	Input	#Param.↓	SSM FLOPs↓	Top-1 (%)↑	Top-5 (%)↑
ViT-B/16 Dosovitskiy et al. (2020)	Attention	384^2	86M	–	77.9	95.3
Swin-T Liu et al. (2021)	Attention	224^2	29M	–	81.3	95.6
QuadMamba-Li Xie et al. (2024)	SSM	224^2	6.5M	318M	74.2	92.1
Vim-Ti-F Zhang et al. (2025b)	SSM	224^2	7.1M	497M	76.7	93.6
LocalViM-T Huang et al. (2024)	SSM	224^2	8M	825M	76.3	93.1
LocalViM-S Huang et al. (2024)	SSM	224^2	27.8M	1938.8M	81.8	95.6
S4ND-ViT-B Nguyen et al. (2022)	SSM + Attn.	224^2	89M	414M	80.4	–
PlainMamba-L2 Yang et al. (2024)	SSM	224^2	25.7M	1459M	81.6	95.6
Mamba-ND Li et al. (2024)	SSM	224^2	24M	1308M	81.7	–
VMamba-T Liu et al. (2024)	SSM	224^2	30.2M	253M	82.5	95.9
Vim-T Zhu et al. (2024)	SSM	224^2	7.1M	497.5M	76.1	93.0
TCP-Vim-T1	SSM	224^2	7.5M	**295.3M** (↓ 40.6%)	76.0	**93.1**
TCP-Vim-T2	SSM	224^2	8.1M	459.5M (↓ 7.6%)	**76.6**	**93.3**
EVM-T Pei et al. (2025)	SSM	224^2	6.5M	109.3M	76.3	93.1
TCP-EVM-T1	SSM	224^2	**6.4M**	**62.9M** (↓ 42.5%)	**76.3**	93.0
EVM-B Pei et al. (2025)	SSM	224^2	33.7M	233M	81.8	95.6
TCP-EVM-B3	SSM	224^2	**33.5M**	**129.9M** (↓ 44.2%)	**82.0**	**95.8**
TCP-EVM-B5	SSM	224^2	**33.5M**	231.4M (↓ 0.7%)	**82.6**	**96.0**

The denominator coefficients $q_{t,g,i}$, numerator coefficients α_t, and gates γ_t are generated from these learned parameters and the current token x_t. Letting $o_{b,t}^{(\ell)}$ and $\tilde{o}_{b,t}^{(\ell)}$ denote the student and teacher outputs at layer ℓ, batch b, and token t, we train the model using an L_1 loss:

$$\mathcal{L}_{\text{distill}} = \frac{1}{L_{\text{blk}}} \sum_{\ell=1}^{L_{\text{blk}}} \frac{1}{BM_\ell E_\ell} \sum_{b=1}^{B} \sum_{t=1}^{M_\ell} \left\| o_{b,t}^{(\ell)} - \text{sg}\left(\tilde{o}_{b,t}^{(\ell)}\right) \right\|_1 . \tag{15}$$

Here L_{blk} is the number of distilled layers, M_ℓ the token count, E_ℓ the channel dimension, and $\text{sg}(\cdot)$ the stop-gradient. The total loss $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{distill}}\mathcal{L}_{\text{distill}}$ leverages this intermediate guidance to encourage our efficient operator to emulate the teacher's representations while jointly optimizing the final downstream task objective $\mathcal{L}_{\text{task}}$.

4 Experiments

4.1 Image Classification

Settings: We evaluate on the ImageNet-1K dataset Deng et al. (2009) and report single-crop top-1 and top-5 validation accuracies. For fair comparison, we follow the DeiT Touvron et al. (2021a) training recipe with AdamW optimization, cosine learning-rate decay, and standard augmentations including random resized cropping, horizontal flipping, label smoothing, and MixUp Zhang et al. (2017). All experiments are performed on six RTX PRO 6000 Blackwell GPUs. Detailed training hyperparameters are provided in Table 6, and TCP-SSM configurations are summarized in Table 10.

To study the efficiency trade-off, we replace all 24 SSM layers in Vision Mamba-Tiny Zhu et al. (2024) with TCP-SSM and train two variants, TCP-Vim-T1 and TCP-Vim-T2, with TCP-Vim-T2 using a higher SSM FLOP budget. Both models are trained for 600 epochs with a learning rate of 1×10^{-4}. To evaluate generality, we apply TCP-SSM to EfficientVMamba Pei et al. (2025), an ideal testbed since it modifies the scan pattern rather than the recurrence. We replace the SSM modules in the first two stages and keep the remaining architecture unchanged. We explore three variants. TCP-EVM-T1 trains for 700 epochs, while TCP-EVM-B1 and TCP-EVM-B2 train for 250 epochs. All use a batch size of 256 and a 1×10^{-4} base learning rate.

Results: Table 1 evaluates the proposed TCP design on Vision Mamba and EfficientVMamba, demonstrating its generality across distinct visual SSM backbones. On the Vim backbone, TCP-Vim-T1 reduces SSM FLOPs by 40.6% (497.5M to 295.3M) while maintaining comparable Top-1 (76.0%) and Top-5 (93.1%) accuracy. TCP-Vim-T2 lowers the SSM cost to 459.5M and improves baseline accuracy to 76.6% Top-1 and 93.3% Top-5. EfficientVMamba shows a similar trend. TCP-EVM-T1 cuts SSM FLOPs by 42.5% (109.3M to 62.9M) while preserving the 76.3% baseline Top-1 accuracy. For EfficientVMamba-Base, TCP-EVM-B3 achieves a 44.2% FLOP reduction (233.0M to

Table 2: Comparison on ADE20K semantic segmentation. * denotes our produced results.

Method	Mixer	Input	#Param.↓	SSM FLOPs↓	mIoU (%)↑
SegFormer-B0 Xie et al. (2021)	Attention	512^2	3.7M	–	37.4
DeiT-T Touvron et al. (2021b)	Attention	512^2	11M	–	39.2
Octopus-T Mahatha et al. (2026)	SSM	512^2	63.3M	0.5G	37.9
Octopus-B Mahatha et al. (2026)	SSM	512^2	85.8M	1.5G	39.0
EfficientVMamba-T Pei et al. (2025)	SSM	512^2	14M	0.6G	38.9
Vim-T + DyVM Wu et al. (2025)	SSM	512^2	12.5M	1.8G	40.1
Vim-Ti-F Zhang et al. (2025b)	SSM	512^2	34M	2.58G	40.4
Vim-T* Zhu et al. (2024)	SSM	512^2	12.3M	2.58G	40.6
TCP-Vim-T1	SSM	512^2	12.7M	**1.53G** (↓ 40.6%)	**41.1**

Table 3: Object detection and instance segmentation on COCO dataset.

Method	#Param.↓	SSM FLOPs↓	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
Vim-T Zhu et al. (2024)	60.0M	10.34G	45.7	63.9	49.6	39.2	60.9	41.7
TCP-Vim-T1	60.4M	**6.14G** (↓ 40.6%)	**47.8**	**64.2**	**52.6**	**39.3**	60.6	**43.1**

129.9M) and improves accuracy to 82.0% Top-1 and 95.8% Top-5. When matched to the original compute budget, TCP-EVM-B5 reaches 82.6% Top-1 and 96.0% Top-5 accuracy using 231.4M SSM FLOPs. Ultimately, these results confirm that TCP's benefits are architectural rather than backbone-specific, consistently maintaining strong performance with a graceful accuracy - FLOPs trade-off (Matched-compute controls and runtime results are provided in Appendices E.2 and E.3).

4.2 Semantic Segmentation

Settings: To evaluate semantic segmentation performance, we conduct experiments on the ADE20K dataset Zhou et al. (2017) using UPerNet Xiao et al. (2018). The backbone is initialized with ImageNet-pretrained weights and followed by a UPerHead for dense prediction. We optimize the model using AdamW with a learning rate of 1×10^{-4} and a weight decay of 0.01, training for $180K$ iterations with total batch size of 24.

Results: Table 2 presents semantic segmentation results on the ADE20K validation set, demonstrating the transferability of our proposed TCP design. Since a pretrained Vim-T segmentation checkpoint is publicly unavailable, we train this baseline under our experimental setup to ensure a fair comparison. Building upon this, TCP-Vim-T1 improves the baseline mIoU from 40.6% to 41.1% while simultaneously reducing SSM FLOPs by 40.6% (from 2.58G to 1.53G). This achievement is especially significant as dense prediction dictates much stricter demands on spatial granularity than standard image classification. The concurrent improvement in performance and reduction in computational overhead confirm that our token-conditioned pole mechanism does not compromise representational capacity for efficiency.

4.3 Object Detection and Instance Segmentation

Settings: We evaluate TCP-SSM on COCO 2017 Lin et al. (2014) for object detection and instance segmentation within a ViTDet-style framework using Simple Feature Pyramid Li et al. (2022) and Cascade Mask R-CNN Cai & Vasconcelos (2019) heads. We replace only the Vim backbone with TCP-SSM and report box AP and mask AP. We provide detailed settings in Section E.1.

Results: Table 3 presents object detection and instance segmentation results on COCO, demonstrating that the proposed TCP design transfers effectively to detection-based dense prediction tasks. TCP-Vim-T1 maintains enhanced downstream performance to the Vim-T baseline while reducing SSM FLOPs by 40.6% (10.34G to 6.14G).

4.4 Ablation Study

Performance vs. Efficiency: Figure 3 illustrates the accuracy-computation trade-off for EfficientVMamba-Base when the original SSM modules are replaced by our proposed TCP design across various configurations. This comparison evaluates how recognition performance scales as recurrent computation is progressively reduced while the backbone architecture remains fixed. While the baseline yields 81.8% Top-1 and 95.6% Top-5 accuracy (233M SSM FLOPs), our TCP-EVM-B5 variant elevates this performance to 82.6% Top-1 and 96.0% Top-5 at a slightly reduced computational cost of 0.7%. The efficiency gains become even more pronounced at higher reduction levels. TCP-EVM-B4 delivers 82.4% Top-1 with an 18.5% FLOP reduction, and TCP-EVM-B3

Table 4: Ablation study on $\mathcal{L}_{\text{distill}}$.

Method	Distill.	λ_{distill}	Top-1	Top-5
Vim-T	–	–	76.1	93.0
TCP-Vim-T1	×	–	75.3	92.7
TCP-Vim-T1	✓	10^{-1}	75.8	92.9
TCP-Vim-T1	✓	2×10^{-2}	75.6	92.8
TCP-Vim-T1	✓	6×10^{-2}	**76.0**	**93.1**

Table 5: Ablation study on pole selectivity.

Method	TCP	#Param.	SSM FLOPs	Top-1	Top-5
Vim-T	–	7.1M	497.5M	76.1	93.0
TCP-Vim-T1	×	7.4M	280.6M	75.4	92.7
TCP-Vim-T1	✓ G-spec.	7.9M	375.3M	76.1	93.1
TCP-Vim-T1	✓ **Shared**	**7.5M**	**295.3M**	76.0	93.1

retains 82.0% Top-1 with a 44.2% drop in FLOPs. Most notably, TCP-EVM-B2 and TCP-EVM-B1 aggressively reduce SSM computation by 62.3% and 70.6%, yet still defend a strong 81.9% Top-1 accuracy. Overall, these results demonstrate a highly favorable efficiency trend across a broad range of operating points, confirming that our model successfully preserves strong accuracy despite substantial reductions in recurrent cost. (Detailed configurations in Table 10).

Impact of Distillation Loss: Table 4 investigates the impact of distillation loss in TCP-Vim-T1. In the absence of distillation, performance degrades (75.3% Top-1, 92.7% Top-5), indicating that TCP replacement alone struggles to recover the full accuracy of the Vim-T baseline. Adding distillation consistently improves performance across the tested weights, demonstrating that teacher guidance effectively helps the proposed recurrent module preserve the original backbone behavior. Specifically, $\lambda_{\text{distill}} = 6 \times 10^{-2}$ yields the strongest results. This closes the performance gap while retaining the reduced SSM computation of TCP-Vim-T1.

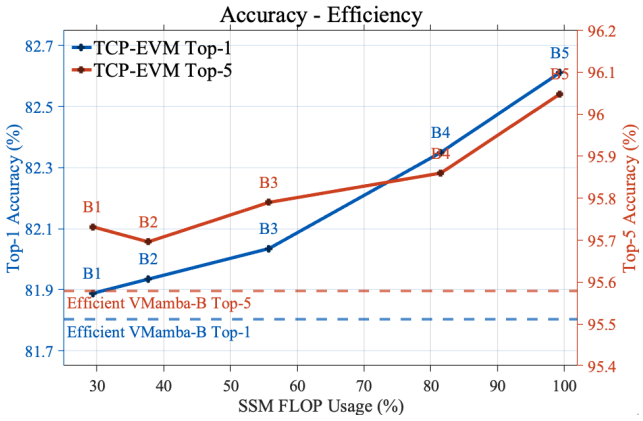


Figure 3: Accuracy-computation trade-off on EVM-B.

Effect of Token-Conditioned Pole Modulation: Table 5 evaluates token-conditioned poles ($G = 12$), which enable recurrent dynamics to adapt to local image content. With a fixed pole structure, TCP-Vim-T1 reaches 75.4% Top-1 accuracy, suggesting that static poles limit the flexibility of the recurrent module. Introducing token-conditioning improves performance, though the group-specific variant increases SSM FLOPs to 375.3M (76.1% Top-1, 93.1% Top-5). The shared-selectivity design provides a better balance, achieving comparable accuracy with a lower cost of 295.3M SSM FLOPs. This shows that sharing the modulation across groups preserves most of the benefit of token-conditioned poles while reducing the recurrent computation (Ablation study of G in Table 11).

5 Conclusion

We present an efficient visual state space modeling framework that uses token-conditioned poles to explicitly parameterize the recurrence dynamics of Vision Mamba-style models through stable real and complex pole components. By separating monotone memory decay from oscillatory frequency responses, our formulation provides a more interpretable and controllable alternative to conventional selective scans. Token-conditioned modulation enables stable, content-adaptive spatial memory, while grouped pole sharing and a lightweight low-rank input pathway preserve linear-time scan complexity. Across classification, segmentation, and object detection, the proposed approach improves the accuracy-efficiency trade-off of compact visual SSMs, reducing SSM computation by up to 44% in Vision Mamba and 70% in EfficientVMamba-Base while maintaining or improving accuracy.

Limitations and Future Work: Our method introduces additional parameters, such as pole order, group size, and low-rank dimension, which may require deployment-specific tuning. While the pole parameterization ensures stable local recurrence dynamics, it does not directly address other sources of inefficiency, such as scan ordering and memory access patterns. Future work can explore native two-dimensional pole-structured recurrences, adaptive scan, and hardware-aware implementations.

AI USE STATEMENT

Generative AI tools were used to improve the clarity and readability of the manuscript, assist with literature retrieval and discovery, and help debug and rewrite limited portions of the research code. All AI-assisted text, references, and code were reviewed and verified by the authors.

REPRODUCIBILITY STATEMENT

To support reproducibility, 1) we provide the code used to reproduce our experiments as a ZIP file in the supplementary material, and 2) we provide detailed experimental settings, implementation details, and hyperparameters in the appendix.

REFERENCES

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Amanda M Beck, Mingjian He, Rodrigo G Gutierrez, Gladia C Hotan, and Patrick L Purdon. An iterative search algorithm to identify oscillatory dynamics in neurophysiological time series. *bioRxiv*, pp. 2022–10, 2022.

Julius Borcea and Petter Brändén. The lee-yang and pólya-schur programs. ii. theory of stable polynomials and applications. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 62(12):1595–1631, 2009.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019.

Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12250–12260. IEEE, 2021.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Bichen Wu, Zijian He, Zhen Wei, Kan Chen, Yuandong Tian, Matthew Yu, Peter Vajda, et al. Fbnetv3: Joint architecture-recipe search using predictor pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16276–16285, 2021.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 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. *arXiv preprint arXiv:2010.11929*, 2020.

Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

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.

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.

Ali Hatamizadeh and Jan Kautz. Mambavision: A hybrid mamba-transformer vision backbone. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 25261–25270, 2025.

Ali Hatamizadeh, Greg Heinrich, Hongxu Yin, Andrew Tao, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. Fastervit: Fast vision transformers with hierarchical attention. *arXiv preprint arXiv:2306.06189*, 2023.

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.

Gao Huang, Zhuang Liu, Geoff Pleiss, Laurens Van Der Maaten, and Kilian Q Weinberger. Convolutional networks with dense connectivity. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):8704–8716, 2019.

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.

Brett Koonce. Mobilenetv3. In *Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization*, pp. 125–144. Springer, 2021.

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

Sanghyeok Lee, Joonmyung Choi, and Hyunwoo J Kim. Efficientvim: Efficient vision mamba with hidden state mixer based state space duality. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14923–14933, 2025.

Shufan Li, Harkanwar Singh, and Aditya Grover. Mamba-nd: Selective state space modeling for multi-dimensional data. In *European Conference on Computer Vision*, pp. 75–92. Springer, 2024.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European conference on computer vision*, pp. 280–296. Springer, 2022.

Yanyu Li, Ju Hu, Yang Wen, Georgios Evangelidis, Kamyar Salahi, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Rethinking vision transformers for mobilenet size and speed. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16889–16900, 2023.

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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

Leiye Liu, Miao Zhang, Jihao Yin, Tingwei Liu, Wei Ji, Yongri Piao, and Huchuan Lu. Defmamba: Deformable visual state space model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8838–8847, 2025.

Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14420–14430, 2023.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 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.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Esteban Maestre, Jonathan S Abel, Julius O Smith, and Gary P Scavone. Constrained pole optimization for modal reverberation. In *Proc. of the 5th International Conference on Digital Audio Effects (DAFx)*, pp. 155, 2017.

Kunal Mahatha, Ali Bahri, Pierre Marza, Sahar Dastani, Maria Vakalopoulou, Stergios Christodoulidis, Jose Dolz, and Christian Desrosiers. Octopus: Enhancing the spatial-awareness of vision ssms with multi-dimensional scans and traversal selection. *arXiv preprint arXiv:2602.00904*, 2026.

Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021.

Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *arXiv preprint arXiv:2206.02680*, 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.

Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 6443–6451, 2025.

Danfeng Qin, Chas Leichner, Manolis Delakis, Marco Fornoni, Shixin Luo, Fan Yang, Weijun Wang, Colby Banbury, Chengxi Ye, Berkin Akin, et al. Mobilenetv4: Universal models for the mobile ecosystem. In *European conference on computer vision*, pp. 78–96. Springer, 2024.

Anders Rantzer. Distributed control of positive systems. In *2011 50th IEEE conference on decision and control and European control conference*, pp. 6608–6611. IEEE, 2011.

Jiacheng Ruan, Jincheng Li, and Suncheng Xiang. Vm-unet: Vision mamba unet for medical image segmentation. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2024.

Abdelrahman Shaker, Muhammad Maaz, Hanoona Rasheed, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 17425–17436, 2023.

Abdelrahman Shaker, Syed Talal Wasim, Salman Khan, Juergen Gall, and Fahad Shahbaz Khan. Groupmamba: Efficient group-based visual state space model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14912–14922, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10819–10829, 2025.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

Petre Stoica and Randolph L Moses. On the unit circle problem: The schur-cohn procedure revisited. *Signal processing*, 26(1):95–118, 1992.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Chao Xu, and Yunhe Wang. Ghostnetv2: Enhance cheap operation with long-range attention. *Advances in Neural Information Processing Systems*, 35:9969–9982, 2022.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, volume 139, pp. 10347–10357, July 2021a.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021b.

Tony Van Gestel, Johan AK Suykens, Paul Van Dooren, and Bart De Moor. Identification of stable models in subspace identification by using regularization. *IEEE Transactions on Automatic control*, 46(9):1416–1420, 2002.

Ao Wang, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Repvit: Revisiting mobile cnn from vit perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15909–15920, 2024a.

Bin Wang, Xinghuo Yu, and Xiangjun Li. Zoh discretization effect on higher-order sliding-mode control systems. *IEEE Transactions on industrial electronics*, 55(11):4055–4064, 2008.

Chengkun Wang, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. V2m: visual 2-dimensional mamba for image representation learning. *arXiv preprint arXiv:2410.10382*, 2024b.

Feng Wang, Jiahao Wang, Sucheng Ren, Guoyizhe Wei, Jieru Mei, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Mamba-reg: Vision mamba also needs registers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14944–14953, 2025.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 568–578, 2021.

Mengxuan Wu, Zekai Li, Zhiyuan Liang, Moyang Li, Xuanlei Zhao, Samir Khaki, Zheng Zhu, Xiaojiang Peng, Konstantinos N Plataniotis, Kai Wang, et al. Dynamic vision mamba. *arXiv preprint arXiv:2504.04787*, 2025.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

Fei Xie, Weijia Zhang, Zhongdao Wang, and Chao Ma. Quadmamba: Learning quadtree-based selective scan for visual state space model. *Advances in Neural Information Processing Systems*, 37:117682–117707, 2024.

Zhaohu Xing, Tian Ye, Yijun Yang, Du Cai, Baowen Gai, Xiao-Jian Wu, Feng Gao, and Lei Zhu. Segmamba-v2: Long-range sequential modeling mamba for general 3d medical image segmentation. *IEEE Transactions on Medical Imaging*, 2025.

Chenhongyi Yang, Zehui Chen, Miguel Espinosa Minano, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J Crowley. Plainmamba: Improving non-hierarchical mamba in visual recognition. In *35th British Machine Vision Conference*. British Machine Vision Conference, 2024.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5728–5739, 2022.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Jingwei Zhang, Anh Tien Nguyen, Xi Han, Vincent Quoc-Huy Trinh, Hong Qin, Dimitris Samaras, and Mahdi S Hosseini. 2dmamba: Efficient state space model for image representation with applications on giga-pixel whole slide image classification. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3583–3592. IEEE, 2025a.

Juntao Zhang, Shaogeng Liu, Jun Zhou, Kun Bian, You Zhou, Jianning Liu, Pei Zhang, and Bingyan Liu. Vim-f: Visual state space model benefiting from learning in the frequency domain. In *Pacific Rim International Conference on Artificial Intelligence*, pp. 364–380. Springer, 2025b.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 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. *arXiv preprint arXiv:2401.09417*, 2024.

A Appendix

B Proof of the Transfer Function and Rational Form

We derive the transfer function for the discrete-time state-space system from Eq. 2. Assuming a zero initial state, taking the z-transform of the state update yields $\mathcal{H}(z) = Az^{-1}\mathcal{H}(z) + bU(z)$, which simplifies to

$$\mathcal{H}(z) = (I - Az^{-1})^{-1}bU(z). \tag{16}$$

Substituting this into the z-transformed output equation, $V(z) = c^\top \mathcal{H}(z) + dU(z)$, produces the transfer function as

$$H(z) \triangleq \frac{V(z)}{U(z)} = c^\top (I - Az^{-1})^{-1}b + d. \tag{17}$$

We next show that this expression can be written as a rational function in z^{-1}. Using the identity, $S^{-1} = \frac{\mathrm{adj}(S)}{\det(S)}$ for an invertible matrix S, we can obtain

$$(I - Az^{-1})^{-1} = \frac{\mathrm{adj}(I - Az^{-1})}{\det(I - Az^{-1})}. \tag{18}$$

Therefore, $H(z)$ can be written as

$$H(z) = \frac{d\det(I - Az^{-1}) + c^\top \mathrm{adj}(I - Az^{-1})b}{\det(I - Az^{-1})}. \tag{19}$$

This establishes the rational form $H(z) = P(z^{-1})/Q(z^{-1})$, where

$$Q(z^{-1}) = \det(I - Az^{-1}) \quad \text{and} \quad P(z^{-1}) = d\det(I - Az^{-1}) + c^\top \operatorname{adj}(I - Az^{-1})b. \tag{20}$$

Since $A \in \mathbb{R}^{N\times N}$, the denominator $Q(z^{-1})$ is a polynomial in z^{-1} of degree at most N with a constant term of one. After accounting for potential pole-zero cancellations, $H(z)$ simplifies to

$$H(z) = \frac{P(z^{-1})}{Q(z^{-1})}, \qquad Q(z^{-1}) = 1 + \sum_{i=1}^{r} q_i z^{-i}, \tag{21}$$

where $r \le N$ is the effective denominator order. The poles of the transfer function occur where $Q(z^{-1}) = \det(I - Az^{-1}) = 0$. Multiplying this equation by z^N yields $\det(zI - A) = 0$, demonstrating that the system poles are inherently the eigenvalues of A. In a minimal realization without pole-zero cancellations, the transfer-function poles and the eigenvalues of A perfectly coincide.

C Proofs of Pole Stability

Schur-Stability: A discrete-time denominator is Schur stable if all of its associated poles lie strictly inside the open unit disk. Equivalently, for a denominator written in the z^{-1} form, the roots in the z-plane of the corresponding pole polynomial must have magnitude strictly smaller than one. This is the standard stability condition underlying the Jury and Schur–Cohn Stoica & Moses (1992) criteria for discrete-time systems and polynomials. In practical terms, Schur stability guarantees that the discrete-time system is asymptotically stable, ensuring its response will not grow without bound over time. Therefore, for the proposed factored denominators, it is sufficient to verify that every pole induced by each real and complex-conjugate factor has magnitude less than one.

Proof of Proposition 1. We show that the parameterization in Eq. 6 and Eq. 7 places every pole associated with the base denominator in Eq. 5 strictly inside the unit disk.

First, consider the complex-conjugate factors. Eq. 6 maps the unconstrained parameters via the sigmoid function $\sigma(\cdot) \in (0, 1)$. Given $0 < \varepsilon < 1$, the pole radii are bounded by $0 < \rho_{g,k} < 1-\varepsilon < 1$, and the angles are constrained to $0 < \theta_{g,k} < \pi$. Consequently, each complex factor in Eq. 5 yields a conjugate pole pair $\rho_{g,k}e^{\pm j\theta_{g,k}}$ with $\left|\rho_{g,k}e^{\pm j\theta_{g,k}}\right| = \rho_{g,k} < 1$. Therefore, all complex-conjugate poles lie strictly inside the unit disk.

Similarly, for the real poles, Eq. 7 utilizes both $\sigma(\cdot) \in (0, 1)$ and $\tanh(\cdot) \in (-1, 1)$ to bound the parameters such that $0 < \bar{\rho}^R_{g,\ell} < 1 - \varepsilon < 1$ and $|\bar{s}_{g,\ell}| < 1$. The resulting real pole is defined as $a_{g,\ell} = \bar{s}_{g,\ell}\bar{\rho}^R_{g,\ell}$, which guarantees that $|a_{g,\ell}| = |\bar{s}_{g,\ell}|\,\bar{\rho}^R_{g,\ell} < 1$. Therefore, every real pole also lies strictly inside the unit disk. Since all real poles and all complex-conjugate poles associated with Q_g have magnitude strictly smaller than one, the fixed grouped denominator Q_g is Schur stable. □

Proof of Proposition 2. To prove Proposition 2, we must demonstrate that for each realized denominator $Q_{t,g}$, all real and complex pole factors in Eq. 10 possess magnitudes strictly less than one. Based on Eq. 8, the radius scale is strictly positive ($s^{(\rho)}_{t,g} > 0$) because $\delta_{\min} > 0$, $\delta_0 > 0$, and $\operatorname{softplus}(\cdot) > 0$. This establishes the uniform lower bound: $s^{(\rho)}_{t,g} \ge s_{\min} \triangleq \frac{\delta_{\min}}{\delta_0} > 0$. Given that the base radii from Eq. 6 and Eq. 7 are bounded above by $1 - \varepsilon$, the token-conditioned radii are uniformly bounded away from one by the margin: $\varepsilon_\rho \triangleq 1 - (1 - \varepsilon)^{s_{\min}} > 0$.

For the real poles, Eq. 7 guarantees $|\bar{s}_{g,\ell}| < 1$ and $0 < \bar{\rho}^R_{g,\ell} < 1 - \varepsilon$. Applying the strictly positive scale $s^{(\rho)}_{t,g}$, the modulated radius becomes:

$$0 < \exp\left(s^{(\rho)}_{t,g} \log \bar{\rho}^R_{g,\ell}\right) = \left(\bar{\rho}^R_{g,\ell}\right)^{s^{(\rho)}_{t,g}} \le (1 - \varepsilon)^{s_{\min}} = 1 - \varepsilon_\rho < 1. \tag{22}$$

Consequently, the magnitude of the updated real pole is: $|a_{t,g,\ell}| = |\bar{s}_{g,\ell}| \left(\bar{\rho}^R_{g,\ell}\right)^{s^{(\rho)}_{t,g}} < 1 - \varepsilon_\rho < 1$.

Similarly, for the complex poles, the base radius $0 < \rho_{g,k} < 1 - \varepsilon$ from Eq. 6 is updated via Eq. 9 to:

$$0 < \rho_{t,g,k} = \rho_{g,k}^{s^{(\rho)}_{t,g}} \le (1 - \varepsilon)^{s_{\min}} = 1 - \varepsilon_\rho < 1. \tag{23}$$

The associated poles have magnitude $\left|\rho_{t,g,k}e^{\pm j\theta_{t,g,k}}\right| = \rho_{t,g,k} \le 1 - \varepsilon_\rho < 1$. Hence, every token-conditioned complex pole also lies strictly inside the unit disk. The clipping operation in Eq. 9 keeps $\theta_{t,g,k}$ in $[0, \pi]$. This changes only the phase of the pole pair and does not change its magnitude.

Therefore, all poles associated with each realized $Q_{t,g}$ lie strictly inside the unit disk. By the Schur-stability condition stated above, every local grouped denominator is Schur stable. □

D Proof of Computational Cost Derivation

We derive the inference cost of a single scan route for both the proposed TCP-SSM and the standard Vision Mamba models. Total computation scales directly with the number of scan routes in the Vision Mamba module, which we keep consistent with the original Mamba architecture. Following the main manuscript, the input sequence is defined as $X \in \mathbb{R}^{B\times M\times E}$, where B is the batch size, M is the scan length, and E is the channel dimension. The proposed method divides channels into G groups, uses a denominator order r, and applies a low-rank numerator dimension r_f. All computation costs are written in normalized $BME(\cdot)$ form to make the per-token-channel complexity explicit.

D.1 Computational Cost of the Token-Conditioned Pole-Structured Selective SSM

The operator comprises a token-conditioned denominator, a low-rank numerator, a grouped recurrent update, and a diagonal direct path.

D.1.1 Low-rank Numerator Cost

For each token $x_t \in \mathbb{R}^E$, the numerator branch computes

$$\phi_t = V^\top x_t \in \mathbb{R}^{r_f}, \qquad V \in \mathbb{R}^{E\times r_f}, \tag{24}$$

$$\alpha_t = W_\alpha x_t \in \mathbb{R}^{r}, \qquad W_\alpha \in \mathbb{R}^{r\times E}, \tag{25}$$

$$\gamma_t = \sigma(W_\gamma x_t) \in \mathbb{R}^{r_f}, \qquad W_\gamma \in \mathbb{R}^{r_f\times E}. \tag{26}$$

It then forms the causal low-rank mixture via $\psi_t = \Phi_t\alpha_t = \sum_{i=1}^{r}\alpha_{t,i}\phi_{t-i} \in \mathbb{R}^{r_f}$, and projects it back to the channel dimension using $\eta_t = U(\gamma_t \odot \psi_t) \in \mathbb{R}^E$ $(U \in \mathbb{R}^{E\times r_f})$.

The multiplication count for one token is

$$V^\top x_t \ : \ Er_f, \qquad W_\alpha x_t \ : \ Er, \qquad W_\gamma x_t \ : \ Er_f, \tag{27}$$

$$\Phi_t\alpha_t \ : \ rr_f, \qquad \gamma_t \odot \psi_t \ : \ r_f, \qquad U(\gamma_t \odot \psi_t) \ : \ Er_f. \tag{28}$$

Therefore, over all B batches and M tokens, we obtain

$$F^{Comp}_{\text{num}} = BME\left(r + 3r_f + \frac{rr_f}{E} + \frac{r_f}{E}\right). \tag{29}$$

Since $r, r_f \ll E$, then the dominant per-token-channel numerator cost is $BME(r + 3r_f)$.

D.1.2 Token-conditioned Denominator Cost

The denominator for group g is derived using Eq. 10. The denominator computation has two parts. First, the model constructs the coefficients $q_{t,g,1:r}$ once per token and group. Second, these coefficients are reused in the recurrent scan for all $E_g = E/G$ channels in group g. As discussed in Section 3.2, we predict positive radius scale $s^{(\rho)}_{t,g}$ and a bounded angle scale $s^{(\theta)}_{t,g}$ (Eq. 8), to introduce the selectivity for the defined poles. We consider two different cases to compute the inference cost of the denominator:

Case 1: Group-specific Scale Heads: In this case, each group has its own learned row such that:

$$W_\rho \in \mathbb{R}^{G\times E}, \qquad W_\theta \in \mathbb{R}^{G\times E}, \qquad b_\rho \in \mathbb{R}^{G}, \qquad b_\theta \in \mathbb{R}^{G}. \tag{30}$$

This will give each group-specific poles more freedom to scale based on the local input. For one token, computing each of $W_\rho x_t$, and $W_\theta x_t$ costs EG. Thus, the total cost for Eq. 8 can be written as:

$$F^{Comp}_{\text{scale,group}} = BME(2G). \tag{31}$$

Case 2: Shared Scale Heads: In this case, the same scale head is used for all groups and then broadcast, such that:

$$W_\rho \in \mathbb{R}^{1\times E}, \qquad W_\theta \in \mathbb{R}^{1\times E}, \qquad b_\rho \in \mathbb{R}, \qquad b_\theta \in \mathbb{R}. \tag{32}$$

Thus, we have $s^{(\rho)}_{t,g} = s^{(\rho)}_t$ and $s^{(\theta)}_{t,g} = s^{(\theta)}_t$ for all g. The base poles remain group-specific in both cases, so $Q_{t,g}$ can still differ across groups. The shared-head design only shares the token-conditioned scale prediction. For one token, computing each of $W_\rho x_t$ and $W_\theta x_t$ costs E. Therefore, the total

cost for Eq. 8 can be written as $F^{Comp}_{\text{scale,shared}} = 2BME$. Based on the ablation study in Table 5, we use shared scale in the final model. The shared scale-head design preserves strong performance while reducing the scale-conditioning overhead from $BME(2G)$ to $BME(2)$.

Real-pole Modulation: As mentioned in the main paper, the token-conditioned pole is $a_{t,g,\ell} = \bar{s}_{g,\ell} \exp(s^{(\rho)}_{t,g} \log \bar{\rho}^{R}_{g,\ell})$ for each real pole ℓ. Thus, computing the scaled real poles for all G group and L poles is $F^{Comp}_{\text{real,mod}} = BME\left(\frac{2GL}{E}\right)$

Complex-pole Modulation: For each complex pole k, we follow the Eq. 9 to compute the token-conditioned radius and angle. Thus, computing the scaled complex poles for all G group and K poles is $F^{Comp}_{\text{complex,mod}} = BME\left(\frac{5GK}{E}\right)$.

Polynomial Expansion: After pole modulation, the factorized denominator in Eq. 10 is converted into its coefficient form using $Q_{t,g}(z^{-1}) = 1 + \sum_{i=1}^{r} q_{t,g,i} z^{-i}$. We obtain these coefficients by sequentially multiplying the real and complex factors in the grouped denominator.

For a real factor $1 - az^{-1}$, the coefficient update is

$$q \leftarrow \text{pad}(q, 1) - a\ \text{shift}_1(q). \tag{33}$$

After inserting the j-th real factor, the polynomial has length $j+1$. The corresponding update therefore requires $j+1$ multiplications. Summing over all L real poles yields $F^{Comp}_{\text{real,expand}} = BMG\sum_{j=1}^{L}(j+1)$. Since we have $\sum_{j=1}^{L}(j+1) = \frac{L^2+3L}{2}$, therefore the total computation of coefficient generation for real factor is $F^{Comp}_{\text{real,expand}} = BME\left[\frac{G}{E}\left(\frac{L^2+3L}{2}\right)\right]$. Combining real-pole modulation with real-factor expansion gives a total computation of $F^{Comp}_{\text{real}} = BME\left[\frac{G}{E}\left(\frac{L^2+7L}{2}\right)\right]$.

For a complex factor $(1 - p_1 z^{-1} + p_2 z^{-2})$, the coefficient update becomes

$$q \leftarrow \text{pad}(q, 2) - p_1\ \text{shift}_1(q) + p_2\ \text{shift}_2(q). \tag{34}$$

After all real factors have been inserted, the polynomial degree is L. Inserting the j-th complex factor increases the length to $L + 2j + 1$. Since both p_1 and p_2 scale the coefficient vector, the j-th complex update requires $2(L + 2j + 1)$ multiplications. Summing over all K complex poles gives $F^{Comp}_{\text{complex,expand}} = BMG\sum_{j=1}^{K} 2(L+2j+1)$. The summation can be simplify to $F^{Comp}_{\text{complex,expand}} = BMG(2LK + 2K^2 + 4K)$. Combining complex-pole modulation with complex-factor expansion gives

$$F^{Comp}_{\text{complex}} = BME\left[\frac{G}{E}(2LK + 2K^2 + 9K)\right]. \tag{35}$$

Combining the real and complex contributions, we define

$$\Phi_{\text{den}} = \frac{L^2 + 7L}{2} + 2LK + 2K^2 + 9K. \tag{36}$$

The cost of pole modulation together with denominator expansion is then

$$F^{Comp}_{\text{coeff}} = BME\left(\frac{G\Phi_{\text{den}}}{E}\right). \tag{37}$$

Combining Eq. 37 with the cost of selective pole generation (shared W_ρ and W_θ), the total cost of denominator-construction becomes

$$F^{Comp}_{\text{den,construct,shared}} = BME\left(2 + \frac{G\Phi_{\text{den}}}{E}\right). \tag{38}$$

Denominator Recurrent Scan Cost: Once the coefficients $q_{t,g,1:r}$ have been constructed, the grouped recurrent branch is given by

$$y^{(g)}_t = \eta^{(g)}_t - \sum_{i=1}^{r} q_{t,g,i}\, y^{(g)}_{t-i}. \tag{39}$$

At token t, only $\min(r, t-1)$ past states are available. The computation count of the recurrent scan is therefore $F^{Comp}_{\text{scan}} = BE\sum_{t=1}^{M}\min(r, t-1)$. For $M \geq r$, the summation simplifies to

$$\sum_{t=1}^{M}\min(r, t-1) = Mr - \frac{r(r+1)}{2}. \tag{40}$$

Hence, we obtain

$$F_{\text{scan}}^{Comp} = BME\left(r - \frac{r(r+1)}{2M}\right). \tag{41}$$

D.1.3 Direct Path Cost

The final output is $o_t = y_t + Dx_t$, where $D \in \mathbb{R}^{E\times E}$ is diagonal. Thus, the cost of direct path is $F_{\text{direct}}^{Comp} = BME(1)$.

D.1.4 Total Cost of the Proposed Method

The total multiplication count of the proposed method is obtained by summing the costs of the numerator branch, denominator construction, recurrent scan, and direct path,

$$F_{\text{TCP}}^{Comp} = F_{\text{num}}^{Comp} + F_{\text{den,construct,shared}}^{Comp} + F_{\text{scan}}^{Comp} + F_{\text{direct}}^{Comp}. \tag{42}$$

There, for shared W_ρ and W_θ, we obtain

$$F_{\text{TCP}}^{Comp} = BME\left(r + 3r_f + \frac{rr_f}{E} + \frac{r_f}{E}\right) + BME\left(2 + \frac{G\Phi_{\text{den}}}{E}\right) + BME\left(r - \frac{r(r+1)}{2M}\right) + BME. \tag{43}$$

Collecting terms gives

$$F_{\text{TCP}}^{Comp} = BME\left(2r + 3r_f + 3 + \frac{rr_f + r_f + G\Phi_{\text{den}}}{E} - \frac{r(r+1)}{2M}\right). \tag{44}$$

Since $r, r_f, L, K \ll E$, we can simplify the F_{TCP}^{Comp} by only considering the dominant terms, resulting in $F_{\text{TCP}}^{Comp} \approx BME(2r + 3r_f + 3)$.

D.2 Computational Cost of Vision Mamba baseline

We now derive the inference cost of a single SSM scan for the selective Vision Mamba baseline.

Let N represent the latent SSM state size and r_Δ represent the step-size projection rank. For each token x_t, a standard Vision Mamba selective layer first generates a low-rank step-size representation alongside the token-dependent input and output vectors B_t and C_t. The model then expands the step size to the channel dimension before applying the selective scan. The initial projection maps the input token $x_t \in \mathbb{R}^E$ to three quantities. These are $\tilde{\Delta}_t \in \mathbb{R}^{r_\Delta}$, $B_t \in \mathbb{R}^N$, and $C_t \in \mathbb{R}^N$. Equivalently, this joint projection is written as

$$\begin{bmatrix} \tilde{\Delta}_t \\ B_t \\ C_t \end{bmatrix} = W_x x_t, \qquad W_x \in \mathbb{R}^{(r_\Delta + 2N)\times E}. \tag{45}$$

We can also decompose this operation into three separate projections. These are $\tilde{\Delta}_t = W_x^\Delta x_t$ with $W_x^\Delta \in \mathbb{R}^{r_\Delta \times E}$, $B_t = W_B x_t$ with $W_B \in \mathbb{R}^{N\times E}$, and $C_t = W_C x_t$ with $W_C \in \mathbb{R}^{N\times E}$. For a single token, the total cost for this step breaks down to Er_Δ for $\tilde{\Delta}_t$, EN for B_t, and EN for C_t. Scaling these operations over all B batches and M tokens yields a total projection cost of $F_{\text{x-proj}}^{Comp} = BM(Er_\Delta + 2EN)$. Factoring by BME simplifies this expression to $F_{\text{x-proj}}^{Comp} = BME(r_\Delta + 2N)$.

The low-rank step-size representation is subsequently projected back to the channel dimension via $\Delta_t = W_\Delta \tilde{\Delta}_t$, where $W_\Delta \in \mathbb{R}^{E\times r_\Delta}$ and $\Delta_t \in \mathbb{R}^E$. For a single token, this expansion requires Er_Δ multiplications. Scaling this operation across all dimensions yields a total step-size projection cost of $F_{\Delta\text{-proj}}^{Comp} = BME(r_\Delta)$.

For each channel $e \in \{1, \ldots, E\}$ and state index $n \in \{1, \ldots, N\}$, the selective scan forms the discretized transition $\bar{A}_{t,e,n} = \exp(\Delta_{t,e} A_{e,n})$. While the exponential operation itself is excluded from the cost count, calculating the scalar product $\Delta_{t,e} A_{e,n}$ requires one multiplication. Scaling this operation across all dimensions yields a total cost of $F_{\Delta A}^{Comp} = BME(N)$.

The recurrent state update $h_{t,e,n} = \bar{A}_{t,e,n} h_{t-1,e,n} + \bar{B} x_{t,e,n}$ incorporates the input injection term $\bar{B} x_{t,e,n} = \Delta_{t,e} B_{t,n} x_{t,e}$. Computing this injection requires two multiplications per state entry, namely $\Delta_{t,e} B_{t,n}$ and $(\Delta_{t,e} B_{t,n}) x_{t,e}$, which scales to $BME(2N)$. When combined with the prior transition scaling, the total computational cost for the recurrent state update becomes $F_{state}^{Comp} = BME(3N)$.

Table 6: ImageNet-1K classification training hyperparameters for TCP-SSM models. For Vim-Tiny, the base learning rate is linearly scaled by the global batch size as lr $\leftarrow$ lr $\times$ batch size $\times$ world size$/512$.

Hyperparameter	**TCP-Vim-Tiny**	**TCP-EfficientVMamba-Tiny**	**TCP-EfficientVMamba-Base**
Dataset	ImageNet-1K	ImageNet-1K	ImageNet-1K
Input resolution	224×224	224×224	224×224
Batch size	384	256	256
Number of workers	20	4	4
Training epochs	600	700	250
Optimizer betas	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.999)$
Base learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}
Warmup learning rate	1×10^{-6}	1×10^{-6}	1×10^{-6}
Warmup epochs	50	5	10
Learning-rate schedule	Cosine	Cosine	Cosine
Weight decay	1×10^{-4}	1×10^{-3}	1×10^{-4}
Gradient clipping	50	50	50
Drop path rate	0.1	0.2	0.2
EMA decay	0.99996	0.9999	0.9999
Training precision	BF16 AMP	BF16 AMP	BF16 AMP
TCP selective params	$0.5\times$ LR, no weight decay	$0.5\times$ LR, no weight decay	$0.5\times$ LR, no weight decay
ϵ in Eq. 6 and 7	1×10^{-3}	1×10^{-3}	1×10^{-3}
δ_0 in Eq. 8	1	1	1
δ_{min} in Eq. 8	1×10^{-6}	1×10^{-6}	1×10^{-6}
λ_θ in Eq. 8	0.25	0.25	0.25

The intermediate output is derived by reading out the state using the token-dependent vector C_t via $\tilde{y}_{t,e} = \sum_{n=1}^{N} h_{t,e,n} C_{t,n}$. This step requires N multiplications per token-channel, yielding a cost of $F^{Comp}_{C\text{-readout}} = BME(N)$. Following this, the final output incorporates a diagonal direct path $o_t = y_t + Dx_t$ with $D \in \mathbb{R}^{E\times E}$, incurring a cost of $F^{Comp}_{\text{direct}} = BME(1)$.

Summing the costs of the initial projections, step-size expansion, selective-scan core, and direct path provides the total inference cost for a single scan route as

$$F^{Comp}_{\text{Vim}} = BME(2r_\Delta + 7N + 1). \tag{46}$$

The original Vision Mamba sets $r_\Delta = \lceil E/32 \rceil$ and $N = 16$. In TCP-SSM, the denominator order r and numerator rank r_f are independent design parameters selected according to the target computation–accuracy trade-off. Lower-compute configurations use smaller r and r_f, while higher-capacity variants increase these values. Therefore, the efficiency of TCP-SSM follows from the overall operator cost in Eq. 44.

E Experimental Details

E.1 Implementation Details

We provide the complete training hyperparameters used in our experiments. Table 6 summarizes the ImageNet-1K classification settings for TCP-Vim-Tiny and TCP-EfficientVMamba variants. Table 7 reports the corresponding downstream settings for semantic segmentation on ADE20K and object detection/instance segmentation on COCO using TCP-Vim-Tiny.

During training, the token-conditioned poles change the denominator coefficients at every token and group. This can also change the overall gain of the recurrent branch, not only its memory behavior. To keep the scale of the scan stable, we apply a lightweight denominator gain normalization before the recurrent scan.

After expanding the token-conditioned denominator as $Q_{t,g}(z^{-1}) = 1 + \sum_{i=1}^{r} q_{t,g,i} z^{-i}$, we compute its value at $z = 1$, such that $Q_{t,g}(1) = 1 + \sum_{i=1}^{r} q_{t,g,i}$. We also compute the corresponding base value $Q_{1,g}(1)$ using identity modulation, where $s^{(\rho)}_{t,g} = 1$ and $s^{(\theta)}_{t,g} = 1$. The driving signal for group g is then multiplied by

$$\omega_{t,g} = \text{clip}\left(\frac{Q_{t,g}(1)}{Q_{1,g}(1) + 10^{-6}}, \omega_{min}, \omega_{max}\right). \tag{47}$$

Table 7: Downstream training hyperparameters for TCP-Vim-Tiny on semantic segmentation and object detection/instance segmentation.

Hyperparameter	Semantic Segmentation	Object Detection / Instance Segmentation
Dataset	ADE20K	COCO
Task head	UPerNet	Cascade Mask R-CNN
Backbone	TCP-Vim-Tiny	TCP-Vim-Tiny
Input resolution	512×512	1024×1024
Training augmentation	Random resize $(0.5, 2.0)$, random crop, random flip 0.5, photometric distortion	LSJ: random flip, resize scale $(0.1, 2.0)$, fixed-size crop 1024×1024
Total batch size	24	64
Workers per GPU	16	16
Base learning rate	1×10^{-4}	1×10^{-4}
Weight decay	0.01	0.1
Optimizer betas	$(0.9, 0.999)$	$(0.9, 0.999)$
Layer-wise LR decay	0.92	0.837
LR schedule	Polynomial decay	Multi-step decay
Training iteration	180k	185K
ϵ in Eq. 6 and 7	1×10^{-3}	1×10^{-3}
δ_0 in Eq. 8	1	1
δ_{min} in Eq. 8	1×10^{-6}	1×10^{-6}
λ_θ in Eq. 8	0.25	0.25

Table 8: Matched-compute comparison with compact conventional SSMs under identical feature-level distillation.

Model	Distill.	#Param.	SSM FLOPs	Top-1	Top-5
Vim-T ($N = 16$)	Teacher	7.1M	497.5M	76.1	93.0
Vim-T ($N = 8$)	✓	6.7M	294.1M	75.1	92.5
TCP-Vim-T1	✓	**7.5M**	**295.3M**	**76.0**	**93.1**
EVM-T ($N_1 = N_2 = 16$)	Teacher	6.5M	109.3M	76.3	93.1
EVM-T ($N_1 = 8, N_2 = 10$)	✓	6.3M	62.9M	75.4	92.6
TCP-EVM-T1	✓	**6.4M**	**62.9M**	**76.3**	**93.0**

This normalization approximately preserves the low-frequency gain of the recurrent branch relative to the base denominator. In particular, since a unit-numerator recurrence has gain proportional to $1/Q_{t,g}(1)$ at $z = 1$, scaling the driving signal by $Q_{t,g}(1)/Q_{1,g}(1)$ keeps the effective gain close to the base configuration. The normalization factor is detached from the computational graph and clipped for numerical robustness. It therefore stabilizes the recurrent scale without changing the learned pole parameterization.

E.2 Matched-Compute Conventional SSM Controls

To isolate the contribution of the proposed recurrence from feature-level distillation, we train compact conventional SSM baselines using the same teacher, distilled layers, training schedule, and optimization settings as TCP-SSM. Only the conventional state dimension is reduced to match the recurrent computation of TCP. Importantly, all TCP-specific recurrence parameters, including the pole bank, token-modulation heads, grouped denominator, low-rank numerator, and direct path, are newly initialized; the original model is used only as a frozen, stop-gradient teacher during training and is absent at inference.

For Vim-T, reducing the conventional state dimension from $N = 16$ to $N = 8$ reduces SSM FLOPs from 497.5M to 294.1M, but decreases Top-1 accuracy from 76.1% to 75.1%. At essentially the same recurrent computation (295.3M FLOPs), TCP-Vim-T1 reaches 76.0%, recovering 0.9 of the 1.0 percentage point lost by conventional state reduction. Moreover, TCP-Vim-T1 without distillation already reaches 75.3% (Table 4), exceeding the distilled compact conventional baseline at 75.1%. Identical feature guidance then increases TCP to 76.0%.

The same behavior is independently observed on EfficientVMamba. The compact conventional EVM-T baseline exactly matches TCP-EVM-T1 at 62.9M SSM FLOPs and has a similar parameter

Table 9: End-to-end inference efficiency on one NVIDIA A40. Lower latency and peak memory and higher throughput are better.

Model Pair	Batch	Base ms	TCP ms	Speedup	Base img/s	TCP img/s	Base MiB	TCP MiB
Vim-T / TCP-Vim-T1	128	35.2	**24.3**	**1.5**×	3635.3	**5259.5**	370.8	**335.2**
	256	73.8	**52.3**	**1.4**×	3469.2	**4894.1**	695.1	**616.5**
	512	163.4	**124.6**	**1.3**×	3134.2	**4109.0**	1342.9	**1176.7**
EVM-T / TCP-EVM-T1	128	22.3	**18.4**	**1.2**×	5730.4	**6958.9**	377.9	**323.4**
	256	50.6	**42.1**	**1.2**×	5057.1	**6082.7**	706.4	**692.4**
	512	110.0	**94.4**	**1.2**×	4652.8	**5420.9**	926.9	**874.4**
EVM-B / TCP-EVM-B3	128	59.0	**50.8**	**1.2**×	2170.5	**2521.8**	724.3	**669.2**
	256	131.3	**116.0**	**1.1**×	1950.3	**2207.1**	1294.5	**1184.0**
	512	266.6	**240.5**	**1.1**×	1920.7	**2129.0**	1364.5	**1298.2**

count, but reaches 75.4% Top-1 compared with 76.3% for TCP. Thus, TCP recovers the entire 0.9-point accuracy loss caused by conventional state reduction at the same recurrent-compute budget.

These comparisons hold teacher supervision, training schedule, and recurrent computation fixed. The consistent gains across two distinct backbone families therefore show that the accuracy preservation of TCP-SSM is not explained by distillation alone, but depends on the proposed recurrent parameterization.

E.3 End-to-End Runtime and Memory Efficiency

SSM FLOPs isolate the recurrent operator modified by TCP-SSM and therefore directly quantify the computational reduction introduced by our method. To verify that these operator-level savings also translate into practical end-to-end gains, we additionally measure complete-model inference latency, throughput, and peak GPU memory on one NVIDIA A40 under identical software, precision, and inference settings for each baseline–TCP pair.

The reduction in recurrent computation translates consistently into end-to-end gains across all evaluated backbones and batch sizes. TCP-Vim-T1 reduces latency by 23.7–31.0% and increases throughput by 31.1–44.7%. TCP-EVM-T1 reduces latency by 14.2–17.5% and increases throughput by 16.5–21.4%, while TCP-EVM-B3 reduces latency by 9.8–13.9% and increases throughput by 10.8–16.2%. Peak GPU memory is also reduced for every model pair and batch size. These measurements confirm that the lower recurrent arithmetic cost of TCP-SSM translates into measurable full-model acceleration and memory savings, rather than remaining only a theoretical FLOP reduction.

These results are obtained using our current reference implementation, without a dedicated fused TCP kernel, parallel scan kernel, or hardware-specific scheduling. The order-r recurrence is evaluated sequentially along the scan dimension, while computation across batches, groups, and channels is parallelized on the GPU. Thus, the measurements serve as a practical end-to-end validation of the computational savings provided by TCP-SSM under the current implementation, rather than as a hardware-optimized upper bound on its performance.

As discussed in the Limitations and Future Work section, optimizing memory access, scan execution, kernel fusion, and hardware-specific scheduling is outside the scope of this work and remains an important direction for future study. Importantly, TCP-SSM already provides consistent end-to-end speedups of up to $1.5\times$ and lower peak memory without such dedicated optimization, demonstrating that its structured reduction in recurrent computation produces practical gains even before hardware-aware implementation.

E.4 Detailed TCP-SSM Configurations

Table 10 summarizes the detailed TCP-SSM configurations applied across all Vim and EfficientVMamba variants. For every model, we detail the number of real pole families L, the number of complex pole families K, the total denominator order r, the numerator rank r_f, and the number of groups G. The Vim variants maintain a uniform configuration throughout the backbone. In contrast, the EfficientVMamba variants employ stage-specific settings, denoted as Stage 1 and Stage 2 within each entry. Generally across these models, increasing L, K, and r_f produces a richer recurrent parameterization and a higher-capacity numerator branch, whereas the grouping parameter G dictates the extent of shared structured recurrence among channels. Thus, this table clarifies how the various

Table 10: Configurations of the proposed TCP-SSM. Num. denotes numerator.

Model	Stage	Real poles L	Complex poles K	Total poles r	Num. rank r_f	Groups G
TCP-Vim-T1	–	2	3	8	20	12
TCP-Vim-T2	–	4	5	14	30	12
TCP-EVM-T1	S1	3	2	7	16	3
	S2	3	2	7	18	6
TCP-EVM-B1	S1	3	2	7	16	6
	S2	4	2	8	20	12
TCP-EVM-B2	S1	6	6	18	26	6
	S2	6	6	18	28	12
TCP-EVM-B3	S1	4	5	14	22	6
	S2	4	5	14	24	12
TCP-EVM-B4	S1	2	2	6	10	6
	S2	2	2	6	12	12
TCP-EVM-B5	S1	2	1	4	8	6
	S2	2	1	4	10	12

Table 11: Ablation study on the number of groups G for TCP-Vim-T1.

Method	Groups G	#Param.	SSM FLOPs	Top-1 (%)	Top-5 (%)
Vim-T	–	7.1M	497.5M	76.1	93.0
TCP-Vim-T1	6	7.49M	291.6M	75.8	93.0
TCP-Vim-T1	24	7.51M	302.9M	75.9	93.1
TCP-Vim-T1	**12**	**7.50M**	**295.3M**	**76.0**	**93.1**

TCP-EVM-B variants transition from lightweight to computationally heavy recurrent configurations during the ablation study, and it illustrates how the optimal settings achieve a balance between modeling capacity and efficiency.

E.5 More Ablation Study

Effect of Group Number (G): Table 11 studies the effect of the number of groups G in TCP-Vim-T1. Varying G changes how channels share the grouped denominator, and therefore controls the granularity of the recurrent parameterization. With $G = 6$, the model achieves 75.8% Top-1 and 93.0% Top-5, while increasing the number of groups to $G = 24$ gives 75.9% Top-1 and 93.1% Top-5. In practice, the $G = 24$ setting is more challenging to converge and shows a modest degradation relative to the selected configuration. We therefore use $G = 12$, which achieves the best Top-1 accuracy of 76.0% while keeping the SSM FLOPs close to the other grouped variants. This suggests that an intermediate grouping level provides a better balance between sharing recurrent structure and preserving sufficient channel-wise flexibility.

E.6 Qualitative Analysis

Figures 4, 5 visualize the token-conditioned dynamics learned by TCP-SSM in TCP-Vim-T1 and TCP-Vim-T2. Panel (a) shows the input image together with the patch-token grid from ImageNet validation set. Panel (b) overlays the token-wise memory horizon, computed from the dominant effective pole radius as $\tau = -1/\log(\rho_{\max})$, where larger values indicate tokens whose recurrent dynamics preserve information over longer scan ranges. The highlighted tokens are selected to represent distinct dynamical regimes, including long-memory, short-memory, and high-frequency/oscillatory behavior. Panel (c) plots the corresponding effective TCP poles in the complex plane: black crosses denote the learned base poles, while colored markers show how the token-conditioned modulation moves the poles for the selected tokens while keeping them inside the unit circle for stability. Panel (d) reports the normalized local impulse responses obtained by freezing the token-conditioned denominator at each selected token, illustrating how different pole placements translate into different temporal behaviors, such as slow decay, rapid forgetting, or oscillatory responses. Together, these visualizations

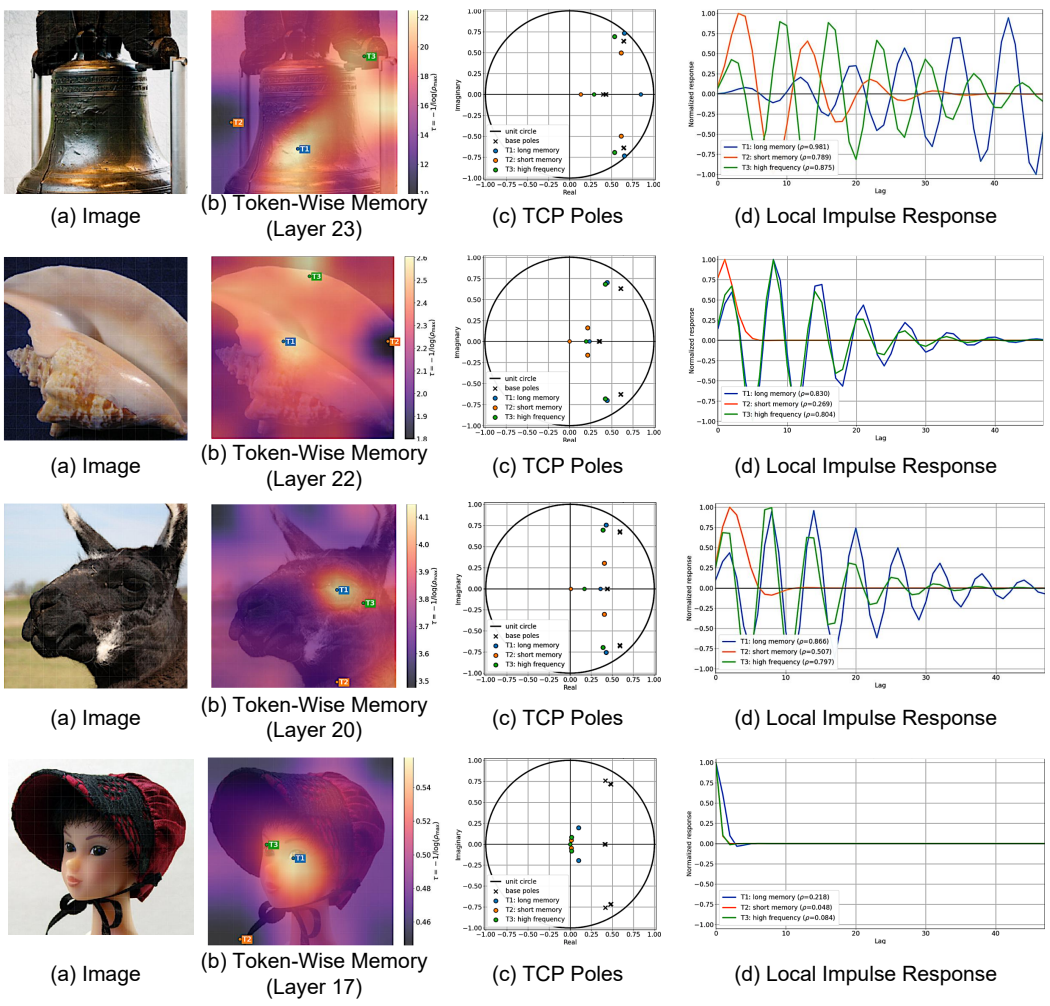


Figure 4: Qualitative Analysis of TCP-Vim-T1.

show that TCP-SSM learns spatially varying and interpretable recurrent dynamics, allowing different image regions and layers to adapt their memory profiles according to local visual content.

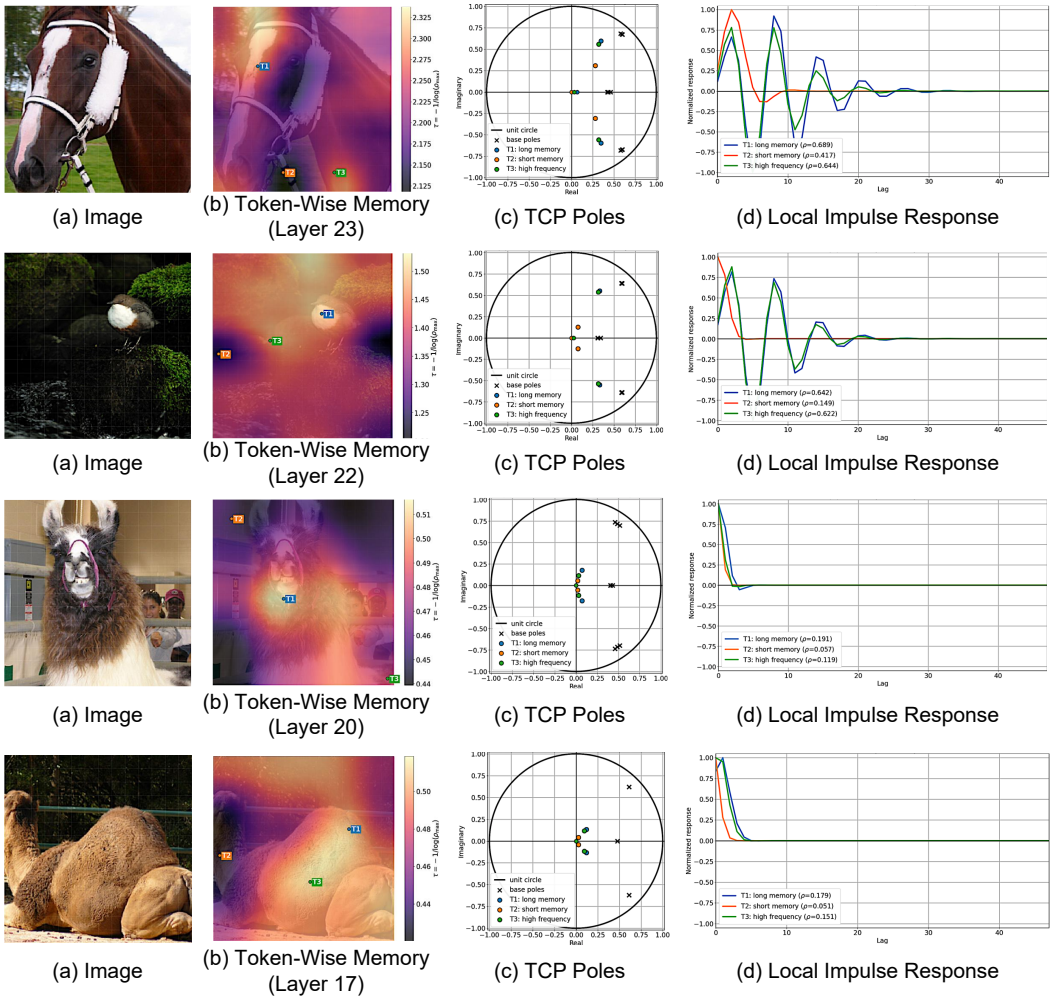


Figure 5: Qualitative Analysis of TCP-Vim-T2.