

# Sequential Token Merging: Revisiting Hidden States

**Yan Wen<sup>1,2</sup>, Peng Ye<sup>3,4</sup>, Lin Zhang<sup>1</sup>, Baopu Li<sup>5</sup>,  
Jiakang Yuan<sup>1</sup>, Yaxin Yang<sup>1</sup>, Tao Chen<sup>1,2†</sup>**

<sup>1</sup>College of Future Information Technology, Fudan University

<sup>2</sup>Shanghai Innovation Institute

<sup>3</sup>Shanghai Artificial Intelligence Laboratory

<sup>4</sup>The Chinese University of Hong Kong <sup>5</sup>Independent Researcher  
21307130037@m.fudan.edu.cn eetchen@fudan.edu.cn

## Abstract

Vision Mambas (ViMs) achieve remarkable success with sub-quadratic complexity, but their efficiency remains constrained by quadratic token scaling with image resolution. While existing methods address token redundancy, they overlook ViMs’ intrinsic Limited Directional Sequential Dependence (LDSD)—a critical information flow mechanism revealed in our analysis. We further identify Mamba’s selective scan enables gradual information aggregation in hidden states. Based on these insights, we propose Sequential Token Merging (STM), featuring: 1) Bidirectional nearest neighbor merging to preserve sequential dependencies through symmetric spatial aggregation, and 2) Hidden states protection to stabilize the hidden states around the class token. STM strategically leverages Mamba’s layer-wise loss convergence to convert temporal forgetfulness into stability. Experiments demonstrate STM’s superiority: 1.0% accuracy drop for ViM-Ti at 20% token reduction, and only 1.4% degradation for ViM-S at 40% reduction. Our method achieves state-of-the-art efficiency with minimal complexity, while providing new insights into state-space model dynamics. Codes will be released soon.

## 1 Introduction

Vision Mambas (ViMs) are generic vision backbones based on Mamba, a hardware-accelerated State Space Model (SSM) [10, 46]. ViMs are famous for their sub-quadratic complexity [12, 20] compared to Vision Transformers (ViTs) [8, 22, 37, 44] in computer vision. Similar to ViTs, larger ViMs encounter deployment difficulties stemming from substantial memory usage and latency limitations [4]. Many efforts have been devoted to more efficient and accelerated ViMs for visual tasks [5, 17, 28, 29, 34]. Token reduction is a common technique in ViTs to balance the tradeoff between computational efficiency and model accuracy [16, 25, 30, 31, 41]. Given its success in ViTs, token reduction has shown to be a useful method to enhance the efficiency of ViMs because the token lengths are usually fixed and independent of the model architectures. [35, 42] center around the reduction of unnecessary tokens by transferring techniques from ViTs to ViMs, such as inter-token similarity metrics and attention-derived importance scores. However, these methods might not follow the mechanisms in ViMs, leading to an unsatisfactory accuracy and deeply relying on fine-tuning or retraining for restoration.

The above issue may be attributed to an unproven assumption in ViMs that some tokens can be removed because they contribute little to the output. However, since Mamba’s hidden states selectively propagate information, such detection-and-removal strategies can lead to unpredictable and uncontrollable loss in hidden states. Figure 1 visualizes heatmaps of hidden states and attention

---

\*<sup>†</sup> Corresponding author.

scores [1], computed from their general forms derived via SSM, offering an intuitive comparison of their respective roles.

We identify two key properties of ViMs: (1) information is transmitted via Limited Directional Sequential Dependence (LDSD), where each hidden state depends on all preceding states and inputs from multiple directions; and (2) important information is selectively compressed into hidden states during their self-update stages, as Mamba’s selective scan recursively encodes the selected features into evolving hidden states. These insights motivate a sequentially-aware approach to token reduction that centers on hidden states. We propose Bidirectional Nearest Neighbor Merging (BNNM) to introduce controllable and self-converging losses within hidden states, and Hidden States Protection (HSP) to stabilize the fluctuations of hidden states near the class token by leveraging input-dependent parameters for selective compression. Together, BNNM and HSP provide a new perspective on sequential token reduction and the dynamic flow in Mamba, transforming its inherent forgetfulness into a form of stability. We evaluate our method on ImageNet-1K [7] using ViM-Ti, ViM-S, and ViM-B [46], achieving consistently high accuracy and strong robustness under aggressive token reduction. Notably, on ViM-Ti, our method keeps the top-1 accuracy drop within 4% under 40% token compression, while reducing FLOPs by 22.3%. Our contributions can be summarized as follows:

1. We identify LDSD as the core mechanism of information transmission in ViMs and highlight that information selection and compression occur progressively, requiring consistent maintenance of dynamic representations. These insights prompt us to revisit hidden states and redefine the goal as stabilizing hidden states around the class token.
2. Following the analysis, we propose a novel sequential token reduction pipeline. We introduce BNNM to construct a predictable and controllable loss in the hidden states because BNNM aligns with the original transmission characteristics in ViMs, paving a feasible way for our next step.
3. We leverage selective scan to maintain the online information compression by proposing HSP. HSP minimizes the controllable perturbation loss introduced by BNNM via retrieving previous selective parameters and exploiting the perturbation’s self-converging property.
4. Experiments show our BNNM strategy preserves spatial structure via symmetric token aggregation, significantly mitigating accuracy loss under token reduction. The HSP scheme stabilizes hidden states across varying reduction rates, maintaining high accuracy. Our method achieves state-of-the-art performance with the lowest complexity among comparable approaches.

## 2 Related work

**Vision Mamba** Mamba [9] has been shown to be a promising alternative [6] to Transformer [38] with only linear complexity of tokens. Recently, abundant works explore the effectiveness of Mamba in computer vision [3, 11, 14, 15, 18, 21, 27, 28, 32, 36, 39], represented by ViMs [46], which focus on bidirectional token scanning. However, these works mainly contribute to backbone mechanism design; further optimization remains still unexplored. Our proposal is an effective inference acceleration method with negligible overhead from a novel sequential token merging perspective.

**Token Reduction** Token reduction is a successful strategy in model compression to balance the computational load and enhance efficiency by reducing amount of the tokens processed. It is popular in efficient Transformers [2, 24, 30, 40] because it doesn’t require specific hardware design nor additional weights [42], but its potential hasn’t been fully recognized in Mamba. Several informative works have been done towards this direction [33, 35, 42, 43]. Famba-V [33] is the first who transfer ToMe [2] in ViT to efficient Mamba training, fusing the most similar pairs in a cross-layer manner. The authors in [42] emphasize the importance of token order for performance restoration. They use the decay factor  $\bar{\Lambda}$  to align hidden states and prune tokens based on attention scores across token channels. R-MeeTo [35] finds that token merging preserves more information than pruning especially in ViM. It also follows ToMe [2] by merging token pairs with the nearest distance, relying on a brief retraining stage to restore accuracy. However, these previously successful methods in ViTs prove ineffective when applied to ViMs, primarily due to the distinct and unidentified information flow mechanism in Mamba, particularly regarding token behavior and their interaction with hidden states.

### 3 Revisiting Hidden States

#### 3.1 Vision Mamba

State Space Models (SSMs) map an input sequence  $x(t) \in \mathbb{R}^L$  to an output sequence  $y(t) \in \mathbb{R}^L$  via a hidden state  $h(t) \in \mathbb{R}^N$ , with dynamics defined as:

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \quad y(t) = \mathbf{C}h(t). \quad (1)$$

Mamba is derived from continuous SSMs by discretizing with a timescale  $\Delta$  and the zero-order hold (ZOH) rule (equation (2)), converting the continuous differential equations (1) into the linear recurrence (3), and further into the general form (4).

$$\bar{\mathbf{A}} = \exp(\Delta \mathbf{A}), \quad \bar{\mathbf{B}} = (\Delta \mathbf{A})^{-1}(\exp(\Delta \mathbf{A}) - \mathbf{I}) \cdot \Delta \mathbf{B}. \quad (2)$$

$$h_t = \bar{\mathbf{A}}_t h_{t-1} + \bar{\mathbf{B}}_t x_t, \quad y_t = \mathbf{C}_t h_t. \quad (3)$$

$$h_t = \sum_{j=1}^t \left( \prod_{k=j+1}^t \bar{\mathbf{A}}_k \right) \bar{\mathbf{B}}_j x_j \quad (4)$$

**Hidden Attention in Mamba** [1] shows that SSM-based models can be interpreted as attention-driven, where hidden states function can be interpreted as an attention mechanism under suitable transformations with query, key, and value formulations. Specifically, they derive the attention matrices in selective state spaces layers generally as  $\tilde{\alpha}_{i,j}$  in (5)

$$\tilde{\alpha}_{i,j} = C_i \left( \prod_{k=j+1}^i \bar{\mathbf{A}}_k \right) \bar{\mathbf{B}}_j \quad (5)$$

**The pipeline of Vision Mamba** ViM [46] projects flattened 2D patches  $t_p$  using a learnable matrix  $W$ , with  $t_{cls}$  denoting the class token representing the entire sequence. We use the middle class token, and  $T_{l-1}$  denotes the input token sequence for the  $l^{th}$  ViM encoder layer.

$$T_0 = [t_p^1 W; t_p^2 W; \dots; t_{cls}; \dots; t_p^J W] + Epos \quad (6)$$

#### 3.2 Mamba Hidden States Analysis and Discovery

Mamba replaces multi-head attention [13] with hidden states, prompting us to revisit their role for a more adaptive token reduction strategy. This section presents our analysis of hidden states in ViMs, introducing LDSD in 3.2.1 to characterize information flow between adjacent hidden states, and showing in 3.2.2 that Mamba performs online sensing and compression of important information.

##### 3.2.1 Theoretical Exploration: Limited Directional Sequential Dependence

Hidden states in Mambas and multi-head attention in Transformers represent fundamentally different mechanisms, despite serving similar objectives. While prior works have explored their theoretical differences and potential connections [1, 12], and even suggested structural similarities [6], these insights offer limited practical guidance for token reduction in ViMs. This is mainly due to the absence of a comparative framework that explicitly focuses on the mechanisms of information transmission in ViMs, particularly how tokens and hidden states interact with sequence structures. We introduce Limited Directional Sequential Dependence (LDSD) to address this gap. It captures a distinctive property of hidden states in Mamba-based architectures: each hidden state exhibits localized and directional dependence on the entire sequence, in contrast to the uniform pairwise interactions in Transformers. LDSD provides a new lens for understanding token-sequence relationships in ViMs, which is both theoretically grounded and practically useful, forming the basis for our token merging strategy detailed in Section 4. Concretely, we define LDSD as a structured hidden state evolution process, where a focal hidden state  $h'$  is formed by aggregating directionally transformed previous

Table 1: Detailed Analysis about Limited Directional Sequential Dependence

Perspective	Description	ViMs	ViTs
Information flow	Information mechanism	Hidden states	Multi-head attention
	Information direction	Directional	Non-directional
	Memory ability	Forgetfulness	Strong memory
Token assumption	Logic behind tokens	Sequentialized	Atomized
	Dependence on sequence	Strong dependence	Weak relation
	Complexity	$O(n)$	$O(n^2)$

hidden states  $h$ , with each transformation influenced by its corresponding input  $x$ . This captures both the temporal sequentiality and spatial directionality unique to Mamba models. Formally, this dependence can be written as:

$$h' = \sum_{Directions, h} SSM(h, x) \quad (7)$$

Table 1 illustrates the two key components of LDSD: the directional information flow and the core assumption that all input tokens are part of a stream. This implies strong dependencies between each token and the entire sequence, contributing jointly to the final representation. For clarity, we contrast these properties with those of ViTs.

**Information flow** Hidden states in ViMs provide a unique flow of sequential information and enable linear-time inference. This process can be compared to a chain of individuals passing information, where each person (i.e., hidden state) refines the message before forwarding it to the next person. In bidirectional or multi-directional Mamba models, multiple such chains exist, passing information in parallel. Unlike ViTs, which rely on computationally intensive attention mechanisms, ViMs aggregate patch information into a single class token through the hidden state chains, resulting in sub-quadratic inference complexity. However, this efficiency introduces trade-offs. ViMs tend to forget early inputs due to the sequential nature of information flow, and the connection between the class token and individual image patches is indirect. We use the term "Limited Directional" to describe this indirectness, because the class token perceives the entire image only through constrained directional branches formed by hidden state propagation. This makes Mamba's perceptual system inherently limited and potentially fragile. While some recent works attempt to restore global information by introducing multi-directional flows [21, 23, 45], we argue that these approaches only partially address the issue. Without an effective coordination mechanism, the directional biases from different scans remain incoherent [29]. Even with coordination, the directions remain fundamentally limited.

**Sequentialized Token Assumption.** The SSM formulation (3) implicitly assumes that tokens are sequentialized, as each input  $x$  updates the hidden state between adjacent time steps. This reflects Mamba's assumption of a strong dependence between individual tokens and their sequence, as described in (7). This assumption also makes token reduction not a straightforward task because reducing tokens may yield a tricky perturbation to the information chain. We figure out the position of tokens plays an equivalent vital role in forming the information flow as well as the information of tokens itself due to the hidden state chains. For example, if we exchange two arbitrary tokens' positions without changing their values, the output is still interfered because the pretrained parameters of different positions in hidden states only adapt to the inputs from the corresponding positions. Even worse, this perturbation may be pronounced due to the aggregation nature of the chains because each hidden state has to resort to its adjacent states and has no other means to calibrate the information it receives. Therefore, when reducing tokens, we not only need to determine the information conveyed by each token but also recognize the global effect each token gives to the chain of hidden states.

In essence, ViMs rely on the effective transmission of semantic information through hidden state chains and sequentialized tokens. This insight prompts a rethinking of token reduction in ViMs, where preserving the class token's hidden state becomes the key objective since all token information is aggregated there. Guided by the principle of LDSD, we propose BNNM in 4.1.



Figure 1: Hidden states show much more semantic information than attention. This phenomenon originates from the selective scan in Mamba.

### 3.2.2 Experimental Exploration: Online Information Selective Compression

Selective scan plays a dominant role in selecting and compressing the important information from tokens into hidden states [9], and this process happens during the self-update stages of hidden states.

**Dominant status of selective scan.** The success of Mamba is largely credited to selective scan. In brief, selective scan sets that  $B$ ,  $C$ , and  $\Delta$  are all liner projections of input  $x$ , with  $A$  and  $B$  being further discretized with  $\Delta$ . The function of selective scan in ViMs is intuitively reflected in Figure 1. Heat maps of hidden states reveal significantly more semantic and input-dependent information than those of attention matrices. In contrast to the attention formulation in equation (5), where the input tokens  $x$  are decoupled from the update process, the hidden states  $h$  are updated through a formulation that fuses the input  $x$ , as shown in equation (4). This fundamental difference accounts for the superior semantic expressiveness of hidden states, as illustrated in Figure 1. The key reason lies in the selective scan mechanism: to fully activate this mechanism, the input must be integrated into the hidden states. This recursive process enables dynamic and input-variant representations, as  $x$  is first projected into  $\bar{A}, \bar{B}$ , which in turn selectively extract features from  $x$ . Nevertheless, heat maps of attention consistently exhibit the highest scores around the class token regardless of inputs. This phenomenon stems from the isolation of  $x$  in the attention formulation (5), where the decay factor  $\bar{A}$  dominates the computation. In our analysis, we refer to  $\bar{A}, \bar{B}$  as the core components of the selective scan mechanism and omit  $C$  during inference, as we focus solely on the hidden states and do not require intermediate outputs.

**Online compression property.** The online compression property of SSMs originates from their finite-state nature [9], which inherently compresses information within the hidden states. Unlike attention-based models, SSMs operate without global context (e.g., key-value caches) during inference. They process input tokens asynchronously, progressively selecting and compressing information via recurrent updates between adjacent hidden states. In contrast, ViTs perform token selection and compression in a synchronous manner by computing attention scores across all tokens simultaneously. Importantly, no token in an SSM is entirely negligible; any meaningful importance metric should holistically reflect each token’s cumulative influence on hidden state evolution and the input-to-output transformation.

**Enlightenment** Our insights from revisiting hidden states highlight the importance of preserving LDSD to maintain effective information transmission. Furthermore, the information of the reduced tokens not fed into Mamba should still be selectively compressed to retain features integrity that contributes to the final output.

## 4 Sequential Token Merging

Based on the enlightenment, we propose a general Sequential Token Merging (STM) method for ViMs. As illustrated in Figure 2, our BNNM preserves LDSD, transforming the uncontrollable loss in traditional token reduction into a controllable perturbation within hidden states. To further minimize information loss, we introduce HSP via selective approximation based on our fine-grained analysis

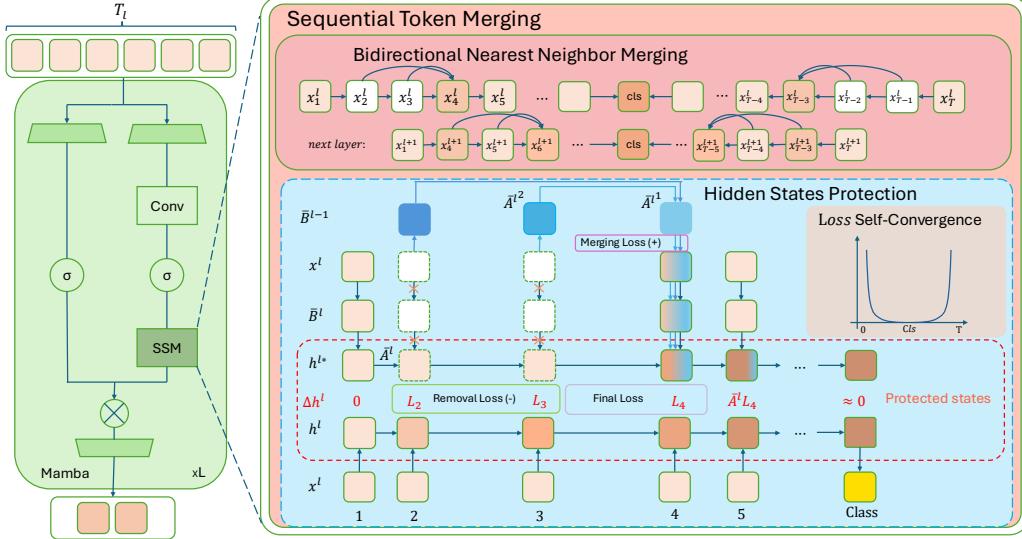


Figure 2: Overview of our proposed Sequential Token Merging (STM) method. It contains two parts: Bidirectional Nearest Neighbor Merging (BNNM) and Hidden States Protection (HSP). Blue colors indicate tokens with external information, and darker colors indicate tokens with more aggregated information.

of online information selective compression. Finally, by leveraging the self-converging nature of merging loss, STM turns Mamba’s forgetfulness into a source of great stability.

#### 4.1 Bidirectional Nearest Neighbor Merging

We propose the Bidirectional Nearest Neighbor Merging (BNNM) to maintain LDSD. At each layer, every reduced token is merged with the nearest remaining neighbor token in its forward direction toward the class token. As illustrated in the BNNM component of Figure 2, this merging process is applied recursively across layers. This approach stabilizes the hidden states and makes perturbations predictable, enabling loss in hidden states to be effectively controlled, as further discussed in Section 4.2. Unlike prior methods [35, 42], BNNM preserves token order and introduces perturbations at predictable locations, making them estimable.

**Notation** We follow the notation system introduced in [42] for our elaboration. In ViM, the input to the  $l^{th}$  layer is a token sequence  $T_{l-1} \in \mathbb{R}^{B \times N \times D}$  (equation (6)), where  $B$  is the batch size,  $N$  is the number of tokens, and  $D$  is the hidden dimension. Each sequence in the batch consists of  $N$  tokens, denoted as  $\{x_j\}_{j=0}^{N-1}$ . We assume that only  $K$  tokens are retained after merging, while the other  $N - K$  tokens are discarded. Following the notation in [42], we represent the indices of the retained tokens as  $\{q_k\}_{k=0}^{K-1}$ ,  $q_s < q_t$  for all  $s < t$ . To make the problem more tractable, consider two adjacent remaining tokens  $x_{q_k}$  and  $x_{q_{k+1}}$ . If  $q_{k+1} - q_k > 1$ , this indicates that some tokens have been reduced between them. Let  $R_k \geq 1$  denote the number of reduced tokens between  $q_k$  and  $q_{k+1}$ , whose indices are given by  $\{q_k + i\}_{i=1}^{R_k}$ . These indices satisfy  $q_k < q_k + 1 < \dots < q_k + R_k < q_{k+1}$ . To emphasize the distinction between  $q_{k+1}$  and  $q_k + 1$ , we adopt the convention of using round brackets in expressions such as  $q_{(k+1)}$  and  $q_{(k)} + 1$ , though their meanings remain unchanged. We use  $h^*$  to denote the modified hidden states, and we use  $h$  to denote the original hidden states.

**Removal Loss and Merging Loss.** The removal loss measures the change in hidden states caused solely by removing a token at its original position  $q_{(k)} + i$ , capturing the information loss from that token. In contrast, the merging loss quantifies the perturbation introduced at the merging position  $q_{(k+1)}$  due to incorporating additional information from the merged token. Because hidden states are updated sequentially, the merging loss cannot be directly measured at  $q_{(k+1)}$  without including accumulated effects. Thus, we define it as the final loss at  $q_{(k+1)}$  minus the propagated removal losses from all merged tokens. Specifically, we denote the merging loss as  $L_{q_{(k+1)}M}$  in equation (10),

in contrast to the overall loss  $L_{q_{(k+1)}}$  (9). The removal loss originally caused by discarding a reduced token at position  $q_{(k)} + i$ , and then propagated to  $q_{(k+1)}$ , is denoted as  $\text{SSM}(L_{q_{(k)}+i})$ . This formulation reflects how removing a token affects subsequent hidden states through sequential dependencies, as modeled by the SSM. Given that both the removed position  $q_{(k)} + i$  and the merging target position  $q_{(k+1)}$  are predetermined, we can explicitly compute these two types of loss without introducing other effects. This controllability ensures that BNNM provides a stable and predictable loss behavior, forming a reliable foundation for further loss minimization in later stages, especially during selective-aware token merging (Section 4.2.1).

$$L_{q_{(k)}+i} = h_{q_{(k)}+i} - h_{q_{(k)}+i}^* \quad (\text{Removal Loss}) \quad (8)$$

$$L_{q_{(k+1)}} = \sum_{\text{Directions}} (h_{q_{(k+1)}} - h_{q_{(k+1)}}^*) \quad (\text{Final Loss}) \quad (9)$$

$$L_{q_{(k+1)} M} = L_{q_{(k+1)}} - \sum_{\text{Directions}, i} \text{SSM}(L_{q_{(k)}+i}) \quad (\text{Merging Loss}) \quad (10)$$

The following section details how we leverage the selective scan mechanism to estimate and compress the information of reduced tokens into hidden states, even without passing through Mamba. Our method strategically employs the merging loss to compensate for the removal loss.

## 4.2 Hidden States Protection

To align with Mamba’s online compression property (Section 3.2.2), we propose selective-aware token merging, which aims to minimize the final hidden state loss as defined in equation (9). Additionally, we transform the forgetfulness of Mamba into stability against perturbations.

### 4.2.1 Selective-aware token merging

The general formulation of our token merging strategy is presented below. It is derived by comparing the hidden states before and after token removal to obtain the compensation term, with detailed derivations provided in Appendix A.

$$\text{Merge}_{fwd}(x_{q_{(k-1)}+1}, x_{q_{(k-1)}+2}, \dots, x_{q_{(k)}}) = \underbrace{\sum_{j=1}^{R_{k-1}} \left( \prod_{n=j+1}^{q_{(k)}} \bar{A}_n \right) \underbrace{\frac{\bar{B}_{q_{(k-1)}+j}^{(l-1)}}{\bar{B}_{q_{(k)}}^{(l-1)}} x_{q_{(k-1)}+j}}_{\text{Removal loss estimation } (-)} + \underbrace{x_{q_{(k)}}}_{\text{Original token}}}_{\text{Merging loss compensation } (+)} \quad (11)$$

**Removal loss estimation (-) & Merging loss compensation (+).** To incorporate token selectiveness into the merging process, we begin by estimating the removal loss using the previous layer’s  $\bar{B}^{(l-1)}$  terms corresponding to the reduced token at  $q_{(k-1)} + j$  and the remaining token at  $q_{(k)}$ . This estimated hidden state loss is then propagated through the current layer to compute the merging loss compensation, i.e., each term in the forward merging function is scaled by an exponentially decaying factor constructed from multiple  $\bar{A}$  values spanning from the reduced position to the merging point. Therefore, our approach leverages the dominant status of the selective scan mechanism (Section 3.2.2) in Mamba. By integrating  $\bar{A}, \bar{B}$  into the merging strategy, it fully utilizes Mamba’s inherent selectivity.

### 4.2.2 From Forgetfulness to Stability: The Self-Convergence of Loss

**Corollary (Distance Fading Rule):**

$$L_{q_{(k+1)}+1} = \bar{A}_{q_{(k+1)}+1} L_{q_{(k+1)}} \quad (12)$$

$$L_{cls} = \bar{A}^{|cls-p|} L_p \quad (13)$$

After compensation, we can obtain the final loss with equation (9) and derive the corollary. Due to the linear property of Mamba, the final loss at the merging position  $L_{q_{(k+1)}}$  decays exponentially over time, with the hidden state difference at the next token position scaled by  $\bar{A}_{q_{(k+1)}+1}$ . The proof is provided in Appendix B. Combining this property with the bidirectional Mamba architecture,

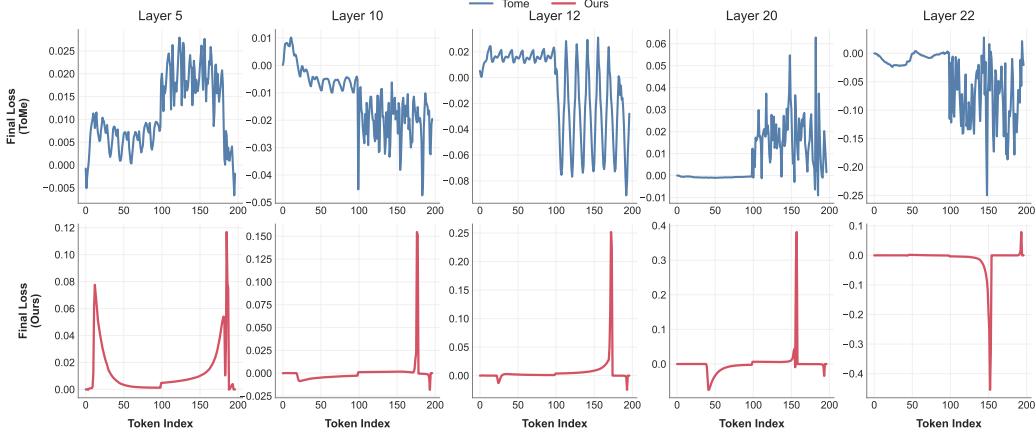


Figure 3: This figure presents a layer-wise comparison of the final loss introduced by ToMe-based methods in ViM [2][33][35], and our proposed method (STM). As illustrated, the merging compensation in STM ensures that hidden states undergo only minimal perturbations at merging points and rapidly converge to zero, stabilizing the hidden states around the class token.

we calculate the ultimate impact of a token merged at position  $p$  on the class token. Using the middle class token in the ViM setting as an example, we derive the distance fading rule (13), where  $cls$  denotes the position of the class token. For simplicity, we replace the product  $\prod_{j=p+1}^{cls} \bar{A}_j$  with  $\bar{A}^{|cls-p|}$ , since the selective nature of  $\bar{A}$  arises from  $\Delta$  [9], making  $\bar{A}$  approximately a constant factor. This is also supported by the exponential degradation curves shown in Figure 3. This formulation highlights how the distance between the merged token and the class token effects the final hidden representation. Importantly, this property proves useful in our experiments. Corollary (13) finally motivates a layer-wise merging strategy, where tokens are merged progressively from distant to closer positions, thereby leveraging the loss’s self-convergence. As a result, our method facilitates a truly data-free design, meaning it doesn’t rely on any data for fine-tuning or retraining the compressed model.

## 5 Experiment Results

We implement our method using PyTorch [26] for scientific computation. To validate both our theoretical analyses and the empirical effectiveness of STM, extensive image classification experiments are conducted on the ImageNet-1K [7] validation dataset, using 4 NVIDIA RTX 4090 GPUs. We conduct a comparison against three state-of-the-art token reduction techniques in ViM: Token Recognition [19], Hidden State Alignment [42], and R-MeeTo [35]. To enable a fair comparison and demonstrate our super performance, the result of [42] and [35] are yielded directly after token reduction without fine-tuning nor retraining, while scores of [19] are the fine-tuned results reported in [42]. For the bidirectional ViM-Ti, ViM-S, and ViM-B [46] models, we apply a uniform token reduction rate of 20% across all settings. STM consistently applies forward merging function (11) to merge two tokens in each direction at every layer throughout the model. ViM-Ti validation takes under 2 hours, ViM-S around 4 hours, and ViM-B around 6 hours. Table 2 provides ImageNet validation top-1 accuracies and the accompanying FLOPs.

**Results** As shown in Table 2, our method consistently achieves the highest accuracy among all compared approaches across all models. ViM-Ti only has a 1.0% drop in accuracy compared to the unreduced baseline, with 25.8% accuracy higher than Hidden States Alignment and 22.8% higher than R-MeeTo. For ViM-S, our method achieves 5.6% higher performance restoration rate than Token Recognition because Token Recognition prunes tokens excessively. For ViM-B, our method continues to outperform other approaches with wide margins in accuracy, along with a 16% FLOPs reduction compared to 12.1% of Hidden States Alignment and 4.7% of R-MeeTo respectively.

Table 2: Comparison of different methods on top-1 accuracy and FLOPs. **Bold** is the best, and underline is the second best.

Method	top-1 acc. (%)				FLOPs (G)			
	Vim-Ti	Vim-S	Vim-B	Vim-Ti	Vim-S	Vim-B		
Vim (Baseline)	76.1	0.0	80.5	0.0	81.9	0.0	1.45	0.00
Token Recognition	<u>71.3</u>	<u>4.8</u> ↓	74.8	5.7 ↓	—	—	<u>1.29</u>	<u>0.16</u> ↓
Hidden State Alignment	49.3	26.8 ↓	72.0	8.5 ↓	<u>79.6</u>	<u>2.3</u> ↓	1.29	0.16 ↓
R-MeeTo	52.3	23.8 ↓	<u>78.7</u>	<u>1.8</u> ↓	79.1	2.8 ↓	1.41	0.04 ↓
Ours	<b>75.1</b>	<b>1.0</b> ↓	<b>79.3</b>	<b>1.2</b> ↓	<b>79.8</b>	<b>2.1</b> ↓	<b>1.28</b>	<b>0.17</b> ↓
							<u>4.23</u>	<u>0.85</u> ↓
							<b>15.73</b>	<b>3.14</b> ↓

**Complexity Comparison** Despite achieving the highest accuracy across all models, our method requires significantly fewer FLOPs than Hidden State Alignment and R-MeeTo. This efficiency is attributed to the uniform merging strategy inherited from BNNM, which avoids the costly computation of pairwise distances or importance scores for token selection. These computations grow quadratically with the number of tokens and hidden dimensions, leading to substantial overhead in larger models. In contrast, our method maintains a lightweight inference process. Experiments validate that this simple yet principled merging approach offers a strong balance between efficiency and accuracy.

### 5.1 Ablation & Analysis

**Both sides v.s. One side** In Table 3, we compare the impact of merging tokens on one side versus both sides symmetrically in bidirectional Mamba. STM merges 1, 2, 2, and 2 tokens per direction at each layer to achieve token reductions of 15%, 20%, 30%, and 40%, respectively. On ViM-Ti, the both-sides merging strategy reduces degradation from 45.0%↓ to 1.0%↓ at 15% token reduction. This results demonstrate the effectiveness of our bidirectional merging, which better preserves spatial structure by symmetrically aggregating contextual information with minimal perturbation. On ViM-S, the difference in accuracy between one-side and both-sides merging is negligible, but one-side merging results in slightly higher FLOPs. Given ViM-S’s larger capacity, it is more resilient to token reduction, with minimal performance degradation. Therefore, we do not include this ablation on ViM-B, as its even larger size shows negligible degradation under our method.

**Effect of different reduction rate** Table 3 shows that STM maintains strong performance under varying token reduction rates. Notably, STM incurs only minimal performance drops of 3.9%↓ and 1.4%↓ under a 40% token reduction for ViM-Ti and ViM-S respectively, highlighting the stability and effectiveness of our merging strategy. For a detailed analysis about BNNM, we focus on both-sides merging. ViM-Ti shows a consistent trade-off between accuracy and FLOPs as reduction rate increases. ViM-S, however, peaks in both accuracy and FLOPs at 20% reduction, with minor accuracy fluctuations and a unique FLOPs rebound due to ViM-S’s higher feature dimension, indicating that merging in ViM-S becomes increasingly costly relative to the computation saved.

Table 3: Ablation comparison between one side merging and both sides merging, along with performance under varying token reduction rates. Top-1 acc. (%) and FLOPs (G) are reported. **Bold** is the best.

Model	Mode	Token Merging Rate (%)				FLOPs (G)			
		15%	20%	30%	40%	15%	20%	30%	40%
ViM-Ti	One sides	31.1	45.0 ↓	31.1	45.0 ↓	34.6	41.5 ↓	34.6	41.5 ↓
	Both sides	<b>75.1</b>	<b>1.0</b> ↓	75.0	1.1 ↓	73.3	2.8 ↓	72.2	3.9 ↓
ViM-S	One sides	79.1	1.4 ↓	79.2	1.3 ↓	79.1	1.4 ↓	79.1	1.4 ↓
	Both sides	79.1	1.4 ↓	<b>79.3</b>	<b>1.2</b> ↓	79.1	1.4 ↓	79.1	1.4 ↓

## 6 Conclusion and Limitation

In this paper, we introduce a novel Sequential Token Merging method for ViMs. To develop a more adaptive token reduction technique, we first analyze the hidden states in Mamba and propose the concept of LDSD to describe the unique information transmission mechanism across hidden state chains. We identify the information selection and compression process occurs during interactions between inputs and hidden states. Guided by our insights into hidden states, we apply the BNNM

strategy to preserve the original dependence relationships and retain essential information through retrieving ViM’s selective scan. By compensating for perturbations and leveraging the self-convergence property of hidden states, we stabilize the key hidden state at the class token position. Our extensive experiments demonstrate the effectiveness of our STM strategy and provide insights into the dynamic information flow in Mamba, offering directions for future research on dynamic mechanisms. Although our method is effective, the inter-layer continuity of  $\bar{B}$  may not always hold, which limits the accuracy of the current estimation. While our uniform merging strategy is computationally efficient, it can be further optimized to achieve better balance and performance.

## References

- [1] A. Ali, I. Zimerman, and L. Wolf. The Hidden Attention of Mamba Models, Mar. 2024. arXiv:2403.01590 [cs].
- [2] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, C. Feichtenhofer, and J. Hoffman. Token Merging: Your ViT But Faster, Mar. 2023. arXiv:2210.09461 [cs].
- [3] K. Chen, B. Chen, C. Liu, W. Li, Z. Zou, and Z. Shi. RSMamba: Remote Sensing Image Classification With State Space Model. *IEEE Geoscience and Remote Sensing Letters*, 21:1–5, 2024.
- [4] H.-Y. Chiang, C.-C. Chang, N. Frumkin, K.-C. Wu, and D. Marculescu. Quamba: A Post-Training Quantization Recipe for Selective State Space Models, Dec. 2024. arXiv:2410.13229 [cs].
- [5] Y. Cho, C. Lee, S. Kim, and E. Park. PTQ4VM: Post-Training Quantization for Visual Mamba, Dec. 2024. arXiv:2412.20386 [cs].
- [6] T. Dao and A. Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality, May 2024. arXiv:2405.21060 [cs].
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.
- [8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minervini, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [9] A. Gu and T. Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, May 2024. arXiv:2312.00752 [cs].
- [10] A. Gu, K. Goel, and C. Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.
- [11] H. Guo, J. Li, T. Dai, Z. Ouyang, X. Ren, and S.-T. Xia. MambaIR: A Simple Baseline for Image Restoration with State-Space Model. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, editors, *Computer Vision – ECCV 2024*, pages 222–241, Cham, 2025. Springer Nature Switzerland.
- [12] D. Han, Z. Wang, Z. Xia, Y. Han, Y. Pu, C. Ge, J. Song, S. Song, B. Zheng, and G. Huang. Demystify Mamba in Vision: A Linear Attention Perspective, Dec. 2024. arXiv:2405.16605 [cs].
- [13] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao. A Survey on Vision Transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):87–110, Jan. 2023.
- [14] A. Hatamizadeh and J. Kautz. MambaVision: A Hybrid Mamba-Transformer Vision Backbone, Mar. 2025. arXiv:2407.08083 [cs].
- [15] T. Huang, X. Pei, S. You, F. Wang, C. Qian, and C. Xu. LocalMamba: Visual State Space Model with Windowed Selective Scan, Mar. 2024. arXiv:2403.09338 [cs].
- [16] M. Kim, S. Gao, Y.-C. Hsu, Y. Shen, and H. Jin. Token Fusion: Bridging the Gap between Token Pruning and Token Merging. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1372–1381, Waikoloa, HI, USA, Jan. 2024. IEEE.
- [17] X. Lei, W. Zhang, and W. Cao. DVMSR: Distillated Vision Mamba for Efficient Super-Resolution. pages 6536–6546, 2024.

- [18] K. Li, X. Li, Y. Wang, Y. He, Y. Wang, L. Wang, and Y. Qiao. VideoMamba: State Space Model for Efficient Video Understanding. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, editors, *Computer Vision – ECCV 2024*, pages 237–255, Cham, 2025. Springer Nature Switzerland.
- [19] Y. Liang, C. Ge, Z. Tong, Y. Song, J. Wang, and P. Xie. Not All Patches are What You Need: Expediting Vision Transformers via Token Reorganizations, Apr. 2022. arXiv:2202.07800 [cs].
- [20] X. Liu, C. Zhang, and L. Zhang. Vision Mamba: A Comprehensive Survey and Taxonomy, May 2024. arXiv:2405.04404 [cs].
- [21] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, J. Jiao, and Y. Liu. VMamba: Visual State Space Model. *Advances in Neural Information Processing Systems*, 37:103031–103063, Dec. 2024.
- [22] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022.
- [23] X. Ma, X. Zhang, and M.-O. Pun. RS3Mamba: Visual State Space Model for Remote Sensing Image Semantic Segmentation. *IEEE Geoscience and Remote Sensing Letters*, 21:1–5, 2024. Conference Name: IEEE Geoscience and Remote Sensing Letters.
- [24] L. Meng, H. Li, B.-C. Chen, S. Lan, Z. Wu, Y.-G. Jiang, and S.-N. Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12309–12318, 2022.
- [25] B. Pan, R. Panda, Y. Jiang, Z. Wang, R. Feris, and A. Oliva. IA-RED<sup>2</sup>: Interpretability-Aware Redundancy Reduction for Vision Transformers. *Advances in neural information processing systems*, 34:24898–24911, 2021.
- [26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [27] B. N. Patro and V. S. Agneeswaran. SiMBA: Simplified Mamba-Based Architecture for Vision and Multivariate Time series, Apr. 2024. arXiv:2403.15360 [cs].
- [28] X. Pei, T. Huang, and C. Xu. EfficientVMamba: Atrous Selective Scan for Light Weight Visual Mamba. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(6):6443–6451, Apr. 2025. Number: 6.
- [29] A. Ramachandran, M. Lee, H. Xu, S. Kundu, and T. Krishna. OuroMamba: A Data-Free Quantization Framework for Vision Mamba Models, Mar. 2025. arXiv:2503.10959 [cs].
- [30] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.
- [31] C. Renggli, A. S. Pinto, N. Houlsby, B. Mustafa, J. Puigcerver, and C. Riquelme. Learning to merge tokens in vision transformers. *arXiv preprint arXiv:2202.12015*, 2022.
- [32] J. Ruan, J. Li, and S. Xiang. VM-UNet: Vision Mamba UNet for Medical Image Segmentation, Nov. 2024. arXiv:2402.02491 [eess].
- [33] H. Shen, Z. Wan, X. Wang, and M. Zhang. Famba-V: Fast Vision Mamba with Cross-Layer Token Fusion, Oct. 2024. arXiv:2409.09808 [cs].
- [34] B.-Y. Shi, Y.-C. Lo, An-Yeu, Wu, and Y.-M. Tsai. Post-Training Quantization for Vision Mamba with k-Scaled Quantization and Reparameterization, Feb. 2025. arXiv:2501.16738 [eess].
- [35] M. Shi, Y. Zhou, R. Yu, Z. Li, Z. Liang, X. Zhao, X. Peng, T. Rajpurohit, S. R. Vedantam, W. Zhao, K. Wang, and Y. You. Faster Vision Mamba is Rebuilt in Minutes via Merged Token Re-training, Dec. 2024. arXiv:2412.12496 [cs].
- [36] Y. Shi, M. Dong, and C. Xu. Multi-Scale VMamba: Hierarchy in Hierarchy Visual State Space Model, May 2024. arXiv:2405.14174 [cs].
- [37] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

- [38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [39] C. Yang, Z. Chen, M. Espinosa, L. Ericsson, Z. Wang, J. Liu, and E. J. Crowley. PlainMamba: Improving Non-Hierarchical Mamba in Visual Recognition, Aug. 2024. arXiv:2403.17695 [cs].
- [40] H. Yin, A. Vahdat, J. M. Alvarez, A. Mallya, J. Kautz, and P. Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10809–10818, 2022.
- [41] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567, 2021.
- [42] Z. Zhan, Z. Kong, Y. Gong, Y. Wu, Z. Meng, H. Zheng, X. Shen, S. Ioannidis, W. Niu, P. Zhao, and Y. Wang. Exploring Token Pruning in Vision State Space Models, Sept. 2024. arXiv:2409.18962 [cs].
- [43] Z. Zhan, Y. Wu, Z. Kong, C. Yang, Y. Gong, X. Shen, X. Lin, P. Zhao, and Y. Wang. Rethinking Token Reduction for State Space Models, Oct. 2024. arXiv:2410.14725 [cs].
- [44] X. Zhang, Y. Tian, L. Xie, W. Huang, Q. Dai, Q. Ye, and Q. Tian. Hivit: A simpler and more efficient design of hierarchical vision transformer. In *The Eleventh International Conference on Learning Representations*, 2023.
- [45] S. Zhao, H. Chen, X. Zhang, P. Xiao, L. Bai, and W. Ouyang. RS-Mamba for Large Remote Sensing Image Dense Prediction. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–14, 2024. Conference Name: IEEE Transactions on Geoscience and Remote Sensing.
- [46] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model, Nov. 2024. arXiv:2401.09417 [cs].

## Appendix for Mathematical Formulations

### A Merging Process

The forward merging formula is derived as follows. For clarity, we first extend the hidden state equation (4) at two remaining position  $q_{(k-1)}$  and  $q_{(k)}$ :

$$\begin{aligned} h_{q_{(k-1)}} &= \bar{A}_{q_{(k-1)}} \bar{A}_{q_{(k-1)}-1} \dots \bar{A}_2 \bar{B}_1 x_1 \\ &\quad + \bar{A}_{q_{(k-1)}} \bar{A}_{q_{(k-1)}-1} \dots \bar{A}_3 \bar{B}_2 x_2 \\ &\quad + \dots \\ &\quad + \bar{A}_{q_{(k-1)}} \bar{B}_{q_{(k-1)}-1} x_{q_{(k-1)}-1} \\ &\quad + \bar{B}_{q_{(k-1)}} x_{q_{(k-1)}} \end{aligned} \tag{14}$$

$$\begin{aligned} h_{q_{(k)}} &= \bar{A}_{q_{(k)}} \bar{A}_{q_{(k)}-1} \dots \bar{A}_{q_{(k-1)}+1} \bar{A}_{q_{(k-1)}} \bar{A}_{q_{(k-1)}-1} \dots \bar{A}_2 \bar{B}_1 x_1 \\ &\quad + \bar{A}_{q_{(k)}} \bar{A}_{q_{(k)}-1} \dots \bar{A}_{q_{(k-1)}+1} \bar{A}_{q_{(k-1)}} \bar{A}_{q_{(k-1)}-1} \dots \bar{A}_3 \bar{B}_2 x_2 \\ &\quad + \dots \\ &\quad + \bar{A}_{q_{(k)}} \bar{A}_{q_{(k)}-1} \dots \bar{A}_{q_{(k-1)}+1} \bar{A}_{q_{(k-1)}} \bar{B}_{q_{(k-1)}-1} x_{q_{(k-1)}-1} \\ &\quad + \bar{A}_{q_{(k)}} \bar{A}_{q_{(k)}-1} \dots \bar{A}_{q_{(k-1)}+1} \bar{B}_{q_{(k-1)}} x_{q_{(k-1)}} \\ &\quad + \bar{A}_{q_{(k)}} \bar{A}_{q_{(k)}-1} \dots \bar{A}_{q_{(k-1)}+2} \bar{B}_{q_{(k-1)}+1} x_{q_{(k-1)}+1} \\ &\quad + \dots \\ &\quad + \bar{A}_{q_{(k)}} \bar{B}_{q_{(k)}-1} x_{q_{(k)}-1} \\ &\quad + \bar{B}_{q_{(k)}} x_{q_{(k)}} \end{aligned} \tag{15}$$

By examining the two equations, we observe that the second can be expressed in terms of the first as follows,

$$\begin{aligned} h_{q_{(k)}} &= \left( \prod_{j=q_{(k-1)}+1}^{q_{(k)}} \bar{A}_j \right) h_{q_{(k-1)}} \\ &\quad + \bar{A}_{q_{(k)}} \bar{A}_{q_{(k)}-1} \dots \bar{A}_{q_{(k-1)}+2} \bar{B}_{q_{(k-1)}+1} x_{q_{(k-1)}+1} \\ &\quad + \dots \\ &\quad + \bar{A}_{q_{(k)}} \bar{B}_{q_{(k)}-1} x_{q_{(k)}-1} \\ &\quad + \bar{B}_{q_{(k)}} x_{q_{(k)}} \end{aligned} \tag{16}$$

Now we merge the reduced tokens  $\{x_{q_{(k-1)}+j}\}_{j=1}^{R_{k-1}}$  into the nearest remaining neighborhood token  $x_{q_{(k)}}$ , and let us introduce a merging function  $x_{q_{(k)}}^* = f(x_{q_{(k-1)}+1}, x_{q_{(k-1)}+2}, x_{q_{(k-1)}+3}, \dots, x_{q_{(k)}})$  to describe the merging process. Aligning with our notation in 4.1,  $R_{k-1}$  is the number of the reduced tokens between  $x_{q_{(k-1)}}$  and  $x_{q_{(k)}}$ . We denote the modified hidden state at the merging position  $q_{(k)}$  as  $h_{q_{(k)}}^*$ , and it can be computed as follows,

$$\begin{aligned} h_{q_{(k)}}^* &= \left( \prod_{j=q_{(k-1)}+1}^{q_{(k)}} \bar{A}_j \right) h_{q_{(k-1)}} \\ &\quad + \bar{B}_{q_{(k)}} x_{q_{(k)}}^* \end{aligned} \tag{17}$$

To ensure that the new hidden state matches the original  $h_{q_{(k)}}^*$  (equation (15)), we aim to design a merging function that maintains this consistency.

$$\text{Objective : } x_{q_{(k)}}^* = f(x_{q_{(k-1)}+1}, x_{q_{(k-1)}+2}, x_{q_{(k-1)}+3}, \dots, x_{q_{(k)}}) \quad \text{s.t.} \quad h_{q_{(k)}}^* = h_{q_{(k)}} \tag{18}$$

By comparing equations (16) and (17), the differing term can be identified as follows:

$$\begin{aligned}\bar{B}_{q(k)} x_{q(k)}^* &= \bar{A}_{q(k)} \bar{A}_{q(k)-1} \cdots \bar{A}_{q(k-1)+2} \bar{B}_{q(k-1)+1} x_{q(k-1)+1} \\ &\quad + \cdots \\ &\quad + \bar{A}_{q(k)} \bar{B}_{q(k)-1} x_{q(k)-1} \\ &\quad + \bar{B}_{q(k)} x_{q(k)}\end{aligned}\tag{19}$$

Solving the equation yields:

$$\begin{aligned}x_{q(k)}^* &= \bar{A}_{q(k)} \bar{A}_{q(k)-1} \cdots \bar{A}_{q(k-1)+2} \frac{\bar{B}_{q(k-1)+1}}{\bar{B}_{q(k)}} x_{q(k-1)} \\ &\quad + \cdots \\ &\quad + \bar{A}_{q(k)} \frac{\bar{B}_{q(k)-1}}{\bar{B}_{q(k)}} x_{q(k)-1} \\ &\quad + x_{q(k)} \\ &= \sum_{j=q(k-1)+1}^{q(k)} \left( \prod_{n=j+1}^{q(k)} \bar{A}_n \right) \bar{B}_j x_j\end{aligned}\tag{20}$$

Rewrite the summation notation, we obtain:

$$\begin{aligned}x_{q(k)}^* &= f(x_{q(k-1)+1}, x_{q(k-1)+2}, x_{q(k-1)+3}, \dots, x_{q(k)}) \\ &= \sum_{j=1}^{q(k)-q(k-1)} \left( \prod_{n=j+1}^{q(k)} \bar{A}_n \right) \frac{\bar{B}_{q(k-1)+j}}{\bar{B}_{q(k)}} x_{q(k-1)+j} \\ &= \sum_{j=1}^{R_{k-1}} \left( \prod_{n=j+1}^{q(k)} \bar{A}_n \right) \frac{\bar{B}_{q(k-1)+j}}{\bar{B}_{q(k)}} x_{q(k-1)+j} + x_{q(k)}\end{aligned}\tag{21}$$

The last term is exactly the same as (11), except for the notation of  $(l-1)$  of  $\bar{B}$ . Because the selective mechanism of Mamba, we cannot compute  $\bar{B}^{(l)}$  if we reduce the corresponding token  $x$  at the layer it's reduced. So we have to refer to the  $\bar{B}^{(l-1)}$  in the last layer to estimate  $\bar{B}^{(l)}$ . This is where almost all of loss come from.

## B Proof of the Corollary

We leverage the linearity of Mamba to derive the distance fading rule of the loss. According to (9), the final loss is defined as:

$$\begin{aligned}L_{q(k+1)} &= h_{q(k+1)} - h_{q(k+1)}^* \\ L_{q(k+1)+1} &= h_{q(k+1)+1} - h_{q(k+1)+1}^*\end{aligned}\tag{22}$$

Then, based on the discrete SSM formulation in equation (3), we have:

$$\begin{aligned}h_{q(k+1)+1}^* &= \bar{A}_{q(k+1)+1} h_{q(k+1)}^* + \bar{B}_{q(k+1)+1} x_{q(k+1)+1} \\ h_{q(k+1)+1} &= \bar{A}_{q(k+1)+1} h_{q(k+1)} + \bar{B}_{q(k+1)+1} x_{q(k+1)+1}\end{aligned}\tag{23}$$

So we get,

$$\begin{aligned}L_{q(k+1)+1} &= \bar{A}_{q(k+1)+1} [h_{q(k+1)} - h_{q(k+1)}^*] \\ &= \bar{A}_{q(k+1)+1} L_{q(k+1)}\end{aligned}\tag{24}$$

This completes the derivation of the corollary.