

SuPER-Mamba: Surprisal Prioritization for Extended-Context Routing in Mamba

Zhetong Zhang and Gayan K Kulatilleke*

School of Information Technology and Electrical Engineering

The University of Queensland

zhetong.zhang@student.uq.edu.au, g.kulatilleke@uq.edu.au

Abstract

State space models provide efficient linear-complexity processing for long sequences, but their performance frequently degrades as context length grows. We propose SuPER-Mamba, a training-free framework that uses Surprisal as a model-intrinsic signal for long-range state routing in pretrained Mamba models. SuPER-Mamba computes position-wise Surprisal from next-token predictions, aggregates neighboring responses through Local Pooling, and uses the resulting scores to prioritize Global-Channel updates without query-dependent scoring or additional training. We evaluate SuPER-Mamba on three state-space models across the LongBench-E benchmark. The results show that SuPER-Mamba improves the average score from 8.06 to 15.69, corresponding to a 94.7% relative gain over the Base Models and a 13.1% improvement over the state-of-the-art prior baseline. Ablation and length-wise analyses further support the effectiveness of the proposed routing framework.

Code, training scripts and datasets : <https://github.com/gayanku/SuPER-Mamba>

1 Introduction

Long-context modeling requires a model to process large amounts of input while effectively preserving and utilizing information that remains relevant over long distances. Mamba and its subsequent variants provide an efficient alternative to Transformer-based architectures for long-sequence modeling through recurrent state-space computation with linear sequence complexity (Gu and Dao, 2024; Dao and Gu, 2024). However, historical information must be continuously compressed into a finite recurrent state, making effective long-range information preservation increasingly difficult as the context grows. Existing studies have shown

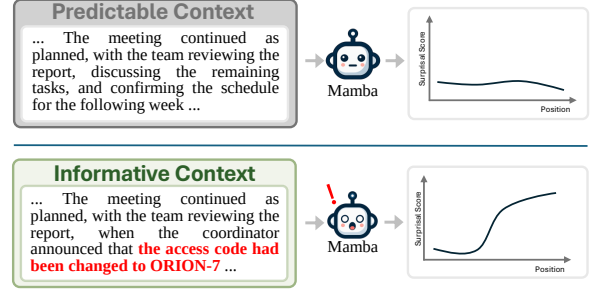


Figure 1: Motivation of SuPER-Mamba. A pretrained Mamba produces position-wise Surprisal from next-token prediction probabilities while processing the input context. Relatively predictable local content typically produces lower and more stable Surprisal responses, whereas less predictable local changes may produce larger responses. SuPER-Mamba treats these responses as model-intrinsic predictive feedback and further investigates the relationship between such local predictive feedback and signals for long-range state allocation.

that different hidden channels in Mamba exhibit different long-range retention characteristics, and that selectively controlling long-range state updates can alleviate information decay and improve long-context modeling performance (Ye et al., 2025b,a). These findings regarding Mamba’s long-context capabilities motivate the investigation of intrinsic model signals that can guide the allocation of limited long-range state updates without relying on downstream queries.

Pretrained autoregressive language models continuously produce predictive distributions over the next token while processing a context, reflecting the model’s predictive expectations at different positions. As illustrated in Figure 1, relatively predictable local content tends to produce more stable predictive responses, whereas less predictable changes may induce more pronounced Surprisal responses. The mismatch between the model’s predictive expectation and the observed next token motivates our hypothesis that these position-dependent

* Corresponding author.

predictive responses can provide a model-intrinsic signal for long-range state allocation.

Based on this hypothesis, we propose SuPER-Mamba, which uses Surprisal derived from the complete input context to quantify position-wise predictive responses and combines them with local information aggregation to construct context scores for guiding long-range state updates. Figure 2 provides an overview of the framework. SuPER-Mamba first computes position-wise Surprisal and aggregates neighboring responses through Local Pooling, then uses the resulting scores to route selected content positions to long-range state updates while preserving the complete input in the original recurrent computation. The method requires no additional training or model fine-tuning. We systematically evaluate SuPER-Mamba on 13 LongBench-E tasks across three representative state-space-based architectures. The results show that, without using the downstream query for context selection, SuPER-Mamba substantially improves the long-context performance of the corresponding Base Models while remaining competitive with state-of-the-art baselines.

Our contributions are as follows.

- We propose SuPER-Mamba, a query-independent routing framework for long-context modeling in pretrained Mamba models. The framework directly uses predictive responses produced by the pretrained model to construct context-scoring signals to guide long-range state updates.
- We first introduce Surprisal-guided routing for long-context modeling in pretrained Mamba models. SuPER-Mamba combines Surprisal with local information aggregation to transform the model’s predictive feedback into a routing signal for long-range state allocation without additional selector training.
- We evaluate SuPER-Mamba on 13 LongBench-E tasks across three representative state-space-based architectures. Without using the downstream query for context selection, SuPER-Mamba improves the average score from 8.06 to 15.69, corresponding to a 94.7% relative improvement over the Base Models, while achieving 13.1% higher average performance than the state-of-the-art baseline evaluated across the same architectures.

2 Related Work

2.1 Long-Context Memory and Routing

Mamba achieves linear-complexity sequence modeling through input-dependent state-space updates, but its fixed-size recurrent state requires growing historical information to be compressed into a bounded representation (Gu and Dao, 2024; Dao and Gu, 2024). Recent studies have examined how information is propagated and retained under this recurrent computation. The Hidden Attention of Mamba Models analyzes interactions between context positions from an attention-like perspective, while LaTIM characterizes fine-grained latent token-to-token interactions in Mamba-1 and Mamba-2 (Ali et al., 2025; Pitorro and Treviso, 2025). Other work reports selective information loss in finite recurrent representations and shows that Mamba handles locally concentrated key information more effectively than information distributed across distant positions (Hossain et al., 2025; You et al., 2025). These findings indicate that long-context performance depends not only on input length, but also on how historical information is preserved within limited recurrent states.

Recent work has improved Mamba’s long-context capability through state-dynamics modulation, selective compression, training-based memory enhancement, hierarchical historical access, and additional memory structures (Lu et al., 2025; Yuan et al., 2025; Ma et al., 2025; Hu et al., 2025; Wang et al., 2026; Lahoti et al., 2026; Shen et al., 2026). LongMamba and LAMB instead focus directly on selective long-range state updates. LongMamba identifies hidden channels with different receptive-field characteristics and applies selective token filtering to the channels responsible for longer-range information propagation (Ye et al., 2025b). LAMB follows this framework and derives attention-guided importance scores from Mamba’s attention-like representation to determine which positions should update long-range states (Ye et al., 2025a). These methods establish selective allocation of long-range updates as an effective training-free direction for pretrained Mamba models.

2.2 Content-Driven Information Selection for Long Contexts

Beyond recurrent state-space models, long-context research increasingly treats historical information as a non-uniform resource-allocation problem. H₂O, Scissorhands, SnapKV, PyramidKV, FastGen,

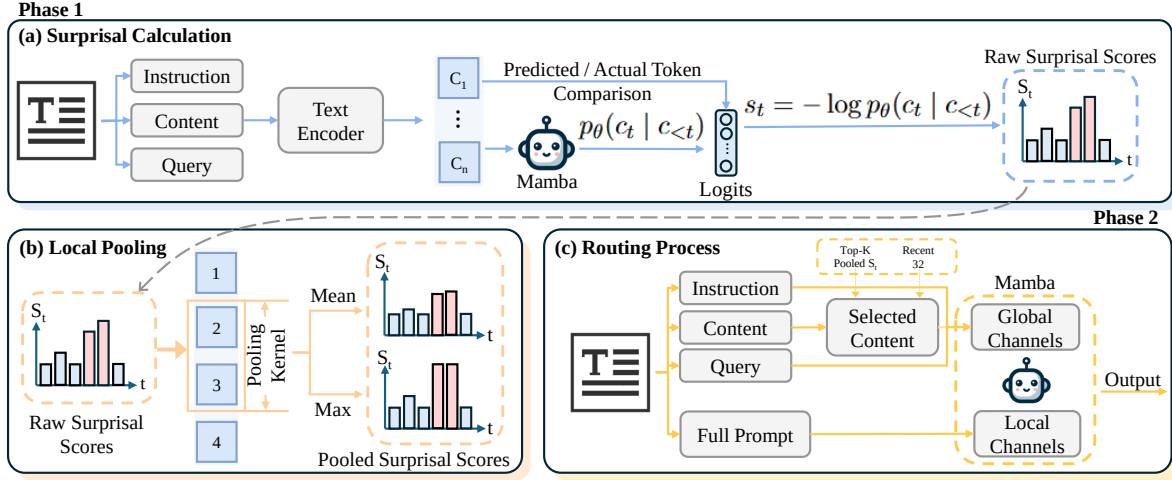


Figure 2: Overview of the SuPER-Mamba framework. Phase 1 performs offline Context Scoring. (a) Surprisal Calculation computes position-wise Raw Surprisal Scores from the next-token prediction probabilities of the pretrained Mamba. (b) Local Pooling aggregates neighboring predictive responses using category-specific Mean or Max Pooling to obtain Pooled Surprisal Scores. Phase 2 uses these scores for long-range state routing. The historical Top- K Content positions selected by the Pooled Surprisal Scores are combined with the most recent $R = 32$ Content positions to form the final routing set. For long-range state updates, Instruction and Query positions do not participate in Content Selection and retain their original state updates, while among Content positions, only the selected positions update the predefined Global Channels. All Content positions continue to participate in the original recurrent computation, so the complete input is neither removed nor truncated.

and AttentionPredictor retain or predictively select historical KV states using attention statistics, persistence of importance, prompt observations, head-specific behavior, or temporal attention patterns (Zhang et al., 2023; Liu et al., 2023; Li et al., 2024; Cai et al., 2025; Ge et al., 2024; Yang et al., 2025). More recent work also considers local structure rather than isolated token scores. ChunkKV performs selection over contiguous chunks to better preserve local semantic structure, while DAST allocates compression capacity according to segment-level information density and combines local predictive information with global signals (Liu et al., 2025; Chen et al., 2025).

A related line of work reduces long inputs through context selection or prompt compression. Selective Context uses language-model self-information to remove predictable lexical units, while LLMLingua and LongLLMLingua use model probabilities, token-level information, and downstream questions for fine-grained compression (Li et al., 2023; Jiang et al., 2023, 2024). Other methods use task instructions, query relevance, user intention, answerability, or reasoning states to determine which passages, KV blocks, or evidence should be retained (Tang et al., 2025; Taguchi et al., 2025; Zhao et al., 2026; Yun and Kim, 2026). Analyses of prompt compression further show that re-

ducing input can discard contextual details that are difficult to recover (Łajewska et al., 2025). These approaches therefore differ both in their scoring signals and in whether they operate on explicit KV states, modify the input, or condition selection on downstream information needs.

2.3 Surprisal and Predictive Feedback for Context Selection

Surprisal is an information-theoretic quantity defined from conditional prediction probabilities and has been used to characterize contextual predictability in incremental language processing (Hale, 2001; Levy, 2008). For pretrained autoregressive models, it can be obtained directly from the next-token predictive distribution produced during context processing, providing a position-dependent response conditioned on the input and model state. Importantly, Surprisal measures predictive difficulty rather than semantic importance for a specific downstream task.

Predictive information has already been used for context selection. Selective Context uses self-information to distinguish more predictable context units, the LLMLingua family incorporates model probabilities or perplexity into compression decisions, and DAST combines local predictive information with global signals (Li et al., 2023;

Category	Dataset	Base Mamba2	LongMamba	LAMB	SuPER-Mamba (Ours)
Single-doc QA	Qasper	1.76	4.75	5.63	<u>5.00</u>
	MultiFieldQA-en	5.03	11.12	14.76	<u>13.81</u>
Multi-doc QA	HotpotQA	2.89	5.45	7.71	<u>6.43</u>
	2WikiMQA	2.31	5.19	8.01	10.09
Summarization	GovReport	7.66	14.67	16.38	13.70
	MultiNews	11.31	14.72	17.49	14.45
Few-shot	TREC	14.67	21.33	46.04	<u>40.33</u>
	TriviaQA	10.08	47.58	51.04	<u>47.69</u>
	SAMSum	4.66	13.79	26.02	28.30
Synthetic	Passage Count	1.29	1.72	3.79	4.00
	Passage Retrieval-en	0.81	5.60	4.30	<u>5.19</u>
Code	LCC	22.85	40.39	41.65	49.03
	RepoBench-P	19.89	33.52	34.02	42.20
Average		8.09	16.91	21.30	21.56

Table 1: Main LongBench-E results on Mamba2-1.3B. Results are grouped by task category. Different datasets are evaluated using their official LongBench-E task-specific metrics, including QA F1 for Qasper, MultiFieldQA-en, HotpotQA, 2WikiMQA, and TriviaQA, ROUGE-L F1 for GovReport, MultiNews, and SAMSum, classification score for TREC, count score for Passage Count, retrieval score for Passage Retrieval-en, and code similarity for LCC and RepoBench-P. Each dataset score is the arithmetic mean over the 0–4k, 4–8k, and 8k+ context-length ranges. The Average row reports the arithmetic mean over all 13 dataset-level scores. Best results are shown in **bold**, and second-best results are underlined.

Jiang et al., 2023, 2024; Chen et al., 2025). However, predictive uncertainty and task-specific relevance are not equivalent. QUITO-X argues from an information-bottleneck perspective that high self-information or perplexity does not necessarily imply greater relevance to a downstream query (Wang et al., 2025). Related work on information preservation in prompt compression similarly cautions against treating prediction-derived scores as direct semantic-importance labels (Łajewska et al., 2025).

Position-wise predictive responses can also vary because of lexical choice, special symbols, or local expression changes. This motivates considering local structure rather than isolated Surprisal peaks. Related work such as ChunkKV and DAST similarly suggests that contiguous or locally structured information can provide a more stable basis for selection than independent token-level scores (Liu et al., 2025; Chen et al., 2025).

Taken together, prior work shows that Mamba long-context modeling depends on information preservation in finite recurrent states, that selective long-range updates can improve performance, and that pretrained models already provide predictive signals for differentiating context positions. SuPER-Mamba connects these directions by preserving the complete input, deriving position-wise predictive feedback from Surprisal, aggregating neighboring responses through Local Pooling, and using the resulting context scores to allocate long-

range state updates without additional selector training or downstream-query-dependent context scoring.

3 Method

SuPER-Mamba constructs routing scores from the predictive feedback produced by a pretrained model during context processing. Given a long context, Surprisal first converts next-token predictions into position-wise predictive responses. Local information aggregation then combines neighboring responses into context scores, which are used to determine which positions update the predefined Global Channels.

The context-scoring procedure consists of Surprisal estimation and local information aggregation, followed by long-range state routing. Figure 2 illustrates the complete procedure.

3.1 Surprisal Based Contextual Prediction

Given a context

$$C = (c_1, c_2, \dots, c_N), \quad (1)$$

where c_t denotes the token at context position t , an autoregressive language model assigns a conditional probability to c_t based on the preceding sequence $c_{<t}$,

$$p_\theta(c_t \mid c_{<t}), \quad (2)$$

where θ denotes the pretrained model parameters. The Surprisal at position t is defined as

$$s_t = -\log p_\theta(c_t \mid c_{<t}). \quad (3)$$

Surprisal characterizes the unpredictability of the current token given its preceding context. A lower conditional probability corresponds to a higher Surprisal value and therefore indicates a stronger predictive response at the corresponding position. Conversely, when the current token is more consistent with the prediction formed from the preceding context, its conditional probability increases and the corresponding Surprisal decreases.

This formulation converts the predictive feedback continuously produced during autoregressive processing into an explicit position wise numerical signal. For a context of length N , the Surprisal values across all positions form the response sequence

$$\mathbf{s} = (s_1, s_2, \dots, s_N). \quad (4)$$

The sequence $\mathbf{s}$ describes how the predictive response of the model varies across context positions. Different positions therefore receive differentiated response values according to the next token distributions produced by the pretrained model. Since these values are derived directly from the predictive distributions of the model, they provide input dependent position wise feedback without requiring a separate scoring model.

Surprisal nevertheless represents the predictive outcome at an individual position, and its values may vary substantially across neighboring tokens. Individual tokens that are difficult to predict may also produce isolated high responses. The Surprisal sequence therefore serves as the position wise predictive representation from which subsequent local information aggregation constructs the context scores used for routing.

3.2 Local Information Aggregation

Token-level Surprisal reflects the predictive response at an individual position and may vary substantially across neighboring tokens. Isolated high responses can arise from local lexical changes, special symbols, or other position-specific factors. Directly using these responses for routing may therefore produce unstable position selection.

Given the Surprisal sequence $\mathbf{s}$, we apply a local pooling operator

$$\mathbf{u} = P_k(\mathbf{s}), \quad (5)$$

where P_k denotes Mean or Max pooling with window size k . The resulting score at each position incorporates predictive responses from its local neighborhood, reducing the influence of isolated peaks while preserving local response patterns. The aggregated sequence $\mathbf{u}$ is subsequently used for long-range state routing.

3.3 Long-Range State Routing

After local aggregation, the pooled score sequence is used to determine which context positions are allowed to update the predefined Global Channels. Let R denote the recent-position window and let $\mathbf{u}$ denote the locally aggregated score sequence. SuPER-Mamba excludes the most recent R positions from score-based selection and selects the K highest-scoring positions from the remaining historical context. The selected historical set is defined as

$$\mathcal{T}_K = \text{TopK}(\mathbf{u}_{1:N-R}, K). \quad (6)$$

The most recent R positions are always preserved. The final routing set is therefore

$$\mathcal{G} = \mathcal{T}_K \cup \{N - R + 1, \dots, N\}. \quad (7)$$

When the input length is smaller than $K + R$, all positions are retained.

The same routing mask is applied across Mamba layers while preserving the predefined Global Channels of each layer. Let $\mathcal{C}_G$ denote the set of predefined Global Channels and let $\Delta_{t,j}$ denote the original state-update step at context position t and channel j . SuPER-Mamba applies the routing mask as

$$\tilde{\Delta}_{t,j} = \begin{cases} \Delta_{t,j}, & j \notin \mathcal{C}_G, \\ \Delta_{t,j}, & j \in \mathcal{C}_G, t \in \mathcal{G}, \\ 0, & j \in \mathcal{C}_G, t \notin \mathcal{G}. \end{cases} \quad (8)$$

Thus, for selected context positions, the original update on the Global Channels is retained, whereas unselected context positions do not update these channels. All non-Global Channels remain unchanged. Non-context prompt positions, including the Instruction and Query, bypass Content selection and retain their original updates. SuPER-Mamba therefore modifies long-range state allocation without removing or truncating input tokens, and the complete sequence remains available to the original recurrent computation.

Category	K	Kernel	Pooling
Single-doc QA	1024	18	Max
Multi-doc QA	1024	10	Mean
Summarization	2048	18	Mean
Few-shot	1024	18	Max
Synthetic	256	18	Mean
Code	1024	12	Mean

Table 2: Category routing configurations used in the full LongBench-E evaluation. K denotes the number of historical Content positions selected based on the Pooled Surprisal Scores and allowed to update the Global Channels. Kernel denotes the Local Pooling window size k , and Pooling denotes Mean or Max aggregation. All task categories additionally retain the most recent $R = 32$ Content positions.

4 Experimental Setup

4.1 Models, Baselines, and Datasets

Models and Datasets. We evaluate SuPER-Mamba on three representative state-space-based models: Mamba1-1.4B (Gu and Dao, 2024), Mamba2-1.3B (Dao and Gu, 2024), and Zamba2-1.2B (Glorioso et al., 2024). All models use their corresponding pretrained parameters without additional training or parameter updates. We evaluate on all 13 tasks in LongBench-E (Bai et al., 2024), covering a total of 3,668 test samples.

Baselines. We compare SuPER-Mamba with the corresponding base pretrained models and LAMB (Ye et al., 2025a). For Mamba2, we additionally report LongMamba (Ye et al., 2025b) to provide a more complete comparison with existing long-context Mamba methods. Since corresponding LongMamba results for Mamba1 and Zamba2 are not included in our evaluation, LongMamba is excluded from the cross-model average.

Evaluation. For each LongBench-E task, we compute the official task-specific metric separately over the three context-length ranges of 0–4k, 4–8k, and 8k+. We then take the arithmetic mean of the three range-level scores as the task-level score used in our aggregate results. Overall performance is reported as the arithmetic mean of the 13 task-level scores.

4.2 Implementation Details

Surprisal Scoring. SuPER-Mamba computes Surprisal from the predictive responses of the pretrained model. Context scoring is performed using only the context and does not use the downstream query. The resulting Surprisal scores are fixed before routed inference. No additional selector is

Architecture	Base Model	LAMB	SuPER-Mamba
Mamba1-1.4B	8.52	12.98	17.93
Mamba2-1.3B	8.09	21.30	21.56
Zamba2-1.2B	7.57	7.32	7.59
Average	8.06	13.87	15.69

Table 3: Overall LongBench-E results of SuPER-Mamba on Mamba1-1.4B, Mamba2-1.3B, and Zamba2-1.2B. Each dataset is evaluated using its official LongBench-E task-specific metric, and the results are equally averaged across the three context-length ranges. The Average row reports the arithmetic mean of the overall scores of the three models.

trained, and the pretrained model parameters remain unchanged.

Routing Configuration. We use predefined Global Channels and apply the selection mask produced by SuPER-Mamba to control Global-Channel updates at different context positions. Token-level Surprisal is first aggregated by local pooling, after which context positions are selected under a fixed routing budget. The most recent 32 context positions are always preserved. We do not apply additional normalization or denoising to the Surprisal scores.

The routing hyperparameters are configured by task category, as summarized in Table 2. The dataset-to-category mapping is provided in Appendix A.

Inference Settings. All methods use the same evaluation samples, generation settings, and evaluation protocol. Inference is performed with batch size 1. SuPER-Mamba does not perform input-level context compression. The complete context remains available to the model, while the routing mechanism only controls which context positions update the Global Channels.

5 Results

5.1 Main Results

Table 3 summarizes the overall LongBench-E performance across Mamba1-1.4B, Mamba2-1.3B, and Zamba2-1.2B. Averaged across the three architectures, SuPER-Mamba improves the Base Model average from 8.06 to 15.69, corresponding to a relative improvement of 94.7%. It also improves over the LAMB average of 13.87 by 13.1%. The gain is largest on Mamba1-1.4B and Mamba2-1.3B, while the improvement on Zamba2-1.2B is limited. This suggests that predictive feedback can support routing across different Mamba architectures, although its benefit remains dependent on the under-

Variant	Average Score
Base Model	8.09
Random Control	19.33
Raw Surprisal	16.33
+ Local Pooling	20.85
Full SuPER-Mamba	21.56

Table 4: SuPER-Mamba component ablation results on Mamba2-1.3B. All results are evaluated on the complete set of 13 LongBench-E datasets using their official task-specific metrics. Each dataset score is obtained by equally averaging the results over the three context-length ranges.

lying backbone. Since LAMB does not provide scripts for Mamba1-1.4B and Zamba2-1.2B, the corresponding results are reproduced based on the method described in the paper.

More fine-grained results on Mamba2-1.3B are reported in Table 1. SuPER-Mamba outperforms LAMB on 6 of the 13 datasets, with the clearest advantages on Code tasks. LCC improves from 41.65 to 49.03 and RepoBench-P improves from 34.02 to 42.20, corresponding to relative gains of 17.7% and 24.0%. Competitive results are also maintained across several remaining datasets, indicating that the overall improvement is not driven by a single task.

5.2 Ablation Study

Table 4 examines the roles of Surprisal and Local Pooling. Raw Surprisal reaches 16.33, improving over the Base Model by 101.9%, showing that predictive feedback produced by the pretrained model provides an effective signal for long-range position scoring. The Random Control reaches 19.33 under the evaluated setting, indicating that selective allocation of long-range state updates itself can also provide substantial benefits.

Local Pooling further increases the score from 16.33 to 20.85, corresponding to a 27.7% improvement over Raw Surprisal and a 7.9% improvement over the Random Control. This result shows that aggregating neighboring predictive responses substantially strengthens Surprisal-based routing. Full SuPER-Mamba further reaches 21.56, demonstrating that predictive feedback, local aggregation, and the complete routing configuration together provide an effective mechanism for long-context state allocation.

5.3 Hyperparameter Sensitivity

Figure 3 presents the controlled sensitivity analysis of routing budget K and Local Pooling kernel k on

Method	0 to 4k	4 to 8k	8k+
Base Model	14.98	5.45	3.85
LAMB	22.41	22.32	19.15
SuPER-Mamba	23.64	21.18	19.85

Table 5: LongBench-E results of Mamba2-1.3B across different context-length ranges. The 0–4k, 4–8k, and 8k+ columns correspond to the three LongBench-E context-length ranges. Each column uses the official task-specific metric for each dataset and reports the equal average across all 13 datasets.

a subset of LongBench-E samples. Different task categories exhibit distinct score distributions and best observed settings. Summarization and Synthetic tasks show different responses to the routing budget, while Multi-Document QA and Code show clearer variation across kernel settings. These results indicate that routing budget and local aggregation range affect task categories differently.

Based on these observations, we use category-level configurations rather than a single global setting. For categories requiring finer shared selection, we further perform a restricted joint refinement in which datasets within the same category share one configuration. The selected configurations are then frozen and kept unchanged during the full LongBench-E evaluation. The controlled analysis is therefore used to characterize broad parameter sensitivity, while the subsequent refinement resolves the final shared configuration within a narrower search space. Details of the category-level hyperparameter selection protocol are provided in Appendix D.

5.4 Length-wise Analysis

Table 5 shows that SuPER-Mamba remains effective as context length increases. The Base Model decreases from 14.98 for inputs from 0 to 4k to 3.85 for inputs above 8k, whereas SuPER-Mamba maintains scores of 23.64 and 19.85 in the same ranges. In the longest range, SuPER-Mamba also exceeds LAMB by approximately 3.7%. The slower degradation of SuPER-Mamba indicates that selective long range state updates remain effective for longer contexts and help mitigate the loss of earlier information as recurrent processing continues.

6 Discussion

SuPER-Mamba demonstrates that predictive feedback produced by pretrained Mamba models can serve as an effective control signal for long-context state allocation. By constructing routing scores

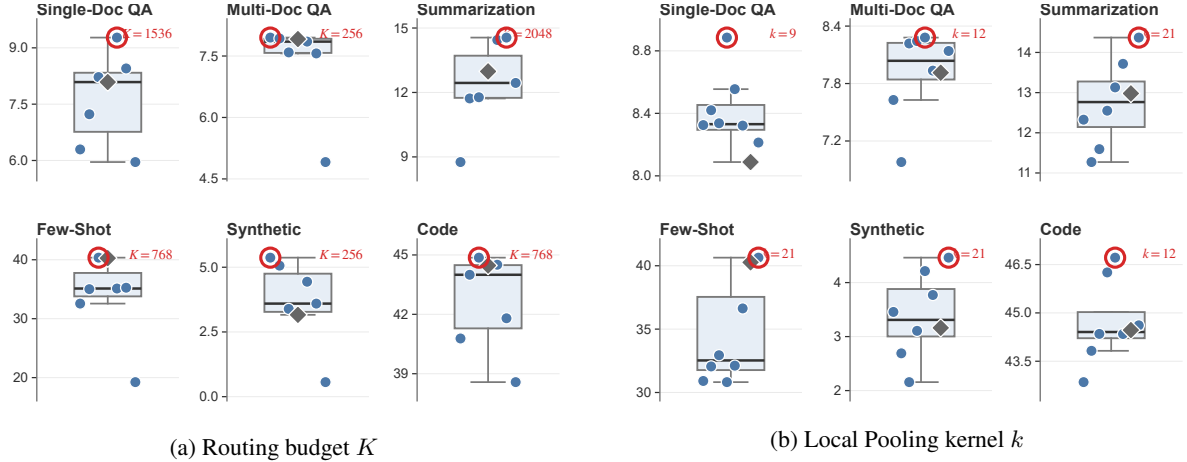


Figure 3: Hyperparameter sensitivity analysis on a LongBench-E subset using Mamba2-1.3B. The horizontal axis represents the six task categories, and the vertical axis reports the corresponding category-level LongBench-E score obtained by equally averaging the results of the datasets within each category. Each configuration is evaluated on 50 samples from each of the 13 datasets. (a) analyzes the routing budget K with $k = 18$ and Mean Pooling fixed, testing $K \in \{256, 512, 768, 1024, 1536, 2048, 3072\}$. (b) analyzes the Local Pooling kernel k with $K = 1024$ and Mean Pooling fixed, testing $k \in \{3, 5, 7, 9, 12, 15, 18, 21\}$. Boxes summarize the result distributions across all tested settings for each category, gray diamonds indicate the common reference configuration with $K = 1024$, $k = 18$, and Mean Pooling, and red circles indicate the best observed setting for each category in the corresponding parameter analysis.

from Surprisal during context processing, SuPER-Mamba determines long-range state updates without downstream-query-dependent context scoring or additional selector training. The results across Mamba1-1.4B, Mamba2-1.3B, and Zamba2-1.2B suggest that such model-intrinsic predictive signals provide a practical basis for training-free long-context routing across different architectures.

Our experiments show that restricting Global-Channel updates under a fixed budget is beneficial. Performance rises from 8.09 (Base Model, where every position updates the Global Channels; Table 1) to 19.33 when only a limited number of randomly selected positions are allowed to perform these updates (Appendix Table 8). This is consistent with the observation that selectively restricting long-range state updates improves performance (Ye et al., 2025a,b), which raises the question of how much additional value the Surprisal signal provides beyond any reasonable selection policy.

Our results show that Raw Surprisal alone (16.33) underperforms random selection, suggesting that unaggregated predictive responses are noisy. Local Pooling substantially improves the signal (20.85), surpassing the Random Control, and the full configuration reaches 21.56. Thus, while selective allocation accounts for a large fraction of the gains, the combination of Surprisal with local aggregation yields a consistently stronger rout-

ing policy than random selection under the same budget. The Surprisal-based scores therefore contribute meaningful structure beyond mere sparsity.

7 Conclusion

This paper presents SuPER-Mamba, a Surprisal-guided query-independent routing method for long-context modeling in pretrained Mamba models. SuPER-Mamba constructs positional scores from predictive feedback produced during context processing and applies Local Pooling to aggregate neighboring responses, enabling the allocation of long-range state updates without additional selector training or downstream-query-dependent context scoring. Experiments on LongBench-E show that SuPER-Mamba consistently improves the Base Model and achieves competitive performance across Mamba1-1.4B, Mamba2-1.3B, and Zamba2-1.2B. The ablation study further demonstrates that Surprisal provides an effective predictive signal for long-range routing and that Local Pooling substantially strengthens its effectiveness, while the length-wise analysis shows that the method remains effective under longer context conditions. Overall, SuPER-Mamba demonstrates that predictive feedback from pretrained models can serve as an effective signal for long-range information routing, providing a training-free approach to long-context modeling in Mamba.

8 Limitations

SUPER-Mamba currently uses fixed task-category-specific routing configurations, which cannot adapt to differences across individual inputs. In addition, the offline design requires an extra context-scoring pass before routed inference, introducing additional computation and latency. Future work could explore input-adaptive routing and more efficient score computation while preserving the query-independent and training-free properties of the framework.

References

- Ameen Ali Ali, Itamar Zimmerman, and Lior Wolf. 2025. [The hidden attention of Mamba models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1516–1534, Vienna, Austria. Association for Computational Linguistics.
- 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, multi-task 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.
- Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. 2025. [PyramidKV: Dynamic KV cache compression based on pyramidal information funneling](#). In *Second Conference on Language Modeling*.
- Shaoshen Chen, Yangning Li, Zishan Xu, Yongqin Zeng, Shunlong Wu, Xinshuo Hu, Zifei Shan, Xin Su, Jiwei Tang, Yinghui Li, and Hai-Tao Zheng. 2025. [DAST: Context-aware compression in LLMs via dynamic allocation of soft tokens](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20544–20552, Vienna, Austria. Association for Computational Linguistics.
- Tri Dao and Albert Gu. 2024. [Transformers are SSMS: Generalized models and efficient algorithms through structured state space duality](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 10041–10071. PMLR.
- Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2024. [Model tells you what to discard: Adaptive KV cache compression for LLMs](#). In *The Twelfth International Conference on Learning Representations*.
- Paolo Glorioso, Quentin Anthony, Yury Tokpanov, Anna Golubeva, Vasudev Shyam, James Whittington, Jonathan Pilault, and Beren Millidge. 2024. [The zamba2 suite: Technical report](#). Preprint, arXiv:2411.15242.
- Albert Gu and Tri Dao. 2024. [Mamba: Linear-time sequence modeling with selective state spaces](#). In *First Conference on Language Modeling*.
- John Hale. 2001. [A probabilistic earley parser as a psycholinguistic model](#). In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.
- Tamanna Hossain, Robert L. Logan IV, Chandrasekhara Ganesh Jagadeesan, Sameer Singh, Joel R. Tetreault, and Alejandro Jaimes. 2025. [Characterizing Mamba’s selective memory using auto-encoders](#). In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2007–2022, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.
- Xiang Hu, Jiaqi Leng, Jun Zhao, Kewei Tu, and Wei Wu. 2025. [Hardware-aligned hierarchical sparse attention for efficient long-term memory access](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 98788–98813.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. [LLMLingua: Compressing prompts for accelerated inference of large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.
- Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. [LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.
- Aakash Sunil Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, Zico Kolter, Tri Dao, and Albert Gu. 2026. [Mamba-3: Improved sequence modeling using state space principles](#). In *The Fourteenth International Conference on Learning Representations*.
- Weronika Łajewska, Momchil Hardalov, Laura Aina, Neha Anna John, Hang Su, and Lluís Marquez. 2025. [Understanding and improving information preservation in prompt compression for LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17520–17541, Suzhou, China. Association for Computational Linguistics.
- Roger Levy. 2008. [Expectation-based syntactic comprehension](#). *Cognition*, 106(3):1126–1177.

- Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. [Compressing context to enhance inference efficiency of large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, Singapore. Association for Computational Linguistics.
- 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.
- Xiang Liu, Zhenheng Tang, Peijie Dong, Zeyu Li, Yue Liu, Bo Li, Xuming Hu, and Xiaowen Chu. 2025. [ChunkKV: Semantic-preserving KV cache compression for efficient long-context LLM inference](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 32533–32583.
- Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023. [Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 52342–52364.
- Peng Lu, Jerry Huang, Qiuhaio Zeng, Xinyu Wang, Boxing Chen, Philippe Langlais, and Yufei Cui. 2025. [Mamba modulation: On the length generalization of Mamba models](#). In *Advances in Neural Information Processing Systems*, volume 38.
- Jun-Yu Ma, Tianqing Fang, Zhisong Zhang, Hongming Zhang, Haitao Mi, and Dong Yu. 2025. [Recall with reasoning: Chain-of-thought distillation for Mamba’s long-context memory and extrapolation](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4714–4720, Suzhou, China. Association for Computational Linguistics.
- Hugo Pitorro and Marcos Vinicius Treviso. 2025. [LaTIM: Measuring latent token-to-token interactions in Mamba models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24478–24493, Vienna, Austria. Association for Computational Linguistics.
- Haochen Shen, Davis Wertheimer, Zheng Wang, Garrett Goon, Derrick Liu, Naigang Wang, Mudhakar Srivatsa, Raghu Ganti, and Minjia Zhang. 2026. [From collapse to control: Understanding and extending context length in emerging hybrid models via universal position interpolation](#). In *The Fourteenth International Conference on Learning Representations*.
- Chihiro Taguchi, Seiji Maekawa, and Nikita Bhutani. 2025. [Efficient context selection for long-context QA: No tuning, no iteration, just adaptive- \$k\$](#) . In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20105–20130, Suzhou, China. Association for Computational Linguistics.
- Jiwei Tang, Jin Xu, Tingwei Lu, Zhicheng Zhang, Yiming Zhao, Lin Hai, and Hai-Tao Zheng. 2025. [Perception compressor: A training-free prompt compression framework in long context scenarios](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4093–4108, Albuquerque, New Mexico. Association for Computational Linguistics.
- Qinwen Wang, Jieping Luo, Aoxiang Qin, Ruoyu Zhao, Jianxiong Tang, Wei Zhang, Zhichao Lu, and Luziwei Leng. 2026. [Mamba with hierarchical memory: Solving representation bottleneck in long sequence modeling](#). *arXiv preprint arXiv:2608.02347*.
- Yihang Wang, Xu Huang, Bowen Tian, Yueyang Su, Lei Yu, Huaming Liao, Yixing Fan, Jiafeng Guo, and Xueqi Cheng. 2025. [QUITO-X: A new perspective on context compression from the information bottleneck theory](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6841–6856, Suzhou, China. Association for Computational Linguistics.
- Qingyue Yang, Jie Wang, Xing Li, Zhihai Wang, Chen Chen, Lei Chen, Xianzhi Yu, Wulong Liu, Jianye Hao, Mingxuan Yuan, and Bin Li. 2025. [Attention-Predictor: Temporal patterns matter for KV cache compression](#). In *Advances in Neural Information Processing Systems*, volume 38.
- Zhifan Ye, Zheng Wang, Kejing Xia, Jihoon Hong, Leshu Li, Lexington Whalen, Cheng Wan, Yonggan Fu, Yingyan Celine Lin, and Souvik Kundu. 2025a. [LAMB: A training-free method to enhance the long-context understanding of SSMS via attention-guided token filtering](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1200–1209, Vienna, Austria. Association for Computational Linguistics.
- Zhifan Ye, Kejing Xia, Yonggan Fu, Xin Dong, Jihoon Hong, Xiangchi Yuan, Shizhe Diao, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. 2025b. [Long-Mamba: Enhancing mamba’s long-context capabilities via training-free receptive field enlargement](#). In *The Thirteenth International Conference on Learning Representations*.
- WangJie You, Zecheng Tang, Juntao Li, Lili Yao, and Min Zhang. 2025. [Revealing and mitigating the local pattern shortcuts of Mamba](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12156–12178, Vienna, Austria. Association for Computational Linguistics.
- Danlong Yuan, Jiahao Liu, Bei Li, Huishuai Zhang, Jingang Wang, Xunliang Cai, and Dongyan Zhao. 2025. [ReMamba: Equip mamba with effective long-sequence modeling](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6830–6840, Suzhou, China. Association for Computational Linguistics.

JungMin Yun and YoungBin Kim. 2026. [IterCOMP: Reasoning-aware adaptive prompt compression for multi-hop question answering](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33827–33840, San Diego, California, United States. Association for Computational Linguistics.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuan-dong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H₂O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36.

Liang Zhao, Xiaocheng Feng, Weihong Zhong, Lei Huang, Kun Zhu, Baoxin Wang, Dayong Wu, Guoping Hu, Ting Liu, and Bing Qin. 2026. [Question tells you where the answer is: Intention-aware long-context KV cache compression](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27153–27169, San Diego, California, United States. Association for Computational Linguistics.

A Dataset Categories and Routing Configurations

We group the 13 LongBench-E datasets into six task categories. Datasets within the same category share the selection budget K , Local Pooling kernel size k , and pooling operator. All category-level configurations are determined before the full LongBench-E evaluation and remain fixed throughout the main experiments. Table 6 provides the complete dataset-to-category mapping and the corresponding routing configurations. All configurations use a fixed recent-position window of $R = 32$.

B Detailed Routing Implementation

SuPER-Mamba follows an offline two-pass inference procedure. The first pass constructs the routing scores and the corresponding routing mask, while the second pass performs routed inference using the fixed mask. The complete prompt is preserved throughout both passes. SuPER-Mamba does not remove or truncate Content positions and instead controls which positions are allowed to update the predefined Global Channels.

During the first pass, Surprisal scoring is performed only on the Content and does not use the downstream Query. Position-wise predictive responses are computed for the Content and aggregated by Local Pooling to obtain the routing scores. These scores are then associated with their corresponding Content positions in the complete prompt. Instruction, Query, and other non-Content prompt positions do not participate in the Content Top- K selection and therefore do not compete for the selection budget.

For Content routing, the most recent $R = 32$ positions are excluded from score-based historical selection and are always retained. Among the remaining historical Content positions, SuPER-Mamba selects the K positions with the highest pooled scores. The final Content routing set therefore consists of the historical Top- K positions together with the Recent- R positions. When the Content length does not exceed $K + R$, all Content positions are retained.

The resulting Content mask is then projected to the complete prompt. All non-Content positions retain their original updates. Selected Content positions retain their original updates on the Global Channels, whereas unselected Content positions are prevented from updating only these chan-

nels. Their remaining recurrent computation is unchanged. The same position-level routing mask is applied to the predefined Global Channels across Mamba layers. During the second pass, the model receives the complete Instruction, Content, and Query, while the routing mask obtained from the first pass remains fixed. This implementation keeps context scoring independent of the downstream Query while preserving the full input sequence during routed inference.

C Surprisal and Local Pooling Implementation Details

Surprisal Alignment. During the first pass, SuPER-Mamba tokenizes the Content independently and performs a forward pass using the pre-trained Mamba without routing modification. For each position $t > 1$, the logits produced at position $t - 1$ are compared with the ground-truth token at position t using cross-entropy, so the resulting Surprisal score is aligned with the predicted token position rather than the position producing the logits. The first Content token has no preceding prediction within the independently processed Content sequence and is therefore assigned a Surprisal score of zero. Logits may be computed in chunks to reduce memory usage without changing the resulting position-wise Surprisal scores.

Projection to the Complete Prompt. Because independently tokenizing the Content does not necessarily produce token boundaries identical to those of the same Content span inside the complete prompt, SuPER-Mamba does not assume a direct one-to-one correspondence between the two tokenizations. The implementation first identifies the exact character span of the Content in the rendered prompt and obtains tokenizer offset mappings for both the Content-only sequence and the complete prompt. Prompt tokens overlapping the Content span are then identified. For each such prompt token, Surprisal scores from overlapping Content-only tokens are combined using a character-overlap-weighted average. If no direct character overlap is available, the score of the nearest Content token by character-center distance is used. This projection produces one routing score for each Content position in the complete prompt.

Local Pooling. For a given kernel size k , SuPER-Mamba applies stride-1 Mean Pooling or Max Pooling to aggregate Surprisal responses over neighboring positions, using padding of $k/2$ according to

Dataset	Category	K	Kernel	Pooling
Qasper	Single-doc QA	1024	18	Max
MultiFieldQA-en	Single-doc QA	1024	18	Max
HotpotQA	Multi-doc QA	1024	10	Mean
2WikiMQA	Multi-doc QA	1024	10	Mean
GovReport	Summarization	2048	18	Mean
MultiNews	Summarization	2048	18	Mean
TREC	Few-shot	1024	18	Max
TriviaQA	Few-shot	1024	18	Max
SAMSum	Few-shot	1024	18	Max
Passage Count	Synthetic	256	18	Mean
Passage Retrieval-en	Synthetic	256	18	Mean
LCC	Code	1024	12	Mean
RepoBench-P	Code	1024	12	Mean

Table 6: Complete mapping from LongBench-E datasets to task categories and their corresponding category routing configurations. Different categories use different routing parameter configurations while following the same routing mechanism during inference.

Category	K	Kernel k	Pooling
Single-doc QA	1024	18	Max
Multi-doc QA	1024	10	Mean
Summarization	2048	18	Mean
Few-shot	1024	18	Max
Synthetic	256	18	Mean
Code	1024	12	Mean

Table 7: Final category routing configurations used for the full LongBench-E evaluation. K denotes the selection budget for historical Content positions, Kernel k denotes the Local Pooling window size, and Pooling denotes Mean or Max aggregation. These configurations are determined from the parameter analysis on a LongBench-E subset and remain fixed throughout the subsequent full evaluation.

the pooling implementation. For even kernel sizes such as $k = 18$, the underlying pooling operation produces one additional output position. The final extra position is therefore removed so that the pooled score sequence has the same length as the original Content position sequence. The resulting Pooled Surprisal Scores are subsequently used for Content routing.

D Hyperparameter Selection Protocol

SUPER-Mamba adopts shared routing configurations at the task-category level. The 13 LongBench-E datasets are first grouped into six categories according to their task type and input-context structure: Single-doc QA, Multi-doc QA, Summarization, Few-shot, Synthetic, and Code. This categorization is defined independently of the subsequent hyperparameter experiments. Datasets within the same category share the same routing budget K ,

Local Pooling kernel size k , and pooling operator, avoiding dataset-specific hyperparameter optimization.

After defining the task categories, we analyze the effects of the routing budget K , Local Pooling kernel size k , and pooling operator using Mamba2 on a subset of LongBench-E. We adopt a one-variable-at-a-time parameter sweep with $K = 1024$, $k = 18$, and Mean Pooling as the common reference configuration. For the routing budget, we fix $k = 18$ and Mean Pooling and evaluate

$$K \in \{256, 512, 768, 1024, 1536, 2048, 3072\}. \quad (9)$$

For the Local Pooling kernel, we fix $K = 1024$ and Mean Pooling and evaluate

$$k \in \{3, 5, 7, 9, 12, 15, 18, 21\}. \quad (10)$$

The pooling operator is evaluated by comparing Mean and Max Pooling under $K = 1024$ and $k = 18$. All 13 LongBench-E datasets are included in the parameter analysis, with 50 samples evaluated for each dataset under each configuration. The shared reference configuration is reused across the individual sweeps, resulting in 15 distinct configurations in total.

Hyperparameter selection is performed within the predefined task categories. For each category, we jointly consider the performance of its constituent datasets under different configurations and select a shared setting that maintains consistently strong performance across the category. As reported in Section 5.3, the sensitivity analysis aggregates the dataset-level scores within each category

Dataset	Random Control	Raw Surprisal	+ Local Pooling	Full SuPER-Mamba
Qasper	4.12	3.36	4.85	5.00
MultiFieldQA-en	13.00	11.70	14.40	13.81
HotpotQA	6.54	5.71	6.82	6.43
2WikiMQA	8.83	8.39	10.00	10.09
GovReport	12.58	10.65	13.80	13.70
MultiNews	12.89	11.85	14.31	14.45
TREC	29.33	5.33	39.67	40.33
TriviaQA	42.14	39.89	44.82	47.69
SAMSum	25.11	26.67	27.84	28.30
Passage Count	3.90	4.22	4.29	4.00
Passage Retrieval-en	4.63	4.73	4.25	5.19
LCC	44.80	40.32	43.94	49.03
RepoBench-P	43.38	39.44	42.07	42.20
Average	19.33	16.33	20.85	21.56

Table 8: Dataset-level component ablation results on LongBench-E using Mamba2-1.3B. Each dataset is evaluated using its official LongBench-E task-specific metric. Samples are divided into the 0–4k, 4–8k, and 8k+ context-length ranges, and the three range-level scores are averaged arithmetically to obtain the dataset-level score. The Average row reports the arithmetic mean over all 13 dataset-level scores. Bold values indicate the best result among the four component settings.

using an equal average to characterize the effect of individual hyperparameters. The final configurations are determined from the experimental results at the category level rather than by applying a single automatic optimization rule.

The resulting category-level routing configurations are summarized in Table 7. These configurations are fixed before the full LongBench-E evaluation and remain unchanged throughout the subsequent experiments.

E Additional Experimental Results

This section provides additional dataset-level and length-wise results for the component ablation study reported in Section 5.2. All results are obtained on Mamba2 using the same LongBench-E evaluation protocol and task-category-specific routing budgets as in the main experiments.

E.1 Dataset-level Component Ablation

Table 8 reports the dataset-level results for Random Control, Raw Surprisal, Surprisal with Local Pooling, and Full SuPER-Mamba across all 13 LongBench-E datasets. The four variants are evaluated under the same task-category-specific routing budgets, allowing the effects of different routing signals and aggregation components to be compared at the dataset level.

The dataset-level results show that the effects of individual routing components vary across tasks. Raw Surprisal substantially improves overall performance over the Base Model reported in the main

text, but its effectiveness is not uniform across datasets. Introducing Local Pooling increases the average score from 16.33 to 20.85 and improves over Raw Surprisal on 12 of the 13 datasets. This result provides further evidence that aggregating neighboring predictive responses substantially strengthens Surprisal-based routing.

Full SuPER-Mamba further increases the average score to 21.56 and outperforms the Local Pooling variant on 9 of the 13 datasets. The strongest variant is not identical across every individual task, indicating that the contribution of the complete routing configuration can vary with task characteristics. Nevertheless, the aggregate improvement is distributed across multiple task categories rather than being attributable to a single dataset.

E.2 Length-wise Component Results

Table 9 further reports the component ablation results across the three LongBench-E context-length ranges. For each range, the reported score is the arithmetic mean over the same 13 datasets.

The length-wise results are consistent with the overall ablation findings. Raw Surprisal performs below the Local Pooling variant across all three context-length ranges. Local Pooling increases the average scores to 23.05, 20.45, and 19.06 for the 0–4k, 4–8k, and 8k+ ranges, respectively, indicating that the benefit of local aggregation is not restricted to a particular input-length regime.

Full SuPER-Mamba achieves the highest average score in all three ranges, reaching 23.64, 21.18, and 19.85, respectively. In the 8k+ range, perfor-

Variant	0-4k	4-8k	8k+	Average
Random Control	21.45	18.51	18.03	19.33
Raw Surprisal	19.17	15.87	13.94	16.33
+ Local Pooling	23.05	20.45	19.06	20.85
Full SuPER-Mamba	23.64	21.18	19.85	21.56

Table 9: Length-wise component ablation results on LongBench-E using Mamba2-1.3B. The 0-4k, 4-8k, and 8k+ columns correspond to the three context-length ranges. Each range uses the official LongBench-E task-specific metric for each dataset and reports the equal average across all 13 datasets. Bold values indicate the best result in each column.

mance increases from 13.94 with Raw Surprisal to 19.85 with the complete method. These results further indicate that combining predictive feedback with Local Pooling and the complete routing configuration provides more robust long-context performance than position-level Surprisal alone.