SERESA: SELECTIVE RECURRENCE RESTRICTION FOR STATE SPACE DUALITY ACCELERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Mamba-2's State Space Duality (SSD) enables hardware-efficient chunk-wise recurrence, yet its computational efficiency does not translate proportionally into prefill latency. Despite accounting for less than 5% of FLOPs, SSD consumes nearly 20% of prefill latency, motivating us to examine temporal dependencies across SSD heads. While SSD applies full recurrence uniformly, output-relevant recurrent contributions vary substantially in temporal range across heads. We quantify this heterogeneity by defining the effective recurrence distribution and its expected lag, Effective Recurrence Length (ERL). Our ERL analysis reveals that most heads concentrate their contributions at short temporal lags, whereas long-range contributions are confined to a small subset of heads. The head-wise ERL ordering also remains stable across input distributions, revealing a quasi-static recurrence structure. Building on this, we propose SERESA, a training-free framework that selectively restricts recurrence. SERESA determines the number of restricted heads from the ERL distribution and selects heads that retain the most contribution within a prescribed recurrence length. It reduces their intra- and inter-chunk computation while recovering boundary-crossing contributions through boundary correction, leaving full recurrence for the remaining heads. Without additional training or input-dependent reselection, SERESA restricts 78–83% of heads across Mamba-2 and Nemotron-H while closely matching full-recurrence baseline perplexity and downstream accuracy. It accelerates the SSD core by 1.75–$2.02\times$ and achieves up to $1.29\times$ end-to-end prefill speedup with projection quantization. Code will be made publicly available upon acceptance.

1 INTRODUCTION

Scaling sequence models to long contexts requires efficient sequence processing. Despite their strong modeling capacity, attention-based Transformers incur quadratic complexity with sequence length (Vaswani et al., 2017), making such scaling computationally expensive (Dao et al., 2022). Mamba reduces this complexity to linear through input-dependent state space recurrence (Gu & Dao, 2024), while Mamba-2 recasts the recurrence as chunk-wise structured computation via State Space Duality (SSD) (Dao & Gu, 2024). SSD evaluates intra-chunk recurrent contributions through matrix multiplication while propagating information across chunks via recurrent states, efficiently exploiting modern accelerators (Dao & Gu, 2024). Mamba-3 further generalizes this recurrence through exponential-trapezoidal discretization while preserving the SSD formulation (Lahoti et al., 2026). Yet, SSD's hardware-efficient design does not fully translate into proportional latency savings during prefill. Our profiling reveals a striking disparity: SSD accounts for only 4.74% of total FLOPs and 3.81% of the memory footprint, yet consumes nearly 20% of the total prefill latency. This discrepancy between computational cost and actual latency warrants a closer examination of SSD computation, suggesting substantial room for prefill optimization.

Prior work has explored SSD optimization along distinct computational dimensions. SSDi8 quantizes SSD activations to INT8, executing the original computation at lower precision (Kim et al., 2026). While this reduces per-operation execution cost, it leaves the structure and volume of full SSD computation unchanged. Accordingly, our prefill evaluation shows only a 1.26–$1.31\times$ SSD core speedup. GHOST, in contrast, applies structured pruning to the recurrent state dimension, reducing state-update cost (Menezes & Kyrillidis, 2026). Pruning 50% of the state dimensions yields a 1.50–$1.66\times$ SSD core speedup, but reduces state capacity and increases perplexity, with pronounced

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.

degradation on Mamba2-8B. These results indicate substantial headroom for further reducing SSD inference latency beyond precision reduction and state pruning.

To explore an alternative optimization axis, we shift our focus to the head-wise recurrence structure of SSD layers. SSD partitions its hidden representation across multiple SSD heads, each with an independent recurrent state and head-specific transition dynamics (Dao & Gu, 2024). Consequently, the dynamics governing the retention and decay of past information can vary substantially across heads. Yet, SSD applies full recurrence computation uniformly to all heads, irrespective of their contribution to the output. This motivates our central question: *Does every SSD head require full recurrence to preserve model performance?*

We formalize this question through the effective recurrence distribution, which characterizes how output-relevant recurrent contributions are distributed over temporal lags. Specifically, we aggregate and normalize the contribution of the value at position j to the output at position i through the recurrent path as a function of temporal lag $i - j$, thereby defining the effective recurrence distribution over temporal lags, whose expected lag we use as the Effective Recurrence Length (ERL). Our analysis reveals pronounced head-wise heterogeneity: most low-ERL heads concentrate their contributions at short temporal lags, whereas long-range recurrent contributions are concentrated in a small subset of high-ERL heads. This imbalance suggests that applying full recurrence uniformly across all heads may be unnecessary.

Motivated by this head-wise heterogeneity, we propose SERESA (SElective REcurrence restriction for State space duality Acceleration), a training-free framework for SSD prefill acceleration. SERESA determines the fraction of heads to restrict from the head-wise ERL distribution and identifies the restricted heads based on their contribution coverage within a prescribed recurrence length. For these heads, SERESA restricts recurrence to the prescribed length, eliminating SSD computation beyond this recurrence range while preserving full recurrence for the remaining heads. This head-wise restriction is directly incorporated into chunk-wise SSD computation to realize tangible latency reductions. Moreover, head-wise ERL ordering exhibits a quasi-static property that remains stable across input domains, allowing the head partition to be fixed throughout deployment without per-input restriction decisions or additional training.

Across Mamba2-1.3B, 2.7B, and 8B, SERESA restricts recurrence computation in approximately 79–83% of all heads while maintaining performance comparable to the full-recurrence baseline, achieving up to $2\times$ SSD speedup. On the hybrid Nemotron-H-8B architecture, SERESA further achieves approximately $1.75\times$ SSD speedup, demonstrating its effectiveness across both model scales and hybrid architectures. By eliminating recurrence computation itself, SERESA operates along a complementary optimization axis distinct from quantization and state-dimension reduction. When combined with quantization, SERESA yields up to $1.29\times$ end-to-end prefill speedup. These results demonstrate that full recurrence is not uniformly required across SSD heads and that incorporating head-wise recurrence structure into the execution policy provides an effective approach to accelerating SSD prefill. Our main contributions are summarized as follows.

- We introduce the effective recurrence distribution and its expected lag, Effective Recurrence Length (ERL), revealing pronounced yet quasi-static head-wise heterogeneity, with long-range contributions concentrated in a small subset of heads.
- We propose SERESA, a training-free framework that introduces head-wise selective recurrence restriction as a new optimization axis for SSD acceleration, eliminating unnecessary recurrence computation while retaining full recurrence for long-range heads.
- We validate SERESA across model scales and a hybrid architecture, achieving up to $2\times$ SSD speedup on Mamba-2 and $1.75\times$ on Nemotron-H while largely preserving baseline performance without fine-tuning. Combined with quantization, SERESA further achieves $1.29\times$ end-to-end prefill speedup, demonstrating complementarity with quantization.

2 Related Work

2.1 Mamba

Mamba introduces selective state space models (SSMs) by making conventional linear time-invariant SSM parameters input-dependent, enabling data-dependent state retention and updates

with linear work via associative scan (Gu & Dao, 2024). However, its scan-based recurrence is less amenable to GPU tensor cores optimized for matrix multiplication. Mamba-2 improves hardware efficiency by simplifying state transitions to a per-head scalar-times-identity form and introducing State Space Duality (SSD), which represents recurrence as a lower-triangular semiseparable matrix (Dao & Gu, 2024). Through chunk-wise decomposition, SSD evaluates the same full recurrence in linear complexity while using intra-chunk matrix multiplications. Mamba-3 preserves this parallel SSD formulation while generalizing exponential-Euler discretization to a more expressive exponential-trapezoidal scheme (Lahoti et al., 2026). Further technical details on Mamba formulations are provided in Appendix A.

2.2 State Space Duality Optimization

Prior work on SSD optimization has primarily focused on reducing numerical precision or the state dimension. SSDi8 applies activation-aware quantization to accelerate prefill by executing the full SSD computation at lower precision, while leaving the amount of computation unchanged (Kim et al., 2026). GHOST structurally prunes the state dimension to reduce recurrent-state memory and computation, but may incur quality degradation due to reduced state capacity (Menezes & Kyrillidis, 2026). However, both methods retain full recurrence for all heads and do not optimize recurrence length on a per-head basis. SERESA instead targets the recurrence-length axis, selectively eliminating unnecessary SSD prefill computation on a per-head basis while preserving both precision and state dimension. It therefore provides an optimization axis complementary to quantization and state-dimension pruning.

2.3 Sparse Attention

Efficient Transformer attention reduces quadratic computation by restricting the query–key support. Existing methods construct this support using predefined (Child et al., 2019), head-wise (Xiao et al., 2025), or input-adaptive patterns (Jiang et al., 2024), thereby reducing the number of token pairs or blocks evaluated during attention. Unlike these attention methods, SERESA restricts the temporal extent of SSD recurrence rather than query–key support. It derives head-wise computation policies from each head's effective recurrence distribution, which characterizes output-relevant recurrent contributions across temporal lags. SERESA retains full recurrence for heads requiring long-range contributions while restricting the remaining heads to a limited recurrence length, converting head-wise recurrence heterogeneity into a length-restricted computation structure.

3 Preliminaries: Head-Wise SSD Formulation and Chunk-Wise Decomposition

For a sequence of length L, SSD head h has input and output $X_h, Y_h \in \mathbb{R}^{L\times d}$ satisfying $Y_h = M_h X_h$, where d is the head dimension. The same SSD matrix M_h, determined by the discretized SSM parameters, applies to all d channels within the head. At sequence position $t \in \{1,\dots,L\}$, the discrete decay factor $\bar{A}_{t,h} = \exp(A_h \Delta_{t,h})$ is determined by the head-specific transition parameter $A_h < 0$ and input-dependent step size $\Delta_{t,h}$. Meanwhile, B and C are parameterized over G groups and shared within each group. For the group $g(h)$ containing head h, the discretized B is $\bar{B}_{t,h} = \Delta_{t,h} B_{t,g(h)}$, incorporating the head-specific step size.

The full recurrence of a single SSD head can be written as the following causal lower-triangular matrix transformation, where we omit the head index for clarity:

$$Y = MX = \begin{bmatrix} C_1^\top \bar{B}_1 & 0 & \cdots & 0 \\ \bar{A}_2 C_2^\top \bar{B}_1 & C_2^\top \bar{B}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \left(\prod_{k=2}^{L} \bar{A}_k\right) C_L^\top \bar{B}_1 & \left(\prod_{k=3}^{L} \bar{A}_k\right) C_L^\top \bar{B}_2 & \cdots & C_L^\top \bar{B}_L \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_L \end{bmatrix}. \tag{1}$$

For $j \le i$, $M_{i,j} X_j$ represents the contribution of the input at position j to the output at position i through the recurrent path. The decay product $\prod_{k=j+1}^{i} \bar{A}_k$ determines how much of the information written at position j is retained as it propagates to position i. Consequently, Y_i aggregates recurrent contributions $M_{i,j} X_j$ from all positions $j \le i$, each associated with a temporal lag $i - j$.

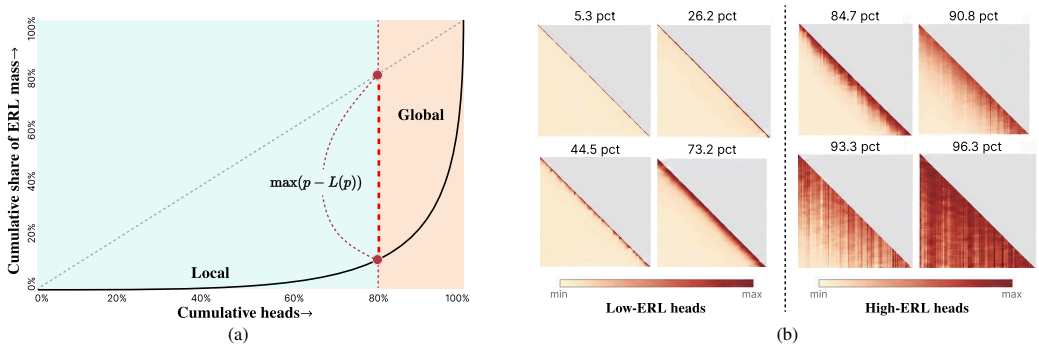


Figure 1: Head-wise effective recurrence structure in Mamba2-2.7B. (a) Cumulative ERL share $L(p)$ versus head share p; the maximizer of $p - L(p)$ determines the local-head budget. (b) Representative SSD matrices at different ERL percentiles, illustrating distinct temporal contribution patterns.

Intra- and Inter-Chunk Computation. Rather than explicitly constructing M_h, SSD partitions the sequence into chunks of size Q (Dao & Gu, 2024). Conceptually, M_h^{intra} retains the block-diagonal regions of M_h, while M_h^{inter} captures the off-diagonal regions corresponding to contributions from preceding chunks. Intra-chunk contributions are computed via independent matrix multiplications, whereas inter-chunk contributions are propagated across chunks through recurrent states. Defining $Y_h^{\text{intra}} = M_h^{\text{intra}} X_h$ and $Y_h^{\text{inter}} = M_h^{\text{inter}} X_h$, the SSD output is $Y_h = Y_h^{\text{intra}} + Y_h^{\text{inter}}$.

4 Analysis of Effective Recurrence Length

Mamba-2 applies full recurrence uniformly across SSD heads (Dao & Gu, 2024). However, each head's decay factor $\bar{A}_{t,h} = \exp(A_h \Delta_{t,h})$ depends on a head-specific A_h and an input-dependent $\Delta_{t,h}$, such that heads may differ in their retention of past information. We quantify this variation through each head's distribution of output-relevant recurrent contributions across temporal lags, jointly accounting for the decay, input X_h, and input-dependent B and C. We find that long-range contributions are concentrated in a small subset of heads and that this structure remains stable across input domains.

4.1 Effective Recurrence Distribution

The sequence length, number of heads, and head dimension are denoted by L, H, and d, respectively. Sequence positions are indexed by $i, j \in \{1, \ldots, L\}$, and heads by $h \in \{1, \ldots, H\}$. We first characterize how each head's output-relevant recurrent contributions are distributed over temporal lags, and then use this distribution to quantify its effective recurrence length.

For head h, the recurrent contribution from position j to output position i depends on both the SSD coefficient $M_{i,j,h}$ and the source input $X_{j,h}$. We therefore define the pairwise recurrent contribution as $c_h(i,j) = \|M_{i,j,h} X_{j,h}\|_2$, where $M_{i,j,h} \in \mathbb{R}$ and $X_{j,h} \in \mathbb{R}^d$. We associate each contribution with its temporal lag $\ell = i - j$. By causality, only pairs with $j \le i$ are valid, yielding $\ell \in \{0, \ldots, L-1\}$. We aggregate contributions from all position pairs sharing the same lag:

$$m_h(\ell) = \sum_{i-j=\ell} c_h(i,j), \qquad \ell = 0, 1, \ldots, L-1. \tag{2}$$

Normalizing this contribution mass yields the effective recurrence distribution of head h:

$$p_h(\ell) = \frac{m_h(\ell)}{\sum_{\ell=0}^{L-1} m_h(\ell)}, \qquad \ell = 0, \ldots, L-1, \tag{3}$$

where $p_h(\ell)$ represents the fraction of output-relevant recurrent contribution at lag ℓ. This distribution captures the head's full temporal contribution profile. We summarize it by its expected lag, Effective Recurrence Length (ERL):

$$\text{ERL}(h) = \sum_{\ell=0}^{L-1} \ell p_h(\ell) = \frac{\sum_{\ell=0}^{L-1} \ell m_h(\ell)}{\sum_{\ell=0}^{L-1} m_h(\ell)}. \tag{4}$$

Thus, the effective recurrence distribution describes how contributions are spread across lags, while ERL provides a scalar summary for comparing their temporal range across heads.

4.2 Quasi-Static Head-Wise Recurrence-Length Imbalance

Recurrence-Length Imbalance. We first examine how effective recurrence is distributed across SSD heads to assess whether full recurrence is uniformly warranted. Figure 1(a) characterizes the head-wise ERL distribution of Mamba2-2.7B using a Lorenz curve, with heads ordered by increasing ERL. Its pronounced curvature indicates substantial ERL concentration among a small fraction of heads. Mamba2-2.7B yields a Gini coefficient of 0.834, with the top 10% of heads accounting for 74.6% of the total ERL; corresponding analyses are provided in Appendix B.1. The recurrent contribution patterns further substantiate this heterogeneity. As shown in Figure 1(b), low-ERL heads exhibit contributions concentrated near the diagonal, indicating predominantly short-range recurrence. As ERL increases, the contribution mass extends toward longer temporal lags. These observations reveal a highly asymmetric recurrence structure: short-range recurrence dominates most heads, whereas long-range recurrence is concentrated in a small high-ERL subset.

Quasi-Static ERL Structure. We next examine whether this recurrence-length disparity reflects a persistent head-wise structure beyond input-dependent variation. We decompose log-ERL variation measured on WikiText-2 (Merity et al., 2017), CodeParrot (CodeParrot, 2022), GSM8K (Cobbe et al., 2021), and ZH-Wiki (Wikimedia Foundation, 2023) into head, domain, and head×domain interaction effects. As detailed in Appendix B.2, head differences account for 98.41% of the total variation, compared with only 0.51% for domain effects and 1.08% for their interaction, indicating that domain-induced variation is minor relative to head-wise disparity. This head-dominant structure also persists in relative ordering: across all domain pairs, head-wise ERL in Mamba2-2.7B yields Spearman rank correlations of 0.9773–0.9912, with the ordering of low- and high-ERL heads largely preserved. Both the recurrence-length disparity and relative head hierarchy thus remain stable across input distributions.

5 SERESA

ERL is highly heterogeneous across SSD heads, while the head-wise ERL ordering remains stable across input distributions. SERESA exploits this quasi-static heterogeneity by partitioning SSD heads into local heads, which perform length-restricted recurrence that retains contributions within a prescribed length, and global heads, which preserve full SSD recurrence. This partition is incorporated into chunk-wise SSD computation to eliminate unnecessary recurrence computation while preserving the required boundary-crossing contributions.

5.1 Distribution-Guided Recurrence Policy

Recurrence Length Restriction. The cumulative decay in SSD suppresses recurrent contributions with increasing temporal lag, concentrating them within a limited range. Since the decay rate depends on head-specific parameters and input-dependent transitions, the effective recurrence span varies across heads. SERESA exploits this heterogeneity by restricting local heads to a recurrence length R, retaining contributions with $i-j < R$ and eliminating those with $i-j \geq R$. Global heads retain full SSD recurrence to preserve long-range contributions, with the resulting head partition fixed throughout inference.

Local-Head Budget from ERL. ERL captures the overall temporal range of each head as the expected lag under its effective recurrence distribution $p_h(\ell)$. As shown in Section 4.2, its head-wise distribution is strongly skewed: most heads have low ERL, whereas long-range recurrence is concentrated in a small high-ERL subset. The Lorenz curve in Figure 1(a) ranks heads by ascending ERL and plots the cumulative ERL fraction $L(p)$ against the cumulative head fraction p. The gap $p - L(p)$ measures the disparity between a fraction of heads and its share of total ERL mass. Its maximum reflects both the prevalence of low-ERL heads and the concentration of ERL mass in the high-ERL tail, providing a distribution-derived reference for the local-head budget. Since the Lorenz-curve slope is $L'(p) = \mathrm{ERL}(p)/\overline{\mathrm{ERL}}$, the maximum occurs where $\mathrm{ERL}(p^\star) = \overline{\mathrm{ERL}}$.

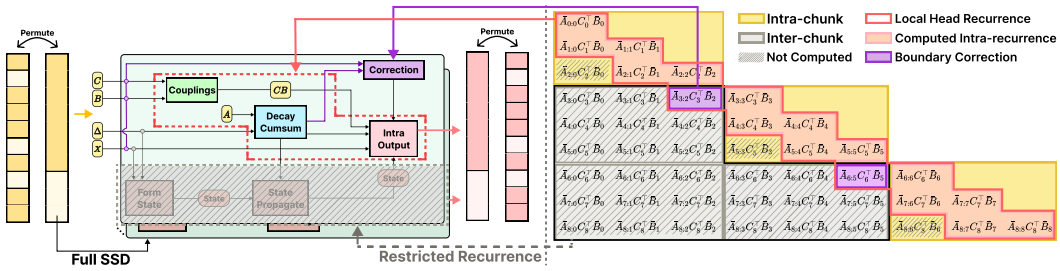


Figure 2: Overview of SERESA. Global heads retain the full SSD computation, whereas local heads omit the full inter-chunk recurrence and recover the required boundary-crossing contributions through boundary correction. The right panel shows the resulting length-restricted structure of the SSD matrix.

SERESA therefore defines the local-head budget, i.e., the number of heads subject to recurrence restriction, as

$$K = \left|\left\{h \mid \mathrm{ERL}(h) < \overline{\mathrm{ERL}}\right\}\right|. \tag{5}$$

Local-Head Selection by Coverage. The local-head budget K determines only how many heads are subject to recurrence restriction. Which heads to restrict is determined by how much of each head's recurrent contribution is retained within the recurrence length R. We define the coverage of head h as

$$\textit{Coverage}_h(R) = \sum_{\ell<R} p_h(\ell). \tag{6}$$

$\textit{Coverage}_h$ measures the fraction of recurrent contribution retained within lag R; higher coverage therefore implies less contribution loss under recurrence restriction. The contribution beyond R is $1 - \textit{Coverage}_h(R)$, and ERL satisfies the tail-sum identity

$$\mathrm{ERL}(h) = \sum_{\ell=0}^{L-1} \ell p_h(\ell) = \sum_{R=1}^{L-1}\sum_{\ell=R}^{L-1} p_h(\ell) = \sum_{R=1}^{L-1} \left(1 - \textit{Coverage}_h(R)\right). \tag{7}$$

Thus, ERL aggregates contributions beyond all possible R, summarizing a head's overall temporal range, whereas $\textit{Coverage}_h$ quantifies the contribution retained at a specific R. Accordingly, ERL decides the local-head budget K, while $\textit{Coverage}_h$ selects the individual heads within that budget. Given the full head set $\mathcal{H}$, SERESA selects the K heads with the highest $\textit{Coverage}_h$:

$$\mathcal{H}_{\text{local}} = \underset{\mathcal{S}\subseteq\mathcal{H},\,|\mathcal{S}|=K}{\arg\max} \sum_{h\in\mathcal{S}} \textit{Coverage}_h(R). \tag{8}$$

This selection maximizes the recurrent contribution retained under length restriction, subject to the ERL-derived local-head budget. The head-partitioning algorithm is provided in Appendix C.

5.2 Efficient Computation of Length-Restricted SSD Recurrence

SERESA realizes the recurrence restriction policy directly within SSD's intra- and inter-chunk computation, eliminating computation for recurrent contributions with temporal lag $i - j \geq R$ in local heads. Figure 2 illustrates the overall computation flow and the corresponding SSD matrix structure. SERESA permutes heads according to the local/global partition, processes the two groups through distinct computation paths, and restores the original head ordering via inverse permutation.

Inter-Chunk Recurrence Restriction. $\textit{Coverage}_h$ of local heads rapidly saturates with R, reaching 99.58% on average at $R = Q$, where Q denotes the chunk size, as analyzed in Appendix D. This indicates negligible recurrent contributions beyond R. SERESA therefore imposes the length-R restriction directly on inter-chunk computation. Global heads construct recurrent states from each chunk and propagate them across subsequent chunks, realizing full recurrence over M_h^{inter}, which captures contributions across chunks. Local heads, however, require only contributions with $i-j < R$. SERESA bypasses their full inter-chunk recurrence, eliminating operations for $i-j \geq R$.

Table 1: Perplexity and SSD core latency across models. Config indicates bit precision for SSDi8, state-dimension pruning for GHOST, and local-head ratio for SERESA; speedup is measured relative to the corresponding baseline. See Appendix G for GHOST at 25% pruning.

Model	Method	Config.	Perplexity			Speed	
			Pile	WikiText-2	C4	SSD (ms)	Speedup
Mamba2-1.3B	Baseline	–	6.99	10.42	14.78	7.47	1.00×
	SSDi8	8-bit	7.04	10.55	14.89	5.86	1.27×
	GHOST	N (50%)	7.25	11.77	15.71	4.81	1.55×
	SERESA (ours)	83.19%	**7.01**	**10.43**	**14.80**	**3.70**	**2.02**×
Mamba2-2.7B	Baseline	–	6.28	9.06	13.26	9.37	1.00×
	SSDi8	8-bit	6.30	9.13	13.32	7.17	1.31×
	GHOST	N (50%)	6.61	10.11	13.74	6.25	1.50×
	SERESA (ours)	81.25%	**6.29**	**9.07**	**13.27**	**4.76**	**1.97**×
Mamba2-8B	Baseline	–	5.99	7.25	9.96	15.76	1.00×
	SSDi8	8-bit	6.06	7.41	10.08	12.33	1.28×
	GHOST	N (50%)	11.87	17.44	21.95	9.50	1.66×
	SERESA (ours)	79.49%	**6.01**	**7.26**	**9.98**	**8.85**	**1.78**×
Nemotron-H-8B	Baseline	–	7.59	8.43	13.57	15.34	1.00×
	SSDi8	8-bit	7.83	8.72	14.09	12.17	1.26×
	GHOST	N (50%)	8.56	9.30	14.98	9.57	1.60×
	SERESA (ours)	78.38%	**7.59**	**8.36**	**13.54**	**8.77**	**1.75**×

Consequently, the K local heads require neither inter-chunk recurrent states nor recurrence across inter-chunk states, thereby reducing computation. This removal eliminates the entire M_h^{inter}, shown as the gray region in Figure 2. However, this region also contains boundary-crossing contributions within R, highlighted in purple. These contributions must therefore be preserved even when i and j reside in different chunks.

Boundary Correction. As illustrated in Figure 2, SERESA separately restores the removed boundary-crossing contributions through boundary correction. For adjacent chunks of length Q, the temporal lag between two positions is $Q+i-j$, and only contributions satisfying $Q+i-j < R$ are restored. When $R \leq Q$, these contributions are confined to a triangular region spanning the end of the preceding chunk and the beginning of the current chunk. SERESA constructs this boundary-correction block using the precomputed cumulative decay, B, and C, and adds it to the local-head output via batched matrix multiplication. This restores the required cross-chunk contributions within recurrence length R without reinstating full inter-chunk recurrence.

Length-Restricted Intra-Chunk Computation. Removing inter-chunk recurrence alone leaves the full causal block of M_h^{intra} computed within each chunk size Q. Since output-relevant contributions of local heads are concentrated at shorter temporal lags, SERESA further restricts intra-chunk computation to entries satisfying $i - j < R$. Each local-head output thus receives intra-chunk contributions only from the current and preceding $R - 1$ positions. Consequently, local heads require only length-restricted intra-chunk computation and boundary correction.

6 Experiments

In the main experiments, we evaluate SERESA on Mamba2-1.3B, Mamba2-2.7B, Mamba2-8B (Waleffe et al., 2024), and Nemotron-H-8B (NVIDIA et al., 2025), using $R = 64$ by default. For each model, head-wise ERL and *Coverage*$_h$ are averaged over five length-2,048 Pile (Gao et al., 2020) calibration sequences to determine a fixed head partition. This yields local-head ratios of 83.19%, 81.25%, 79.49%, and 78.38%, respectively. We evaluate perplexity on Pile, WikiText-2 (Merity et al., 2017), and C4 (Raffel et al., 2020), along with downstream accuracy. We compare against full SSD, SSDi8 (Kim et al., 2026) with its 8-bit kernel, and GHOST (Menezes & Kyrillidis, 2026) with the state dimension pruned from $N = 128$ to $N = 64$. Ablations use Mamba2-2.7B unless noted otherwise. The main experiments use an NVIDIA A5000 at batch size 16 and sequence

length 2,048, with results averaged over 50 runs after 15 warm-ups. Implementation details are provided in Appendix E; calibration-set-size sensitivity is examined in Appendix F.

6.1 Language Modeling Performance and SSD Acceleration

We report language-modeling perplexity and SSD core speedup across four models in Table 1. SERESA restricts recurrence in 78–83% of SSD heads while maintaining perplexity close to the baseline on Pile, WikiText-2, and C4. It achieves SSD core speedups of 2.02×, 1.97×, 1.78×, and 1.75× on Mamba2-1.3B, Mamba2-2.7B, Mamba2-8B, and Nemotron-H-8B, respectively, outperforming both SSDi8 with INT8 execution and GHOST with 50% state-dimension pruning. In particular, GHOST substantially increases perplexity on Mamba2-8B, whereas SERESA remains close to the baseline. Mamba-3 shows a similar trend with Mamba-2, with results and ERL analysis in Appendix H.

Table 2: Accuracy (%) on WikiHop, Hyperpartisan, and ContractNLI for baseline and SERESA (ours).

Task	Model	Baseline	SERESA
WikiHop	Mamba2-1.3B	25.3%	25.0%
	Mamba2-2.7B	27.3%	27.0%
	Mamba2-8B	36.0%	35.0%
	Nemotron-H-8B	43.3%	43.0%
Hyperpartisan	Mamba2-1.3B	63.5%	62.8%
	Mamba2-2.7B	63.8%	63.8%
	Mamba2-8B	45.3%	45.8%
	Nemotron-H-8B	74.3%	73.3%
ContractNLI	Mamba2-1.3B	13.0%	13.3%
	Mamba2-2.7B	17.3%	16.8%
	Mamba2-8B	48.8%	48.8%
	Nemotron-H-8B	28.5%	30.3%

We evaluate downstream accuracy on WikiHop (Welbl et al., 2018), Hyperpartisan (Kiesel et al., 2019), and ContractNLI (Koreeda & Manning, 2021). Table 2 shows that absolute accuracy differences from the baseline remain within 1.8 percentage points across 12 model–task pairs. The largest decrease is 1.0 point for Mamba2-8B on WikiHop and Nemotron-H-8B on Hyperpartisan, while several pairs match or exceed the baseline. SERESA also maintains long-context performance, as shown in Appendix I.

6.2 Combination with Existing Compression Methods

We evaluate SERESA with optimizations outside the SSD core in Figure 3. Quamba2-based quantization (Chiang et al., 2025) quantizes the input and output projections and RMSNorm, while SparseGPT (Frantar & Alistarh, 2023) prunes the projections. As pruning uses no sparse kernel, we assess additional quality loss rather than direct latency reduction. On Mamba2-2.7B, SERESA achieves approximately 2× SSD core and 1.1× end-to-end speedups with negligible PPL degradation, consistent with Amdahl's law (Amdahl, 1967). Adding SERESA to quantization raises end-to-end speedup from 1.202× to 1.287×, while the relative WikiText-2 PPL increase changes only from 3.80% to 3.86%; with pruning, it remains approximately 3.64%. Thus, SERESA complements quantization and projection pruning while incurring minimal additional quality degradation.

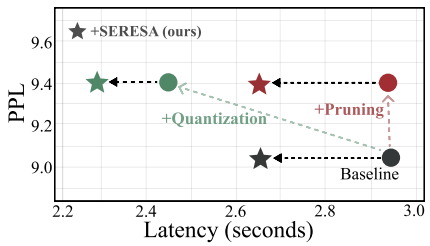


Figure 3: Combining SERESA with projection optimizations.

6.3 Ablation Study

Robustness to Calibration Distribution. We assess sensitivity to the calibration distribution in Table 3. For each dataset, we determine a head partition and evaluate it on the same and other datasets. Relative PPL changes remain within 0.0%–0.3% across all calibration–deployment pairs, including cross-dataset settings. This result is consistent with the quasi-static head-wise recurrence structure and indicates limited sensitivity to the calibration dataset.

Table 3: Perplexity increase (%) over the baseline across calibration–test pairs.

Calibration Dataset	Test Dataset			
	Pile	CodeParrot	GSM8K	ZH-Wiki
Pile	0.1%	0.3%	0.1%	0.2%
CodeParrot	0.0%	0.2%	0.1%	0.2%
GSM8K	0.1%	0.2%	0.1%	0.2%
ZH-Wiki	0.0%	0.2%	0.0%	0.2%

Head-Selection Criterion. We compare head-selection criteria while fixing the local-head ratio at 81.25% and $R = 64$, with exact boundary correction applied throughout. The alternatives are decay-only selection, the learning-based criterion of DuoAttention (Xiao et al., 2025), y-magnitude, and random selection. In Figure 4(a), SERESA yields the smallest relative PPL increases over the

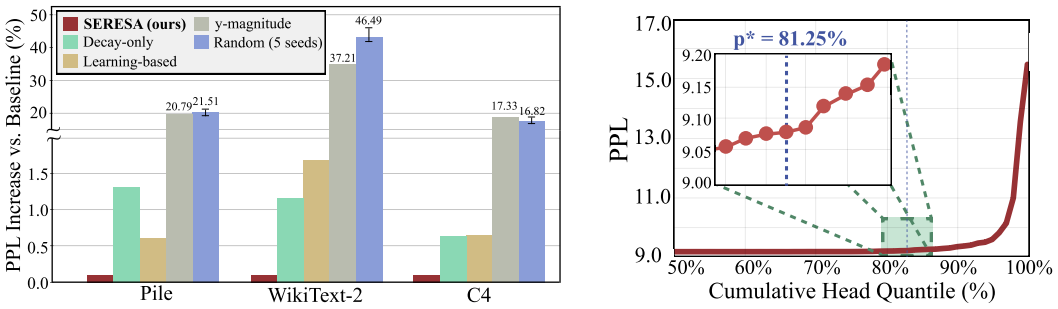


Figure 4: Ablations on Mamba2-2.7B. (a) Perplexity increase over baseline by head-selection criterion. (b) WikiText-2 perplexity versus local-head ratio; $p^\star = 81.25\%$ is selected by mean ERL.

baseline: 0.26%, 0.12%, and 0.08% on Pile, WikiText-2, and C4. The corresponding increases are 1.32%, 1.24%, and 0.59% for decay-only selection, and 0.61%, 1.65%, and 0.60% for learning-based selection. These results show that jointly accounting for output-relevant recurrent contribution and temporal lag improves head selection for recurrence restriction.

Local-Head Budget. We vary the local-head ratio in approximately 0.05 percentage-point increments. In Figure 4(b), PPL remains near the baseline at lower ratios and begins to increase around 80%. Using the mean ERL as the threshold yields a local-head ratio of 81.25%, where the relative PPL increases are 0.123% on WikiText-2 and 0.257% on Pile. These increases reach 0.492% and 0.923% at 84.1%, and 1.235% and 1.702% at 90%. Thus, the mean-ERL threshold provides a conservative local/global boundary without a performance-based search.

Component Ablation. We evaluate SERESA's components in Table 4. Removing full inter-chunk recurrence yields a perplexity of 9.19 and a $1.91\times$ SSD speedup. Adding intra-chunk restriction raises speedup to $2.16\times$, with perplexity at 9.20. Exact boundary correction reduces perplexity to 9.07 while retaining a $1.97\times$ speedup. Thus, inter- and intra-chunk restrictions provide the main computational savings, while boundary correction recovers most of the quality loss.

Table 4: Perplexity and SSD core speedup as SERESA components are added sequentially.

Config.	Perplexity	SSD×
Baseline	9.06	1.00×
+ Inter skip	9.19	1.91×
+ Intra restriction	9.20	2.16×
+ Boundary correction	**9.07**	**1.97**×

Recurrence Length. We fix the local-head partition and vary only R. Table 5 reports perplexities of 9.36, 9.14, and 9.07 at $R = 16$, 32, and 64, with corresponding SSD speedups of $2.52\times$, $2.28\times$, and $1.97\times$. At $R = 128$ and 256, perplexity matches the baseline of 9.06, while speedup decreases to $1.68\times$ and $1.27\times$, respectively. We therefore adopt $R = 64$ as the default, as it provides a favorable balance between near-baseline perplexity and an approximately $2\times$ SSD speedup. Additional ablation studies are reported in Appendices J and K.

Table 5: Perplexity and SSD core speedup across recurrence lengths.

Restriction	Perplexity	SSD×
16	9.36	2.52×
32	9.14	2.28×
64	9.07	1.97×
128	9.06	1.68×
256	9.06	1.27×

7 Conclusion

We revisit the uniform-recurrence assumption in SSD and reveal substantial head-wise heterogeneity in effective recurrence length. Long-range contributions concentrate in a small subset of heads, while most operate over short temporal ranges, with this structure remaining stable across input distributions. SERESA exploits this quasi-static structure to selectively restrict recurrence without additional training or input-dependent reselection. Across models with Mamba-based layers, SERESA preserves language modeling and downstream performance while substantially accelerating SSD computation, with robustness across calibration distributions. Its compatibility with quantization and pruning further demonstrates complementarity to existing optimization axes.

REFERENCES

Gene M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the Spring Joint Computer Conference*, pp. 483–485, 1967.

Anthropic. Claude. https://www.anthropic.com/claude, 2026. Accessed: 2026-09-26.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An open-source autoregressive language model. In Angela Fan, Suzana Ilic, Thomas Wolf, and Matthias Gallé (eds.), *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pp. 95–136, virtual+Dublin, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.bigscience-1.9. URL https://aclanthology.org/2022.bigscience-1.9/.

Hung-Yueh Chiang, Chi-Chih Chang, Natalia Frumkin, Kai-Chiang Wu, Mohamed S. Abdelfattah, and Diana Marculescu. Quamba2: A robust and scalable post-training quantization framework for selective state space models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10411–10427. PMLR, 2025. URL https://proceedings.mlr.press/v267/chiang25a.html.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

CodeParrot. codeparrot-clean-valid. https://huggingface.co/datasets/codeparrot/codeparrot-clean-valid, 2022.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359, 2022.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337. PMLR, 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020. URL https://arxiv.org/abs/2101.00027.

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

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/5dfbe6f5671e82c76841ba687a8a9ecb-Abstract-Conference.html.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. SemEval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 829–839. Association for Computational Linguistics, 2019. doi: 10.18653/v1/S19-2145. URL https://aclanthology.org/S19-2145/.

Hyunwoo Kim, Byoungchan Ko, Minseok Kang, Minwoo Kim, Dongjin Lee, Jaehoon Lee, Sungroh Yoon, and Dahuin Jung. SSDi8: Accurate and efficient 8-bit quantization for state space duality. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/9d3852f2b1fc883fc41011cb0257f0d0-Abstract-Conference.html.

Yuta Koreeda and Christopher Manning. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 1907–1919. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-emnlp.164. URL https://aclanthology.org/2021.findings-emnlp.164/.

Aakash Sunil Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8abd2043b71a074278d5f687947bff9c-Abstract-Conference.html.

Michael Menezes and Anastasios Kyrillidis. GHOST: Unmasking phantom states in Mamba2 via grouped hidden-state output-aware selection & truncation. *arXiv preprint arXiv:2602.11408*, 2026. URL https://arxiv.org/abs/2602.11408.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

NVIDIA et al. Nemotron-H: A family of accurate and efficient hybrid Mamba-Transformer models. *arXiv preprint arXiv:2504.03624*, 2025. URL https://arxiv.org/abs/2504.03624.

OpenAI. GPT-5. https://openai.com/gpt-5/, 2025. Accessed: 2026-09-26.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylKikSYDH.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, et al. An empirical study of Mamba-based language models. *arXiv preprint arXiv:2406.07887*, 2024. URL https://arxiv.org/abs/2406.07887.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302, 2018. doi: 10.1162/tacl_a_00021. URL https://aclanthology.org/Q18-1021/.

Wikimedia Foundation. Wikimedia downloads. https://dumps.wikimedia.org, 2023. Chinese subset 20231101.zh from https://huggingface.co/datasets/wikimedia/wikipedia.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cFu7ze7xUm.

APPENDIX

CONTENTS

A Additional Background on Mamba **14**

B Statistical Analysis of Head-Wise ERL Structure **15**

B.1 Head-Wise ERL Imbalance 15

B.2 Quasi-Static Structure 15

C SERESA Head Partitioning Procedure **16**

D Coverage Saturation of Local Heads **17**

E Implementation and Latency Measurement Details **18**

F Calibration Cost and Partition Stability **19**

G Additional Results **20**

H Mamba-3 Head Structure and Quality **21**

I Long-Context Perplexity and Passkey Retrieval **22**

J Batch Sizes and Sequence Lengths **23**

K Additional Evaluation on NVIDIA A100 **24**

L LLM Usage Statement **25**

A Additional Background on Mamba

Mamba: Selective State Space Models. A continuous-time SSM relates the hidden state $h(t)$, input $x(t)$, and output $y(t)$ as follows (Gu et al., 2022):

$$\frac{dh(t)}{dt} = Ah(t) + Bx(t), \qquad y(t) = C^\top h(t).$$

Here, A governs state evolution, B writes the input into the state, and C reads the state into the output. Discretization with step size Δ gives the following recurrence, where $t \in \{1, \dots, L\}$ indexes sequence positions:

$$h_t = \bar{A}h_{t-1} + \bar{B}x_t, \qquad y_t = C^\top h_t.$$

Mamba extends this linear time-invariant SSM by making B_t, C_t, and Δ_t input-dependent:

$$B_t = \mathrm{Linear}_B(x_t), \qquad C_t = \mathrm{Linear}_C(x_t), \qquad \Delta_t = \mathrm{softplus}\left(\mathrm{Linear}_\Delta(x_t)\right).$$

The resulting selective recurrence is

$$h_t = \bar{A}_t h_{t-1} + \bar{B}_t x_t, \qquad y_t = C_t^\top h_t.$$

Here, Δ_t determines $\bar{A}_t = \exp(\Delta_t A)$, while Δ_t and B_t jointly determine the discretized input projection $\bar{B}_t$. State retention and input updates can therefore vary across positions and channels.

Assuming $h_0 = 0$, unrolling the recurrence gives

$$y_i = \sum_{j=1}^{i} C_i^\top \left(\prod_{k=j+1}^{i} \bar{A}_k \right) \bar{B}_j x_j.$$

Each term in this sum represents the recurrent contribution of the input at position j to the output at position i. The input projection $\bar{B}_j$ controls the input-to-state update, the decay product determines how much information is retained, and C_i determines how the propagated state contributes to the output. Mamba evaluates this input-dependent recurrence with an associative scan using work linear in sequence length, although scan-based execution is less amenable to GPU tensor cores optimized for matrix multiplication (Gu & Dao, 2024).

Mamba-2: State Space Duality. Mamba-2 groups channels into heads and constrains the state transition within each head to a scalar-times-identity structure. For head h at position t, the transition matrix is

$$\bar{\mathbf{A}}_{t,h} = \bar{A}_{t,h} I_N, \qquad \bar{A}_{t,h} = \exp\left(A_h \Delta_{t,h}\right).$$

Here, I_N denotes the $N \times N$ identity matrix, where N is the state dimension. Channels within a head share the scalar decay factor $\bar{A}_{t,h}$, while head-specific A_h and independent recurrent states yield distinct dynamics across heads. Consequently, the cumulative decay between two positions factorizes into a product of scalars, giving the sequence-level recurrence a lower-triangular semiseparable matrix structure.

State Space Duality (SSD) exploits this structure through chunk-wise decomposition. Explicitly constructing and multiplying the full matrix would incur quadratic cost in sequence length, whereas SSD evaluates the same full recurrence in linear complexity. matrix multiplication computes recurrent contributions within each chunk, while recurrent states summarize and propagate information across chunks. This preserves the full recurrence while enabling intra-chunk computation on GPU tensor cores (Dao & Gu, 2024).

Mamba-3: Exponential-Trapezoidal Discretization. Mamba-3 generalizes Mamba-2's exponential-Euler discretization to exponential-trapezoidal discretization while retaining an SSD-compatible parallel formulation. It thereby preserves parallel computation while enabling more expressive recurrent dynamics (Lahoti et al., 2026).

B Statistical Analysis of Head-Wise ERL Structure

This section reports head-wise ERL imbalance and cross-domain stability across model variants. We measure ERL concentration using the Gini coefficient and the share of total ERL held by the top $p\%$ of heads. We assess cross-domain stability by decomposing log-ERL variation into head, domain, and head–domain interaction effects and computing Spearman rank correlations between domain pairs.

B.1 Head-Wise ERL Imbalance

Across the four models, Gini coefficients range from 0.7926 to 0.8409, as shown in Table 6. The top 10% of heads account for 66.42–76.16% of total ERL, while the top 30% account for over 92%. Even in Nemotron-H-8B, which has the lowest concentration, these shares are 66.42% and 92.59%, respectively. Concentration decreases modestly with model size within the Mamba-2 family, yet the top 10% of Mamba2-8B heads still account for 69.29% of total ERL. Thus, long-range recurrent contributions remain concentrated in relatively few heads across model scales and variants.

Table 6: Head-wise ERL concentration across model variants. Top $p\%$ denotes the share of total ERL held by the heads with the highest ERL.

Model	Gini	Top 1%	Top 5%	Top 10%	Top 30%
Mamba2-1.3B	0.8409	15.25	55.26	76.16	95.38
Mamba2-2.7B	0.8340	14.87	53.27	74.60	93.57
Mamba2-8B	0.8100	14.87	49.87	69.29	93.27
Nemotron-H-8B	0.7926	10.27	42.36	66.42	92.59

B.2 Quasi-Static Structure

Table 7: Cross-domain stability of head-wise ERL. Head, Domain, and H×D give the shares of total log-ERL variation attributed to each effect. Spearman ρ gives the range of rank correlations across domain pairs.

Model	Head	Domain	H×D	Spearman ρ range
Mamba2-1.3B	98.68%	0.32%	1.00%	0.9775–0.9894
Mamba2-2.7B	98.41%	0.51%	1.08%	0.9773–0.9912
Mamba2-8B	97.79%	0.12%	2.09%	0.9684–0.9872
Nemotron-H-8B	98.91%	0.01%	1.08%	0.9853–0.9918

We perform a two-way ANOVA on log-ERL with head identity and domain as factors. Across WikiText-2, CodeParrot, GSM8K, and ZH-Wiki, the head effect explains 97.79–98.91% of the total variation, as shown in Table 7. In contrast, the domain effect explains at most 0.51%, while the head–domain interaction accounts for only 1.00–2.09%. Consistent with this result, Spearman rank correlations of head-wise ERL range from 0.9684 to 0.9918 across all domain pairs. These results indicate that head-wise ERL differences are dominated by head identity, with only minor variation induced by the input domain, and that the relative ordering of heads remains stable across English text, code, math problems, and Chinese text. Together, the two-way ANOVA and rank-correlation analysis support a quasi-static head-wise recurrence-length structure.

C SERESA HEAD PARTITIONING PROCEDURE

Algorithm 1 Head Partitioning via Effective Recurrence Distributions

Require: Head set $\mathcal{H}$, effective recurrence distributions $\{p_h(\ell)\}_{h\in\mathcal{H}}$, maximum recurrence range L, target recurrence length R
Ensure: Local-head set $\mathcal{H}_{\text{local}}$ and global-head set $\mathcal{H}_{\text{global}}$
for all $h \in \mathcal{H}$ **do**
  $\text{ERL}(h) \leftarrow \sum_{\ell=0}^{L-1} \ell\, p_h(\ell)$
end for
Compute the mean ERL:
$\overline{\text{ERL}} \leftarrow \frac{1}{|\mathcal{H}|} \sum_{h\in\mathcal{H}} \text{ERL}(h)$
Determine the number of local heads:
$K \leftarrow \left|\left\{h \in \mathcal{H} \mid \text{ERL}(h) < \overline{\text{ERL}}\right\}\right|$
for all $h \in \mathcal{H}$ **do**
  $\textit{Coverage}_h(R) \leftarrow \sum_{\ell=0}^{R-1} p_h(\ell)$
end for
Select the K heads with the largest coverage:
$\mathcal{H}_{\text{local}} \leftarrow \text{TopK}_{h\in\mathcal{H}}\left(\textit{Coverage}_h(R), K\right)$
Assign the remaining heads as global heads:
$\mathcal{H}_{\text{global}} \leftarrow \mathcal{H} \setminus \mathcal{H}_{\text{local}}$
Apply recurrence length R to $\mathcal{H}_{\text{local}}$
Retain full SSD computation for $\mathcal{H}_{\text{global}}$
return $\mathcal{H}_{\text{local}}, \mathcal{H}_{\text{global}}$

Algorithm 1 summarizes the head partitioning procedure of SERESA. Given the effective recurrence distribution $p_h(\ell)$ of each head, SERESA first computes the head-wise ERL and the mean ERL across all heads. The local-head budget K is set to the number of heads whose ERL is below the mean. This step uses the overall shape of the ERL distribution to determine the number of heads to which recurrence restriction is applied.

SERESA then computes $\textit{Coverage}_h(R)$, the fraction of effective recurrent contribution preserved within the target recurrence length R. The K heads with the highest $\textit{Coverage}_h$ are selected as $\mathcal{H}_{\text{local}}$. Recurrence length R is applied to the selected heads, while the remaining heads, denoted by $\mathcal{H}_{\text{global}}$, retain the original Full SSD computation.

D COVERAGE SATURATION OF LOCAL HEADS

We examine how rapidly effective recurrent contributions accumulate with recurrence length in local heads. For Mamba2-2.7B, we fix the local-head set at 81.25% of all SSD heads and measure $\textit{Coverage}_h(R)$ as R varies from 4 to 2,048.

Table 8: $\textit{Coverage}_h$ statistics of the fixed local-head set in Mamba2-2.7B across recurrence cutoffs R. The partition is calibrated on Pile sequences of length 2,048 and contains 81.25% of all SSD heads. P10 is the 10th percentile across local heads. The last two columns give the fractions of heads with $\textit{Coverage}_h$ of at least 95% and 99%.

R	Mean Coverage	Median	P10	Frac. $\geq 95\%$	Frac. $\geq 99\%$
4	84.62%	90.85%	59.21%	30.3%	6.8%
8	89.38%	93.75%	72.54%	42.0%	11.7%
16	93.67%	96.53%	84.19%	60.1%	25.2%
32	96.68%	98.63%	90.32%	76.1%	45.2%
64	98.38%	99.78%	94.52%	88.1%	63.1%
128	99.19%	99.99%	96.88%	95.4%	77.7%
256	99.58%	100.00%	98.39%	99.2%	85.7%
512	99.81%	100.00%	99.42%	100.0%	93.2%
1024	99.95%	100.00%	99.90%	100.0%	98.4%
2048	100.00%	100.00%	100.00%	100.0%	100.0%

$\textit{Coverage}_h$ varies substantially across local heads at small cutoffs, as shown in Table 8. At $R = 4$, mean $\textit{Coverage}_h$ is 84.62%, but P10 is only 59.21%. At $R = 16$, these values rise to 93.67% and 84.19%, and 60.1% of local heads retain at least 95% of their contributions.

At $R = 64$, mean $\textit{Coverage}_h$ reaches 98.38%, while P10 reaches 94.52%. When R equals the Mamba2-2.7B chunk size of $Q = 256$, mean $\textit{Coverage}_h$ and P10 rise to 99.58% and 98.39%, and 99.2% of local heads retain at least 95% of their contributions. All local heads reach 95% $\textit{Coverage}_h$ at $R = 512$, while 98.4% reach 99% $\textit{Coverage}_h$ at $R = 1024$. Thus, most local contributions accumulate at short lags, while larger cutoffs progressively capture heads with longer tails.

E Implementation and Latency Measurement Details

All latency experiments use an NVIDIA A5000 with CUDA 12.1, PyTorch 2.4.0, and Triton 3.0.0. The three Mamba-2 models use FP16 and chunk size 256; Nemotron-H-8B uses its native BF16 precision and chunk size 128. For each model, baseline and SERESA use identical precision, tensor layout, chunk size, and compilation settings.

We evaluate on three downstream tasks. WikiHop is a QAngaroo multi-hop question answering task over multiple Wikipedia passages, with a variable number of answer candidates per question (2–30). Hyperpartisan is the SemEval 2019 hyperpartisan news detection task, formulated as binary classification. ContractNLI is a three-way NLI task from SCROLLS over contract clauses, with entailment, contradiction, and not-mentioned as the three classes. All three tasks are evaluated using log-likelihood. Due to the long-context limitations observed in Mamba-2, downstream evaluations are conducted only on samples with input lengths of at most 2K tokens. We filter for samples that naturally fit within this limit, without truncation.

After kernel compilation, we perform 15 warm-up iterations and report mean latency over 50 runs. Each run is timed with CUDA events and explicit device synchronization. SSD core latency includes full computation for global heads, restricted intra-chunk computation for local heads, and boundary correction. Head permutation is applied to the relevant weights once at model loading and incurs no additional operation during the forward pass. Input and output projections, causal convolution, and other model components are excluded. We compute SSD core speedups against each model's native SSD kernel `mamba_chunk_scan_combined` under identical settings.

For DuoAttention, we train per-head gates α_h for 300 steps (warm-started at 150) using $\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + 0.02\|\boldsymbol{\alpha}\|_1$ and Adam with cosine LR $0.08 \to 0.005$, then threshold α_h to match the 81.25% local-head budget. Local heads use a recurrence length of $R = 64$.

F CALIBRATION COST AND PARTITION STABILITY

SERESA determines a local/global head partition once from head-wise ERL and *Coverage*$_h$ measured on calibration data, then fixes it throughout deployment. We evaluate partition stability across calibration sizes and the one-time wall-clock calibration cost.

Each calibration sample is one sequence of length $2{,}048$ at batch size 1. The default $n = 5$ therefore uses $10{,}240$ tokens. Let S_n denote the local-head set selected from n calibration sequences. We measure partition stability against the reference set S_{20} using Jaccard similarity:

$$J(S_n, S_{20}) = \frac{|S_n \cap S_{20}|}{|S_n \cup S_{20}|}.$$

Table 9: Head-partition stability and calibration cost across calibration sizes. n is the number of length-$2{,}048$ sequences at batch size 1. Jaccard similarity uses the $n = 20$ partition as reference. Random gives the chance-level approximation $\frac{p}{2-p}$ for local-head ratio p. Time is the one-time offline calibration time for $n = 5$ on a single NVIDIA A5000.

Model	Local-head ratio	$n = 1$	$n = 5$	Random	Time at $n = 5$
Mamba2-1.3B	83.19%	0.9813	0.9934	0.712	46 s
Mamba2-2.7B	81.25%	0.9695	0.9892	0.684	62 s
Mamba2-8B	79.49%	0.9597	0.9866	0.660	87 s
Nemotron-H-8B	78.38%	0.9798	0.9930	0.644	48 s
Range	78.38–83.19%	**0.960–0.981**	**0.987–0.993**	0.644–0.712	46–87 s

As shown in Table 9, a single sequence yields Jaccard similarity of 0.9597–0.9813 with the reference partition. Five sequences raise it to 0.9866–0.9934, well above the random baseline of 0.644–0.712. Thus, the high overlap cannot be attributed solely to the large local-head ratio.

The default calibration takes 46–87 seconds per model on one NVIDIA A5000 and is performed only once before deployment. Inference requires neither ERL computation nor further head selection. ERL and *Coverage*$_h$ accumulate recurrent-contribution statistics sequentially across heads, without materializing all head-wise mixing matrices. Peak auxiliary memory is therefore $O(L^2)$ rather than $O(HL^2)$. The stable partition obtained from few samples supports a quasi-static head-wise recurrence structure rather than a sample-sensitive, task-specific search.

G Additional Results

Table 10: Perplexity and SSD core latency with an additional GHOST 25% pruning configuration. Bold indicates the best result among non-baseline methods in each column.

Model	Method	Config.	Pile	WikiText-2	C4	SSD (ms)	Speedup
Mamba2-1.3B	Baseline	–	6.99	10.42	14.78	7.47	1.00×
	SSDi8	8-bit	7.04	10.55	14.89	5.86	1.27×
	GHOST	N (25%)	7.16	11.12	14.95	5.88	1.27×
	GHOST	N (50%)	7.25	11.77	15.71	4.81	1.55×
	SERESA (ours)	83.19%	**7.01**	**10.43**	**14.80**	**3.70**	**2.02**×
Mamba2-2.7B	Baseline	–	6.28	9.06	13.26	9.37	1.00×
	SSDi8	8-bit	6.30	9.13	13.32	7.17	1.31×
	GHOST	N (25%)	6.35	9.65	13.30	7.35	1.27×
	GHOST	N (50%)	6.61	10.11	13.74	6.25	1.50×
	SERESA (ours)	81.25%	**6.29**	**9.07**	**13.27**	**4.76**	**1.97**×
Mamba2-8B	Baseline	–	5.99	7.25	9.96	15.76	1.00×
	SSDi8	8-bit	6.06	7.41	10.08	12.33	1.28×
	GHOST	N (25%)	9.97	14.61	18.45	12.35	1.28×
	GHOST	N (50%)	11.87	17.44	21.95	9.50	1.66×
	SERESA (ours)	79.49%	**6.01**	**7.26**	**9.98**	**8.85**	**1.78**×
Nemotron-H-8B	Baseline	–	7.59	8.43	13.57	15.34	1.00×
	SSDi8	8-bit	7.83	8.72	14.09	12.17	1.26×
	GHOST	N (25%)	8.08	8.67	14.07	12.34	1.24×
	GHOST	N (50%)	8.56	9.30	14.98	9.57	1.60×
	SERESA (ours)	78.38%	**7.59**	**8.36**	**13.54**	**8.77**	**1.75**×

Table 10 extends the comparison in Table 1 with a less aggressive GHOST configuration that prunes 25% of the state dimensions. All SSD core latency measurements use batch size 16 and sequence length 2,048; speedups are relative to the full SSD baseline. Lower perplexity and latency are better.

Reducing GHOST's pruning ratio from 50% to 25% improves perplexity across all three datasets and four models, but reduces its SSD core speedup from 1.50–$1.66\times$ to 1.24–$1.28\times$. Despite this milder pruning, the quality degradation remains substantial for Mamba2-8B, with perplexities of 9.97, 14.61, and 18.45 on Pile, WikiText-2, and C4, compared with 5.99, 7.25, and 9.96 for baseline. In contrast, SERESA remains close to baseline across all four models while achieving substantially higher SSD core speedups of 1.75–$2.02\times$.

H MAMBA-3 HEAD STRUCTURE AND QUALITY

We analyze the distribution and cross-domain ordering of head-wise effective recurrence length in Mamba-3. Using the recurrent-contribution-based ERL definition from the main text, we evaluate SISO-1.5B and MIMO-1.5B (Lahoti et al., 2026) on Pile, CodeParrot, GSM8K, and ZH-Wiki.

Table 11: Head-wise ERL concentration and cross-domain ordering stability in Mamba-3. Gini values are ranges across the four domains, rounded to two decimal places. Spearman ρ gives the range of head-wise rank correlations across domain pairs.

Model	Gini	Spearman ρ
SISO-1.5B	0.75–0.76	0.9734–0.9928
MIMO-1.5B	0.75–0.77	0.9801–0.9923

Both variants exhibit Gini coefficients of 0.75–0.77, indicating substantial head-wise ERL imbalance, as shown in Table 11. Spearman correlations remain high across all domain pairs, ranging from 0.9734 to 0.9928 for SISO-1.5B and from 0.9801 to 0.9923 for MIMO-1.5B. The ordering of high-ERL heads thus remains stable across natural language, code, math problems, and Chinese text. These results extend the quasi-static head-wise recurrence structure observed in Mamba-2 to both Mamba-3 variants.

Quality under a Fixed Recurrence Length. We apply the ERL- and $\mathit{Coverage}_h$-based partition policy from the main text and restrict local-head recurrence to $R = 64$.

Table 12: Quality under length-64 recurrence restriction in Mamba-3. The local-head ratio is averaged over five calibration draws, and the local-head count is the corresponding rounded average. Relative ΔPPL is the percentage increase over each model's baseline.

Model	Local-head ratio	Local heads	Relative ΔPPL in %		
			Pile	WikiText-2	C4
SISO-1.5B	72.54%	$1114/1536$	$+1.123$	$+1.073$	$+0.368$
MIMO-1.5B	73.89%	$1135/1536$	$+1.098$	$+0.996$	$+0.388$

Both models restrict recurrence in approximately 73% of heads while keeping the relative PPL increase below 1.2% on Pile, WikiText-2, and C4, as shown in Table 12. The increase is below 0.4% on C4 and approximately 1% on Pile and WikiText-2. These results are consistent with local-head contributions being concentrated within $R = 64$ and show that the head-wise recurrence policy preserves quality across both Mamba-3 variants.

I LONG-CONTEXT PERPLEXITY AND PASSKEY RETRIEVAL

We further evaluate SERESA under long-context settings on Nemotron-H-8B, using PG-19 perplexity (Rae et al., 2020) and passkey retrieval. We use Nemotron-H-8B to avoid confounding the evaluation with the long-context limitations observed in Mamba-2. Using 2K-token calibration sequences, the ERL-derived budget sets the local-head ratio to 78.38%, and $\mathit{Coverage}_h(128)$ selects the local heads. We keep this partition and the recurrence length $R = 128$ fixed across sequence lengths from 2K to 32K.

For passkey retrieval, we use the RULER NIAH multi-key task (Hsieh et al., 2024), which combines multiple key–value pairs with decoy entries. Following the official RULER protocol, we report target-value retrieval accuracy. Sequence length is defined by the GPT-NeoX-tokenized haystack length (Black et al., 2022). For both the baseline and SERESA, autoregressive answer generation is performed without KV/state caching: at each generation step, we recompute a full prefill over the entire sequence accumulated up to that step, rather than using incremental decoding from cached states.

Table 13: Long-context perplexity and passkey retrieval results.

	PG19 Perplexity		Passkey Retrieval	
Seq. Len.	Baseline	SERESA	Baseline	SERESA
2K	11.10	11.12	98.7%	99.3%
4K	11.74	11.73	98.0%	98.0%
8K	11.85	11.89	94.0%	92.7%
16K	10.73	10.79	88.0%	89.3%
32K	10.40	10.43	64.7%	64.0%

Table 13 shows that SERESA closely preserves the language-modeling quality of the baseline across sequence lengths from 2K to 32K. The relative PG19 perplexity difference remains within 0.6% at every evaluated length, including 32K, despite the recurrence policy being determined at a much shorter calibration length. Passkey retrieval exhibits a similarly small gap between the two models: the absolute difference remains within 1.3 percentage points across all sequence lengths. At 32K, retrieval accuracy decreases for both the baseline and SERESA to 64.7% and 64.0%, respectively, suggesting that the degradation at the longest context is primarily associated with sequence length rather than the recurrence restriction introduced by SERESA. Overall, these results show that SERESA preserves both long-context language modeling and information retrieval behavior while restricting recurrence in the majority of SSD heads.

J BATCH SIZES AND SEQUENCE LENGTHS

Table 14: SSD core latency of Mamba2-2.7B across batch sizes at sequence length 2K. SERESA uses a local-head ratio of 81.25% and recurrence length $R = 64$. All latency values are in milliseconds.

Batch Size	Baseline	SERESA	Speedup
1	0.50	0.29	$1.724\times$
2	0.99	0.53	$1.868\times$
4	2.16	1.08	$2.000\times$
8	4.48	2.21	$2.027\times$
16	9.37	4.76	$1.968\times$
32	17.99	9.09	$1.979\times$
64	36.04	18.09	$1.992\times$

We evaluate the SSD core latency of Mamba2-2.7B across batch sizes at a fixed sequence length of 2K in Table 14. SERESA achieves $1.724\times$ and $1.868\times$ speedups at batch sizes 1 and 2, respectively. The smaller gains at low batch sizes are consistent with fixed kernel-launch and scheduling overhead, which accounts for a larger fraction of latency when the per-kernel workload is small. From batch size 4 onward, the speedup remains within 1.968–$2.027\times$, peaking at batch size 8 and reaching $1.992\times$ at batch size 64. These results show that SERESA provides consistent SSD core acceleration across a wide range of batch sizes.

Table 15: SSD core latency of Mamba2-2.7B across sequence lengths at batch size 16. SERESA uses a local-head ratio of 81.25% and recurrence length $R = 64$. All latency values are in milliseconds.

Sequence Length	Baseline	SERESA	Speedup
2K	9.37	4.76	$1.968\times$
4K	19.54	10.05	$1.944\times$
8K	40.08	20.56	$1.949\times$
12K	60.36	30.95	$1.950\times$
16K	80.05	41.45	$1.931\times$

We evaluate the SSD core latency of Mamba2-2.7B across sequence lengths at a fixed batch size of 16 in Table 15. SERESA achieves a $1.968\times$ speedup at 2K tokens. From 4K to 16K tokens, the speedup remains within 1.931–$1.950\times$, while the latency of both the baseline and SERESA grows approximately linearly with sequence length. This indicates that the computational benefit of head-wise recurrence restriction remains stable as context length scales.

K Additional Evaluation on NVIDIA A100

We additionally evaluate Mamba2-2.7B on an NVIDIA A100 GPU to examine whether the SSD acceleration observed on the NVIDIA A5000 transfers to a different accelerator. We use the same batch size of 16 and sequence length of 2,048 as in the main latency evaluation. Table 16 reports perplexity and SSD core latency, with speedup normalized to the native SSD kernel on the A100.

Table 16: Perplexity and SSD core latency of Mamba2-2.7B on an NVIDIA A100 (batch size 16, sequence length 2,048). Speedup is measured relative to the native SSD kernel on the same GPU.

Method	Config.	Pile	WikiText-2	C4	SSD (ms)	Speedup
Baseline	–	6.2745	9.0608	13.2607	4.32	1.000×
SSDi8	8-bit	6.3004	9.1304	13.3167	4.84	0.893×
GHOST	N (50%)	6.6147	10.1140	13.7354	3.54	1.220×
SERESA (ours)	81.25%	**6.2915**	**9.0705**	**13.2668**	**2.73**	**1.582**×

SERESA achieves a $1.582\times$ SSD speedup over the native kernel on A100 while maintaining perplexity close to the baseline. The relative perplexity increases are only 0.27%, 0.11%, and 0.05% on Pile, WikiText-2, and C4, respectively. In comparison, GHOST achieves a $1.220\times$ speedup but incurs substantially larger perplexity increases, while SSDi8 is slower than the native kernel ($0.893\times$) in this setting.

The relative speedups on A100 are lower than those observed on the NVIDIA A5000 in the main experiments. On Mamba2-2.7B, the A5000 results show speedups of $1.31\times$, $1.50\times$, and $1.97\times$ for SSDi8, GHOST, and SERESA, respectively, whereas the corresponding A100 results are $0.893\times$, $1.220\times$, and $1.582\times$. Thus, the benefit of each optimization depends on the underlying hardware and native SSD implementation. In particular, the INT8 execution used by SSDi8 provides a measurable gain on the A5000 but does not outperform the native kernel on the A100 under this configuration.

Despite these hardware-dependent differences, SERESA remains effective across both GPUs. It provides the largest SSD acceleration among the evaluated methods on both the A5000 and A100 while preserving near-baseline perplexity. This result indicates that eliminating unnecessary recurrence computation provides acceleration beyond a single GPU configuration, although the realized speedup depends on the efficiency of the hardware-specific baseline.

L LLM USAGE STATEMENT

We used OpenAI's ChatGPT (GPT-5; https://chatgpt.com/) (OpenAI, 2025) and Anthropic's Claude (https://claude.ai/) (Anthropic, 2026) to polish the writing of the manuscript, improving its clarity and readability. All LLM-assisted revisions were based on author-provided research content and were reviewed and verified by the authors. The research ideas, experimental results, and final scientific claims were determined by the authors, who take full responsibility for the content of this paper.