REBOUND : READOUT-AWARE BOUNDARY PRUNING FOR EFFICIENT MAMBA PREFILL

Anonymous authors
Paper under double-blind review

ABSTRACT

Mamba language models make autoregressive decoding efficient, but prefill still processes every prompt token through every layer. We propose **ReBound** , a training-free readout-aware boundary pruning method that runs the selected Mamba layer at full length and then removes low-impact boundary representations before downstream layers. Its score combines channel-mean Δ, selective-update energy, and proximity to the final autoregressive readout, while a calibration-only KL sweep selects the earliest behaviorally safe pruning layer. At 10% FLOP reduction on Mamba-1 2.8B, ReBound reaches 68.0% average lm-evaluation-harness accuracy, only 0.62 points below baseline, and raises WikiText-103 continuation perplexity by only 0.61%. A complete seven-way score ablation shows that the full $\Delta + E + P$ score is best, and calibration generalizes across architectures, with sharp safe-layer transitions at L44 for Mamba-1 2.8B and L52 for Mamba-2 2.7B. On five LongBench tasks, ReBound preserves long-context quality substantially better than Δ-only DTP on both architectures. Measured hardware performance realizes these savings: at L44 with 50% token keep, latency improves by $1.15\times$ and throughput by $1.157\times$, close to the $1.174\times$ FLOP-based ceiling. These results show that Mamba token compression should preserve the final readout rather than rely on local token importance alone.

1 INTRODUCTION

State Space Models (SSMs), and selective variants such as Mamba, have emerged as compelling alternatives to Transformer architectures for long-sequence modeling (Vaswani et al., 2017; Gu & Dao, 2024; Dao & Gu, 2024). By replacing quadratic self-attention with a recurrent state-space update, Mamba achieves linear-time sequence processing while maintaining competitive language modeling quality. This makes Mamba especially attractive for long-context inference, where Transformer decoding must repeatedly attend over an expanding key-value cache. In Mamba, decoding is substantially cheaper because generation proceeds by updating a fixed-size recurrent state rather than revisiting the full prompt.

However, Mamba's efficiency advantage is asymmetric across inference phases. During decoding, the recurrent state has fixed size, so the per-token cost does not grow with the prompt length. During prefill, by contrast, the model must still process every input token through every layer in order to construct the hidden trajectory that initializes generation. As prompts grow to thousands of tokens in document question answering, retrieval-augmented generation, long-form summarization, and code completion, this linear prefill cost becomes a major component of end-to-end latency. Token compression during prefill is therefore a direct and practical way to improve Mamba inference: if redundant tokens can be removed after some layer, all downstream layers can operate on a shorter sequence. Token reduction is well studied for Transformers, including pruning, merging, and learned token selection methods (Rao et al., 2021; Liang et al., 2022; Bolya et al., 2023; Cao et al., 2023). These methods are natural inspiration, but they cannot be transferred directly to Mamba. Transformer token reduction typically relies on attention scores or pairwise token similarity; Mamba has no explicit attention matrix, and its tokens interact through an ordered recurrent scan. Removing a token in a Transformer deletes one participant from a parallel attention computation, while removing a token in Mamba changes the state-transition path through which information flows to all subsequent positions shown in Figure 1. Prior work on token reduction for SSMs has empirically confirmed this mismatch: directly applying Transformer-style pruning or merging to Mamba causes severe

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

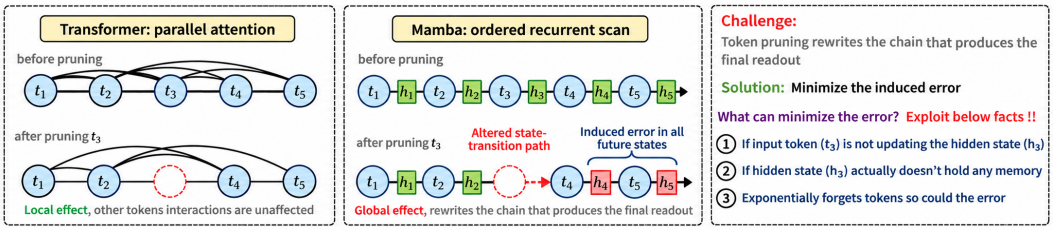


Figure 1: Pruning is local in Transformers but globally rewrites the recurrent chain in Mamba; ReBound minimizes the induced final-readout error using three selective-SSM properties: Δ - strength, selective-update energy, and exponential forgetting toward the readout.

performance degradation (Zhan et al., 2024; Ma et al., 2025). This motivates compression methods that respect Mamba's state-space structure rather than treating its hidden sequence as an unordered set of token embeddings.

Recent work has begun to develop Mamba-specific token reduction, especially in vision and multimodal settings. Methods such as DyVM, Famba-V, MTR, QuarterMap, and DTP exploit spatial redundancy, bidirectional scanning, class-token aggregation, or visual-token privilege to remove or merge image patches with limited task degradation (Wu et al., 2025; Shen et al., 2024; Ma et al., 2025; Chi et al., 2025; Park et al., 2026). These works show that Mamba compression should use architecture-aware signals such as the input-dependent timescale parameter Δ_t. Similarily In vision, many outputs are spatially redundant, and downstream prediction often depends on a class token, pooled representation, or task-specific visual head. In multimodal models, pruning can often be restricted to visual tokens while preserving the text stream. However, these methodologies do not fully transfer to causal language-model prefill: there is no separate visual stream, no spatial redundancy structure, and no class or pooled token that summarizes the input. Every prompt token belongs to a single causal sequence, and pruning any token may perturb the autoregressive state used for generation.

This paper argues that Mamba language-model pruning should therefore be framed around a different objective: preserving the final autoregressive readout. In prefill, the quantity that directly determines the next-token distribution is the final hidden representation at the last prompt position. Thus, a token should not be judged only by local importance signals such as Δ; it should be pruned only when its removal minimizes the error in the final readout token. This readout-preservation view is the central distinction of our work.

We highlight three main contributions:

- **Readout-aware formulation.** We formulate Mamba prefill pruning as final-readout preservation and derive a score combining Δ, selective-update energy, and readout proximity.
- **Training-free boundary pruning and calibration.** ReBound runs the selected layer at full length, compacts only its inter-layer output, and selects the earliest safe layer using held-out final-readout KL rather than evaluation-task tuning.
- **Broad empirical validation.** Across Mamba-1 and Mamba-2, score ablations, lm-evaluation-harness, WikiText-103, LongBench, and H100 measurements show that the proposed score preserves quality while translating FLOP savings into real latency and throughput gains.

At 10% FLOP reduction, ReBound loses only 0.89 and 0.62 average accuracy points on Mamba-1 1.4B and 2.8B, respectively; on Mamba-1 2.8B WikiText-103 continuation, perplexity rises only 0.61% versus 7.50% for DTP and 13.51% for UTRC.

2 BACKGROUND

2.1 SELECTIVE STATE-SPACE MODELS

State-space model maps an input sequence $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^D$ to outputs $\mathbf{y}_1, \ldots, \mathbf{y}_T$ through a hidden state $\mathbf{h}_t \in \mathbb{R}^N$ governed by the continuous-time linear system

$$\dot{\mathbf{h}}(t) = \mathbf{A}\,\mathbf{h}(t) + \mathbf{B}\,\mathbf{x}(t), \qquad \mathbf{y}(t) = \mathbf{C}\,\mathbf{h}(t). \tag{1}$$

Earlier structured SSMs such as S4 (Gu et al., 2022) treat Δ, $\mathbf{B}$, and $\mathbf{C}$ as time-invariant parameters, yielding a linear time-invariant (LTI) recurrence. Mamba (Gu & Dao, 2024) departs from this by making Δ_t, $\mathbf{B}_t$, and $\mathbf{C}_t$ *input-dependent* through learned linear projections of $\mathbf{x}_t$:

$$\Delta_t = \mathrm{softplus}\big(\mathbf{W}_\Delta\,\mathbf{x}_t + \mathbf{b}_\Delta\big), \qquad \mathbf{B}_t = \mathbf{W}_B\,\mathbf{x}_t, \qquad \mathbf{C}_t = \mathbf{W}_C\,\mathbf{x}_t, \tag{2}$$

where $\mathbf{W}_\Delta, \mathbf{W}_B, \mathbf{W}_C$ and $\mathbf{b}_\Delta$ are learned parameters. The state matrix $\mathbf{A}$ remains time-invariant and is parameterized as $\mathbf{A} = -\exp(\mathbf{A}_{\log})$ with diagonal structure, which guarantees $\mathrm{Re}(\mathrm{spec}(\mathbf{A})) < 0$—i.e., the recurrence is provably stable.

Discretizing (1) using zero-order hold with the per-token Δ_t from (2) produces the discrete selective recurrence

$$\mathbf{h}_t = \bar{\mathbf{A}}_t\,\mathbf{h}_{t-1} + \bar{\mathbf{B}}_t\,\mathbf{x}_t, \qquad \mathbf{y}_t = \mathbf{C}_t\,\mathbf{h}_t, \tag{3}$$

where

$$\bar{\mathbf{A}}_t = \exp(\Delta_t\mathbf{A}), \qquad \bar{\mathbf{B}}_t = (\Delta_t\mathbf{A})^{-1}\big(\exp(\Delta_t\mathbf{A}) - \mathbf{I}\big)\,\Delta_t\mathbf{B}_t. \tag{4}$$

Because Δ_t, $\mathbf{B}_t$, and $\mathbf{C}_t$ vary with the input, the recurrence is no longer LTI: every token sees its own $(\bar{\mathbf{A}}_t, \bar{\mathbf{B}}_t, \mathbf{C}_t)$, and the model can decide on a per-position basis how much of the current input to integrate. This input-dependence is the defining property of selective SSMs and the foundation of the analysis that follows—in particular, Δ_t is computable directly from $\mathbf{x}_t$ at inference time, with no labels or auxiliary computation required.

The role of Δ_t. The input-dependent timescale Δ_t acts as a continuous gate over the recurrence. Two limits make this concrete:

- $\Delta_t \to 0 \;\Rightarrow\; \bar{\mathbf{A}}_t \to \mathbf{I}$ and $\bar{\mathbf{B}}_t \to \mathbf{0}$: the state is preserved unchanged ($\mathbf{h}_t \approx \mathbf{h}_{t-1}$) and the input $\mathbf{x}_t$ is ignored.
- Δ_t large $\;\Rightarrow\; \bar{\mathbf{A}}_t \to \mathbf{0}$ and $\bar{\mathbf{B}}_t$ grows: the prior state is forgotten and $\mathbf{x}_t$ overwrites the state.

This generalizes the gating mechanism of classical RNNs (Gu & Dao, 2024, Theorem 1): the model decides, per token and per channel, how much of the current input to integrate versus how much of the past state to retain. Throughout this work, Δ_t is the central quantity: it gates token contributions to the state, and our perturbation analysis (Section 3) shows it is also the natural signal for identifying tokens whose removal causes minimal damage to the final output.

Notation. We use T for prompt length, L for the number of layers, and $\ell \in \{1, \ldots, L\}$ for the layer index. Superscripts $\mathbf{x}_t^\ell$, $\mathbf{h}_t^\ell$, $\mathbf{y}_t^\ell$ denote the layer-ℓ position-t input, hidden state, and output, respectively. Bold lowercase symbols denote vectors, bold uppercase denote matrices; scalars (such as Δ_t when treated as a scalar gate, or the dimensions T, D, N, L) are non-bold. We write $\bar{\Delta}_t = \frac{1}{D}\sum_d \Delta_{t,d}$ for the channel-mean of the timescale at position t, and $\mu^\ell = \min_d |\mathrm{Re}(\mathbf{A}_d^\ell)|$ for the per-layer stability constant. When the layer is clear from context we drop the superscript.

3 METHODOLOGY

We propose a training-free token pruning method for Mamba language models during prefill. The method is organized around two questions. First, *where should pruning be applied?* Pruning too early can remove information before it has propagated through the recurrent state, while pruning too late is safe but yields little compute reduction. We therefore select a single pruning layer using a calibration-only readout preservation criterion. Second, *how should tokens be selected for pruning once the layer is fixed?* At the selected layer, we run the Mamba block at full sequence length, score the resulting boundary representations, and physically compact the sequence before downstream layers. Figure 2 summarizes the two parts: the upper panel selects the pruning layer, and the lower panel scores and removes low-impact tokens at that layer. The final prompt token is always preserved.

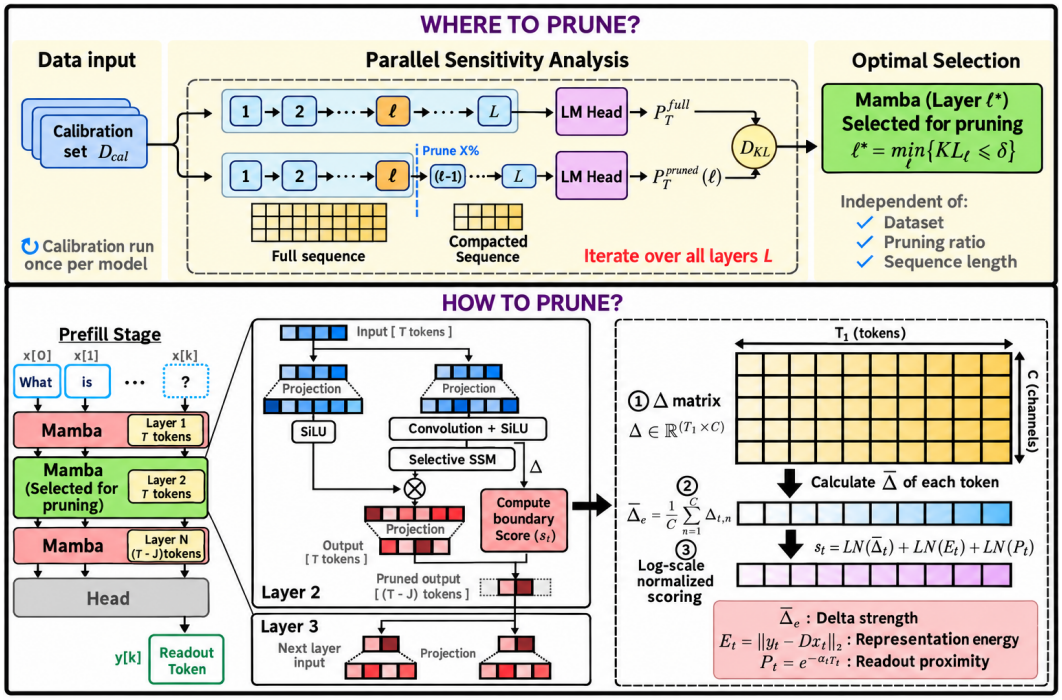


Figure 2: **Overview of ReBound pruning for Mamba prefill.**

3.1 Where to prune: calibration-based layer selection

The pruning layer determines the accuracy–efficiency trade-off. Let p_T^{full} denote the next-token distribution of the unpruned model at the final prompt position, and let $p_T^{\text{pruned}}(\ell)$ denote the distribution obtained when boundary pruning is applied after candidate layer ℓ:

$$p_T^{\text{full}} = \text{softmax}\Big(\text{LMHead}(y_T^{L,\text{full}})\Big), \qquad p_T^{\text{pruned}}(\ell) = \text{softmax}\Big(\text{LMHead}(y_T^{L,\text{pruned}(\ell)})\Big).$$

On a held-out calibration set $\mathcal{D}_{\text{cal}}$, we measure the behavioral effect of pruning layer ℓ by the final-readout KL divergence:

$$\text{KL}_\ell = \mathbb{E}_{x\sim\mathcal{D}_{\text{cal}}}\left[D_{\text{KL}}\left(p_T^{\text{full}} \,\|\, p_T^{\text{pruned}}(\ell)\right)\right], \qquad \ell^\star = \min_\ell\{\text{KL}_\ell \leq \delta\}.$$

The selected layer $\ell^\star$ is fixed after calibration and reused for inference. Calibration is performed once per model and does not use downstream evaluation labels or task accuracy. Empirically, the safe region is stable across calibration datasets, pruning ratios, and sequence lengths but is model-specific: the procedure identifies L44 for Mamba-1 2.8B and an analogous sharp transition at L52 for Mamba-2 2.7B.

3.2 Inter-layer pruning after a full scan

Consider a Mamba language model with L layers and a prompt of length T. Let $X^\ell = [x_1^\ell, \ldots, x_T^\ell] \in \mathbb{R}^{T\times d}$ be the input to layer ℓ, and let $Y^\ell = F_\ell(X^\ell) = [y_1^\ell, \ldots, y_T^\ell]$ be the corresponding output sequence. Standard prefill passes the full sequence through every layer:

$$X^{\ell+1} = Y^\ell, \qquad \ell = 1, \ldots, L-1.$$

ReBound applies pruning once, at the selected boundary after $\ell^\star$. We first run the selected layer at full length,

$$Y^{\ell^\star} = F_{\ell^\star}(X^{\ell^\star}),$$

with no token removal inside the layer. We then compute a boundary sensitivity score, remove a fixed fraction of low-score non-final tokens, and pass the compacted sequence to the next layer:

$$X^{\ell^\star+1} = \text{Gather}(Y^{\ell^\star}, S^{\ell^\star}), \qquad S^{\ell^\star} \subseteq \{1, \ldots, T\}.$$

The final prompt position is always preserved,

$$T \in S^{\ell^\star},$$

because the final prompt representation determines the next-token distribution and initializes autoregressive decoding. The survivor order is unchanged:

$$i < j,\ i, j \in S^{\ell^\star} \quad \Rightarrow \quad y_i^{\ell^\star} \text{ appears before } y_j^{\ell^\star}.$$

Thus, pruning physically shortens the sequence while preserving the original causal order of the retained tokens. This full-scan-then-prune design keeps the trajectory of layer $\ell^\star$ exact, while reducing the sequence length for all downstream layers.

3.3 How to prune: readout-aware boundary score

Because pruning is applied only after a full scan, the approximation occurs at the inter-layer boundary: some full-scan token representations are not propagated to downstream layers. We therefore score each non-final token using three quantities available after the selected layer: Δ-strength, selective-update energy, and readout proximity. We have provided a mathematical motivation behind the ReBound scoring method in Appendix B.

Δ-strength. Mamba's selective recurrence is controlled by the input-dependent timescale Δ_t^ℓ. Larger values correspond to stronger state updates, while smaller values indicate weaker integration into the recurrent state. Since Δ_t^ℓ is channel-wise, we use its channel mean:

$$\bar{\Delta}_t^\ell = \frac{1}{d_{\text{inner}}} \sum_{c=1}^{d_{\text{inner}}} \Delta_{t,c}^\ell.$$

We refer to $\bar{\Delta}_t^\ell$ as the Δ-strength of token t.

Selective-update energy. A token with small Δ-strength can still be important if the SSM branch writes a large non-residual update. We therefore do not use the full output energy $\|y_t^\ell\|_2$. Instead, we subtract the Mamba skip/residual term $D^\ell x_t^\ell$ and measure the energy of the selective update:

$$u_t^\ell = y_t^\ell - D^\ell x_t^\ell, \qquad E_t^\ell = \|u_t^\ell\|_2.$$

Here $D^\ell x_t^\ell$ denotes the residual/skip contribution in the Mamba SSM output. The term E_t^ℓ measures the magnitude of the information added by the selective SSM branch that would be withheld from downstream layers if token t were removed.

Readout proximity. For autoregressive language modeling, the key output of prefill is the final prompt representation. Perturbations closer to the final token are more likely to affect the next-token distribution. We encode this using a cumulative Δ-weighted distance from token t to the final position:

$$\tau_{t,T}^\ell = \sum_{k=t+1}^{T} \bar{\Delta}_k^\ell, \qquad P_t^\ell = \exp(-\alpha \tau_{t,T}^\ell),$$

where $\alpha > 0$ is a decay coefficient. Tokens far from the final readout receive smaller P_t^ℓ, while tokens close to the final readout receive larger P_t^ℓ and are therefore less likely to be pruned.

3.4 Log-scale normalized scoring

A direct multiplicative score $\bar{\Delta}_t^\ell E_t^\ell P_t^\ell$ can be dominated by one factor because the three quantities have different dynamic ranges. We therefore combine them in log-normalized form. For any positive token-wise quantity Z_t^ℓ, define

$$\mathrm{LN}(Z_t^\ell) = \frac{\log(Z_t^\ell + \epsilon) - \mu_Z^\ell}{\sigma_Z^\ell + \epsilon}, \qquad \mu_Z^\ell, \sigma_Z^\ell = \mathrm{MeanStd}_{t<T}\left[\log(Z_t^\ell + \epsilon)\right].$$

The final boundary sensitivity score is

$$s_t^\ell = \mathrm{LN}(\bar{\Delta}_t^\ell) + \mathrm{LN}(E_t^\ell) + \mathrm{LN}(P_t^\ell), \qquad t < T.$$

Algorithm 1 Calibration: selecting where to prune

Require: Mamba model f, calibration set $\mathcal{D}_{\text{cal}}$, pruning ratio r_{cal}, KL tolerance δ

for $\ell = 1, \dots, L-1$ **do**
  for $x \in \mathcal{D}_{\text{cal}}$ **do**
    Run full model and compute p_T^{full}
    Run layers $1, \dots, \ell$ at full sequence length
    Compute s_t^ℓ for $t < T$ using Eq. (3.4)
    Prune bottom $\lfloor r_{\text{cal}}(T-1) \rfloor$ non-final tokens and compact
    Run layers $\ell+1, \dots, L$ and compute $p_T^{\text{pruned}}(\ell)$
  end for
  Compute $\text{KL}_\ell = \mathbb{E}_{x \sim \mathcal{D}_{\text{cal}}}[D_{\text{KL}}(p_T^{\text{full}} \| p_T^{\text{pruned}}(\ell))]$
end for
return $\ell^\star = \min_\ell \{\text{KL}_\ell \le \delta\}$

Algorithm 2 Inference: pruning after the selected layer

Require: Mamba model f, prompt x, fixed pruning layer $\ell^\star$, pruning ratio r

Run layers $1, \dots, \ell^\star$ at full sequence length
Compute $\bar{\Delta}_t^{\ell^\star}$, $E_t^{\ell^\star}$, $P_t^{\ell^\star}$, and $s_t^{\ell^\star}$ for $t < T$
$J^{\ell^\star} = \text{BottomK}(\{s_t^{\ell^\star}\}_{t=1}^{T-1}, \lfloor r(T-1) \rfloor)$
$S^{\ell^\star} = \{1, \dots, T\} \setminus J^{\ell^\star}$ ▷ final token always kept
$X^{\ell^\star+1} = \text{Gather}(Y^{\ell^\star}, S^{\ell^\star})$
Run layers $\ell^\star + 1, \dots, L$ on the compacted sequence
Compute next-token distribution from the final retained prompt position

Lower scores indicate safer tokens to remove. Given pruning ratio r, we prune the lowest-scoring non-final positions:

$$J^\ell = \text{BottomK}\left(\{s_t^\ell\}_{t=1}^{T-1}, \lfloor r(T-1) \rfloor\right), \qquad S^\ell = \{1, \dots, T\} \setminus J^\ell.$$

The final token is excluded from pruning by construction, so the compacted sequence still ends close to the original final prompt token.

3.5 Algorithms

Algorithm 1 describes the calibration procedure for selecting where to prune. Algorithm 2 describes the inference-time procedure for how to prune after $\ell^\star$ has been fixed.

3.6 Computational and memory impact

After calibration, inference uses a fixed pruning layer $\ell^\star$ and pruning ratio r. Layers $1, \dots, \ell^\star$ process the full sequence of length T, while layers $\ell^\star + 1, \dots, L$ process the compacted sequence of length

$$T' = (1-r)T.$$

The idealized fraction of sequence-dependent downstream computation removed is therefore

$$r \cdot \frac{L - \ell^\star}{L}.$$

The savings apply to downstream hidden activations, projections, convolution inputs, generated Δ, B, and C tensors, selective-scan inputs, and scan-related workspace whose size scales with sequence length. The method does not reduce parameter count: model weights remain resident, so when weights dominate peak memory, pruning is primarily a prefill compute/latency optimization rather than a model-memory reduction technique.

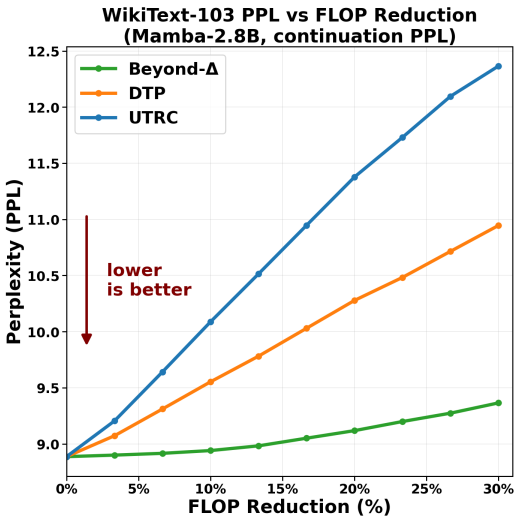
4 EXPERIMENTS

We test five questions: whether ReBound improves quality over UTRC (Zhan et al., 2024) and DTP (Park et al., 2026), whether its three score components are complementary, whether calibration identifies stable model-specific pruning boundaries, whether gains transfer to long-context LongBench evaluation, and whether FLOP savings become hardware speedups.

4.1 EXPERIMENTAL SETUP

Models and pruning. We evaluate Mamba-1 1.4B and 2.8B (extttmamba-2.8b) for the main study and Mamba-2 2.7B (extttmamba2-2.7b) for cross-architecture calibration and LongBench. Pruning is prefill-only, occurs after a full scan of the selected layer, physically compacts surviving tokens, and always retains the final prompt token.

Calibration and evaluation. Layer selection uses $n = 256$ held-out WikiText-103 passages and the final-readout KL criterion in Section 3.1, with tolerance $\delta = 0.01$; no evaluation-task labels are used. We report LAMBADA perplexity plus accuracy on LAMBADA, PIQA, ARC-Easy, HellaSwag, and WinoGrande (Gao et al., 2023); the five-task accuracy mean excludes perplexity. We additionally evaluate WikiText-103 continuation (Merity et al., 2017) across FLOP budgets and five LongBench tasks (Bai et al., 2024): Qasper, GovReport, PassageRetrieval-en, LCC, and RepoBench-P.



FLOPs Red.	WikiText-103 PPL↓		
	ReBound	DTP (Park et al., 2026)	UTRC (Zhan et al., 2024)
6.67%	**8.9158**	9.3110	9.6416
10.00%	**8.9404**	9.5528	10.0871
13.33%	**8.9824**	9.7812	10.5141

Figure 3: WikiText-103 continuation perplexity sweep on Mamba-1 2.8B; uncompressed PPL is 8.8863.

Baselines and hardware. We compare against no pruning, UTRC (Zhan et al., 2024), DTP (Park et al., 2026), and random pruning; controlled random/DTP comparisons use the same layer and ratio as ours. All methods are training-free. Calibration, evaluation, and latency/throughput measurements use a single NVIDIA H100 80GB GPU.

4.2 MAIN RESULTS

Table 1 and Figure 3 show complementary short- and longer-context behavior. At 10% FLOP reduction, ReBound loses only 0.89 and 0.62 average accuracy points on Mamba-1 1.4B and 2.8B, versus $7.25/9.06$ for DTP and $2.40/2.38$ for UTRC. On Mamba-1 2.8B WikiText-103 continuation, PPL rises from 8.8863 to only 8.9404 at 10% FLOP reduction and 8.9824 at 13.33%, compared with 9.5528/9.7812 for DTP and 10.0871/10.5141 for UTRC. Thus ReBound avoids DTP's accuracy collapse and UTRC's larger continuation-perplexity degradation.

4.3 SCORE ABLATION, LONGBENCH GENERALIZATION, AND REALIZED SPEEDUP

The ablation isolates the score design: the full $\Delta + E + P$ score gives the smallest PPL increase (+0.69%), and the two strongest pairwise variants both retain readout proximity, showing that local Δ and energy alone are insufficient. LongBench gives the same conclusion under long-context QA, summarization, retrieval, and code completion. Across all five tasks and both architectures, ReBound degrades less than DTP; the gap is especially large on PassageRetrieval-en, where DTP drops 34.8% on Mamba-1 and 46.2% on Mamba-2 while ReBound is effectively unchanged.

The FLOP reduction also materializes on hardware. For Mamba-1 2.8B with pruning after L44 and 50% token keep, a single forward pass at $B = 8, L = 2048$ is $1.15\times$ faster; at $L = 1024$, throughput

Table 1: Main results at 10% FLOP reduction across Mamba-1 1.4B and Mamba-1 2.8B. LAMBADA reports perplexity, perplexity change relative to the corresponding baseline, and accuracy. Avg. is computed over the common five accuracy tasks: LAMBADA, HellaSwag, PIQA, ARC-E, and WinoGrande. ΔAvg. reports the average accuracy change in percentage points relative to the corresponding uncompressed baseline.

Method	FLOPs Reduction	LAMBADA			HellaSwag Acc↑(%)	PIQA Acc↑(%)	ARC-E Acc↑(%)	WinoGrande Acc↑(%)	Avg. Acc↑(%)	ΔAvg.
		PPL↓	ΔPPL↓	Acc↑(%)						
Mamba-1 1.4B	0%	5.04	0.00	64.90	59.10	74.20	65.50	61.50	65.04	0.00
+ UTRC (Zhan et al., 2024)	10%	11.46	+6.42	56.50	58.90	71.30	65.10	61.40	62.64	-2.40
+ Random	10%	5.26	+0.22	64.47	42.97	65.02	60.14	59.91	58.50	-6.54
+ DTP (Δ) (Park et al., 2026)	10%	5.23	+0.19	64.56	39.52	62.40	61.87	60.62	57.79	-7.25
+ ReBound (Ours)	**10%**	**5.20**	**+0.16**	**64.47**	**58.38**	**72.12**	**65.03**	**60.77**	**64.15**	**-0.89**
Mamba-1 2.8B	0%	4.23	0.00	69.10	66.10	75.10	69.70	63.10	68.62	0.00
+ UTRC (Zhan et al., 2024)	10%	9.53	+5.30	59.90	66.00	72.00	69.80	63.50	66.24	-2.38
+ Random	10%	4.45	+0.22	68.10	47.00	65.60	65.30	62.00	61.60	-7.02
+ DTP (Δ) (Park et al., 2026)	10%	4.44	+0.21	68.00	39.50	65.10	63.90	61.30	59.56	-9.06
+ ReBound (Ours)	**10%**	**4.43**	**+0.20**	**68.00**	**65.90**	**74.20**	**68.90**	**63.00**	**68.00**	**-0.62**

Table 2: **Additional results. Top:** Ablation of pruning-score components on WikiText-103 continuation ($L = 2048$), reported as relative PPL increase over the uncompressed baseline (lower is better). Δ denotes delta strength, E selective-update energy, and P readout proximity. **Bottom:** LongBench (Bai et al., 2024) results. Base denotes the uncompressed task score, while ReBound and DTP report relative score change after pruning (higher is better). Mamba-1 uses 15% FLOP reduction and Mamba-2 uses 10%.

Metric	Score Components Used for Pruning						
	$\Delta + E + P$	$E + P$	$\Delta + P$	P only	E only	$\Delta + E$	Δ only
PPL Increase ↓	**+0.69%**	+1.20%	+1.44%	+2.10%	+2.37%	+2.49%	+2.52%

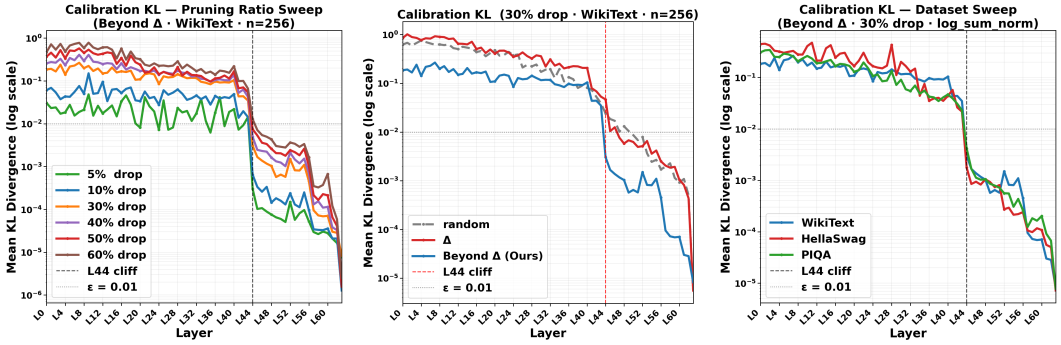
Task	**Mamba-1 2.8B** 15% FLOP Reduction			**Mamba-2 2.7B** 10% FLOP Reduction		
	Base	ReBound	DTP	Base	ReBound	DTP
Qasper (F1)	0.1637	−4.0%	−15.8%	0.1950	0.0%	−3.6%
GovReport (ROUGE-L)	0.1490	0.0%	−2.1%	0.1440	−0.4%	−1.7%
PassageRetrieval-en (Acc)	0.1150	0.0%	−34.8%	0.0650	0.0%	−46.2%
LCC (EditSim)	0.4125	−0.2%	−2.0%	0.4949	−0.6%	−2.4%
RepoBench-P (EditSim)	0.4351	0.0%	−20.3%	0.5033	−2.2%	−4.5%

improves by $1.157\times$. Speedup is consistently 1.15–$1.16\times$ across batch sizes, close to the $1.174\times$ FLOP-based ceiling because 19 of 64 layers process the halved sequence after L44. The gain increases with pruning ratio and earlier safe boundaries. Parameter weights remain unchanged and dominate static memory, so peak memory is essentially unchanged; ReBound is a prefill compute/latency optimization, not a parameter-memory compression method.

4.4 Calibration analysis: where to prune and what to prune

Figure 4 separates *where* and *what* to prune. For Mamba-1 2.8B, the final-readout KL drops sharply near L44 and remains low across pruning ratios from 5%–60% and across calibration datasets, identifying a stable earliest-safe region. The same calibration procedure on Mamba-2 2.7B produces an analogous sharp transition at L52. Thus the safe boundary is a property discovered per model, rather than a fixed layer chosen for one dataset or operating point.

At a fixed 30% pruning ratio, panel (b) compares token selection at each candidate layer: ReBound induces the lowest calibration KL, while Δ-only pruning remains substantially more disruptive and is sometimes only modestly better than random. Together with Table 2, this supports the central design:



(a) **Pruning-ratio independence.** (b) **Scoring-method comparison.** (c) **Calibration-dataset independence.**

Figure 4: **Calibration analysis of boundary pruning.** Calibration answers both *where to prune* and *what to prune*. Panels (a) and (c) show that the earliest safe pruning region is stable near L44, independent of the pruning ratio and calibration dataset. Panel (b) shows that, once this boundary is selected, ReBound better leverages it by choosing tokens that cause less final-readout perturbation than random or Δ-only pruning.

reliable Mamba pruning requires both a behaviorally safe boundary and a score that accounts for representation energy and distance to the final autoregressive readout, not local Δ alone.

5 Conclusion

We presented **ReBound** , a training-free, full-scan-then-prune method for Mamba prefill. Its readout-aware score combines Δ, selective-update energy, and readout proximity, while held-out KL calibration discovers the earliest safe model-specific boundary. Across Mamba-1 and Mamba-2, the complete score ablation, lm-evaluation-harness, WikiText-103, and LongBench consistently favor the full readout-aware design over local Δ pruning; the calibration transition moves from L44 on Mamba-1 2.8B to L52 on Mamba-2 2.7B, showing that the procedure rather than a hard-coded layer generalizes. On H100, the predicted compute savings translate to $1.15\times$ latency and $1.157\times$ throughput improvements, close to the FLOP-based ceiling.

6 Limitation and Future Work

ReBound reduces sequence-dependent prefill compute but not parameter count, so peak memory can remain weight-dominated. The current method uses a single calibrated pruning boundary; adaptive multi-layer schedules and extension to newer selective SSMs such as Mamba-3 may require revisiting the scoring terms and calibration rule.

AI Use Statement

In this work, we used generative AI tools for editing and polishing the text, as well as for assisting with software implementation. We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for mathematical proofs, assist in writing proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative or thematic data analysis, or interpret experimental results.

The use of generative AI for generating synthetic datasets, translation, or dataset cleaning and reformatting was not applicable to this work. We additionally used generative AI tools to create or edit software code, summarize or analyze existing literature, and improve the readability and presentation of the manuscript.

All AI-assisted work was reviewed by the authors. LLM-generated code was independently verified and tested for correctness, and all AI-assisted manuscript refinements were checked for technical correctness and accuracy. The authors take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the assistance of generative AI.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.09461.

Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. PuMer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 12890–12903, 2023. doi: 10.18653/v1/2023.acl-long.721.

Tien-Yu Chi, Hung-Yueh Chiang, Diana Marculescu, and Kai-Chiang Wu. Quartermap: Efficient post-training token pruning for visual state space models. In *Proceedings of the 3rd Workshop on Efficient Systems for Foundation Models at the 42nd International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2507.09514.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2405.21060.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *COLM*, 2024. URL https://arxiv.org/abs/2312.00752.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EViT: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2202.07800.

Qiankun Ma, Ziyao Zhang, Chi Su, Jie Chen, Zhen Song, Hairong Zheng, and Wen Gao. Training-free token reduction for vision mamba. *arXiv preprint arXiv:2507.14042*, 2025.

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.

Seong Yeol Park, Min Jung Kwon, Xianghua Piao, and Yeong Hyeon Gu. DTP: Delta-guided two stage pruning for mamba-based multimodal large language models. In *International Conference on Learning Representations*, 2026.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Hui Shen, Zhongwei Wan, Xin Wang, and Mi Zhang. Famba-V: Fast vision mamba with cross-layer token fusion. In *European Conference on Computer Vision*, pp. 268–278. Springer, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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

Zheng Zhan, Yushu Wu, Zhenglun Kong, Changdi Yang, Yifan Gong, Xuan Shen, Xue Lin, Pu Zhao, and Yanzhi Wang. Rethinking token reduction for state space models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1686–1697, 2024.

A LICENSES

Mamba-1 1.4B and Mamba-1 2.8B [Gu and Dao, 2024] are released under the Apache 2.0 license. WikiText-103 is released under the Creative Commons Attribution-ShareAlike (CC BY-SA 3.0) license. LAMBADA is released under the Modified MIT license. PIQA, ARC-Easy, HellaSwag, and WinoGrande are released under permissive research-use licenses (CC BY-4.0, CC BY-SA 4.0, MIT, and Apache 2.0 / CC BY-4.0 respectively, per their original repositories). The lm-evaluation-harness used for downstream evaluation is released under the MIT license. All assets are used in accordance with their respective terms of use.

B PERTURBATION ANALYSIS

We analyze the effect of removing a single token from a selective SSM stream. The analysis has two parts. First, we bound the suffix error induced inside one selective SSM layer when a token is removed before the scan. Second, we bound how this suffix error propagates through subsequent Mamba blocks. Our actual method prunes only after a full scan at the selected layer; therefore, the selected pruning layer is exact, and the first nonzero error appears in the next layer that receives the compacted sequence.

B.1 SINGLE-LAYER ERROR FROM DELETING ONE TOKEN

Consider one selective SSM layer. Let $u_1, \dots, u_T$ denote the sequence entering the selective scan after the layer's input projection and convolution. The scan is

$$h_t = \bar{A}_t h_{t-1} + \bar{B}_t u_t, \qquad o_t = C_t h_t + D_t u_t, \tag{5}$$

where $h_t \in \mathbb{R}^N$ is the SSM state, o_t is the scan output, and

$$\bar{A}_t = \exp(\Delta_t A).$$

For notational simplicity, define the state write

$$b_t = \bar{B}_t u_t. \tag{6}$$

Then the recurrence becomes

$$h_t = \bar{A}_t h_{t-1} + b_t. \tag{7}$$

Suppose token j is deleted before this layer. The pruned sequence is

$$u_1, \dots, u_{j-1}, u_{j+1}, \dots, u_T.$$

We compare the full output at original position $t > j$ with the pruned output at position $t-1$, which corresponds to the same retained original token. Let $\tilde{h}_{t-1}$ denote the pruned state corresponding to original token t, and define the aligned state error

$$e_t = h_t - \tilde{h}_{t-1}, \qquad t > j. \tag{8}$$

We first state the result under an idealized retained-token alignment assumption, then extend it to include local mismatch terms caused by convolutional or index changes.

Assumption 1: stability of the selective transition. For each retained position t, assume there exists $\nu_t \geq 0$ such that

$$\|\bar{A}_t\| \leq \exp(-\nu_t). \tag{9}$$

For diagonal Mamba transitions with $A = -\exp(A_{\log})$ and $\Delta_t \geq 0$, this holds in the natural diagonal norm with

$$\nu_t \approx \mu\, \underline{\Delta}_t, \qquad \mu = \min_d |A_d|, \tag{10}$$

where $\underline{\Delta}_t$ is a lower channel statistic such as the minimum or a conservative quantile of $\Delta_{t,d}$. In practice, we use the channel mean $\bar{\Delta}_t$ as a stable scoring statistic, but the deterministic bound is stated using a conservative contraction parameter ν_t.

Assumption 2: retained-token parameters are unchanged. For the first bound, assume that after deleting token j, the retained tokens $t > j$ use the same $(\bar{A}_t, b_t, C_t, D_t)$ as in the full sequence. This isolates the recurrent effect of missing one state update. We relax this assumption below.

Define the missing-update vector

$$\xi_j = h_j - h_{j-1} = (\bar{A}_j - I)h_{j-1} + b_j. \tag{11}$$

This is the state displacement caused by token j. If token j is removed, the next retained token starts from h_{j-1} instead of h_j, and ξ_j is the initial state error injected by the deletion.

Proposition 1 (Single-layer suffix error under exact retained-token alignment)**.** *Under Assumptions 1 and 2, deleting token j induces no error for positions $t < j$. For each retained position $t > j$,*

$$e_t = \left(\prod_{k=j+1}^{t} \bar{A}_k \right) \xi_j, \tag{12}$$

and therefore

$$\|e_t\| \le \exp\left(- \sum_{k=j+1}^{t} \nu_k \right) \|\xi_j\|. \tag{13}$$

If $\|C_t\| \le c_t$, then the aligned output error satisfies

$$\|o_t - \tilde{o}_{t-1}\| \le c_t \exp\left(- \sum_{k=j+1}^{t} \nu_k \right) \|\xi_j\|. \tag{14}$$

Proof. For positions before j, both the full and pruned scans process the same prefix, so their states and outputs are identical.

For the first retained token after the deletion, original position $j+1$, the full scan gives

$$h_{j+1} = \bar{A}_{j+1} h_j + b_{j+1}.$$

The pruned scan processes u_{j+1} immediately after u_{j-1}, so under Assumption 2,

$$\tilde{h}_j = \bar{A}_{j+1} h_{j-1} + b_{j+1}.$$

Therefore,

$$e_{j+1} = h_{j+1} - \tilde{h}_j = \bar{A}_{j+1}(h_j - h_{j-1}) = \bar{A}_{j+1}\xi_j.$$

For any later retained token $t > j+1$,

$$h_t = \bar{A}_t h_{t-1} + b_t, \qquad \tilde{h}_{t-1} = \bar{A}_t \tilde{h}_{t-2} + b_t.$$

Subtracting gives

$$e_t = \bar{A}_t e_{t-1}.$$

Unrolling from e_{j+1} yields

$$e_t = \bar{A}_t \bar{A}_{t-1} \cdots \bar{A}_{j+1} \xi_j = \left(\prod_{k=j+1}^{t} \bar{A}_k \right) \xi_j.$$

Taking norms and applying Assumption 1 gives

$$\|e_t\| \le \prod_{k=j+1}^{t} \|\bar{A}_k\| \, \|\xi_j\| \le \exp\left(- \sum_{k=j+1}^{t} \nu_k \right) \|\xi_j\|.$$

Finally, since the direct term $D_t u_t$ is the same for the aligned retained token under Assumption 2,

$$o_t - \tilde{o}_{t-1} = C_t e_t,$$

which gives

$$\|o_t - \tilde{o}_{t-1}\| \le \|C_t\| \, \|e_t\| \le c_t \exp\left(- \sum_{k=j+1}^{t} \nu_k \right) \|\xi_j\|.$$

□

Proposition 1 shows that a single deletion creates a suffix error. The error magnitude is controlled by two factors: the missing-update size $\|\xi_j\|$ and the transition product from j to t. For the final token T, the bound becomes

$$\|o_T - \tilde{o}_{T-1}\| \le c_T \exp\left(-\sum_{k=j+1}^{T} \nu_k\right) \|\xi_j\|. \tag{15}$$

Bounding the missing update. The missing-update term is

$$\xi_j = (\bar{A}_j - I)h_{j-1} + \bar{B}_j u_j.$$

Assume $\|h_{j-1}\| \le H$, $\|u_j\| \le U_j$, and $\|\bar{B}_j\| \le \beta_j$. Then

$$\|\xi_j\| \le \|\bar{A}_j - I\|\, H + \beta_j U_j. \tag{16}$$

For small-to-moderate Δ_j, using $\bar{A}_j = \exp(\Delta_j A)$ gives

$$\|\bar{A}_j - I\| \le \Delta_j \|A\| \exp(\Delta_j \|A\|), \tag{17}$$

and the zero-order hold form of $\bar{B}_j$ gives the first-order scaling

$$\|\bar{B}_j u_j\| = O(\Delta_j \|B_j\|\, \|u_j\|). \tag{18}$$

Thus,

$$\|\xi_j\| = O(\Delta_j K_j)\,, \tag{19}$$

where K_j is an intrinsic energy term depending on the previous state and the current input magnitude, for example

$$K_j = \|A\|\, \|h_{j-1}\| + \|B_j\|\, \|u_j\|. \tag{20}$$

In implementation, we use the boundary representation energy minus residual as a practical observable energy term,

$$K_j \approx \|y_j^{\ell}\|_2. \tag{21}$$

Combining (15) and (19), the final-position error from deleting token j has the form

$$\|o_T - \tilde{o}_{T-1}\| \lesssim C \Delta_j K_j \exp\left(-\sum_{k=j+1}^{T} \nu_k\right). \tag{22}$$

Taking logarithms gives

$$\log \|o_T - \tilde{o}_{T-1}\| \lesssim \log \Delta_j + \log K_j - \sum_{k=j+1}^{T} \nu_k + \text{const}. \tag{23}$$

This motivates our ReBound score: delta strength, representation energy, and readout proximity appear additively in log space.

Extension with local mismatch. Assumption 2 is exact for an isolated selective scan with fixed retained-token parameters. A full Mamba block also contains a causal convolution and input-dependent projections. If deletion changes the local convolutional neighborhood or the input-dependent parameters for nearby retained tokens, the recurrence includes an additional forcing term.

Let r_t collect all mismatch terms for retained token $t > j$:

$$r_t = (\bar{A}_t - \tilde{A}_t)\tilde{h}_{t-1} + (b_t - \tilde{b}_t), \tag{24}$$

where $\tilde{A}_t$ and $\tilde{b}_t$ are the effective transition and write used by the compacted sequence. Then the aligned error recurrence becomes

$$e_t = \bar{A}_t e_{t-1} + r_t. \tag{25}$$

If $\|r_t\| \le \eta_t$, then

$$\|e_t\| \le \exp\left(-\sum_{k=j+1}^{t} \nu_k\right) \|\xi_j\| + \sum_{s=j+1}^{t} \exp\left(-\sum_{k=s+1}^{t} \nu_k\right) \eta_s. \tag{26}$$

If the convolution has finite width w, then the largest mismatch terms are local: η_s is nonzero or largest only for positions $s \in \{j+1, \ldots, j+w\}$, after which the bound reduces to propagation of the induced suffix error through the stable scan.

B.2 PROPAGATION OF THE ERROR ACROSS LAYERS

We now analyze how the single-layer suffix error propagates through later Mamba blocks. Let pruning occur after layer $\ell^\star$. Since layer $\ell^\star$ is evaluated at full length, its output sequence

$$Y^{\ell^\star} = [y_1^{\ell^\star}, \dots, y_T^{\ell^\star}]$$

is exact. Deleting token j creates the compacted boundary sequence

$$\tilde{X}^{\ell^\star+1} = [y_1^{\ell^\star}, \dots, y_{j-1}^{\ell^\star}, y_{j+1}^{\ell^\star}, \dots, y_T^{\ell^\star}].$$

Thus, the first downstream error appears inside layer $\ell^\star + 1$, because that layer receives a shortened input stream.

Let E^m denote the aligned error sequence after layer m on the compacted index set. In particular, $E^{\ell^\star+1}$ is the suffix error produced by applying the single-layer deletion bound from Section B.1 to layer $\ell^\star + 1$.

Assumption 3: downstream layer Lipschitz gain. For each downstream Mamba block $m > \ell^\star+1$, assume the block is locally Lipschitz on the activation region reached by the full and pruned runs. That is, there exists $\Gamma_m \geq 0$ such that for any aligned perturbation sequence E^{m-1},

$$\|E^m\|_\infty \leq \Gamma_m \|E^{m-1}\|_\infty, \tag{27}$$

where

$$\|E^m\|_\infty = \max_t \|E_t^m\|. \tag{28}$$

This assumption is mild for bounded activations: linear projections, depthwise convolutions, SiLU gates, RMSNorm with nonzero ϵ, softplus, and stable selective scans are locally Lipschitz. Residual connections are included in Γ_m and may make $\Gamma_m > 1$.

Theorem 1 (Cross-layer propagation of one-token pruning error)**.** *Suppose token j is pruned after the full scan of layer $\ell^\star$. Let $B_{\ell^\star+1}(j)$ be an upper bound on the aligned output error produced by layer $\ell^\star + 1$:*

$$\|E^{\ell^\star+1}\|_\infty \leq B_{\ell^\star+1}(j). \tag{29}$$

Under Assumption 3, the final-layer aligned error satisfies

$$\|E^L\|_\infty \leq \left(\prod_{m=\ell^\star+2}^{L} \Gamma_m \right) B_{\ell^\star+1}(j). \tag{30}$$

In particular, for the final readout position,

$$\|E_T^L\| \leq \left(\prod_{m=\ell^\star+2}^{L} \Gamma_m \right) B_{\ell^\star+1}(j). \tag{31}$$

Proof. By definition,

$$\|E^{\ell^\star+1}\|_\infty \leq B_{\ell^\star+1}(j).$$

Applying Assumption 3 to layer $\ell^\star + 2$ gives

$$\|E^{\ell^\star+2}\|_\infty \leq \Gamma_{\ell^\star+2} \|E^{\ell^\star+1}\|_\infty \leq \Gamma_{\ell^\star+2} B_{\ell^\star+1}(j).$$

Repeating this argument through layer L yields

$$\|E^L\|_\infty \leq \Gamma_L \Gamma_{L-1} \cdots \Gamma_{\ell^\star+2} B_{\ell^\star+1}(j),$$

which proves (30). Since the final readout error is one element of the error sequence,

$$\|E_T^L\| \leq \|E^L\|_\infty,$$

which proves (31). □

Combining Theorem 1 with the single-layer bound gives the full one-token pruning bound. Ignoring local mismatch terms for clarity, we obtain

$$\|E_T^L\| \lesssim \left(\prod_{m=\ell^\star+2}^{L} \Gamma_m\right) C_{\ell^\star+1,T} \Delta_j^{\ell^\star+1} K_j^{\ell^\star+1} \exp\left(-\sum_{k=j+1}^{T} \nu_k^{\ell^\star+1}\right). \tag{32}$$

With local mismatch terms included, the bound becomes

$$\|E_T^L\| \le \left(\prod_{m=\ell^\star+2}^{L} \Gamma_m\right) C_{\ell^\star+1,T} \Bigg[\exp\left(-\sum_{k=j+1}^{T} \nu_k^{\ell^\star+1}\right) \|\xi_j^{\ell^\star+1}\| + \sum_{s=j+1}^{T} \exp\left(-\sum_{k=s+1}^{T} \nu_k^{\ell^\star+1}\right) \eta_s^{\ell^\star+1} \Bigg]. \tag{33}$$

Connection to next-token KL. Let the final logits be

$$z_T = W_{\mathrm{lm}} y_T^L. \tag{34}$$

Then

$$\|\Delta z_T\| \le \|W_{\mathrm{lm}}\| \, \|E_T^L\|. \tag{35}$$

The softmax map has bounded curvature, so for sufficiently small logit perturbations,

$$D_{\mathrm{KL}}\left(p_T^{\mathrm{full}} \,\|\, p_T^{\mathrm{pruned}}\right) \le \frac{1}{2}\|\Delta z_T\|_2^2 \le \frac{1}{2}\|W_{\mathrm{lm}}\|^2 \|E_T^L\|_2^2. \tag{36}$$

Thus, controlling the final readout perturbation also controls the change in the model's next-token distribution. This motivates our calibration rule: we select pruning layers by directly measuring the empirical KL between the full and pruned final next-token distributions.

B.3 Implications for the pruning score

Equation (32) shows that the final error caused by deleting token j is controlled by three main terms:

$$\Delta_j, \qquad K_j, \qquad \exp\left(-\sum_{k=j+1}^{T} \nu_k\right). \tag{37}$$

The first term measures how strongly token j writes into the selective state. The second measures the magnitude of the representation being removed. The third measures how much the induced error can propagate to the final readout.

Approximating ν_k by a constant multiple of the channel-mean timescale $\bar{\Delta}_k$ gives

$$\sum_{k=j+1}^{T} \nu_k \approx \mu \sum_{k=j+1}^{T} \bar{\Delta}_k = \mu \tau_j. \tag{38}$$

Therefore,

$$\|E_T^L\| \lesssim \mathrm{const} \cdot \bar{\Delta}_j K_j e^{-\mu\tau_j}. \tag{39}$$

Taking logs yields

$$\log \|E_T^L\| \lesssim \log \bar{\Delta}_j + \log K_j - \mu\tau_j + \mathrm{const}. \tag{40}$$

This is the mathematical motivation for the log-sum normalized score used by ReBound :

$$s_j^\ell = Z\big(\log(\bar{\Delta}_j^\ell + \epsilon)\big) + Z\big(\log(K_j^\ell + \epsilon)\big) + Z\left(-\mu \sum_{k=j+1}^{T} \bar{\Delta}_k^\ell\right), \tag{41}$$

where $Z(\cdot)$ denotes token-wise z-score normalization within the same sequence and layer. Lower values correspond to smaller estimated final-readout perturbation and are therefore safer to prune.

C CONTINUATION PERPLEXITY

For perplexity evaluation, we report *continuation perplexity* rather than perplexity over the full input sequence. Each evaluation example is split into a prefill context and a continuation:

$$x = [x_{1:T_p}, x_{T_p+1:T_p+T_c}],$$

where $x_{1:T_p}$ is the prompt processed during prefill and $x_{T_p+1:T_p+T_c}$ is the target continuation. Our pruning method is applied only during the prefill stage. After the compressed prefill produces the recurrent state used to initialize generation, we evaluate the likelihood of the continuation tokens under the standard teacher-forced language-modeling objective:

$$\mathcal{L}_{\text{cont}} = -\frac{1}{T_c} \sum_{s=1}^{T_c} \log p_\theta \left(x_{T_p+s} \mid x_{1:T_p}, x_{T_p+1:T_p+s-1} \right).$$

The continuation perplexity is then

$$\text{PPL}_{\text{cont}} = \exp(\mathcal{L}_{\text{cont}}).$$

This metric scores only the text generated after prefill; the prompt tokens themselves are used only as conditioning context and are not included in the perplexity denominator. During continuation evaluation, we still use teacher forcing: at each decoding step, the model receives the ground-truth previous continuation token rather than its own sampled prediction. Thus, continuation perplexity measures the quality of the predictive distribution induced by the compressed prefill state, while avoiding confounding effects from autoregressive sampling noise.

Continuation perplexity is especially important when comparing against token merging or pruning methods. Such methods physically alter the prompt representation by deleting or merging intermediate prompt tokens, so perplexity over the prefill tokens is not always well defined: removed tokens may no longer have corresponding output positions, and merged tokens may not admit a unique target-token alignment. Scoring the compressed prompt would therefore depend on method-specific reconstruction or alignment choices. In contrast, the continuation is identical for all methods and remains outside the compressed prefill region. This makes continuation perplexity a fair and architecture-agnostic measure of whether a compression method preserves the information needed for downstream generation.

In our experiments, all methods are evaluated under the same teacher-forced continuation protocol. The only difference between methods is the compressed recurrent state produced after prefill. A low continuation perplexity therefore indicates that the compressed prompt representation preserves the final autoregressive readout needed to predict future text.