RNN as Linear Transformer: A Closer Investigation into Representational Potentials of Visual Mamba Models

Supplementary Material

Appendix

A. More Feature Map Visualization

Due to space limitations in the main text, this section provides supplementary visualizations of the feature maps for the Transformer [51]/Mamba [11] models presented in the main text, as shown in Figure 8, 9, 11, 10, 12.

B. Visualizing Long-Range Dependencies

In this section, utilizing Mamba-Reg [52], we discovered that inserting register tokens at different positions allows Mamba to exhibit a multi-head-like characteristic similar to Transformers, as shown in Figure 7. Each register token functions akin to an attention head, with even the forward-placed register tokens (e.g. Image 2, Index 2) able to perceive distant target objects within the image. This demonstrates Mamba's ability to capture global information, transcending the limitations of local token interactions.

Evaluation dataset for feature maps To accurately evaluate the quality of the model's performance in visualization, we use ImageNet-S [15] as the evaluation dataset. This dataset contains 919 categories, with 1,183,322 images for training, 12,419 for validation, and 27,423 for testing. Each image is accompanied by high-quality semantic segmentation annotations, enabling a detailed analysis of the quality and interpretability of feature maps. In our setting, we use only the validation set for evaluation.

C. Additional Implementation Details

‘ To further investigate the performance of Mamba as a pretrained model in the Dino setting, we conduct two downstream experiments: Semantic Segmentation, and Object Detection and Instance Segmentation.

Semantic Segmentation Settings. We conduct the semantic segmentation task on the ADE20K dataset [64], which comprises 20K training images, 2K validation images, and 3K test images, encompassing 150 fine-grained semantic categories. For our experiments, we utilize the UperNet framework [58] as the baseline. The optimizer is configured as AdamW [35] with a learning rate of 6e-5, momentum parameters of (0.9, 0.999), and a weight decay of 0.05. The learning rate scheduling includes two phases: initially, LinearLR is applied for the first 1500 steps with a starting factor of 1e-6. Following this, the main training phase utilizes a PolyLR scheduler up to 160,000 steps.

Object Detection and Instance Segmentation Settings We perform object detection and instance segmentation experiments on the COCO 2017 dataset [32], which includes 118K training, 5K validation, and 20K test images. Cascade Mask R-CNN [4] is adopted as the base framework. The optimization setup uses AdamW [35] with a learning rate of 0.0001 and a weight decay of 0.1, where specific parameters, such as biases and positional embeddings, have decay multipliers set to zero to avoid excess regularization. The learning rate schedule comprises two phases: initially, LinearLR is applied for the first 500 steps to gradually increase the learning rate, followed by MultiStepLR, which decays the learning rate at epochs 8 and 11, ensuring stable convergence during training. The images are resized to 1333×800 pixels for training, validation, and testing, ensuring consistency across these stages.

LinearViT Baseline. The LinearViT baseline replaces $\mathrm{softmax}(QK^T)$ with $\psi(Q)\psi(K)^T$, where $\psi(\cdot)$ denotes the softmax function (i.e., $\psi(Q) = \mathrm{softmax}(Q)$ and $\psi(K) = \mathrm{softmax}(K)$), following the standard linear attention formulation [30]. All other architectural components remain identical to ViT-B, including model dimension (768), number of heads (12), number of layers (12), and total parameter count (85M). The training pipeline also follows the same DINO pretraining setup described in Section 5, with no additional modifications. This controlled design ensures that any observed performance differences between LinearViT-B and ViT-B are solely attributable to the token-mixing mechanism, providing a clean empirical validation of our theoretical rank hierarchy.

Computational Cost. All models adopt unmodified standard architectures from their respective original works [30, 51, 52, 66], and their computational costs strictly follow prior works. No additional modules or operations are introduced during pretraining, ensuring fair comparison across all architectures.

D. More visualization of PCA

This figure 12 presents the PCA visualization of different attention mechanisms, where color distribution reflects how each model captures features in the representation space. The PCA results of self-attention clearly distinguish different parts of the object, providing a well-defined separation of features. In contrast, Mamba appears slightly more blurred than self-attention, with lower differentiation across object regions. Linear attention, on the other hand, exhibits a more sparse PCA representation and struggles to effectively separate different parts using color. This suggests that each mechanism focuses on different aspects of feature ex-

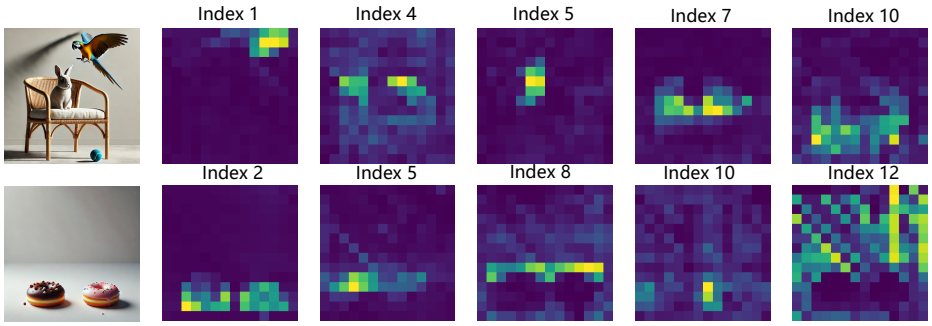

Figure 7. Feature maps corresponding to different register tokens, which are evenly distributed among sequence tokens, reveal a role similar to multi-head attention. Forward-placed register tokens (e.g., Image 2, Index 2) capture global patterns, while later tokens (e.g., Index 10) focus on specific regions, demonstrating Mamba's balance of global and local information.

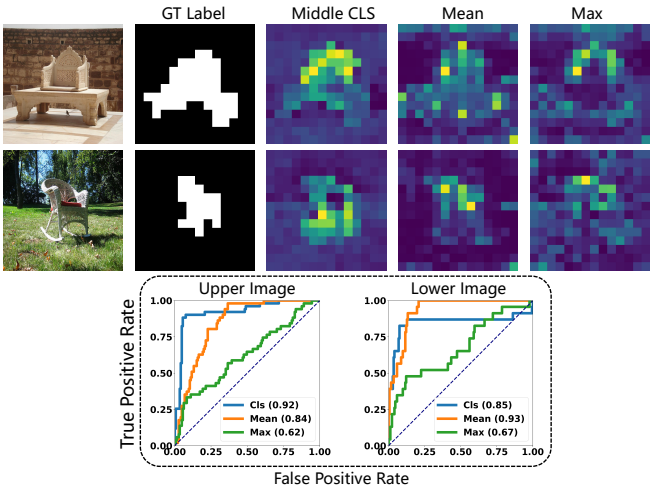

Figure 8. Feature map comparison in DinoVim: The Middle cls captures global information with clearer feature maps. Bottom: ROC curve comparisons.

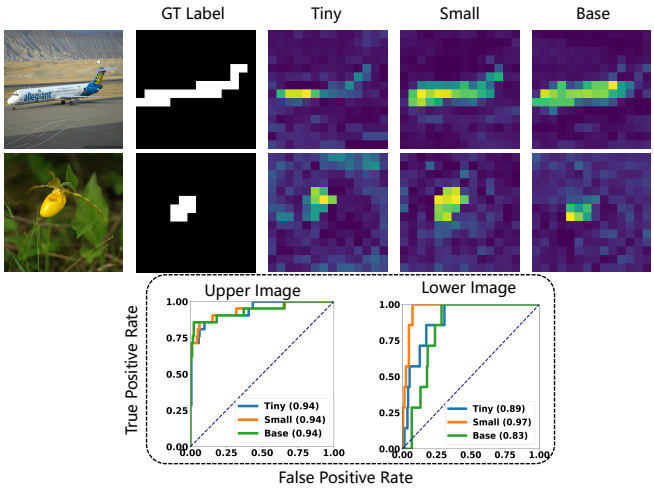

Figure 9. Feature map comparison in DinoVim across different model sizes, the small model displays the clearest feature maps. Bottom: ROC curve comparisons.

traction, and the PCA visualization further highlights their distinct information distributions.

E. Visual Mamba in the DINO Framework

This section supplements the previous discussion with additional details and visualization. As shown in Figure 13, the model employs bidirectional scanning with register tokens for feature aggregation. Vim uses a single register token, while Mamba-Reg introduces multiple tokens for hierarchical learning. The visualization highlights how positional encoding and token interactions enhance representation learning.

F. Hyper-parameter ablation

In the ablation study, we systematically evaluate various design choices through controlled experiments on the small DinoVim model.

Latent feature expansion for linear probing. In linear probing protocols, a common way to improve predictive performance is to concatenate the output of multiple blocks as an expanded latent feature. In our experiments, by default we follow DINO [6]'s approach to combine the output of the last four model blocks, and we summarize the ablation results of this mechanism in the left section of Table 5. As shown, the top-1 accuracy improves with more concatenated blocks, peaking at 77.2% with 4 blocks.

#blocks	acc	wd	acc	lr	acc
1	71.8	0	77.2	2.5e-4	77.1
2	75.6	1e-2	75.7	5e-4	77.2
3	76.7	5e-3	76.4	1e-3	77.3
4	**77.2**	**1e-3**	**77.3**	**2e-3**	**77.4**

Table 5. Comparison of top-1 accuracy for different hyperparameters: **Left**: Number of blocks. **Middle**: Weight decay (fixed lr = 1e-3). **Right**: Learning rate (fixed wd = 1e-3).

Learning rate and weight decay. As shown in the mid-

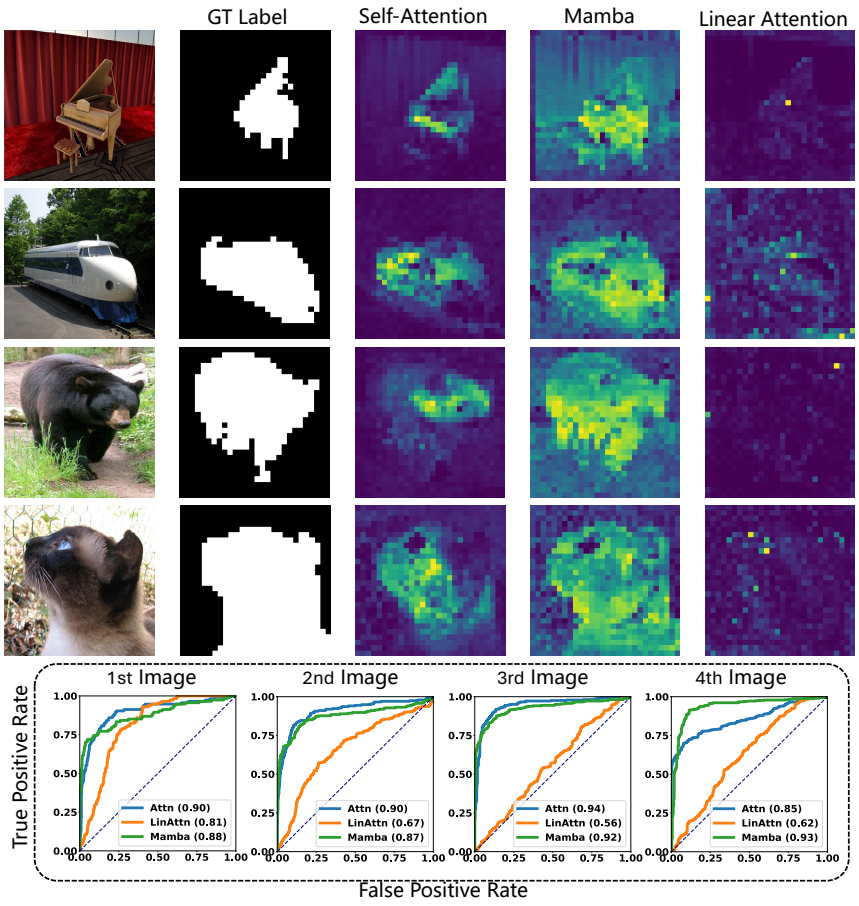

Figure 10. Feature maps comparison among Self-Attention, Mamba, and Linear Attention shows that Self-Attention provides the clearest target-background distinction, Mamba captures moderate clarity with global patterns, while Linear Attention exhibits blurred boundaries and weaker separation.

dle/right section of Table 5, top-1 accuracy varies with different `lr` and `wd` combinations. Fixing `lr` = 1e-3, increasing `wd` from 0 to 1e-3 raises accuracy from 77.2 to 77.3. Likewise, with `wd` = 1e-3, increasing `lr` from 2.5e-4 to 2e-3 achieves the highest accuracy of 77.4. The optimal setting (`lr` = 2e-3, `wd` = 1e-3) yields the best 77.4%.

Ablation of Classification Design As shown in Table 6, the *middle cls* strategy achieves the highest top-1 accuracy at 77.2, outperforming the *mean* (75.9) and *max* (74.6) strategies. Combined strategies, such as *middle cls, mean* (76.5) and *middle cls, max* (76.3), offer slight improvements over *mean* or *max* alone but fall short of the *middle cls*'s standalone performance. This indicates that the *middle cls* alone possesses sufficient representational capacity to effectively summarize the feature map.

Cls strat.	Top-1 acc
mean	75.9
max	74.6
middle cls, mean	76.5
middle cls, max	76.3
middle cls	**77.2**

Table 6. Top-1 accuracy of different classification strategies.

G. Mathematical Foundations for Rank Analysis

This appendix provides the formal mathematical foundations that underpin the rank-based analysis presented in Section 3 of the main paper. We present three well-established lemmas from matrix theory regarding block lower triangular matrices, Hadamard products, and matrix products. We provide their proofs for completeness and demonstrate how they directly support our theoretical analysis.

The following lemma establishes a fundamental lower bound on the rank of block lower triangular matrices, which is essential for our analysis of the matrix $\mathbf{M}$ in Equation 12. This is a well-known result in matrix theory. **Note:** The following lemmas are written using independent notations.

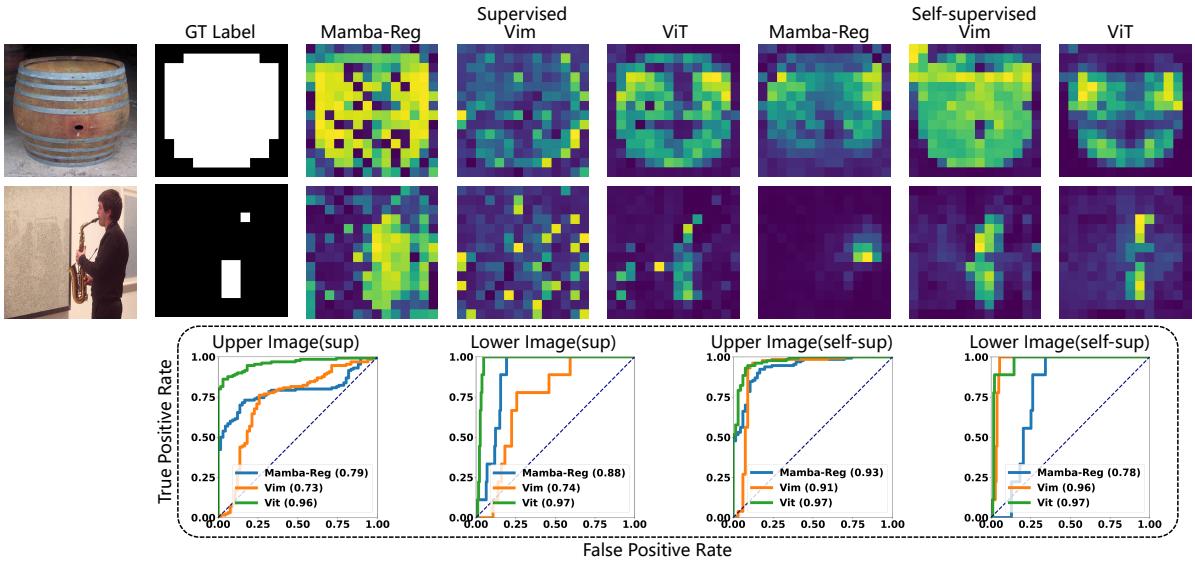

Figure 11. Feature map quality comparison: Supervised vs. Self-supervised. The figure illustrates that feature maps generated by self-supervised learning are clearer compared to those from supervised learning. Additionally, ViT produces less noisy feature maps compared to Mamba. The bottom section presents the ROC curve comparisons.

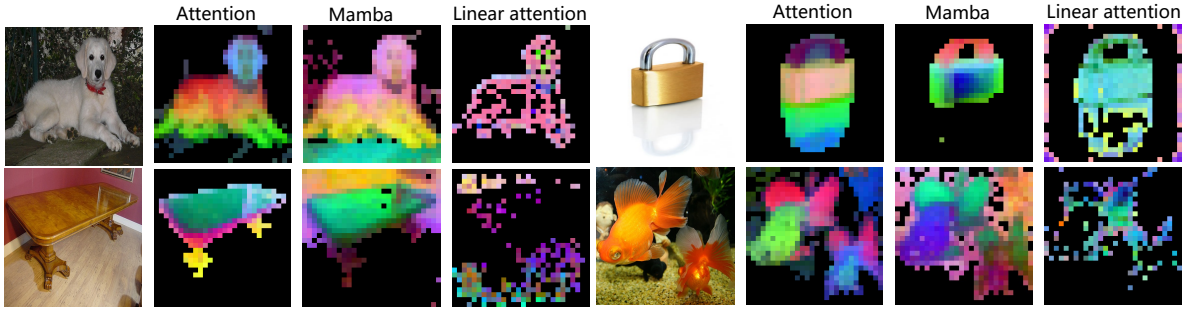

Figure 12. More visualization of the first PCA components of self-attention, Mamba, and linear attention, all generated under a consistent threshold.

Hadamard Product Rank Bounds

The following lemma establishes rank bounds for the Hadamard (element-wise) product of matrices, which is crucial for analyzing the structure of $\mathbf{M}$ in our unified formulation. This bound is a standard result in matrix analysis.

Rank of Matrix Products

The following lemma establishes a fundamental upper bound on the rank of matrix products, which is essential for analyzing the representational capacity of composed transformations in our models. This is one of the most important results in linear algebra for understanding how information flows through sequential operations. **Note**: *The following lemma uses independent abstract notations for clarity*

Lemma 2 (Rank Bound for Matrix Products)**.** *For any two matrices* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{B} \in \mathbb{R}^{n\times p}$*, the rank of their product satisfies:*

$$\operatorname{rank}(\mathbf{AB}) \leq \min\{\operatorname{rank}(\mathbf{A}), \operatorname{rank}(\mathbf{B})\}.$$

Proof. We prove both inequalities separately.

Part 1: $\operatorname{rank}(\mathbf{AB}) \leq \operatorname{rank}(\mathbf{B})$

Let $r_{\mathbf{B}} = \operatorname{rank}(\mathbf{B})$. Then the column space of $\mathbf{B}$ has dimension $r_{\mathbf{B}}$, which means every column of $\mathbf{B}$ can be expressed as a linear combination of $r_{\mathbf{B}}$ basis vectors.

The columns of $\mathbf{AB}$ are linear combinations of the columns of $\mathbf{A}$, where the coefficients come from the columns of $\mathbf{B}$. More precisely, if $\mathbf{b}_j$ denotes the j-th column of $\mathbf{B}$ and $\mathbf{a}_i$ denotes the i-th column of $\mathbf{A}$, then the j-th column of $\mathbf{AB}$ is:

$$(\mathbf{AB})_j = \sum_{i=1}^{n} b_{ij}\mathbf{a}_i.$$

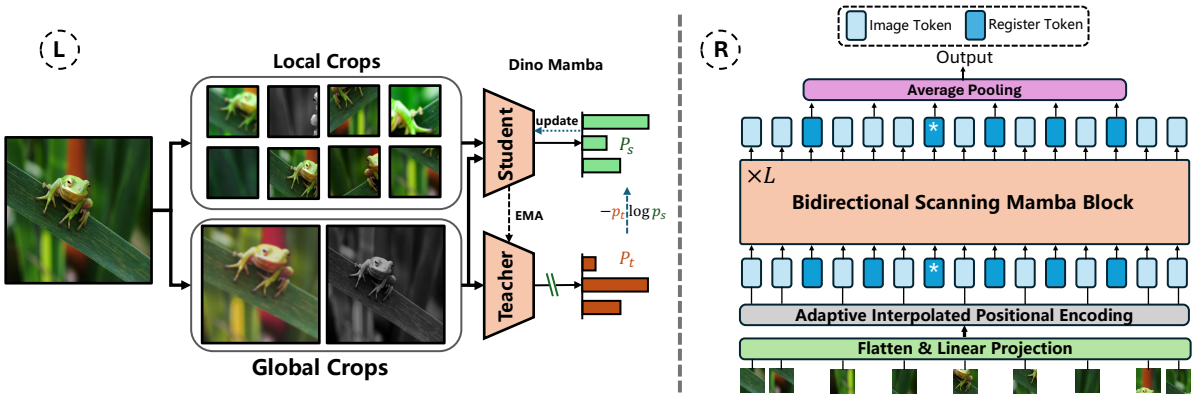

Figure 13. Visual Mamba in the DINO Framework: The **left** side shows the standard DINO architecture. The **right** side illustrates the structure of the student and teacher models, inspired by Mamba-Reg [52], with register tokens inserted at various positions to observe learned representations. When only a single register token is used at the middle position ('*'), this setup corresponds to the Vim [66] model. Additionally, an adaptive iterpolated positional encoding module is included to accommodate inputs of varying sizes (e.g., 224×224, 96×96).

Since each column of $\mathbf{B}$ lies in a space of dimension $r_{\mathbf{B}}$, the columns of $\mathbf{AB}$ must lie in a space of dimension at most $r_{\mathbf{B}}$. Therefore:

$$\operatorname{rank}(\mathbf{AB}) \leq \operatorname{rank}(\mathbf{B}).$$

Part 2: $\operatorname{rank}(\mathbf{AB}) \leq \operatorname{rank}(\mathbf{A})$

Consider the row space perspective. Let $r_{\mathbf{A}} = \operatorname{rank}(\mathbf{A})$. The row space of $\mathbf{A}$ has dimension $r_{\mathbf{A}}$.

Each row of $\mathbf{AB}$ is a linear combination of the rows of $\mathbf{B}$, where the coefficients come from the corresponding row of $\mathbf{A}$. Specifically, if $\mathbf{a}_i^\top$ denotes the i-th row of $\mathbf{A}$ and $\mathbf{b}_j^\top$ denotes the j-th row of $\mathbf{B}$, then the i-th row of $\mathbf{AB}$ is:

$$(\mathbf{AB})_i^\top = \sum_{k=1}^{n} a_{ik} \mathbf{b}_k^\top.$$

Alternatively, we can use the fact that the row rank equals the column rank. By taking transposes:

$$\operatorname{rank}(\mathbf{AB}) = \operatorname{rank}((\mathbf{AB})^\top) = \operatorname{rank}(\mathbf{B}^\top \mathbf{A}^\top).$$

Applying Part 1 to $\mathbf{B}^\top \mathbf{A}^\top$:

$$\operatorname{rank}(\mathbf{B}^\top \mathbf{A}^\top) \leq \operatorname{rank}(\mathbf{A}^\top) = \operatorname{rank}(\mathbf{A}).$$

Therefore:

$$\operatorname{rank}(\mathbf{AB}) \leq \operatorname{rank}(\mathbf{A}).$$

Combining both parts, we have:

$$\operatorname{rank}(\mathbf{AB}) \leq \min\{\operatorname{rank}(\mathbf{A}), \operatorname{rank}(\mathbf{B})\}. \tag{23}$$

□

Application to the Main Analysis in Submatrices

Connection to Section 3: The above lemmas directly support the rank analysis in the main paper:

1. *Hadamard product structure:* In the unified formulation (Equation 12), Linear attention and Mamba mechanisms involve Hadamard products:
   - Mamba: $\mathbf{M} = \mathbf{L}_{\mathbf{M}} \circ (\mathbf{C}^\top \mathbf{B})$
   - Linear Attention: $\mathbf{M} = \mathbf{L}_{\mathbf{Attn}} \circ (\psi(\mathbf{Q})\psi(\mathbf{K})^\top)$

   Lemma 1 provides the upper bounds:

$$\begin{cases} R^{\text{off}}_{\text{LinAttn}} \leq \operatorname{rank}(\mathbf{L}^{\text{off}}_{\mathbf{Attn}}) \cdot \operatorname{rank}(\psi(\mathbf{Q})\psi(\mathbf{K})^\top) \\ R^{\text{off}}_{\text{Mamba}} \leq \operatorname{rank}(\mathbf{L}^{\text{off}}_{\mathbf{M}}) \cdot \operatorname{rank}(\mathbf{C}^\top \mathbf{B}) \end{cases} \tag{24}$$

2. *Matrix product bounds*: For the terms $\psi(\mathbf{Q})\psi(\mathbf{K})^\top$ and $\mathbf{C}^\top \mathbf{B}$ appearing in the above equations, Lemma 2 provides:

$$\begin{aligned} \operatorname{rank}(\psi(\mathbf{Q})\psi(\mathbf{K})^\top) &\leq \min\{\operatorname{rank}(\psi(\mathbf{Q})), \operatorname{rank}(\psi(\mathbf{K})^\top)\} \leq D_{QK}, \\ \operatorname{rank}(\mathbf{C}^\top \mathbf{B}) &\leq \min\{\operatorname{rank}(\mathbf{C}^\top), \operatorname{rank}(\mathbf{B})\} \leq N. \end{aligned} \tag{25}$$

These bounds are crucial because they show that even before considering the masking operations, the representational capacity is already limited by the dimensionality of the intermediate transformations.

The key distinction between mechanisms arises from the structure of the off-diagonal submatrices. For a $C \times C$ off-diagonal block:

Linear Attention off-diagonal block:

$$\mathbf{L}^{\text{off}}_{\mathbf{Attn}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{C \times C}$$

This is a fixed matrix of all ones, with $\operatorname{rank}(\mathbf{L}^{\text{off}}_{\mathbf{Attn}}) = 1$.

Mamba off-diagonal block (with block indices $i > j$):

$$\mathbf{L}_{\mathbf{M}}^{\text{off}} = \begin{bmatrix} A_{i,j+1} & A_{i,j+2} & \cdots & A_{i,j+C} \\ A_{i+1,j+1} & A_{i+1,j+2} & \cdots & A_{i+1,j+C} \\ \vdots & \vdots & \ddots & \vdots \\ A_{i+C-1,j+1} & A_{i+C-1,j+2} & \cdots & A_{i+C-1,j+C} \end{bmatrix}_{C\times C}$$

where $A_{p,q} = \mathbf{A}_p \mathbf{A}_{p-1} \cdots \mathbf{A}_{q+1}$ is a product of learnable parameters, with $\text{rank}(\mathbf{L}_{\mathbf{M}}^{\text{off}}) \geq 1$ and typically much higher.

Combining Equations 24 and 25, we obtain:

$$R_{\text{LinAttn}}^{\text{off}} \leq 1 \cdot D_{QK} = D_{QK}$$
$$R_{\text{Mamba}}^{\text{off}} \leq \text{rank}(\mathbf{L}_{\mathbf{M}}^{\text{off}}) \cdot N$$

This establishes the rank hierarchy:

$$\underbrace{C}_{\substack{\text{Self-Attn} \\ \text{(full rank)}}} > \underbrace{\text{rank}(\mathbf{L}_{\mathbf{M}}^{\text{off}}) \cdot N}_{\substack{\text{Mamba} \\ \text{(learnable mask)}}} > \underbrace{D_{QK}}_{\substack{\text{Linear Attn} \\ \text{(fixed mask)}}} .$$

3. *Block structure analysis:* The matrix $\mathbf{M}$ in Equation (12) is partitioned into a $\frac{L}{C} \times \frac{L}{C}$ grid of $C \times C$ submatrices. We analyze the rank of diagonal and off-diagonal blocks separately for each mechanism.

   Diagonal blocks: All three mechanisms achieve full rank due to the lower triangular structure:

$$\mathbf{L}^{\text{diag}} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \cdot & 1 & 0 & \cdots & 0 \\ \cdot & \cdot & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \cdot & \cdot & \cdot & \cdots & 1 \end{bmatrix}_{C\times C} \quad \text{with } R_{\text{Self-Attn}}^{\text{diag}} = R_{\text{Mamba}}^{\text{diag}} = R_{\text{LinAttn}}^{\text{diag}} = C$$

   Off-diagonal blocks: The key differences emerge in the off-diagonal submatrices:

   Self-Attention: The softmax operation ensures full rank upper bound:

$$R_{\text{Self-Attn}}^{\text{off}} = C.$$

   Mamba: From Equations (24) and (25):

$$R_{\text{Mamba}}^{\text{off}} \leq \text{rank}(\mathbf{L}_{\mathbf{M}}^{\text{off}}) \cdot N.$$

   Linear Attention: From Equations (24) and (25):

$$R_{\text{LinAttn}}^{\text{off}} \leq 1 \cdot D_{QK} = D_{QK}.$$

   For typical base model settings ($C = 256$, $N = D_{QK} = 64$):

Block Type	Self-Attn	Mamba	Linear Attn
Diagonal	$C = 256$	$C = 256$	$C = 256$
Off-diagonal	$C = 256$	$\leq \text{rank}(\mathbf{L}_{\mathbf{M}}) \cdot 64$	≤ 64

   This establishes the rank hierarchy for off-diagonal blocks:

$$\underbrace{256}_{\text{Self-Attn}} > \underbrace{\text{rank}(\mathbf{L}_{\mathbf{M}}^{\text{off}}) \cdot 64}_{\text{Mamba}} > \underbrace{64}_{\text{Linear Attn}} .$$

   Key observations:

   - All mechanisms are equivalent on diagonal blocks ($R^{\text{diag}} = C$).
   - Self-Attention achieves full rank upper bound on off-diagonal blocks due to softmax nonlinearity.
   - Mamba's learnable mask $\mathbf{L}_{\mathbf{M}}$ allows $\text{rank}(\mathbf{L}_{\mathbf{M}}^{\text{off}}) \gg 1$ in practice, significantly exceeding Linear Attention's fixed rank-1 mask.
   - Linear Attention is most constrained with $R_{\text{LinAttn}}^{\text{off}} \leq D_{QK}$ due to its fixed all-ones mask.

Remarks on Practical Implications

The theoretical framework established here explains why:

1. *Self-Attention is most expressive:* The softmax operation makes the effective rank upper bound of off-diagonal blocks full ($R^{\text{off}} = C$), allowing maximum representational capacity.
2. *Mamba balances efficiency and expressiveness:* The learnable matrix $\mathbf{A}$ in $\mathbf{L}_{\mathbf{M}}^{\text{off}}$ increases $\text{rank}(\mathbf{L}_{\mathbf{M}}^{\text{off}})$, enabling $R_{\text{Mamba}}^{\text{off}}$ to approach $N \cdot \text{rank}(\mathbf{L}_{\mathbf{M}}^{\text{off}})$. Since N scales linearly with computation ($O(LNJ)$), Mamba can achieve higher ranks than Linear Attention while maintaining linear complexity. Moreover, Lemma 2 shows that the rank is fundamentally limited by N, but this limit is more attainable than in Linear Attention due to the learnable mask.
3. *Linear Attention is most constrained:* The fixed mask $\mathbf{L}_{\mathbf{Attn}}^{\text{off}}$ with $\text{rank}(\mathbf{L}_{\mathbf{Attn}}^{\text{off}}) = 1$ severely limits $R_{\text{LinAttn}}^{\text{off}} \leq D_{QK}$ by Lemma 1. Moreover, Linear Attention's $O(LD_{QK}^2)$ complexity restricts D_{QK} growth, further limiting its representational capacity. Lemma 2 additionally shows that the product $\psi(\mathbf{Q})\psi(\mathbf{K})^\top$ cannot exceed rank D_{QK} regardless of the quality of the feature mappings.

These mathematical foundations rigorously support the empirical observations in the experiments (Section 5), where feature map quality and downstream task performance follow the predicted hierarchy: ViT > Mamba > Linear ViT.

Remark on Low-Rank Structures. Low-rank structures are sometimes regarded as beneficial regularization rather than a representational limitation. We clarify that these are two distinct notions of rank: low-rank regularization (*e.g.* LoRA) operates in the *parameter* dimension, while the rank analyzed here operates in the *token* dimension, directly bounding inter-token dependency modeling. For large-scale vision tasks, mainstream models are typically under-fitting in token-wise dependencies, as evidenced by ViTs consistently benefiting from larger input sizes where both sequence length and effective rank increase. Therefore, a lower token-wise rank reflects a representational bottleneck, not a regularization advantage.