LAMB: A Training-Free Method to Enhance the Long-Context Understanding of SSMs via Attention-Guided Token Filtering

Zhifan Ye[1*], **Zheng Wang**[1], **Kejing Xia**[1], **Jihoon Hong**[1], **Leshu Li**[1], **Lexington Whalen**[1], **Cheng Wan**[1], **Yonggan Fu**[1], **Yingyan (Celine) Lin**[1], **Souvik Kundu**[2]
[1]Georgia Institute of Technology, USA [2]Intel Labs, USA
{zye327,celine.lin}@gatech.edu, souvikk.kundu@intel.com

Abstract

State space models (SSMs) achieve efficient sub-quadratic compute complexity but often exhibit significant performance drops as context length increases. Recent work attributes this deterioration to an exponential decay in hidden state memory. While token filtering has emerged as a promising remedy, its underlying rationale and limitations remain largely non-understood. In this paper, we first investigate the attention patterns of Mamba to shed light on why token filtering alleviates long-context degradation. Motivated by these findings, we propose LAMB, a training-free, attention-guided token filtering strategy designed to preserve critical tokens during inference. LAMB can boost long-context performance for both pure SSMs and hybrid models, achieving up to an average improvement of 30.35% over state-of-the-art techniques on standard long-context understanding benchmarks. Our analysis and experiments reveal new insights into the interplay between attention, token selection, and memory retention, and are thus expected to inspire broader applications of token filtering in long-sequence modeling. Our code is available at https://github.com/GATECH-EIC/LAMB.

1 Introduction

State-space models (SSMs), including Mamba variants (Gu and Dao, 2024; Dao and Gu, 2024), have emerged as a sub-quadratic alternative to traditional transformers, enabling large language models (LLMs) to process long contexts more efficiently (Bai et al., 2024). However, prior studies (Waleffe et al., 2024; Azizi et al., 2025) have shown that vanilla Mamba models struggle with contexts exceeding their training length, primarily due to the exponential decay of their hidden states. This shortcoming prevents SSMs from fully realizing their sub-quadratic efficiency benefits.

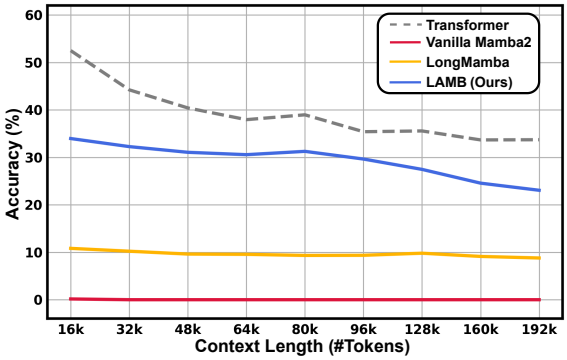

Figure 1: Comparison of average performance on RULER across various context lengths. We evaluate a Transformer baseline (Llama-3.2-1B (Llama Team, 2024)), the vanilla Mamba2-1.3B (Dao and Gu, 2024), the LongMamba-enhanced Mamba2-1.3B model (state-of-the-art baseline for training-free long-context enhancement), and the Mamba2-1.3B model enhanced with the proposed LAMB framework.

Recent works (Ben-Kish et al., 2025; Ye et al., 2025) have mitigated this issue by introducing token filtering techniques to extend Mamba models' effective context length. For instance, LongMamba (Ye et al., 2025) categorizes the hidden state channels of SSMs into global and local channels, capturing global and local information, respectively. It then filters out less important tokens from the global channels to mitigate information decay. While these methods have demonstrated effectiveness across various applications, a comprehensive understanding of their underlying mechanisms, strengths, and limitations remains lacking. In this paper, we make the following contributions:

- We present a new study of token filtering methods by analyzing the attention [1] patterns of Mamba. Specifically, we observe that a small subset of tokens dominates the attention map, indicating that preserving these key tokens

[*]Work done as a part of his internship at Intel.

[1]Note that the "attention" refers to attention-equivalent metrics derived from the original Mamba mechanism, as formulated by Ali et al. and detailed in Sec. 2.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1200–1209

can be sufficient for retaining the major attention pattern. Furthermore, we find that an attention-guided approach can outperform previous approaches when it comes to identifying these key tokens.

- Building on these insights, we introduce LAMB, a ***training-free*** method for **L**ong-context extension driven by **A**ttention-guided token filtering in **M**am**B**a. LAMB features a custom attention metric that could identify important tokens from lengthy input sequences. Extensive experiments demonstrate that LAMB surpasses the state-of-the-art (SOTA) method on both Mamba models (as shown in Fig. 1) and hybrid architectures.

2 Preliminary and Related Works

Mamba SSM. The Mamba model (Gu and Dao, 2024; Dao and Gu, 2024) is built by stacking multiple layers of Mamba blocks, each of which processes an input sequence of L tokens while maintaining its own hidden state h. The hidden state is updated recurrently within each block as the tokens are processed sequentially. For a given channel c and token index t, the update is defined by:

$$h_{t;c} = \bar{A}_{t;c}\, h_{t-1;c} + \bar{B}_{t;c}\, x_{t;c}, \quad (1)$$

$$\bar{A}_{t;c} = \exp\big(\Delta_{t;c}\, A_c\big), \quad \bar{B}_{t;c} = \Delta_{t;c}\, B_{t;c}. \quad (2)$$

where $h_{t;c} \in \mathbb{R}^N$ denotes the hidden state vector after processing token t, $A_c \in \mathbb{R}^{N\times N}$ is a constant negative matrix and $B_{t;c} \in \mathbb{R}^{N\times 1}$ is a token-dependent matrix. $\Delta_{t;c} \in (0,1)$ is a scalar that modulates the update magnitude: When $\Delta_{t;c} \approx 0$, we have $\bar{A}_{t;c} \approx 1$ and $\bar{B}_{t;c} \approx 0$, resulting in little to no change in the hidden state, whereas larger values of $\Delta_{t;c}$ produce more significant updates. The output at token t is given by

$$y_{t;c} = C_{t;c}^{\top} h_{t;c}, \quad (3)$$

where $C_{t;c} \in \mathbb{R}^{N\times 1}$. For brevity, we will omit the subscript c in the remainder of this work unless channel-specific behavior is being emphasized.

Attention in Mamba Models. Previous work (Ali et al., 2024) introduced an attention score to quantify the contribution of j^{th} (x_j) to the i^{th} (y_i) token in Mamba. Specifically, unrolling Eq. 3 yields

$$y_i = \sum_{j=1}^{i} C_i^{\top}\Big(\prod_{k=j+1}^{i} \bar{A}_k\Big)\bar{B}_j x_j = \sum_{j=1}^{i} \alpha_{i,j} x_j, \quad (4)$$

where $\alpha_{i,j} = C_i^{\top}\Big(\prod_{k=j+1}^{i} \bar{A}_k\Big)\bar{B}_j$ represents the contribution of x_j to y_i, serving as an analogue to the attention weights in Transformer-based models.

Long-Context Mamba. Prior studies (Ben-Kish et al., 2025; Azizi et al., 2025) have shown that Mamba models struggle to generalize beyond the sequence lengths encountered during training. This limitation stems from the hidden state update in Eq. 1, where each new token multiplies the previous hidden state h_{t-1} by a decay factor $\bar{A}_t < 1$. As the sequence length increases, this repeated decay progressively attenuates earlier information.

The previous method, LongMamba (Ye et al., 2025), mitigates this effect by first identifying a subset of hidden state channels C_g, termed "global channels", that retain long-range information. A channel is classified as global if the cumulative decay over a sampled sequence of length S, $\Pi_{t=1}^{S}\bar{A}_t$, falls below a predefined threshold θ: $\prod_{t=1}^{S}\bar{A}_t < \theta$; such channels decay slowly and thus preserve global context. LongMamba then discards tokens whose Δ_t is small and would make only negligible updates in these channels. Concretely, it sets $\Delta_t = 0$ for every discarded token, which forces $\bar{A}_t = 1$ and $\bar{B}_t = 0$, so the update rule in Eq. 1 simplifies to $h_t = h_{t-1}$. Hence, the hidden state remains unchanged for discarded tokens, while retained tokens update it normally.

LAMB employs a similar framework: it identifies the same set of global channels and sets Δ_t to zero for discarded tokens. However, LAMB selects the tokens to discard using an attention-guided importance score rather than LongMamba's heuristic based on the magnitude of Δ_t. This principled selection yields stronger long-context performance, as demonstrated in Sec. 5.

3 Analysis of Token Filtering

In this section, we systematically investigate the role of token filtering in SSM models by closely analyzing their attention patterns. Our goal is to understand why certain tokens are more critical than others for maintaining performance on long-context tasks and how to identify these critical tokens.

3.1 Attention Patterns in SSMs

Fig. 2 (a) shows the attention map of a randomly selected global channel in the Mamba2-1.3B model with 512 input tokens. A pronounced column-wise pattern emerges, indicating that only a small subset of tokens (columns) dominates the attention distribution. To quantify this effect, we compute each token's cumulative contribution to future generated

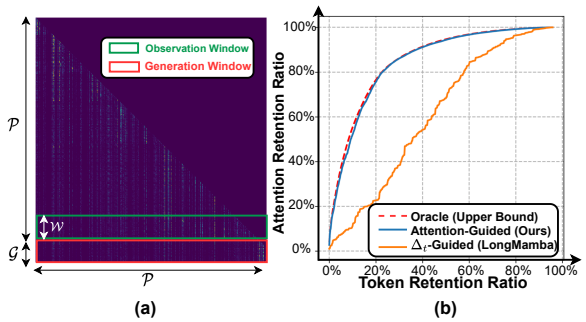

Figure 2: (a) Attention map of a randomly selected global channel in Mamba2-1.3B for an input sequence $\mathcal{P}$ of 512 tokens followed by 32 generated tokens $\mathcal{G}$. (b) Attention retention ratio within the generation window (red box in (a)) as a function of the token retention ratio under three token filtering strategies: *Oracle* (upper bound, requires oracle access to the future attention map), *Attention-Guided* (ours), and Δ_t-*Guided* (Ben-Kish et al., 2025; Ye et al., 2025). $\mathcal{W}$ denotes the last 32 tokens of the input prompt $\mathcal{P}$. Strategies are detailed in Sec. 3.

tokens (highlighted in the red box in Fig. 2 (a)):

$$\alpha_t = \sum_{i \in \mathcal{G}} \alpha_{i,t} = \sum_{i \in \mathcal{G}} C_i^\top \Big(\prod_{k=t+1}^{i} \bar{A}_k \Big) \bar{B}_t, \quad (5)$$

where $\mathcal{G}$ denotes the set of generated tokens. Under the Oracle setting shown in Fig. 2 (b) (red dashed curve), we retain tokens with the highest α_t and plot the attention retention ratio against various token retention budgets. Approximately 20% of the tokens account for 80% of the total attention, suggesting that an effective filtering scheme can retain most of the attention signal while discarding a significant portion of the tokens. Additional results in Appendix E confirm this sparsity pattern across different global channels.

3.2 Token Filtering Metric

In practice, token filtering (Ben-Kish et al., 2025; Ye et al., 2025) must be performed before generation begins to mitigate information decay in input prompts, necessitating proxy metrics for estimating future token importance. Prior studies have considered two approaches:

Δ_t-Guided Filtering. Prior works (Ben-Kish et al., 2025; Ye et al., 2025) estimate token importance based on the magnitude of Δ_t. The underlying rationale is that a large Δ_t increases $\bar{B}_t$ through Eq. 2, which subsequently amplifies α_t via Eq. 5.

Attention-Guided Filtering. For Transformer-based LLMs, SnapKV (Li et al., 2024) proposes using historical attention as a importance metric. We adapt this idea to SSMs: tokens that received

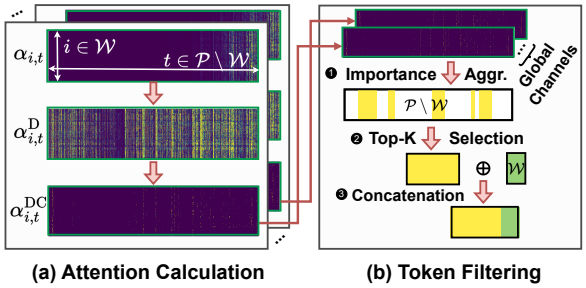

Figure 3: (a) The original Mamba attention $\alpha_{i,t}$ versus the proposed debiased attention $\alpha^{\mathrm{D}}_{i,t}$ and contrastive attention $\alpha^{\mathrm{DC}}_{i,t}$ for a 2048-token prompt. (b) End-to-end token filtering pipeline of LAMB.

substantial attention from the last few tokens before generation, referred to as the "observation window", are presumed influential for future generation. Formally, we define the importance metric as: $\hat{\alpha}_t = \sum_{i \in \mathcal{W}} \alpha_{i,t}$, where $\mathcal{W}$ denotes the observation window (last 32 tokens).

As shown in Fig. 2 (b), attention-guided filtering significantly outperforms Δ_t-guided filtering on the 512-token sequence, approaching the Oracle upper bound. At a 20% token retention ratio, attention-guided filtering retains about four times more attention in the generation window than Δ_t-guided filtering.

Despite its promise, attention-guided filtering presents two challenges in longer sequences:

C. 1 Bias Towards the Last Tokens: As illustrated in the first row of Fig. 3 (a), attention distributions in longer sequences show a strong bias towards the last few tokens (*i.e.,* the tokens closer to $\mathcal{G}$), with attention quickly diminishing for earlier tokens.

C. 2 Noisy Attention Patterns: The second row of Fig. 3 (a) highlights substantial noise in the attention distributions of longer sequences, making it difficult to identify truly influential tokens.

4 The Proposed LAMB Method

Motivated by the findings and challenges identified in Sec. 3, we introduce LAMB, a refined attention-guided token filtering framework designed to robustly handle long-context sequences in SSMs. We address the two challenges via enhanced attention metrics in Sec. 4.1, and then describe the end-to-end token filtering pipeline in Sec. 4.2.

4.1 Enhanced Attention Metric

Debiased Attention. To directly address **C. 1**, we introduce debiased attention, in which the cumulative decay factor $\prod_{k=t+1}^{i} \bar{A}_k$ responsible for the bias is replaced with a precomputed constant factor

$\prod_{k=1}^{L_0} \bar{A}_k$ based on the training sequence length length L_0, resulting in:

$$\alpha_{i,t}^{\mathrm{D}} = C_i^{\top}(\prod_{k=1}^{L_0} \bar{A}_k)\bar{B}_t. \tag{6}$$

As shown in Fig. 3 (a), this modification substantially reduces the bias towards recent tokens, yielding a more balanced representation of token importance.

Contrastive Attention. To address **C. 2**, we propose contrastive attention, which enhances the clarity of importance estimation by suppressing minor fluctuations and highlighting tokens with consistently strong contributions:

$$\alpha_{i,t}^{\mathrm{DC}} = \mathrm{ReLU}(\alpha_{i,t}^{\mathrm{D}} - \gamma \cdot \max_t(\alpha_{i,t}^{\mathrm{D}})), \tag{7}$$

where $\gamma < 1$ controls the strength of noise suppression, $\max_t \alpha_{i,t}^{\mathrm{D}}$ denotes the maximum attention score received by token i, and ReLU ensures non-negativity. The third row of Fig. 3 (a) clearly illustrates how contrastive attention effectively reduces noise and highlights the most influential tokens.

4.2 Aggregated Importance and Token Filtering Pipeline

While Sec. 4.1 introduces attention metrics for a single hidden state channel, an SSM typically comprises multiple channels. We extend the refined metrics across all global channels and present a three-step token filtering pipeline (illustrated in Fig. 3 (b)):

Step ❶ Importance Aggregation. For each token t, we aggregate the enhanced attention metric $\alpha_{i,t}^{\mathrm{DC}}$ across all global channels $c \in C_g$ and all tokens i in the observation window $\mathcal{W}$:

$$I_t^{\mathrm{raw}} = \sum_c \hat{\alpha}_{t,c}^{\mathrm{DC}} = \sum_c \sum_{i \in \mathcal{W}, i > t} \alpha_{i,t,c}^{\mathrm{DC}}. \tag{8}$$

We then apply average pooling to I_t^{raw} across the token dimension:

$$I_t = \mathtt{mean\_pool}(I_t^{\mathrm{raw}}), \tag{9}$$

which preserves local context around the important tokens and improves long-context comprehension (see ablation results provided in Sec. 5.4).

Step ❷ Top-K Token Selection. We select the top-K tokens among $\mathcal{P} \setminus \mathcal{W}$ based on aggregated importance scores I_t, identifying those tokens most likely to influence future model generations.

Step ❸ Concatenation and Selective State Update. The selected influential tokens from $\mathcal{P} \setminus \mathcal{W}$ are concatenated with the tokens from the observation window $\mathcal{W}$. Hidden states are then updated exclusively based on this concatenated set, while the remaining tokens in the input prompt $\mathcal{P}$ are excluded from the state update. This selective update effectively reduces cumulative decay and preserves crucial context.

Together, these components enable LAMB to identify important tokens in SSMs and significantly improve long-context modeling.

5 Experiments

This section presents the evaluation results of LAMB: Sec. 5.3 and Sec. 5.2 detail the results on two long-context benchmarks, HELMET (Hsieh et al., 2024) and RULER (Yen et al., 2024); Sec. 5.4 presents an ablation study of the denoising and pooling techniques; finally, Sec. 5.5 measures the latency overhead of LAMB.

5.1 Experimental Setup

We apply the proposed LAMB method to a representative SSM Mamba-2 (Gu and Dao, 2024) and a hybrid model Zamba-2 (Glorioso et al., 2024b), and benchmark them against both vanilla models and LongMamba (Ye et al., 2025), the SOTA method to enhance the SSMs' long-context capabilities. Further setup details are in Appendix A.

5.2 Benchmark Results on HELMET

Tab. 1 presents the results on HELMET (Yen et al., 2024). The table shows that the proposed LAMB consistently achieves the highest average accuracy across all three context lengths (*i.e.,* 8k, 16k, 32k) on both Mamba2 and Zamba2. This consistent improvement highlights the effectiveness of LAMB at varying context lengths. The detailed per-task performance on HELMET is provided in Appendix C.

Table 1: Benchmark results on HELMET.

Model	Method	Sequence Length		
		8k	16k	32k
Mamba2	Vanilla	3.23	3.08	1.60
	LongMamba	7.27	6.31	6.20
	LAMB (Ours)	**10.63**	**10.25**	**7.82**
Zamba2	Vanilla	5.11	6.76	3.70
	LongMamba	12.21	11.35	9.17
	LAMB (Ours)	**13.93**	**12.35**	**11.28**

Table 2: Comparison of performance (%) on RULER with a 16k context length. We evaluate vanilla models, LongMamba, and LAMB on Mamba2 (Dao and Gu, 2024) and Zamba2 (Glorioso et al., 2024a).

Model	Method	single1	single2	single3	multi1	multi2	multi3	multivalue	multiquery	vt	cwe	fwe	qa1	qa2	Avg.
Mamba2-780M	Vanilla	0	0	0	0	0	0	0	0	0	0	0	0	2	0.15
	LongMamba	0	7	2	7	0	0	1.75	2.5	1.6	0	5	7	7	3.14
	LAMB (Ours)	**98**	**88**	**73**	**57**	**3**	0	**6**	**34**	**6**	**0.7**	**42.67**	**9**	**18**	**33.49**
Mamba2-1.3B	Vanilla	0	0	1	1	0	0	0.5	0	0	0	1	0	1	0.27
	LongMamba	**100**	2	2	5	0	0	0.75	0.75	1.4	0.4	2.33	8	18	10.82
	LAMB (Ours)	99	**90**	**78**	**40**	**6**	0	**6.5**	**24.5**	**14.6**	**0.5**	**47.33**	**14**	**21**	**33.96**
Zamba2-1.2B	Vanilla	30	11	7	6	0	0	8	0	0.2	0	13.67	0	1	5.92
	LongMamba	79	92	**31**	23	0	0	**58**	**49.25**	0.2	**2.3**	0.67	1	11	26.72
	LAMB (Ours)	**83**	**96**	**31**	**26**	**2**	**1**	49.75	43	**0.8**	0.5	**30.67**	**4**	**13**	**29.29**

5.3 Benchmark Results on RULER

Tab. 2 provides the results on RULER (Hsieh et al., 2024). It can be observed that the proposed LAMB provides a +30.35%/+23.14%/+2.57% performance improvement compared to LongMamba on Mamba2-780M/Mamba2-1.3B/Zamba2-1.2B. Notably, LAMB is consistently better than LongMamba on QA tasks (*i.e.,* qa1 and qa2), demonstrating the potential of LAMB on related real-world applications.

5.4 Ablation Study

Tab. 3 presents an ablation study of the proposed techniques on the RULER benchmark. Here, *Denoising* refers to whether the denoised attention $\alpha^{\text{DC}}_{i,t}$ is used instead of the noisy $\alpha^{\text{D}}_{i,t}$, and *Pooling* refers to whether mean pooling (Eq. 9) is applied to the per-token importance. The first row and second row of Tab. 3 demonstrate that *Pooling* is crucial for LAMB, as the average accuracy on RULER drops from 33.96% to less than 5% without it. The table shows that *Denoising* is also necessary to achieve a high accuracy with LAMB, which brings a 6.74% improvement in the average accuracy on RULER (comparing the last two rows of Tab. 3). Ablation studies on the denoising factor γ and pooling kernel size are presented in Appendix B.

5.5 Latency Overhead

Tab. 4 reports the prefill latency of LAMB across a range of prompt lengths. For context lengths ranging from 32k to 192k tokens, LAMB incurs at most 12.31% additional latency. The overhead falls below 7% for contexts of 96k tokens or more, highlighting the efficiency and scalability of LAMB.

Table 3: Ablation study of the proposed techniques on RULER with a 16k sequence length.

Model	Denoising	Pooling	Acc. (%)
Mamba2-1.4B	✗	✗	3.40
	✓	✗	4.52
	✗	✓	27.22
	✓	✓	**33.96**

Table 4: Prefill latency (in seconds) for the vanilla Mamba2-1.3B model (Dao and Gu, 2024) and its LAMB-enhanced variant on an NVIDIA A100 GPU (Choquette et al., 2021) (batch size = 1). LAMB adds *no* overhead during the generation stage.

Prompt Length	32k	64k	96k	128k	160k	192k
Vanilla	0.52	1.05	1.58	2.11	2.63	3.16
LAMB	0.58	1.14	1.69	2.24	2.79	3.34
Overhead (%)	12.31	8.06	6.90	6.21	5.84	5.78

6 Conclusion

We introduced LAMB, a training-free method for enhancing the long-context performance of SSMs and hybrid architectures by selectively filtering out tokens from the hidden state update. LAMB is the first attention-guided method for token filtering and achieves up to an average of 30.35% improvement over the state-of-the-art (SOTA) method. We anticipate our findings to spark broader applications of token filtering in long-sequence modeling.

7 Limitations

LAMB presently relies on fixed values for the denoising factor γ and the pooling kernel size. Such rigidity can constrain its effectiveness across varied tasks. Enabling these hyperparameters to adjust dynamically to the input and the long-context task might unlock additional gains, which we leave to future work.

Acknowledgment

This work was partially supported by CoCoSys, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, and National Science Foundation (NSF) Division of Information & Intelligent Systems (IIS) program (Award ID: 2403297).

References

Ameen Ali, Itamar Zimerman, and Lior Wolf. 2024. The hidden attention of mamba models. *arXiv preprint arXiv:2403.01590*.

Seyedarmin Azizi, Souvik Kundu, Mohammad Erfan Sadeghi, and Massoud Pedram. 2025. Mambaextend: A training-free approach to improve long context extension of mamba. In *The Thirteenth International Conference on Learning Representations*.

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. 2024. 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)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. 2025. Decimamba: Exploring the length extrapolation potential of mamba. In *The Thirteenth International Conference on Learning Representations*.

Jack Choquette, Wishwesh Gandhi, Olivier Giroux, Nick Stam, and Ronny Krashinsky. 2021. Nvidia a100 tensor core gpu: Performance and innovation. *IEEE Micro*, 41(2):29–35.

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

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Lin, Jan Kautz, and Pavlo Molchanov. 2024. Hymba: A hybrid-head architecture for small language models.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, Anna Golubeva, Vasudev Shyam, James Whittington, Jonathan Pilault, and Beren Millidge. 2024a. The zamba2 suite: Technical report. *arXiv preprint arXiv:2411.15242*.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. 2024b. Zamba: A compact 7b ssm hybrid model. *Preprint*, arXiv:2405.16712.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. *Preprint*, arXiv:2312.00752.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. 2022. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983.

Albert Gu, Karan Goel, and Christopher Ré. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

R. E. Kalman. 1960. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, pages 22947–22970. Curran Associates, Inc.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. 2024. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*.

Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. 2023. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pages 26670–26698. PMLR.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. 2024. Samba: Simple hybrid state space models for efficient unlimited context language modeling. *arXiv preprint*.

Chien Van Nguyen, Huy Huu Nguyen, Thang M Pham, Ruiyi Zhang, Hanieh Deilamsalehy, Puneet Mathur, Ryan A Rossi, Trung Bui, Viet Dac Lai, Franck Dernoncourt, et al. 2024. Taipan: Efficient and expressive state space language models with selective attention. *arXiv preprint arXiv:2410.18572*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. 2024. An empirical study of mamba-based language models. *Preprint*, arXiv:2406.07887.

Zhifan Ye, Kejing Xia, Yonggan Fu, Xin Dong, Jihoon Hong, Xiangchi Yuan, Shizhe Diao, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. 2025. Longmamba: Enhancing mamba's long-context capabilities via training-free receptive field enlargement. In

The Thirteenth International Conference on Learning Representations.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2024. Helmet: How to evaluate long-context language models effectively and thoroughly. *Preprint*, arXiv:2410.02694.

A Experimental Setup

Models and Benchmarks. We evaluate LAMB with two open-sourced SSMs (*i.e.,* Mamba2-780M, Mamba2-1.4B (Gu and Dao, 2024)) and a representative hybrid model Zamba2-1.3B (Glorioso et al., 2024b) on two long-context benchmarks: HELMET (Yen et al., 2024) and RULER (Hsieh et al., 2024). Specifically, HELMET encompasses a diverse range of application-centric long-context tasks, covering synthetic recall, long-document question answering, summarization, many-shot in-context learning, retrieval-augmented generation, passage re-ranking, and citation-aware generation. Unlike HELMET, RULER consists exclusively of synthetic tasks, allowing precise control over sequence length and task complexity. Comprehensive evaluation on both benchmarks provides a robust assessment of LAMB's ability to handle diverse long-context applications.

Baselines. We benchmark our proposed LAMB against two baselines: vanilla models without token filtering and LongMamba (Ye et al., 2025), the SOTA training-free method for enhancing Mamba models' long context understanding via token filtering.

Hyperparameters. For all experiments, we use a denoising factor γ=0.9. When testing Mamba2 models on RULER, the pooling kernel size is 51 and we preserve top 512 tokens in each Mamba block. On all other experiments, we preserve top 1024 tokens in each Mamba block and set the pooling kernel size to 9 for the Zamba-1.2B model and 18 for the Mamba2 models. We use the same hyperparameters in LongMamba (Ye et al., 2025) for determining the global channels.

Evaluation Protocol. Our evaluation follows the official implementation and settings of the benchmarks. More specifically, on the RULER benchmark, we generate 100 test sequences for each task under each sequence length. The metrics used for HELMET and RULER follow their original papers (Yen et al., 2024; Hsieh et al., 2024), respectively.

B Ablation Study on Hyperparameters

B.1 Ablation study on the Denoising Factor

Tab. 5 shows the impact of varying the denoising factor γ on LAMB's performance across tasks in the RULER benchmark. As γ increases from 0 to 0.9, we observe a consistent improvement in most metrics. We also find that the average performance differs by less than 2% for γ values between 0.8 and 0.95, indicating that LAMB is robust to the exact choice of γ in this range.

B.2 Ablation study on Pooling Kernel Size

Tab. 6 presents the performance impact of different pooling kernel sizes on the RULER benchmark. Without pooling (kernel size = 1), performance is notably poor. Increasing the kernel size to 51 yields steady improvements, whereas using a size of 71 causes a drop in average accuracy. These results suggest that (1) pooling with a sufficiently large kernel is crucial for enhancing LAMB's performance, and (2) an oversized kernel can degrade effectiveness.

C Detailed Experimental Results

Tab. 7 presents the detailed per-task performance on HELMET. We evaluated the vanilla models, LongMamba, and HELMET using both Zamba2-1.2B and Mamba2-1.3B.

D Extended Related Works

State Space Models. The quadratic space and time complexity of attention mechanisms (Vaswani, 2017) has posed significant challenges in training and inference with long input sequences. To address these issues, a new class of models that replaces attention with convolution-based architectures has recently gained traction, offering a more computationally feasible alternative (Gu et al., 2021). Among these alternatives, state space models (SSMs) have emerged as a powerful framework. SSMs, extensively used in control theory, describe the evolution of dynamic systems through latent states that encapsulate system behavior over time (Kalman, 1960). Several models, including S4 (Gu et al., 2021), S4D (Gu et al., 2022), and LRU (Orvieto et al., 2023), leverage this approach. Notably, Mamba models (Gu and Dao, 2024; Dao and Gu, 2024) have gained attention for its departure from traditional time-invariant formulations by introducing a time-dependent state update mechanism, thereby enhancing expressivity and adaptability in sequence modeling tasks.

Long Context Mamba. While Mamba models have demonstrated strong performance under short context, prior studies (Waleffe et al., 2024; Ben-Kish et al., 2025) have shown that their effectiveness diminishes on long-context tasks due to exponential hidden state decay, which limits their ability to retain information over extended sequences. To

Table 5: LAMB's performance (%) on RULER (16k context length) with different γ values.

γ	single1	single2	single3	multi1	multi2	multi3	multivalue	multiquery	vt	cwe	fwe	qa1	qa2	Avg.
0	95	72	34	25	0	0	5.50	11.50	28.4	0.2	47.33	15	20	27.23
0.5	91	89	78	33	8	0	6.75	12.00	17.2	0.4	31.33	12	16	30.36
0.8	97	94	77	37	6	0	7.25	25.50	10.4	0.6	39.00	10	21	32.67
0.9	99	90	78	40	6	0	6.50	24.50	14.6	0.5	47.33	14	21	33.96
0.95	100	91	74	41	2	0	7.75	27.25	15.8	0.5	44.33	11	21	33.51

Table 6: LAMB's performance (%) on RULER (16k context length) with different pooling kernel sizes.

Kernel	single1	single2	single3	multi1	multi2	multi3	multivalue	multiquery	vt	cwe	fwe	qa1	qa2	Avg.
1	2	2	1	0	0	0	0.50	0.50	4.2	0.6	18.00	11	19	4.52
31	99	96	10	40	3	0	10.00	34.75	8.4	0.3	38.67	13	26	29.16
51	99	90	78	40	6	0	6.50	24.50	14.6	0.5	47.33	14	21	33.96
71	100	91	82	43	5	0	7.00	28.25	16.0	0.5	42.00	8	18	33.90

address this limitation, several methods have been proposed. For instance, DeciMamba (Ben-Kish et al., 2025) mitigates hidden state decay by progressively filtering out tokens across layers based on the magnitude of Δ_t. LongMamba (Ye et al., 2025), on the other hand, introduces the concept of global channels—channels with a wide receptive field that capture long-range dependencies across tokens—and selectively applies token filtering to these channels. Despite these advancements, the underlying principles behind the effectiveness of token filtering and its potential for further optimization remain unclear, which is a key focus of this study.

Hybrid Transformer-SSM Models. To complement the long-range precision of self-attention with the efficiency of SSM blocks, recent work has proposed hybrid architectures that employs both in a single network. For instance, Hymba (Dong et al., 2024) mixes attention and SSM heads in parallel within each layer, while a second line of work, exemplified by Jamba (Lieber et al., 2024), Samba (Ren et al., 2024), and Zamba (Glorioso et al., 2024b,a), alternates attention and Mamba layers throughout the network depth. Both approaches achieve compelling accuracy–efficiency trade-offs compared with purely Transformer- or Mamba-based models. In parallel, Taipan (Van Nguyen et al., 2024) augments Mamba with a lightweight selective-attention module that focuses on long-range dependencies among salient tokens. Orthogonal to these efforts, our proposed LAMB serves as a plug-and-play enhancement for pretrained SSM or hybrid models, boosting long-context performance without further training.

E Additional Visualization

Visualizations of attention maps and the attention retention ratios under different approaches for additional global channels are provided in Fig.4 and Fig.5. In this section, we follow the same experimental setups as in Fig. 2 and randomly sample eight additional global channels from the Mamba2-1.3B model for visualization.

Table 7: Detailed benchmark results on HELMET.

	Methods	Recall			RAG			Cite			Re-rank			LongQA			Summ			ICL			Avg		
		8k	16k	32k	8k	16k	32k	8k	16k	32k	8k	16k	32k	8k	16k	32k	8k	16k	32k	8k	16k	32k	8k	16k	32k
Mamba2	vanilla	0	0	0	11.9	10.2	5.2	0.0	0.1	0.0	0.2	0.0	0.0	5.3	4.2	3.2	1.9	1.3	0.8	11.8	5.8	2.1	3.2	3.1	1.6
	LongMamba	2.1	1.2	3.0	13.1	14.3	15.4	0.0	0.3	0.2	3.9	1.3	0.0	15.3	16.2	15.4	6.3	4.6	4.3	10.2	6.3	5.1	7.3	6.3	6.2
	LAMB	4.1	5.4	4.2	25.6	24.9	20.8	1.4	0.6	0.7	0.1	0	0	11.7	11.1	11.1	7.2	6.4	6.2	24.2	23.4	11.7	**10.6**	**10.3**	**7.8**
Zamba2	vanilla	0.3	1.2	0.0	12.7	14.2	4.9	0.6	1.0	0.3	0.1	0.7	0.5	4.7	4.6	5.0	3.6	2.1	4.4	13.8	23.6	10.8	5.1	6.8	3.7
	LongMamba	13.8	7.9	4.1	29.6	28.8	22.3	0.4	0.8	1.0	9.2	4.2	0.7	5.7	6.6	7.0	4.7	5.5	1.8	22.0	25.6	27.2	12.2	11.4	9.2
	LAMB	12.4	6.8	5.5	26.3	24.0	20.0	1.1	1.1	0.7	16.0	2.3	3.8	10.6	10.5	9.3	7.1	6.8	5.8	24.0	35.2	33.2	**13.9**	**12.4**	**11.3**

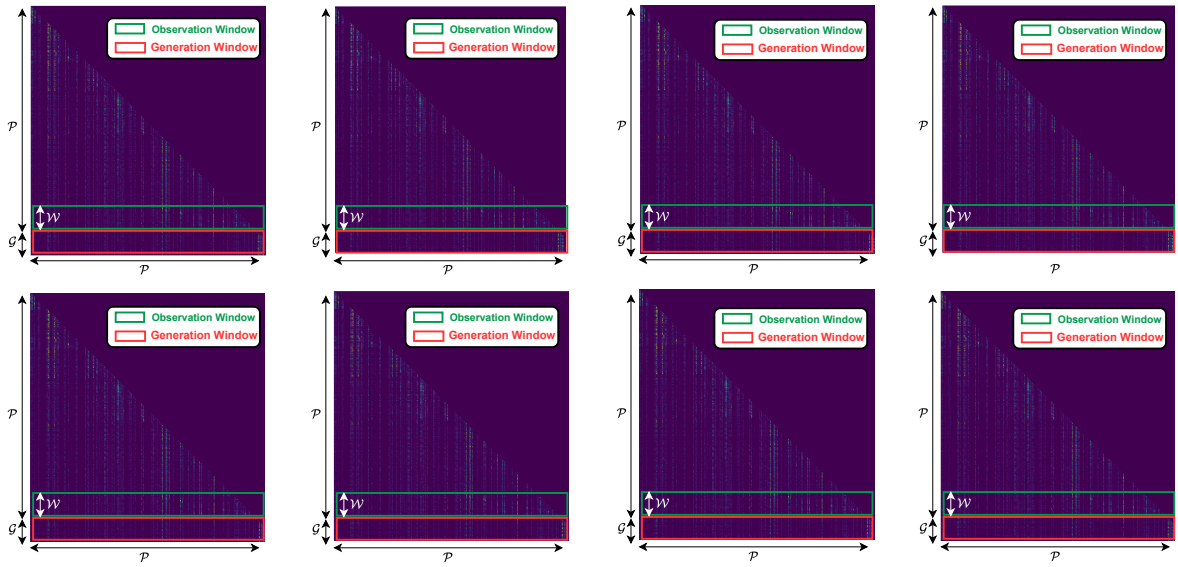

Figure 4: Visualization of attention maps from eight randomly sampled global channels, covering 512 input tokens (denoted as $\mathcal{P}$) and 32 generated tokens (denoted as $\mathcal{G}$). $\mathcal{W}$ denotes the last 32 tokens of $\mathcal{P}$.

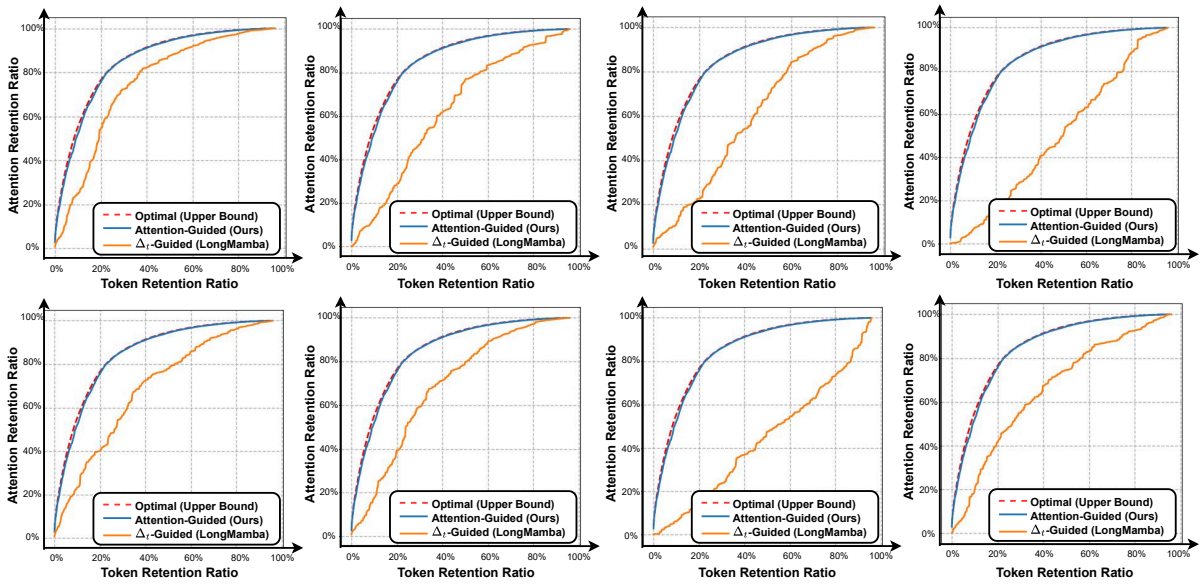

Figure 5: Attention retention ratio within the generation window under varying token retention budgets. Here we evaluate three token filtering strategies: *Oracle* (upper bound, requires oracle access to future attention maps), *Attention-Guided* (ours), and Δ_t-*Guided* (Ben-Kish et al., 2025; Ye et al., 2025).