LONGMAMBA: ENHANCING MAMBA'S LONG-CONTEXT CAPABILITIES VIA TRAINING-FREE RECEPTIVE FIELD ENLARGEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Mamba models have emerged as an efficient alternative to Transformer models for language modeling tasks, offering linear complexity as context length increases. However, despite their efficiency in handling long contexts, recent studies have demonstrated that Mamba models underperform in understanding extended contexts compared to Transformer models. To address this significant shortfall, we propose "LongMamba", a training-free technique that significantly enhances the long-context capabilities of Mamba models. Our approach builds upon the discovery that hidden state channels in Mamba models—categorized into *local* and *global channels* based on their receptive field lengths—exhibit distinct functionalities. Specifically, the *global channels* struggle to adaptively extend their effective receptive fields when input lengths far exceed their training sequence length due to exponential decay in their hidden states. We hypothesize this exponential decay is the root cause of Mamba models' limited performance in extended contexts. LongMamba counters this by effectively expanding the *global channels*' receptive fields to fully encompass the input sequence length, thus enabling them to capture global information more effectively. Through extensive benchmarking across synthetic and real-world long-context scenarios, LongMamba sets a new standard for state-of-the-art performance in Mamba-based long-context tasks, significantly extending the operational range of Mamba models without requiring additional fine-tuning. All code and models will be released upon acceptance.

1 INTRODUCTION

The rapid advancement of Transformer-based large language models (LLMs) has demonstrated significant capabilities across a diverse array of real-world tasks, ranging from question answering (Zhuang et al., 2023), document summarization (Jin et al., 2024), to code completion (Li et al., 2022). These tasks often involve processing long input sequences, such as extensive documents and sizable codebases, thus escalating the demand for LLMs to manage increasingly longer context lengths. Contemporary commercial LLMs, including Mistral Large 2 (MistralAI, 2024) and GPT-4 (Achiam et al., 2023), now feature context windows extending up to 128,000 tokens. Despite their capabilities, Transformer-based models encounter significant scalability issues as sequence lengths increase (Katharopoulos et al., 2020). This is primarily due to their quadratic computational complexity and linear memory complexity, which intensify as the token-by-token decoding process allows each token to attend to all preceding tokens.

In contrast, state space models (SSMs) (Gu et al., 2021; 2022a;b) offer a recursive computation mechanism that maintains linear computational complexity and constant memory usage, which are critical for managing fixed-size hidden states efficiently. Mamba, a recent SSM (Gu & Dao, 2023), incorporates a time-variant selective update mechanism into the hidden state, achieving performance comparable to that of Transformer-based LLMs on various language modeling tasks. Despite this, Mamba models struggle with long-sequence recall compared to similarly sized Transformers, as highlighted in recent empirical studies Waleffe et al. (2024); Ben-Kish et al. (2024).

To understand the aforementioned challenge, we first analyze the per-channel attention patterns (Ali et al., 2024) of Mamba models. We identified that the channels have distinct receptive field lengths:

most channels, termed *local channels*, focus on local contexts, while others, termed *global channels*, have receptive fields as long as the training sequence length, enabling them to capture global information from the entire input sequence. When we extend the sequence length to an order of magnitude larger than the training sequence, we find that the receptive fields of the *global channels* fail to cover the entire sequence length. This failure stems from a cumulative state decay that increases exponentially with context length, thereby rendering the *global channels* incapable of effectively capturing global information. We hypothesize that **this failure of the *global channels* to capture global information under extended sequence lengths is the root cause of Mamba's poor performance in long-context understanding**.

Inspired by these analyses and findings, we propose LongMamba, a training-free method designed to significantly enhance the receptive fields of *global channels* when the sequence length far exceeds the training sequence. Specifically, LongMamba enables the *global channels* to effectively enlarge their receptive fields by adaptively adjusting the decay dynamics within their hidden states. This enlargement ensures that these channels can maintain their function as global information processors when exposed to much longer sequences than they were originally trained on, thereby substantially extending the functional range of Mamba models. Our contributions can be summarized as follows:

- Through visualization and analysis, we are the first to demonstrate that hidden state channels in Mamba SSMs have distinct receptive field lengths. Based on these effective receptive field lengths, the channels can be categorized into two classes: most channels, termed *local channels*, focus on local contexts, while others, termed *global channels*, capture information from the entire input sequence.
- We identify that the inability of global channels to capture global information when exposed to much longer sequences than they were originally trained on, due to exponential hidden state decay with respect to context length, is a critical bottleneck in Mamba's performance on long-context tasks. This insight enhances our understanding of Mamba models and may inspire future innovations in achieving extended context capabilities for Mamba models or other SSMs.
- Building upon our findings, we propose LongMamba, a training-free method that can enhance the receptive fields of *global channels* when the sequence length exceeds the training sequence. Our analysis demonstrates that analytically identifying and removing less important tokens in *global channels* can adaptively alleviate the exponential decay of hidden states—intensifying with increased context length—and thus significantly expand these channels' receptive fields.
- Through comprehensive benchmarking on both synthetic and real-world long-context tasks, we demonstrate that LongMamba significantly extends the operational range of pre-trained Mamba models, outperforming previous methods aimed at enhancing Mamba's context handling capabilities. For instance, on the language modeling task, our method maintains the same level of perplexity at a context length more than 25$\times$ that of the vanilla Mamba model and 10$\times$ longer than the previous methods.

2 Related Works

State Space Models (SSMs). SSMs provide a framework for representing dynamic systems through a temporal sequence of latent states, where the system's output is derived from these states (Durbin et al., 2012). In the realm of deep learning, SSMs have emerged as a promising alternative to Transformer-based architectures for sequential data processing. Initial efforts to integrate SSMs into deep learning architectures encountered significant obstacles, such as stability issues during training. The Structured State Space Sequence model (S4) (Gu et al., 2022b) marks a pivotal advancement in addressing these challenges, enabling the stable training of SSMs in deep neural networks. However, early deep SSM implementations still lack a crucial feature inherent to attention mechanisms: input-dependent information selection. Mamba (Gu & Dao, 2023) addresses this limitation by introducing selective SSM layers with input-dependent update mechanisms. Subsequent iterations, e.g., Mamba-2 (Dao & Gu, 2024b), further refined this approach, demonstrating competitive performance as compared to Transformers. However, the difficulty of effectively handling very long-range dependencies in SSM-based models like Mamba remains a key challenge in modern language modeling, particularly when processing extended contexts beyond their initial training lengths (Ben-Kish et al., 2024; Waleffe et al., 2024).

Mamba Models. The Mamba architecture's efficiency and potential drive its adaptation across diverse applications. In computer vision, Vim (Zhu et al., 2024) uses bidirectional state space modeling for managing long-range dependencies in images, while VMamba (Liu et al., 2024b) enhances selective SSMs with novel scanning algorithms for better information flow. DiM (Teng et al., 2024) customizes Mamba for high-resolution image diffusion. The need for extended context modeling in video, point cloud, and graph sequences boosts the demand for Mamba solutions, spurring further research. VideoMamba (Li et al., 2024) and Graph-Mamba (Wang et al., 2024a) exemplify this by applying Mamba's long temporal and spatial sequence handling. Ongoing advancements and applications further fuel the demand for Mamba's long-context capabilities. Hybrid Mamba-Attention models like Jamba (Lieber et al., 2024) and Zamba(Glorioso et al., 2024) attempt to combine the benefits of attention mechanisms with Mamba's efficiency in long-range modeling (Lieber et al., 2024; Glorioso et al., 2024).

Language Models for Long Context. Language models trained on length-limited contexts often experience performance degradation when extrapolated to longer sequences. Previous research attempted to address this problem through various approaches, including positional interpolation (Peng et al., 2024; Wang et al., 2024b), improvements to the attention mechanism (Xiao et al., 2024b; Yao et al., 2024), and external memory integration (Xiao et al., 2024a; Bulatov et al., 2022). Despite these advancements, such transformer-based solutions frequently encounter computational and memory constraints as context lengths increase significantly. Furthermore, these methods cannot be directly applied to Mamba models due to the fundamental architectural differences between transformers and SSMs, particularly the absence of explicit attention mechanisms in Mamba's recurrent structure. To close this gap, DeciMamba (Ben-Kish et al., 2024) is among the first to explore context-extension capabilities in Mamba models. Specifically, DeciMamba employs a token pruning mechanism that progressively reduces sequence length in deeper layers by selectively removing less critical tokens, with empirically determined pruning ratios that vary across datasets and tasks. In contrast, our approach thoroughly analyzes the root cause of Mamba models' limitations in handling extended sequences and addresses this through a principled method applicable across various tasks. Consequently, our method eliminates the need for meticulous layer-specific adjustments and consistently surpasses DeciMamba in performance across diverse benchmarks.

3 Preliminaries of Mamba Models

In this section, we provide the background of the Mamba algorithm and review the previous efforts (Ali et al., 2024; Ben-Kish et al., 2024) in measuring the attention score and receptive field of Mamba models, which lays the groundwork for our analysis in Sec. 4.

Mamba Algorithm. Mamba models (Gu & Dao, 2023) are built by stacking multiple Mamba blocks that performs sequence-to-sequence mapping. Given an input sequence of L tokens $I \in \mathbb{R}^{L \times d_m}$ (d_m is the input channel dimension), a Mamba block maps the input sequence to output sequence $O \in \mathbb{R}^{L \times d_m}$ through the following computation:

$$X = \sigma(\text{Conv1D}(\text{Linear}_1(I))) \in \mathbb{R}^{L \times d_e} \tag{1}$$

$$Y = \text{SSM}(X) \in \mathbb{R}^{L \times d_e} \tag{2}$$

$$O = \text{Linear}_3(\sigma(\text{Linear}_2(I)) \odot Y) \in \mathbb{R}^{L \times d_m} \tag{3}$$

where Linear_1, Linear_2 and Linear_3 are regular linear projections, Conv1D is a 1D causal convolution with a causal mask, σ is an activation function, and $\odot$ represents element-wise product. SSM is a state-space machine that performs a recurrent computation on the input sequence $X = (x_1, x_2, ..., x_L) \in \mathbb{R}^{L \times d_e}$ (d_e is the output dimension of Linear_1):

$$H_t = \bar{A}_t \odot H_{t-1} + \bar{B}_t \odot X_t \in \mathbb{R}^{d_e \times d_s} \tag{4}$$

$$Y_t = H_t C_t \in \mathbb{R}^{d_e} \tag{5}$$

where $H_t \in \mathbb{R}^{d_e \times d_s}$ is the hidden state at time step t, $\bar{A}_t \in (0,1)^{d_e \times d_s}$ is a decay factor on the hidden state, $\bar{B}_t \in \mathbb{R}^{d_e \times d_s}$ determines the hidden state update at step t, and $C_t \in \mathbb{R}^{d_s}$ is a per-channel output scaling factor.

The key innovation of Mamba is making $\bar{A}_t$, $\bar{B}_t$ and C_t time variant (i.e. predicted from the input token x_t). Specifically:

$$\Delta_t = \text{Softplus}(X_t), \qquad B_t, C_t = \text{Linear}_4(X_t) \in \mathbb{R}^{d_s} \times \mathbb{R}^{d_s} \tag{6}$$

$$\bar{A}_t = \exp(\Delta_t \odot A), \qquad \bar{B}_t = \Delta_t \otimes B_t \tag{7}$$

where $\Delta_t \in \mathbb{R}^{d_e}_{>0}$ is a per-channel positive factor, while $A \in \mathbb{R}^{d_e \times d_s}_{<0}$ is a negative learnable matrix, which makes the hidden state decay factor $\bar{A}_t$ always smaller than 1 (i.e. continously decaying the previous hidden state H_{t-1}). Finally, $\otimes$ represents outer product.

Attention Score of Mamba based SSMs. We can quantify the contribution of a token X_j to the hidden state at time step i by expanding the recurrent computation in Eq. 4 across timesteps:

$$H_i = \Sigma^i_{j=0}(\Pi^i_{k=j+1}\bar{A}_k) \odot \bar{B}_j \odot X_j \tag{8}$$

therefore for the output at time step i:

$$Y_i = C_i \odot \Sigma^i_{j=0}(\Pi^i_{k=j+1}\bar{A}_k) \odot \bar{B}_j \odot X_j \tag{9}$$

$$= \Sigma^i_{j=0}\alpha_{i,j} \odot X_j \tag{10}$$

where

$$\alpha_{i,j} = C_i \odot (\Pi^i_{k=j+1}\bar{A}_k) \odot \bar{B}_j \tag{11}$$

is the weighting factor of the contribution of the j-th token's input X_j to the i-th tokens output Y_i, which has a similar function as the attention score in a Transformer model. Therefore, (Ali et al., 2024) proposes to regard $\alpha_{i,j}$ as the attention score between token i and j.

Receptive Field of Mamba. The above attention score metric opens up the opportunity to quantitatively measure the context length that a token could attend to (i.e. the receptive field of a token). DeciMamba (Ben-Kish et al., 2024) proposes to measure the effective receptive field (ERF) per layer using a metric dubbed Mamba Mean Distance that measures the average distance between tokens weighted by their attention score. For the last token (L-th) in a sequence, the mathematical formulation of ERF is as follows:

$$\mathbb{E}[d(j, L)] = \sum_{j \le L}(L - j)\mathbb{N}(\alpha)_{j,L} \tag{12}$$

where $\mathbb{N}(\alpha)_{j,L}$ is the normalized attention score:

$$\mathbb{N}(\alpha)_{j,L} = \frac{|\alpha_{j,L}|}{\sum^L_{i=1}|\alpha_{i,L}|} \tag{13}$$

DeciMamba (Ben-Kish et al., 2024) averages the attention score on all d hidden state channels and measures the averaged ERF on each layer. In Sec. 4 we show that different channels have distinct ERFs, which suggests opportunities for more detailed analysis of Mamba models' performance with long input sequences.

4 Motivating Analysis

To understand the limited context length extrapolation capability of Mamba models, such as their failure to retrieve past keys (Ben-Kish et al., 2024) or look up entries in a PhoneBook (Waleffe et al., 2024) from contexts longer than the training sequence, we first conduct a per-channel receptive field characterization using the training sequence length in Sec. 4.1. We then extend the experiment to a longer sequence length and analyze the challenges of adapting Mamba for extended contexts in Sec. 4.2.

4.1 Per-Channel Receptive Field Characterization in Vanilla Mamba

Fig. 1 (a) illustrates the attention map and ERF (Effective Receptive Field, marked by red lines) of five sampled channels in a 130M Mamba model, pre-trained on a sequence of 2,000 tokens. Our empirical analysis reveals that while some channels exhibit a local receptive field, others extend their

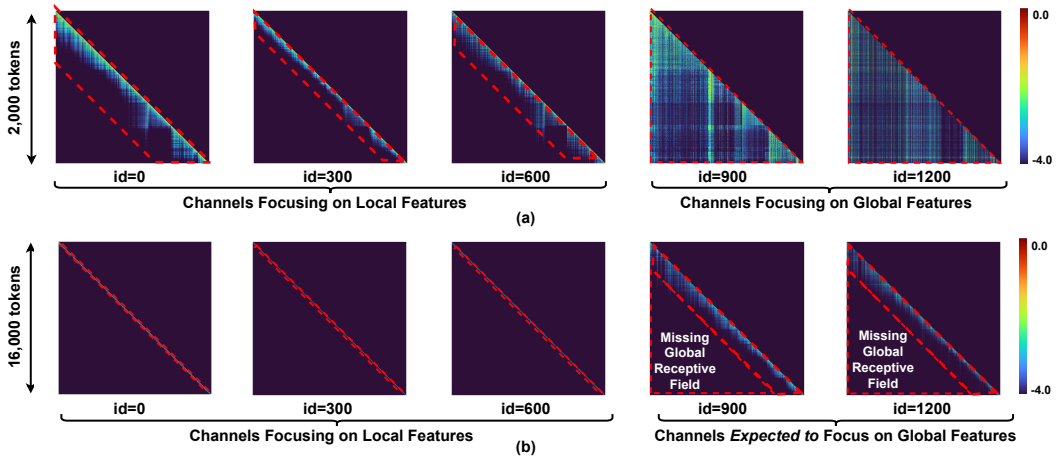

Figure 1: Visualization of the Mamba attention map (log scale) under the (a) training sequence length (2,000 tokens) and (b) extended sequence length (16,000 tokens). We uniformly sampled five hidden state channels in the 12-th layer of a Mamba-130M model and selected a sequence from The Pile (Gao et al., 2020) dataset, which serves as Mamba's training dataset. Here, the red lines delineate the effective receptive field of each channel, and "id" denotes the channel index in the model weights, where we select three/two local/global channels to illustrate their concepts.

receptive field to the full sequence length. More visualization on the per-channel receptive fields is available in the appendix B.4, which aligns with our observation in Fig. 1 (a). Prior research, such as that by (Ben-Kish et al., 2024), has advocated measuring the average receptive field per layer; however, due to the variability in receptive field lengths across channels, a per-channel characterization is necessary. Accordingly, we categorize the channels within each Mamba layer into two classes: those whose ERFs match the training context length and those that do not.

Local Channels. The channels with ERFs significantly shorter than the training context length (e.g., the 0-th, 300-th, and 600-th channels in Fig. 1 (a)), inside which each token only attends to a local context window. Their attention pattern suggests that these channels function like a convolution layer or a sliding window attention (Beltagy et al., 2020) that captures localized information.

Global Channels. The channels whose ERFs are comparable to the training context length (e.g., the 900-th and 1200-th channels in Fig. 1 (a)), so that at these channels a token could attend to almost any previous token within the sequence. This means that during training, the global channels learn to capture information globally from the entire sequence.

4.2 Why Mamba fails in Extended Contexts?

After characterizing the distinct ERFs (Effective Receptive Fields) of different hidden state channels, we next investigate their attention patterns at extended context lengths (e.g., 16,000 tokens by a Mamba model pretrained on a sequence of 2,000 tokens). It can be observed from Fig. 1 (a) that although the global channels (e.g., the 900th and 1200th channels) have an ERF comparable to the training context length, they fail to extend their ERF to 16,000 tokens (see their corresponding ERF in Fig. 1 (b)), rendering them unable to capture global information from a 16,000-token sequence. This observation is consistent with our visualization of more channels in appendix B.4.

To understand this failure, we analyze the mechanism of hidden state updates in Mamba models. Specifically, we find that the term $\Pi_{k=j+1}^{i}\bar{A}_k$ in Eq. 8 exhibits exponential decay with growing context length. Take an extreme case as an example, for the hidden state H_0 at the initial timestep, it suffers from the following cumulative decay at the end of the sequence:

$$\Pi_{k=1}^{L}\bar{A}_k = \exp\left(\left(\sum_{k=1}^{L}\Delta_k\right)\odot A\right). \tag{14}$$

where A is a negative matrix. As a result, the expression above exponentially decays towards zero as the sequence length L increases. Similarly, such significant decay prevents the *global channels* of

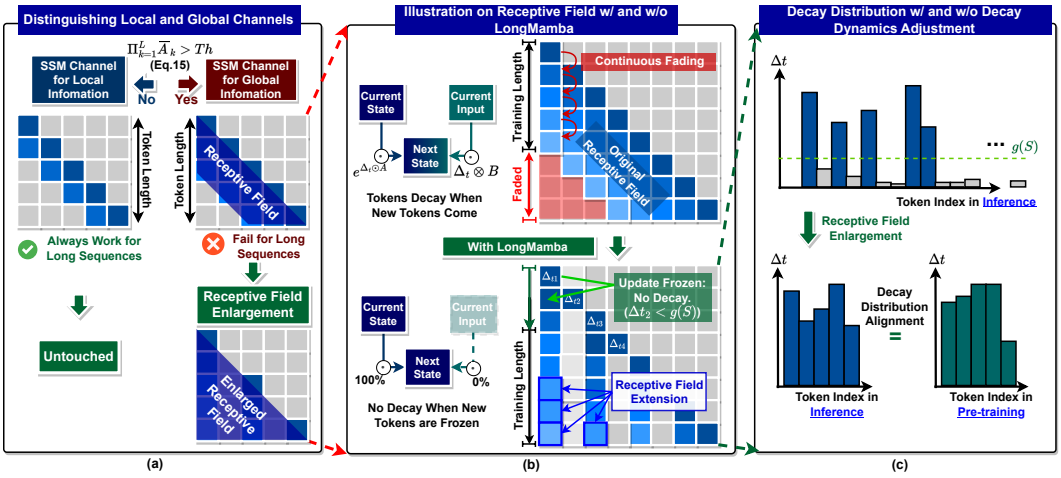

Figure 2: Overview of the LongMamba framework. (a) We first distinguish the *global* and *local* channels by measuring their cumulative state decay; (b) for the global channels, we enlarge their receptive fields by freezing the update of less important tokens, as detailed in Sec. 5.2; (c) to maintain a similar decay distribution after enlarging the receptive fields compared to the distribution during the training process, we scale each unfrozen decay value to align the overall distribution with that of the training phase.

Mamba models from preserving global information in their hidden states, when the sequence length L significantly exceeds the training context length.

5 LongMamba for Receptive Field Enlargement

Following our analysis in Sec. 4.2, we propose LongMamba, a training-free technique to enhance Mamba models' long-context understanding capabilities. As shown in Fig. 2, LongMamba first categorizes each hidden state channel into either *global channel* or *local channel* by computing the learned cumulative decay at training length. Then for the *global channels*, we alleviate expoential hidden state decay by removing less important tokens in their corresponding states. We detail the two steps in Sec. 5.1 and Sec. 5.2, respectively.

5.1 LongMamba: Identifying Global Channels

As analyzed in Sec. 4, it is the failure of the *global channels* to attend to global information from extended context lengths that leads to difficulties in modeling long contexts with Mamba, while the *local channels* perform as expected under these conditions. This motivates us to extend the context length only on the *global channels*, while keeping the *local channels* untouched.

To identify the *global channels*, we measure the cumulative decay of each channel on the training sequence length. Specifically, if the cumulative decay factor of a channel is larger than an empirical threshold:

$$\Pi_{k=1}^{L}\bar{A}_k > Th, \tag{15}$$

which means that if a channel suffers from a hidden state decay no stronger than Th at the training length L, we classify this channel as a global channel.

5.2 LongMamba: Receptive Field Enlargement Through Token Filtering

For the identified *global channels*, our goal is to enlarge their receptive field to ensure their receptive field can extend to match the entire input sequence, when the input sequence length S is larger than the training sequence length L. This can be achieved by aligning the cumulative decay at the input

sequence length with the learned cumulative decay at the training sequence length:

$$\Pi_{i=1}^{S} \bar{A}'_i \approx \Pi_{i=0}^{L} \bar{A}_i, \tag{16}$$

where $\bar{A}'_i$ is the decay factor after applying our technique.

In order to achieve this goal, we propose to filter out token t for state update when Δ_t is smaller than a certain threshold g, as illustrated in Fig. 2 (b), i.e. when $\Delta_t < g$, we set:

$$\bar{A}'_t = 1, \bar{B}'_t = 0, \tag{17}$$

so that $H_t = H_{t-1}$. For the rest of the tokens with $\Delta_t \geq g$, $\bar{A}'_t = \bar{A}_t$ and $\bar{B}'_t = \bar{B}_t$.

In order to satisfy Eq. 16, we set $g = g(S)$, where $g(S)$ is a per-channel look-up table that takes sequence length S as input, s.t. if Δ_t values of the S tokens are uniformly sampled from the distribution of training sequence, then:

$$\Pi_{\{i:\Delta_i > g(S)\}} \bar{A}_i \approx \Pi_{i=0}^{L} \bar{A}_i, \tag{18}$$

Throughout our experiments, the distribution of Δ_t is established by a calibration process on the training dataset.

6 Experiments

In this section, we evaluate our proposed LongMamba across various tasks to assess its ability to understand long contexts. Specifically, our benchmark tasks include Passkey Retrieval, Document Retrieval, Language Modeling, and LongBench, where we compare LongMamba with state-of-the-art (SOTA) methods, as detailed in Sec. 6.2. Additionally, in Sec. 6.3, we conduct ablation studies to explore how different design strategies (e.g., changing channel selection strategy as described in Sec. 5.1) impact its capabilities for understanding and processing extended long sequences.

6.1 Experiment Settings

Datasets. For the Passkey Retrieval, Document Retrieval, and Language Modeling tasks, we follow (Ben-Kish et al., 2024) and use WikiText (Merity et al., 2016), SQuAD (Rajpurkar, 2016), and PG-19 (Rae et al., 2019), respectively, as the datasets. For evaluation on LongBench Bai et al. (2023), which is commonly used to measure language models' long-context understanding capability, we follow the settings in (Wang et al., 2024c; Liu et al., 2024a), using the HotpotQA, Qasper, SAMSum, and VSSUM datasets in LongBench Bai et al. (2023).

Baselines. We include the fine-tuned Mamba model (Gu & Dao, 2023) and DeciMamba Ben-Kish et al. (2024) as baselines. It is worth noting that DeciMamba requires external fine-tuning on the corresponding tasks after applying its decimation technique, while our proposed LongMamba is training-free. All baseline implementations follow the corresponding papers and official codebases.

Implementation Details of Our Model. Our proposed LongMamba is implemented based on the official Mamba codebase[1] and uses checkpoints from the Huggingface Model Hub[2].we adopt Mamba models of various sizes (e.g. Mamba-130M, Mamba-1.4B, and Mamba-2.8B) for the experiments. Specifically, we set the hyperparameters as follows: for **all layers**, we define the *global channels* as the channels whose cumulative decay at sampled training sequence $\Pi_{k=1}^{2K} \bar{A}_k > Th =$ **1e-30**, which is suggested by our ablation study detailed in Sec. 6.3. We randomly sample 10 training sequences from The Pile (Gao et al., 2020) dataset for measuring the per-channel cumulative decay and establish the look-up table $g(S)$. We create a look-up entry in $g(S)$ for every 1,000-unit increase in context length (e.g., at 1,000, 2,000, 3,000, and so on).

6.2 Benchmarking Mamba-based models on Long Contexts

To comprehensively benchmark the capabilities of our proposed LongMamba on long-context inputs, we first conduct long-range language retrieval tasks, followed by demonstrating LongMamba's superiority in more complex language modeling tasks. The detailed results across four tasks are summarized below.

[1] https://github.com/state-spaces/mamba

[2] https://huggingface.co/state-spaces

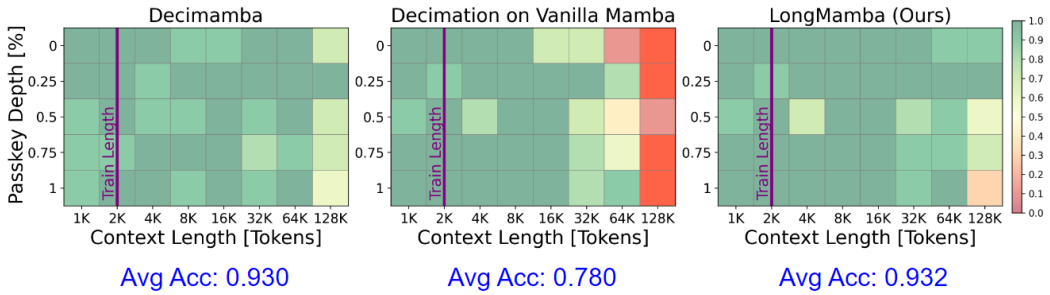

Figure 3: The accuracies of DeciMamba **w/** fine-tuning on the target task, Decimation on Vanilla Mamba (i.e. directly applying DeciMamba on vanilla Mamba's checkpoint **w/o** fine-tuning), and our proposed LongMamba (also directly applied on vanilla Mamba's checkpoint **w/o** fine-tuning) on the Passkey Retrieval task.

Passkey Retrieval. For the Passkey Retrieval task, we implemented LongMamba on top of the officially fine-tuned Mamba-130M model (Ben-Kish et al., 2024) to retrieve a random 5-digit code hidden at various locations within a contiguous 2K-token sample from WikiText (Merity et al., 2016). To evaluate the model's ability to extrapolate to longer contexts, we progressively increased the sequence lengths from 1K to 128K tokens during inference, assessing its performance across various passkey locations. The results presented in Fig. 3 show that our proposed LongMamba performs better than Decimation on Vanilla Mamba (i.e., 15.2% higher average accuracy) and even slightly better than DeciMamba (i.e., 0.2% higher average accuracy). It is worth noting that DeciMamba is **fine-tuned** on WikiText dataset while our proposed LongMamba is a **training-free** solution. Furthermore, while the Mamba-130M model, fine-tuned on this task with 2K input contexts, demonstrated a diminishing success rate with longer sequences and completely failed at 128K, LongMamba successfully extended its inference ability without any additional training or parameter modifications. Unlike DeciMamba, which requires task-specific fine-tuning with decimation enabled, LongMamba allows any Mamba model already fine-tuned on the task to easily adapt to much longer contexts.

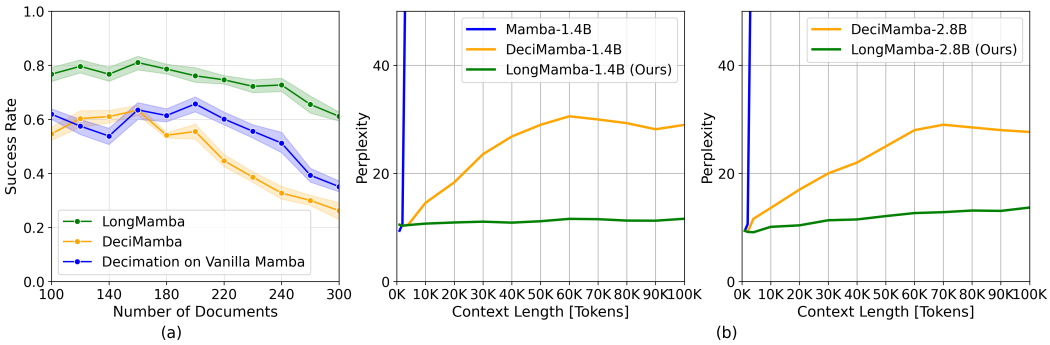

Figure 4: Comparing (a) the success rates on the Document Retrieval task and (b) the perplexity on the Language Modeling task of the vanilla Mamba, DeciMamba, and our proposed LongMamba.

Document Retrieval. Besides the aforementioned Passkey Retrieval task, we increase the task difficulty from capturing special codes to understanding complex text across multiple documents. In this task, the model is provided with a query and N_n randomly selected noisy documents, one of which is a golden document containing the correct answer. The model must identify the ID of the golden document. To evaluate our method, we use a Mamba-130M model fine-tuned on the document retrieval task using the SQuAD (Rajpurkar, 2016) dataset with $N_n = 10$, corresponding to approximately 2K tokens. We then equip the model with LongMamba and evaluate its performance across varying $N_n \in [10, 300]$ (about 2K to 60K tokens). The results, presented in Fig. 4 (a), show the success rate, measuring how often the model correctly identifies the golden document ID, across 100 random queries for each document set size, with 10 independent evaluation iterations. Although the success rate gradually declines as the number of documents and tokens increases, the Mamba-130M model equipped with LongMamba consistently outperforms both DeciMamba and the Mamba-130M model using DeciMamba's decimation technique in all iterations, achieving a success rate consistently 10% higher than the baselines.

Table 1: The results on the LongBench tasks with 1.4B Mamba models. For DeciMamba, we use its hyperparameter settings for Language Modeling tasks.

Datasets	**HotpotQA**	**Qasper**	**SAMSum**	**VCSUM**
Metrics	F1		Rouge-L	
Vanilla Mamba	4.68	13.6	3.7	2.13
DeciMamba	13.88	14.24	8.57	1.4
LongMamba	**16.27**	**14.76**	**9.16**	**2.55**

Language Modeling. We further evaluate the performance of the proposed DeciMamba at a language modeling task.

Compared to the retrieval tasks, sophisticated long-context language modeling challenges models in a more refined way. We evaluate our method on long-range language modeling under zero-shot setting, where we test LongMamba on the test set of the PG-19 dataset using our technique on top of pre-trained Mamba-1.4B and 2.8B models. The results in Fig. 4(b) show that LongMamba substantially outperforms the vanilla Mamba model and another previous SOTA method, DeciMamba, across all context lengths. We attribute this to the difference in how our method and DeciMamba deal with less important tokens. DeciMamba aggressively drops them in the middle of inference, and the token is no longer processed in the following layers. In contrast, LongMamba suppress these tokens by setting their Δ_t to 0, which enables the model to skip its information in the current layer but allows it to be processed by later layers.

LongBench Tasks. To further explore LongMamba's ability to understand long contexts in real-world tasks, we conducted zero-shot experiments on four tasks from LongBench (Bai et al., 2023)[3]. Specifically, we selected the following datasets: **HotpotQA** (which involves answering related questions based on multiple provided documents), **Qasper** (a question answering task on NLP research papers), **SAMSum** (a dialogue summarization task), and **VCSUM** (a Chinese meeting summarization task). The average sequence lengths for these datasets are 4k, 6k, 9k, and 16k tokens, respectively, providing a comprehensive assessment of performance across various sequence lengths in LongBench. We conducted experiments on these tasks using vanilla Mamba-1.4B, DeciMamba, and our LongMamba, all under a zero-shot setting (w/o finetuning). We adopted the same evaluation metrics as those used in LongBench for these datasets, namely F1 score and Rouge-L. The results presented in Tab. 1 show that our method outperforms others across all datasets, demonstrating its efficacy on real-world long-context tasks.

6.3 Ablation Studies: What Enables Long-Context Understanding?

We present the results of an ablation study on our method to better understand what makes it powerful. Specifically, we focus on the three hyper-parameters of LongMamba, namely *Channel Selection Method*, *Layer Selection Method*, and *Choice of Threshold*. All ablation experiments have been conducted on the Passkey Retrieval task.

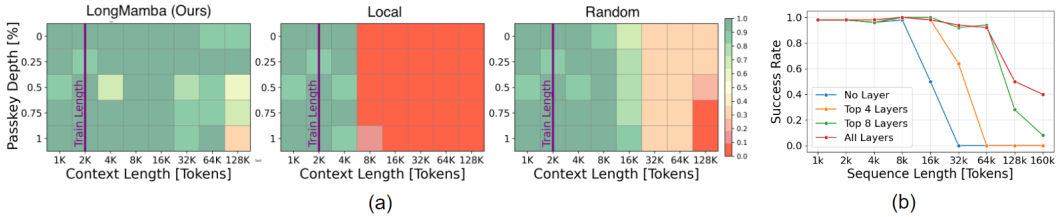

Figure 5: Success rates of our proposed LongMamba on the Passkey Retrieval task across 10 random evaluations. (a) Conducting token removal on global channels outperforms removal on local channels or random channels. (b) Applying token removal across all layers achieves better performance compared to fewer layers.

Channel Selection. We compare three possible ways to select which channels to perform the enlargement of receptive fields in LongMamba: *(a) ours*, *(b) Selection of Local Channels*, *(c) Random Channel Selection*. Recall that we define the channels whose cumulative product $\Pi_{i=1}^{2K}\bar{A}_i$ exceeds a given threshold Th as *global channels* and select them for token removal. To validate that *global*

[3] https://github.com/THUDM/LongBench

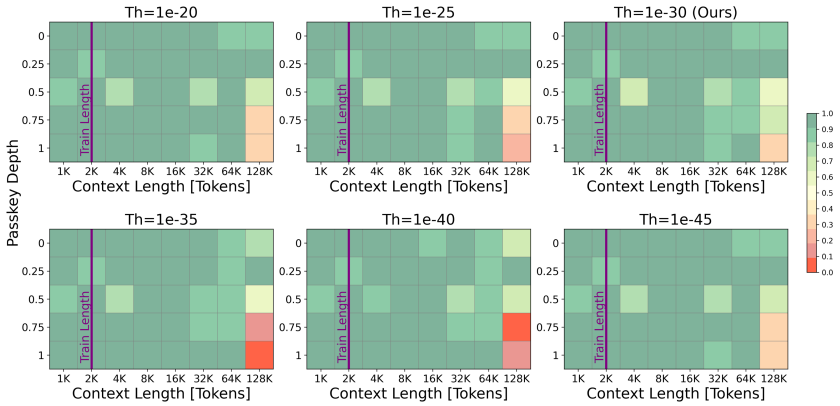

Figure 6: Success rates of our proposed LongMamba on the Passkey Retrieval task across different thresholds (Th).

channels are indeed the correct channels to apply token filtering, we compare LongMamba against the performance when we select *local channels* or *random channels* to conduct token filtering. Here, we use the same threshold to define *local channels* and select those whose cumulative products are smaller than 1e-30 (i.e., Th =1e-30). For random selection, we ensure that the number of channels selected is identical to LongMamba. The average success rates over 10 iterations across varying context lengths and passkey depth for these methods are illustrated in Fig. 5(a). While applying token filtering to *global channels* was able to achieve high success rates over all settings, selection of *random channels* failed in sequences longer than 32k tokens, and selecting *local channels* failed in almost every iteration for sequences longer than 4k. This shows that identifying the *global channels* is one of the key components to the success of LongMamba.

Layer Selection. Next, we validate whether selecting *global channels* and conducting token filtering is necessary across all mamba layers. To this end, we compare against applying this method to only a few layers. These layers are chosen based on the average cumulative product $\Pi_{i=1}^{2K}\bar{A}_i$ across all channels, since a layer with a larger average value will contain more *global channels* that have to be modulated to boost model performance. As can be observed in Fig. 5(b), applying our method only on the top 4 or top 8 layers with the highest average cumulative product fails to match the performance of applying it to all layers. While some layers may contain more *global channels*, attending to these channels in all layers is critical to extending Mamba's performance to long contexts.

Threshold Choice. Finally, we evaluate the performance of LongMamba across multiple thresholds, $Th \in \{1e-20, 1e-25, 1e-30, 1e-35, 1e-40\}$. The results in Fig. 6 illustrate that, under varying thresholds, LongMamba performs comparably on the Passkey Retrieval task to the default setting of $1e-30$ shown in Fig. 5(a). While LongMamba demonstrates robustness across a wide range of Th, a drop in success rate is observed with longer sequences, particularly with the longest context length of 128k. Therefore, we select $Th = 1e-30$ as the default setting due to its highest accuracy in the extreme 128k sequence settings.

7 Conclusion

In this paper, we present LongMamba, a training-free method designed to enhance the capabilities of Mamba SSMs for long-context tasks. Our approach builds on several critical findings: Firstly, we discover that hidden state channels in Mamba models exhibit distinct receptive field lengths; *local channels* focus on nearby contexts, while *global channels* engage with the entire input sequence. Secondly, we identify that the inability of *global channels* to handle global information from sequences longer than their training length—due to exponential hidden state decay—is a significant limitation for Mamba's effectiveness in long-context tasks. Lastly, our analysis demonstrates that analytically identifying and removing less important tokens in *global channels* can adaptively alleviate the exponential decay of hidden states, which intensifies with increased context length, and thus significantly expand these channels' receptive fields. Through extensive benchmarking across synthetic and real-world long-context scenarios, LongMamba sets a new standard for state-of-the-art performance in Mamba-based long-context tasks. Our findings enhance our understanding of Mamba and might inspire new innovations for long-context models.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Ameen Ali, Itamar Zimerman, and Lior Wolf. The hidden attention of mamba models, 2024.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*, 2023.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. Decimamba: Exploring the length extrapolation potential of mamba. *arXiv preprint arXiv:2406.14528*, 2024.

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 2022.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality, 2024a. URL https://arxiv.org/abs/2405.21060.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*, 2024b.

James Durbin et al. *Time series analysis by state space methods*, volume 38. Oxford University Press, 2012.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. Zamba: A compact 7b ssm hybrid model, 2024. URL https://arxiv.org/abs/2405.16712.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021.

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

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

Hanlei Jin, Yang Zhang, Dan Meng, Jun Wang, and Jinghua Tan. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *arXiv preprint arXiv:2403.02901*, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding, 2024.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid transformer-mamba language model, 2024. URL https://arxiv.org/abs/2403.19887.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. Minicache: Kv cache compression in depth dimension for large language models. *arXiv preprint arXiv:2405.14366*, 2024a.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model, 2024b. URL https://arxiv.org/abs/2401.10166.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

MistralAI. Large enough — mistral ai — frontier ai in your hands. https://mistral.ai/news/mistral-large-2407/, 2024. (Accessed on 10/01/2024).

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, Chloe Hillier, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint*, 2019. URL https://arxiv.org/abs/1911.05507.

P Rajpurkar. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Yao Teng, Yue Wu, Han Shi, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Dim: Diffusion mamba for efficient high-resolution image synthesis, 2024.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, et al. An empirical study of mamba-based language models. *arXiv preprint arXiv:2406.07887*, 2024.

Chloe Wang, Oleksii Tsepa, Jun Ma, and Bo Wang. Graph-mamba: Towards long-range graph sequence modeling with selective state spaces. *arXiv preprint arXiv:2402.00789*, 2024a.

Suyuchen Wang, Ivan Kobyzev, Peng Lu, Mehdi Rezagholizadeh, and Bang Liu. Resonance RoPE: Improving context length generalization of large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, 2024b.

Yu Wang, Xiusi Chen, Jingbo Shang, and Julian McAuley. Memoryllm: Towards self-updatable large language models. *arXiv preprint arXiv:2402.04624*, 2024c.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Song Han, and Maosong Sun. Infllm: Unveiling the intrinsic capacity of llms for understanding extremely long sequences with training-free memory. *arXiv preprint arXiv:2402.04617*, 2024a.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024b.

Yao Yao, Zuchao Li, and Hai Zhao. SirLLM: Streaming infinite retentive LLM. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Forty-first International Conference on Machine Learning*, 2024.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36: 50117–50143, 2023.

A HYPERPARAMETER TUNING

The decay threshold, Th, used for selecting global channels in Sec.5.1, is the only hyperparameter that requires tuning in the LongMamba framework. We calibrate this threshold on the Passkey Retrieval task using a grid search, with candidate values 1e-20, 1e-25, 1e-30, 1e-35, 1e-40, as illustrated in Fig.6. The optimal threshold identified (1e-30) is then reused for all other tasks. As demonstrated in Sec. 6.2, this threshold generalizes well across tasks, such as language modeling and LongBench, consistently outperforming the vanilla Mamba and DeciMamba.

B VISUALIZATION

B.1 MAMBA EFFECTIVE RECEPTIVE FIELD AT DIFFERENT SEQUENCE LENGTH

To analyze how the receptive field of each channel evolves with increasing context length, we compute the channel-wise receptive fields of the Mamba-130M model across different sequence lengths and illustrate the results in Fig. 7. The figure shows that for global channels, the receptive field grows linearly with increasing sequence length up to the training sequence length. Beyond this point, the receptive field ceases to increase linearly with increasing sequence length, limiting the model's ability to capture global information from longer sequences.

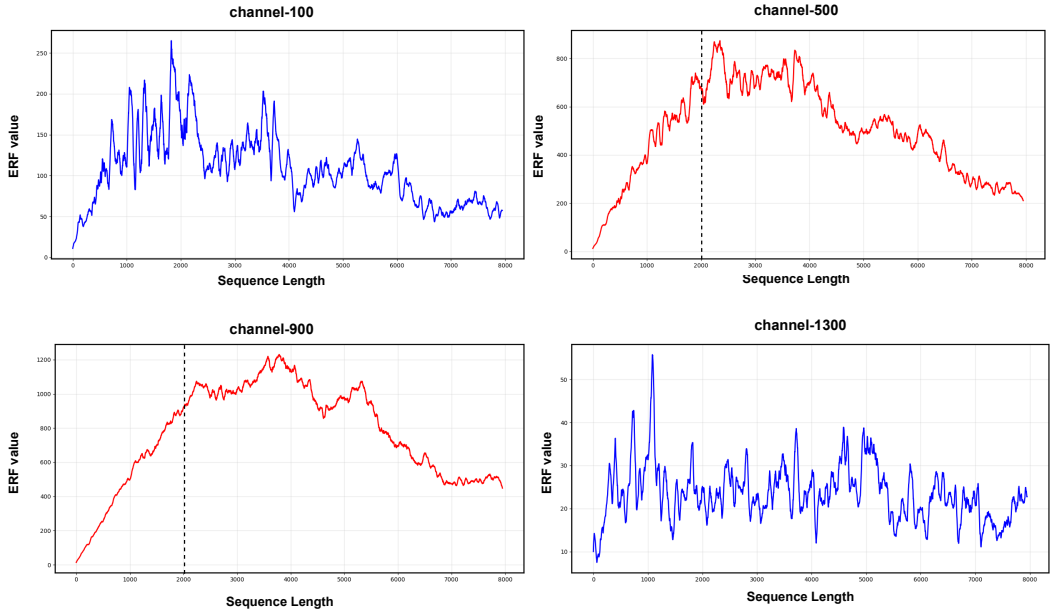

Figure 7: Evolution of Effective Receptive Field (ERF) with respect to different sequence lengths. We use 2 local channels (blue lines) and 2 global channels (red lines) from the 16-th layer of a Mamba-130m model for this visualization. The "channel-x" stands for the channel's index at the model's weight tensor, we use a sequence from the Pile dataset (Gao et al., 2020) for the illustration. Black dashed line represents the training sequence length.

B.2 HIDDEN STATE CUMULATIVE DECAY WITH AND WITHOUT LONGMAMBA

To validate the effectiveness of the proposed LongMamba in mitigating exponential hidden state decay across various context lengths, Fig. 8 illustrates the hidden state decay of global channels with and without LongMamba's decay adjustment. As shown in the figure, LongMamba (represented by the dashed curves) significantly reduces exponential hidden state decay at extended context lengths compared to the vanilla Mamba models (represented by the solid curves).

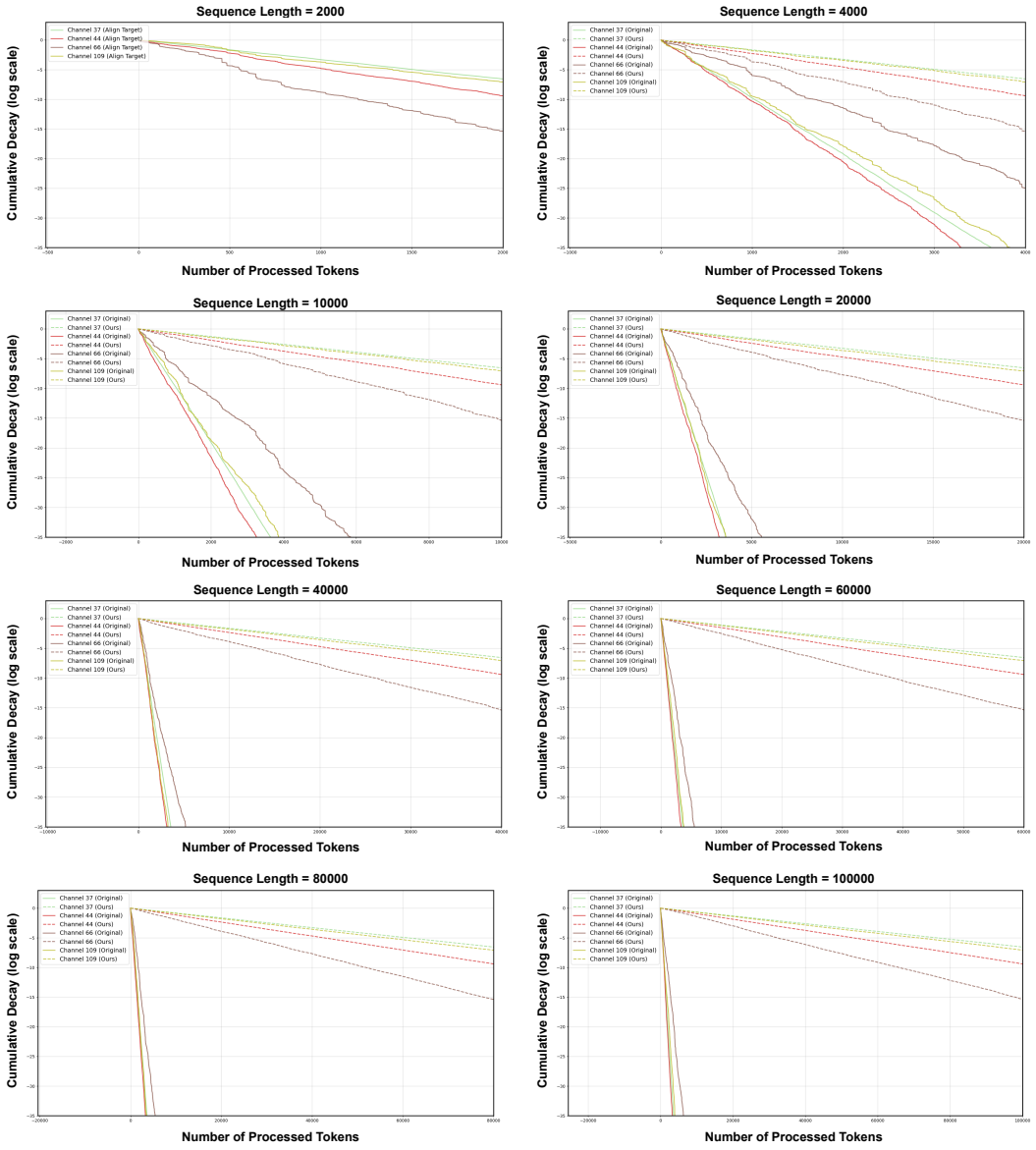

Figure 8: Cumulative decay with and without LongMamba's decay adjustment across different sequence lengths. The visualization includes four global channels from the 16-th layer of the 130M Mamba model, with input sequences sampled from the Pile dataset (Gao et al., 2020). Evaluations are conducted at the 2k training sequence length and extended sequence lengths (4k, 10k, 20k, 40k, 60k, 80k, and 100k). Each subfigure corresponds to a specific sequence length S and uses the corresponding token filtering threshold $g(S)$. The subfigures illustrate how cumulative decay evolves as the number of processed tokens increases. "Original" represents the vanilla Mamba model without LongMamba, while "Ours" refers to LongMamba. Channels are indexed according to their position in the model's weight tensor.

B.3 MAMBA-2 CUMULATIVE HIDDEN STATE DECAY VISUALIZATION

To investigate whether Mamba-2 (Dao & Gu, 2024a) models also exhibit exponential hidden state decay, we visualize how the cumulative hidden state decay, as defined in Eq.14, of a Mamba2-130M model evolves with increasing sequence length in Fig.9, 10, 11, 12, 13, 14. Specifically, each figure shows the hidden state decay evolution for the first 24 channels (indexed according to their position in the model's weight tensor). The analysis is conducted using a sample sequence from the Pile dataset (Gao et al., 2020). Red curves represent global channels, while blue curves represent local channels. For both types of channels, the cumulative decay becomes exponentially close to 0 with the increase of the sequence length.

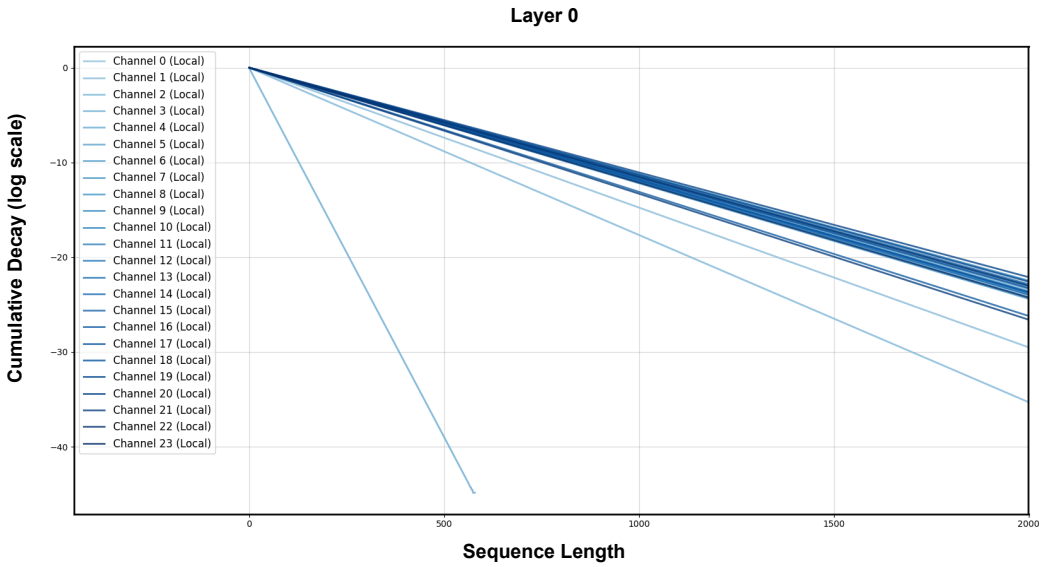

Figure 9: Cumulative hidden state decay of the Mamba2-130m model at layer 0 under varying context lengths.

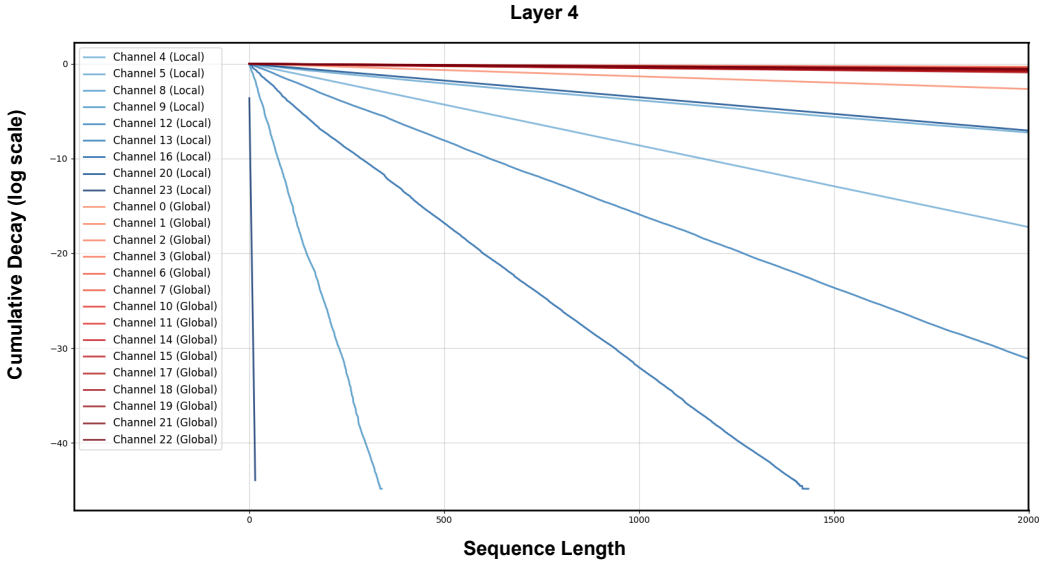

Figure 10: Cumulative hidden state decay of the Mamba2-130m model at layer 4 under varying context lengths.

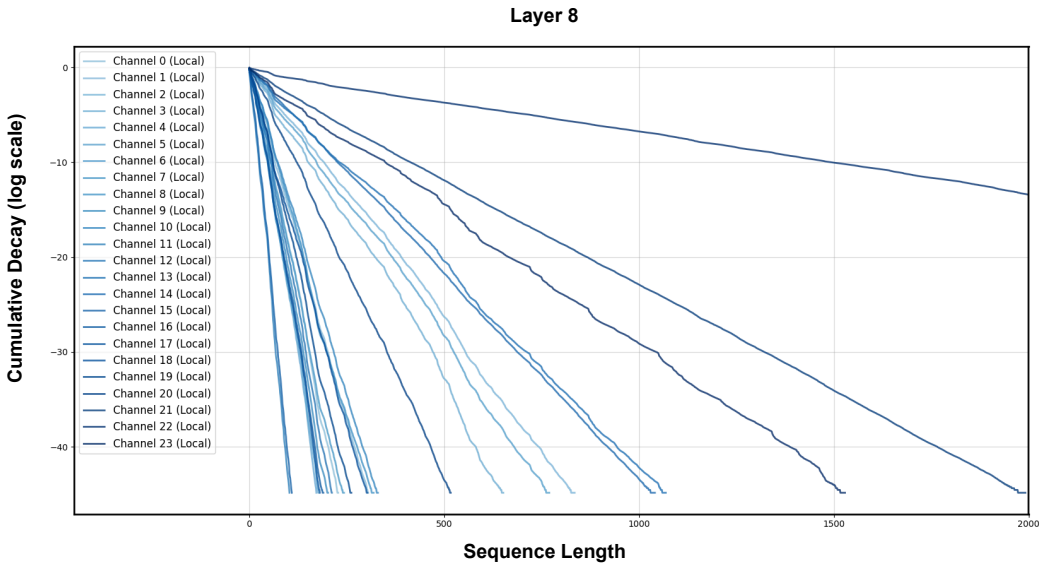

Figure 11: Cumulative hidden state decay of the Mamba2-130m model at layer 8 under varying context lengths.

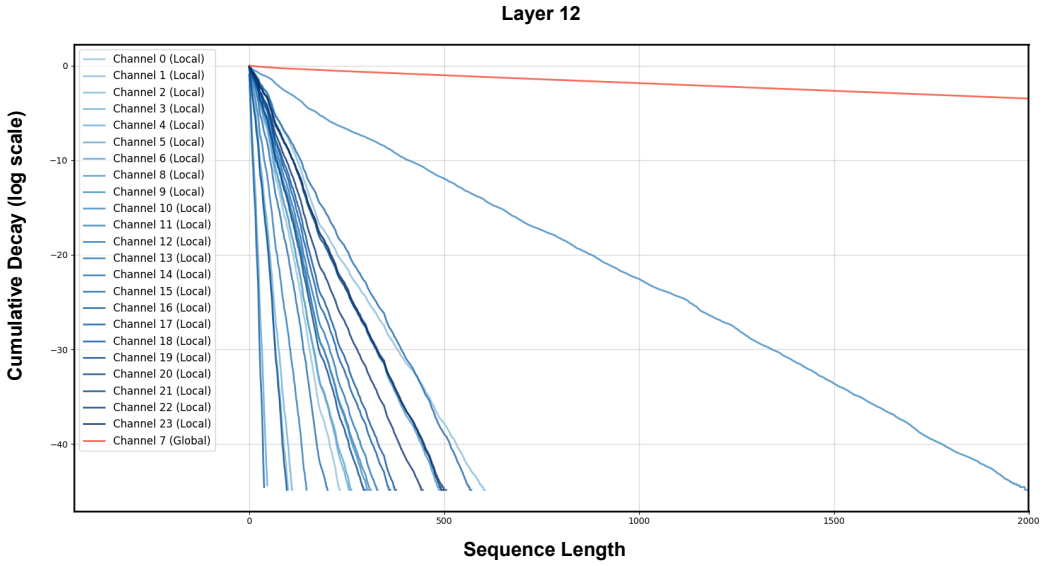

Figure 12: Cumulative hidden state decay of the Mamba2-130m model at layer 12 under varying context lengths.

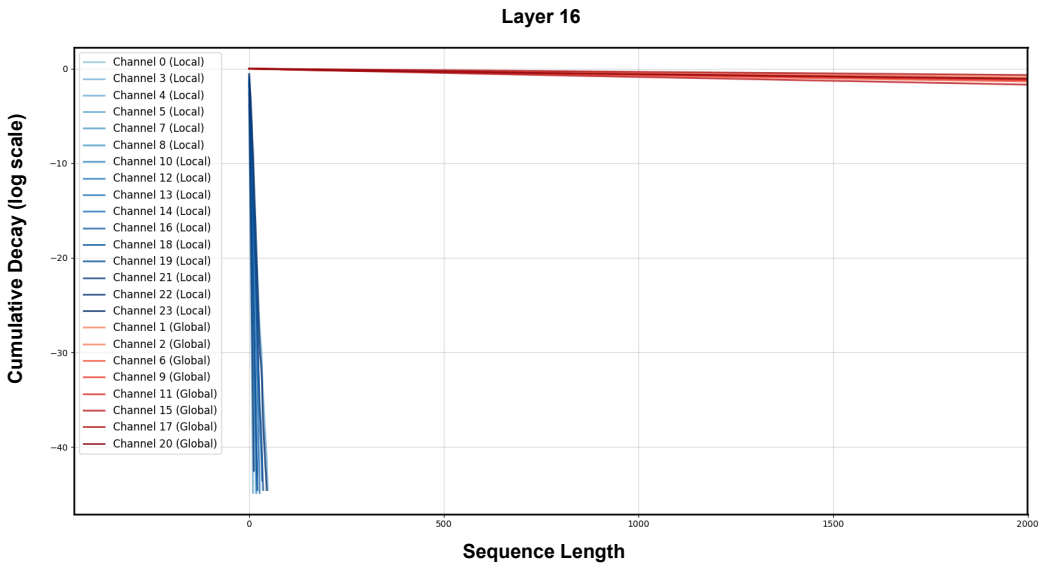

Figure 13: Cumulative hidden state decay of the Mamba2-130m model at layer 16 under varying context lengths.

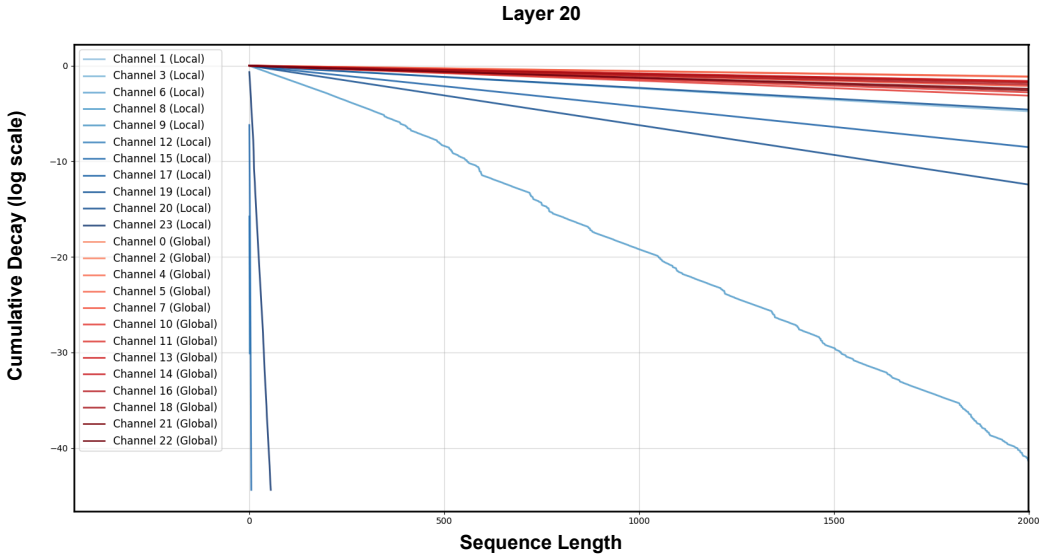

Figure 14: Cumulative hidden state decay of the Mamba2-130m model at layer 20 under varying context lengths.

B.4 More Attention Maps at Training Sequence Length and Extended Sequence Length (2,000 tokens and 16,000 tokens)

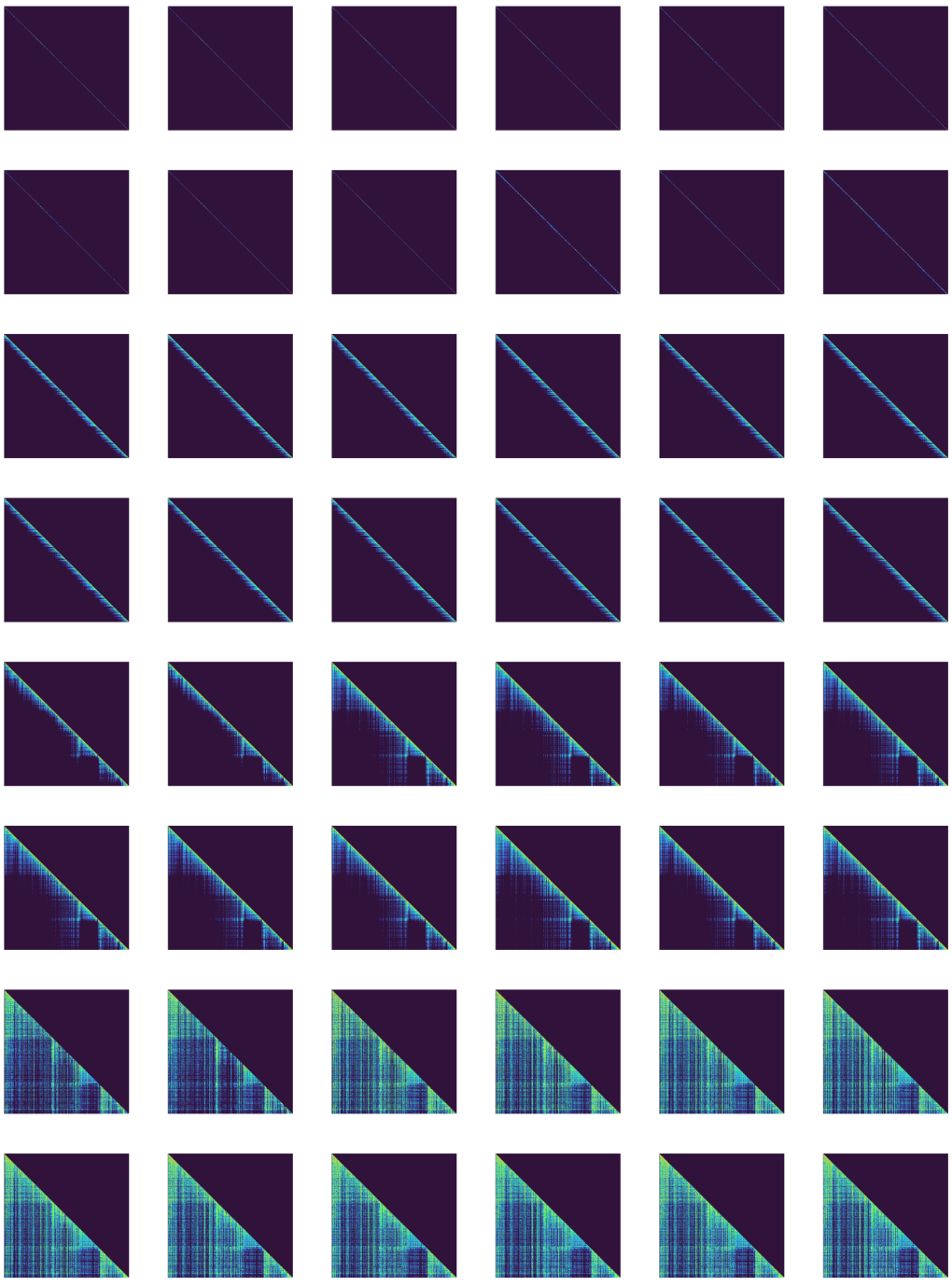

Figure 15: We use a sequence of 2,000 tokens from The Pile (Gao et al., 2020) dataset as input on the Mamba-130m, and randomly chose four layers, each with 12 channels, to visualize their attention maps. We find that each layer has both global and local channels.

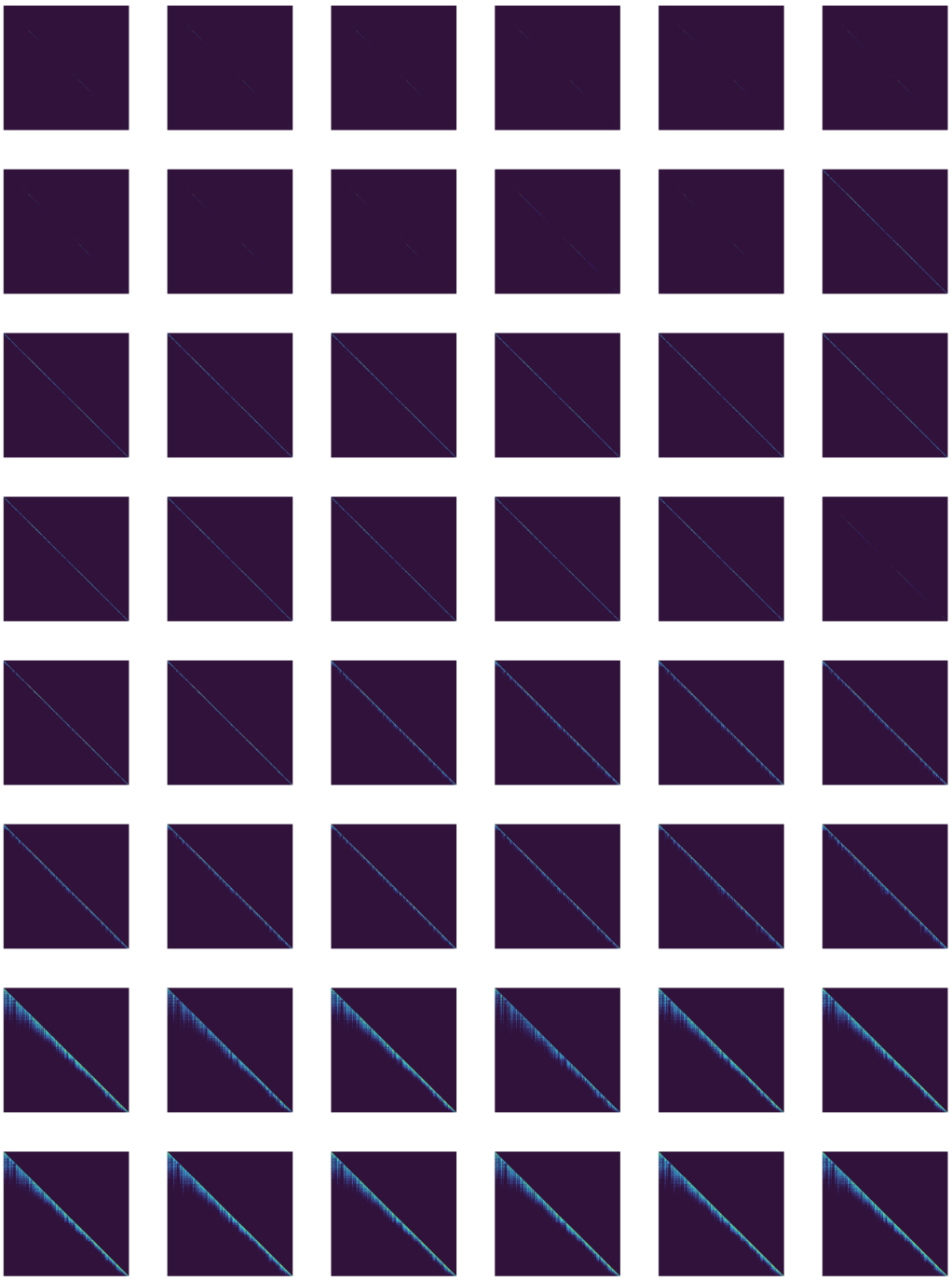

Figure 16: We maintain the same model, layer and channel configuration as previous section Fig 15. The only variation is the input token length. We use a sequence of 16,000 tokens from The Pile (Gao et al., 2020) dataset as input. We find that the corresponding global channel fails to maintain global information at this input length.