S & M 3800

A Flexible State Space Model for Large Language Models: The GroupMamba Approach

Xiling Liu,[1] Qunsheng Ruan,[2] Yingjia Wu,[1] Kai Chen,[1] and Cheng-Fu Yang[3,4]*

[1]School of Information Science and Technology, HaiNan Normal University, HaiNan 571158, China
[2]Department of Nature Science and Computer, Ganzhou Teachers College, Gan Zhou 341004, China
[3]Department of Chemical and Materials Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan
[4]Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413, Taiwan

(Received August 14, 2024; accepted September 27, 2024)

Keywords: transformers, state space models (SSMs), GroupMamba, natural language processing (NLP)

Transformers have consistently excelled in large language models owing to their exceptional scalability, efficient parallel processing, superior contextual comprehension, and versatility across a wide range of tasks. In recent years, state space models (SSMs) have also seen notable advancements, with the Mamba model standing out for its efficient parallel processing capabilities and low computational complexity. However, despite these strengths, SSMs, including Mamba, often struggle to match the performance of transformers in tasks that require deep contextual understanding and the handling of high-dimensional data. In this paper, we introduce GroupMamba, a novel group-based SSM specifically designed to optimize the trade-off between complexity and parallel processing capabilities by strategically grouping SSM modules. These groupings can be customized to suit various tasks, effectively blending the strengths of both Mamba and transformer architectures. Experimental results demonstrate that GroupMamba achieves significant improvements across diverse tasks, including a notable 2% increase in accuracy on public benchmark tests. In this work, we mark a significant advancement in the integration of SSMs and transformers, offering a more adaptable, scalable, and efficient solution for addressing complex natural language processing challenges.

1. Introduction

The rapid advancements in natural language processing (NLP) have been significantly propelled by the development of large language models, with transformers leading this transformative shift.(1) The transformer architecture is renowned for its unparalleled scalability, efficient parallel processing, superior contextual understanding, and exceptional versatility. These attributes have enabled transformers to set new benchmarks across a diverse range of NLP tasks. For instance, in language generation, transformers have achieved unprecedented levels of fluency and coherence.(2) In translation, the model proposed by Zhang *et al.* had surpassed previous models in terms of accuracy and efficiency.(3) Additionally, in comprehension

*Corresponding author: e-mail: cfyang@nuk.edu.tw
https://doi.org/10.18494/SAM5309

tasks, transformers have demonstrated a deep understanding of complex text, outperforming earlier approaches. The widespread adoption and success of transformers underscore their pivotal role in advancing the field of NLP and shaping its future directions.

Despite the transformative impact of transformers, recurrent neural networks (RNNs) and their extensions have continued to evolve, leading to the development of state space models (SSMs),(4,5) which offer unique advantages in certain contexts. Initially, RNNs were widely used for sequential data processing owing to their ability to capture temporal dependences. However, RNNs, including their more advanced variants such as long short-term memory (LSTM) networks and gated recurrent units (GRUs), faced limitations in handling long-range dependences and complex temporal patterns. To address these challenges, researchers began exploring SSMs, which provide a more structured approach to modeling temporal dynamics. In recent years, SSMs have made significant strides, culminating in models such as Mamba.(6) Mamba exemplifies efficient parallel capabilities while maintaining lower computational complexity, showcasing the potential of SSMs to handle sequential data more efficiently than traditional RNNs. However, even with these advancements, SSMs often fall short when compared with transformers in tasks requiring deep contextual understanding and the processing of high-dimensional data. This ongoing evolution from RNNs to SSMs highlights the continuous quest for more robust and efficient models in the field of sequential data processing.

While Mamba excels at retaining historical features, its effectiveness wanes as the sequence length increases, creating a performance gap compared with transformers. To bridge this gap, we propose a novel method of grouping SSM modules, illustrated in Fig. 1. Our approach leverages an across-attention mechanism to integrate hidden states across different groups, enhancing the model's ability to manage longer sequences. By adjusting the number of groups, we can control the density of hidden states within each group: a higher number of groups leads to fewer hidden states per group, while fewer groups result in more hidden states per group. This adaptable grouping strategy enables us to fine-tune the model according to the specific requirements of each task. Consequently, we can optimize performance, improving the model's efficiency and accuracy for diverse applications. This method not only addresses the limitations of Mamba but also demonstrates the potential for SSMs to achieve results that are competitive with, or even surpass, those of transformers in certain contexts.

SSMs can be utilized in design filters, such as Kalman filters, for estimating a system's state. This estimation is typically based on the system's dynamic model and observational data

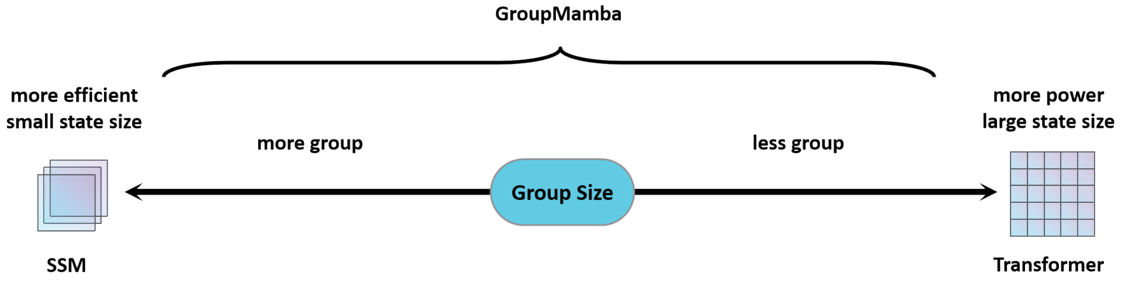

Fig. 1. (Color online) Differences among SSM, transformer, and GroupMamba.

obtained from sensors. Sensors provide real-time observational data, which form the foundation for state estimation. However, measurement errors and noise from sensors can impact the accuracy of these estimates. In control design, SSMs are used to create controllers that regulate system behavior. Designers can leverage these models to ensure that the system reaches the desired state under given inputs. Sensor feedback is employed to adjust the controller's output in real time, ensuring that the system's state meets the set targets.(7) The accuracy of sensor data directly affects the effectiveness of control. In complex systems, multiple sensors may provide data. SSMs can be used to fuse data from different sensors, resulting in more accurate state estimation.(8) Different sensors might measure various aspects of the system, such as position and velocity, and combining these data can enhance the precision and reliability of state estimation. SSMs and sensors play complementary roles in system dynamics analysis, control design, and state estimation. While SSMs offer a framework for describing system dynamics, sensors provide real-time data essential for estimating and controlling the system's state.

SSMs and sensors play complementary roles in system dynamics analysis, control design, and state estimation. SSMs provide a framework for describing system dynamics, while sensors enable real-time data logging for estimating and controlling system states. Therefore, when applying SSMs in the GroupMamba approach to large language models, sensors are crucial auxiliary tools.(9,10) Consequently, in this study, we incorporate the data from various sensors throughout its research. In this paper, our contributions are as follows. First, we propose the GroupMamba model, a novel approach that groups SSM modules and integrates their hidden states using a cross-attention mechanism. This method effectively addresses the limitations of the Mamba model in retaining historical features over long sequences. Second, we introduce a flexible grouping strategy that allows for the adjustment of both the number of groups and the hidden states within each group. These adaptabilities of the proposed methods enable the model to tailor its structure to specific tasks, thereby optimizing performance across diverse applications. Third, the proposed GroupMamba model demonstrates substantial improvements across a wide range of tasks, achieving a 2% increase in accuracy on public benchmark tests. This highlights the model's effectiveness in combining the strengths of both SSMs and transformer architectures for complex NLP tasks. Fourth, by exploring the synergy between SSMs and transformer architectures, in this work, we pave the way for future research and development in the field of NLP, offering a promising direction for building more powerful and versatile language models.

2. Related Works

2.1 Large language models

Large language models have revolutionized NLP, with the advent of transformer-based architectures at the forefront of this transformation. The introduction of the transformers by Vaswani *et al.* marked a significant leap forward by utilizing self-attention mechanisms to effectively capture long-range dependences and nuanced contextual relationships within text.(11) This seminal work laid the groundwork for the development of models such as bidirectional encoder representations from transformers (BERTs) and generative pretrained transformers (GPTs), both of which have set new benchmarks across a wide array of NLP tasks. Building upon the bidirectional encoding approach of BERTs, the GPT series further advanced the field of language generation. Notably, GPT-2(12) and GPT-3,(13) with their autoregressive frameworks, have exhibited exceptional proficiency in generating coherent, contextually appropriate text. These models, scaled up to billions of parameters, have not only improved performance but also demonstrated remarkable versatility in tasks ranging from summarization and translation to creative writing. This evolution underscores the transformative impact of large-scale pretraining on the ability of machines to comprehend and generate human language, driving unprecedented progress in diverse NLP applications.

2.2 Transformer

The introduction of the transformer architecture by Vaswani *et al.* had been a landmark development in the field of NLP.(14) Unlike previous architectures such as RNNs and convolutional neural networks (CNNs), transformers utilize self-attention mechanisms to process input data in parallel, rather than sequentially. This parallelism significantly speeds up training and increasing the accuracy and efficiency of handling long-range dependences. Following the success of the transformer architecture, several notable models have emerged, expanding its applications and improving performance. Devlin *et al.* were among the first to leverage the transformer architecture for bidirectional context understanding, achieving state-of-the-art results in various NLP tasks, including question answering and named entity recognition.(15) The ongoing evolution of transformer models has also been supported by advances in training techniques, such as mixed-precision training and distributed computing frameworks, which facilitate the handling of larger datasets and model sizes. These advancements have broadened the applicability of NLP models, driving progress in areas such as machine translation, summarization, and conversational AI. The transformer architecture has profoundly affected NLP, leading to numerous breakthroughs and establishing new standards for model performance and versatility in language understanding and generation tasks.

2.3 Introduction of SSMs

SSMs have a long-standing tradition in time series analysis and dynamical systems, offering a robust framework for modeling temporal dependences and stochastic processes. Classic SSMs, such as the Kalman filter,(16,17) provide optimal estimation for linear dynamical systems with Gaussian noise. This model has been foundational in various applications, including navigation and control systems, owing to its ability to efficiently estimate hidden states and handle noisy observations. Recent advancements in deep learning have led to the integration of neural networks with SSMs, resulting in innovative models such as deep Kalman filters (DKFs) and recurrent SSMs. These hybrid approaches combine the rigorous mathematical framework of SSMs with the expressive power of deep learning. For instance, DKFs utilize deep neural networks to model transition and observation functions, offering enhanced flexibility in capturing complex, nonlinear dynamics that are challenging for traditional SSMs. The Mamba model represents a significant advancement in this area, demonstrating efficient parallel processing capabilities while maintaining lower computational complexity. Mamba addresses some limitations of traditional SSMs by enabling more scalable and efficient inference, making it suitable for large-scale applications. Despite its strengths in parallel processing and reduced complexity, Mamba often underperforms compared with transformers in tasks requiring deep contextual understanding. Additionally, it faces challenges related to scalability, task adaptability, and implementation complexity. These limitations highlight the ongoing need for research to further enhance SSMs' capabilities and versatility, aiming to bridge gaps between traditional methods and modern, data-driven approaches.

3. Methods

3.1 Preliminaries

3.1.1 Equation of SSMs

An SSM is a sequence-to-sequence transformer that integrates the strengths of both RNNs and transformers. During inference, it functions sequentially like an RNN with O(L) complexity, whereas during training, it can leverage parallel processing similar to a transformer. Drawing inspiration from classical SSMs in time series analysis and control engineering, SSMs can be viewed as a hybrid between RNNs and CNNs. The formulation of the SSMs is as follows:

$$h\left(t\right)=Ah(t)+Bx(t)\,, \tag{1}$$

$$y(t)=Ch(t)\,. \tag{2}$$

In this context, A, B, and C are constants with respect to the input $x(t)$. The SSMs can be considered a continuous mapping from x to y, effectively treating the target as a continuous signal. However, in language and image processing, it is essential to handle discrete values. Thus, by discretizing the above equations, we obtain

$$h_t = Ah_{t-1} + Bx_t\,, \tag{3}$$

$$y_t = Ch_t\,. \tag{4}$$

By recursion, we have

$$h_t^l = \overline{A}h_{t-1}^l + \overline{B}x_t^l = \overline{A}(\overline{A}h_{t-2}^l + \overline{B}x_{t-1}^l) + \overline{B}x_t^l = \overline{A}^2 h_{t-2}^l + \overline{AB}x_{t-1}^l + \overline{B}x_t^l$$

$$= \overline{A}^t h_0^l + \overline{A}^{t-1}\overline{B}x_{t-1}^l + \cdots + \overline{AB}x_{t-1}^l + \overline{B}x_t^l = \overline{A}^t x_0^l + \overline{A}^{t-1}\overline{B}x_1^l + \cdots + \overline{AB}x_{t-1}^l + \overline{B}x_t^l, \tag{5}$$

where l represents the $L-$ th module of the model. Finally, there is

$$h_t^l = \overline{A}^t\ \overline{B}x_0^l + \overline{A}^{t-1}\ \overline{B}x_1^l + \ldots + \overline{AB}x_{t-1}^l + \overline{B}x_t^l = \left(x_0^l, x_1^l, \ldots, x_t^l\right) * \left(\overline{B}, \overline{AB}, \ldots, \overline{A}^t\overline{B}\right). \tag{6}$$

3.1.2 Mamba

Mamba introduces an enhanced SSM that significantly improves its capacity to process input sequences by integrating a dynamic selection mechanism. This mechanism enables the model to adjust its focus on the basis of the relevance of different parts of the input, thereby boosting both the efficiency and accuracy of handling long sequences of data. Unlike traditional SSMs, Mamba distinguishes itself by making certain parameters—specifically A, B, and C—functions of the input. This modification results in changes in tensor shape throughout the processing stages. Notably, these parameters now include a length dimension, which shifts the model from being time-invariant to time-variant. While this transition enhances the model's flexibility, it also impacts efficiency due to the loss of equivalence with convolution operations, reflecting a trade-off between adaptability and computational performance.

3.2 GroupMamba

While Mamba is capable of retaining more historical information compared with RNNs or traditional SSMs, it still lags behind transformers in terms of global memory capabilities. As the length of the input sequence increases, transformers demonstrate superior ability to capture and utilize global context, although this advantage comes at the cost of reduced inference speed. For simpler tasks, transformers can become excessively redundant, whereas Mamba struggles with effectively managing long sequences. To address these limitations, in this paper, we propose a balanced approach by introducing a group-based SSM module model, termed GroupMamba. The framework of GroupMamba is illustrated in Fig. 2. In this model, the input sequence (x_1^l, x_2^l, …, x_t^l) is processed through grouped SSM modules. The hidden states generated by each group are then integrated with those from subsequent groups using cross-attention mechanisms. This integration allows later sequences to benefit from a broader historical context. When the number of groups is large and each group contains fewer modules, the computational load is reduced, as later modules have access to a smaller subset of historical information, balancing the trade-off between information retention and computational efficiency.

Specifically, when each group contains only a single module, the model omits the cross-attention operation, thereby functioning equivalently to the original Mamba model. On the other hand, when there are fewer groups, but each group comprises more modules, the later modules can leverage a greater amount of historical information, although this increases computational demands. When the entire sequence is processed as a single group, all hidden states are subjected

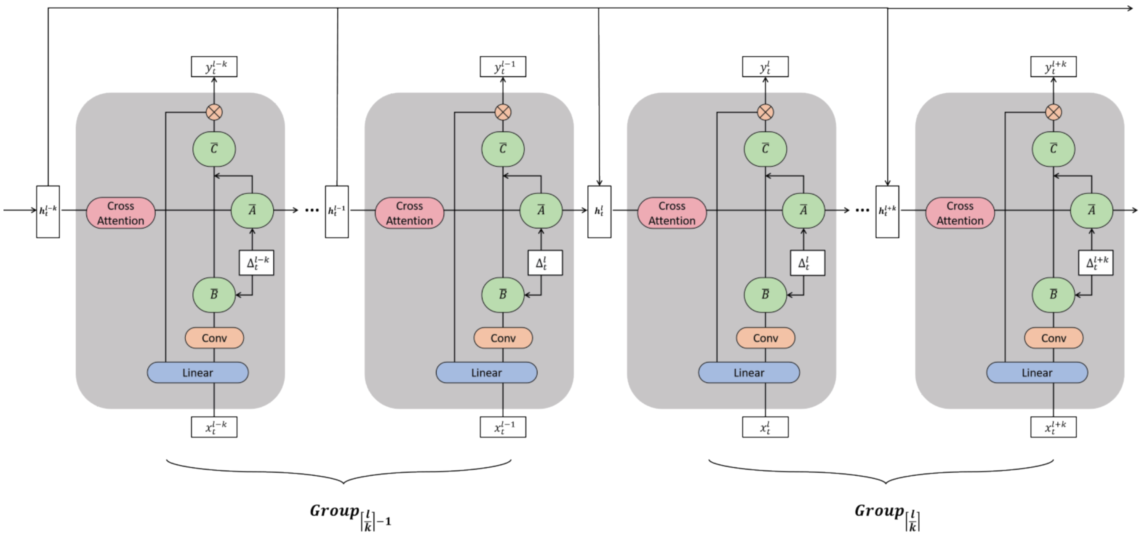

Fig. 2. (Color online) Illustrations of GroupMamba.

to cross-attention computations, which makes the model more similar to a transformer. The framework for cross-attention among hidden states is depicted in Fig. 3. In this process, the hidden states from the preceding group are concatenated and passed through a linear layer to generate the value (V) and key (K) matrices. Simultaneously, the hidden states to be queried are processed through a separate linear layer to produce the query (Q) matrix. The cross-attention computation is then executed using these matrices, facilitating the integration of historical context into the processing of the current sequence. The cross-attention computation is then performed as follows:

$$CrossAttention(h_t^l,(h_t^{l-k} \cdots h_t^l)) = \begin{cases} Soft\max(\frac{QK^T}{\sqrt{d_v}})V & k > 1 \\ h_t^l & k = 1 \end{cases}, \tag{7}$$

where d_v denotes the dimension of the key set (which is also the dimension of the query set) and k represents the number of hidden states within each group. When $k = 1$, the model simplifies to the original Mamba, eliminating the need for cross-attention computation.

A key characteristic of SSMs is their ability to be trained in parallel, and this group-based hidden state integration operation maintains that parallelism without disruption:

$$H_t^l = CorssAttention\left(h_t^l,\left(h_t^{1-k} \ldots h_t^l\right)\right). \tag{8}$$

According to Eqs. (4) and (6), we have

$$y_t^1 = Ch'^1_t = CCrossAttention(\left(x_0^l, x_1^l, \cdots,\right) * (\overline{B}, \overline{AB}, \cdots, \overline{A}^t \, \overline{B})\mathcal{H}_t^1)\ . \tag{9}$$

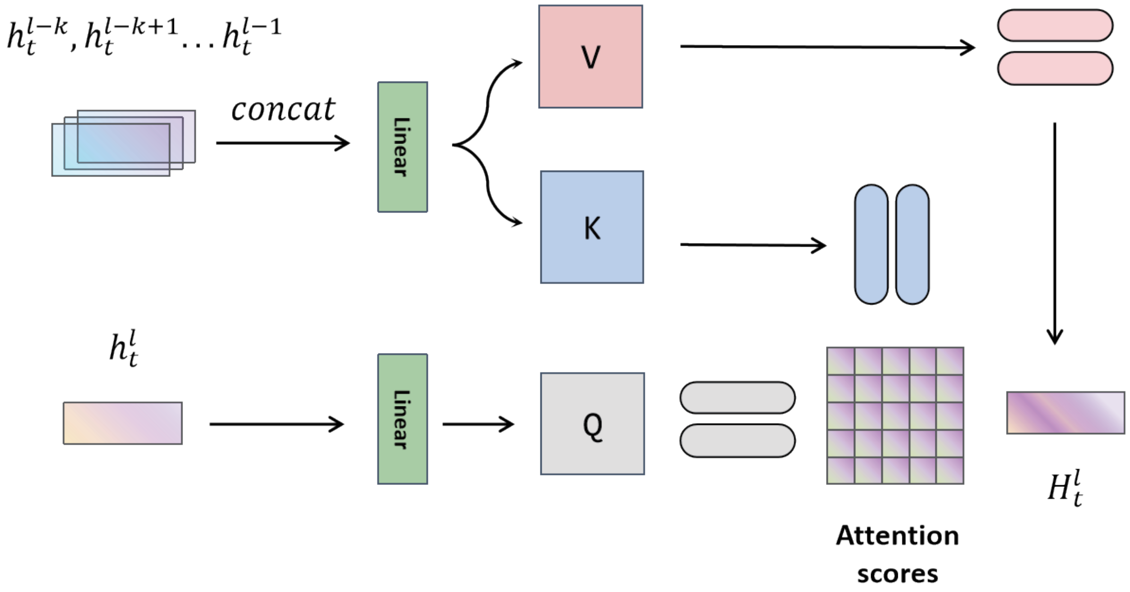

Fig. 3. (Color online) Cross-attention of different hidden states.

The higher the value of k, the higher the model's accuracy. The selection of k and its impact on various tasks will be detailed in the experiments, as shown in Fig. 4.

4. Experiments

In this section, we performed a series of comprehensive experiments to rigorously evaluate the effectiveness of the proposed GroupMamba model. Our validation process involved testing the model across a diverse range of architectures, including the Mamba and transformer architectures, to ensure robustness and generalizability. By comparing the performance of GroupMamba within these different frameworks, we aimed to demonstrate its versatility and potential advantages in various contexts.

4.1 Training setup

We compared our GroupMamba model against both the transformer and Mamba models, with the hyperparameters for each model detailed in Table 1. For GroupMamba, we used $k = 2$ as the default grouping configuration to maintain consistency in the parameters across each group. Note that GroupMamba exhibits a higher parameter count than the other models. This increase is primarily due to the incorporation of cross-attention mechanisms, which contribute to the model's enhanced capability but also result in a greater number of parameters.

4.2 Dataset

We validated the effectiveness of the proposed model using a diverse set of datasets, including the choice of plausible alternatives (COPA), physical interaction question answering (PIQA), the Winograd schema challenge, Winogrande, StoryCloze, and OpenBookQA. Each of these

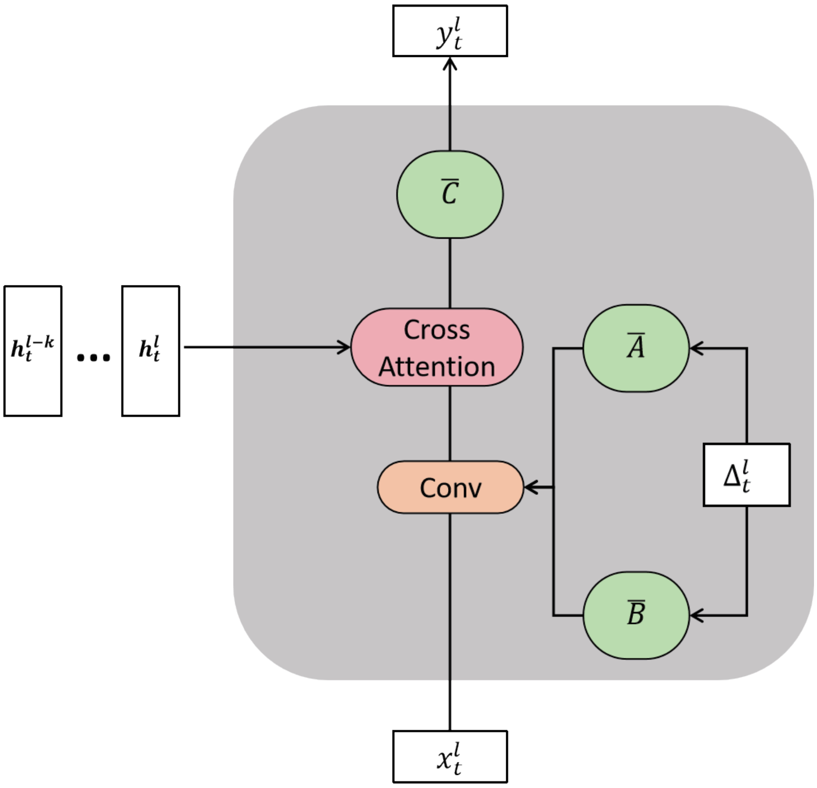

Fig. 4. (Color online) GroupMamba in convolution mode.

Table 1
Hyperparameters: rotary position embedding was used for transformer models.

Hyperparameters		Transformer	Mamba	GroupMamba
Model	Total blocks	8	16	16
	model #	512	512	512
	Parameters	350M	350M	445M
Feedforward	dfeed	2048	—	—
Position embedding		RoPE	—	—
Attention	Nheads	8	—	—
Training	Training steps	150K	150K	150K
	Context length	1024	1024	1024
	Batch size	64	64	64
	Max learning rate	$5e^{-4}$	$1e^{-3}$	$5e^{-4}$
	LR warmup	1%	1%	1%
	LR schedule	Cosine	Cosine	Cosine
	Final LR ratio	0.1	0.1	0.1
	Weight decay	0.1	0.1	0.1
	Gradient clipping	0.5	0.5	0.5

datasets presents unique challenges, ranging from commonsense reasoning and physical interactions to complex story understanding and open-book question answering. A detailed description of each dataset is provided in Table 2. The evaluations were conducted using the LM evaluation harness, which offers a standardized and consistent framework for assessing the

Table 2
Overview of datasets.

Dataset	Description
Choice of plausible alternatives (COPA)	This dataset is designed for evaluating commonsense causal reasoning. It consists of questions paired with two possible answers, where the task is to select the more plausible alternative based on commonsense knowledge.
Physical interaction question answering (PIQA)	This dataset is designed for evaluating physical commonsense reasoning. It contains questions about everyday physical interactions, each with two possible answers. The task is to choose the more plausible alternative based on an understanding of common physical interactions.
Winograd schema challenge	This benchmark is designed to evaluate coreference resolution and commonsense reasoning capabilities. Each question involves determining the correct referent of a pronoun within a given sentence.
Winogrande	As an extension of the Winograd Schema Challenge, this dataset enhances the diversity and difficulty of the original challenge to more effectively evaluate the capabilities of language models.
StoryCloze	This dataset is designed to evaluate story understanding and generation. It consists of a four-sentence story followed by two possible endings, with the task of selecting the ending that best completes the story.
OpenBookQA	This dataset is designed for testing open-book question answering. It requires retrieving relevant information and reasoning over that information to answer multiple-choice questions.

models' performance across different tasks. This method ensures that our comparisons are reliable and that the effectiveness of the proposed model is rigorously tested across a broad spectrum of natural language processing challenges.

4.3 Quantitative analysis

The suitability of datasets for Mamba and transformer models depends on task complexity and model strengths. Mamba models, noted for their efficient handling of historical features and lower computational complexity, are particularly well-suited for datasets such as COPA, PIQA, and the Winograd Schema Challenge. These datasets involve commonsense reasoning and pronoun resolution, where Mamba's efficiency and effective historical feature management are advantageous. In contrast, transformer models excel in tasks requiring superior context comprehension and scalability. They are better suited for more complex and extensive datasets such as Winogrande, StoryCloze, and OpenBookQA, which demand deep contextual understanding and the ability to synthesize information from long sequences. Table 3 provides a comparison of GroupMamba under various group configurations with both transformer and Mamba models, illustrating the performance differences across these datasets.

The results presented in the table show that our observations align with the analysis results: GroupMamba demonstrates superior performance across all datasets with varying k values. Specifically, in the COPA and PIQA datasets, GroupMamba performs best with k = 2. This suggests that these datasets benefit from structures similar to Mamba, and the improved performance of GroupMamba highlights the effectiveness of cross-attention mechanisms. For the WinoGrande and OpenBookQA datasets, the optimal performance is achieved with k = 1, indicating that these datasets gain more from extensive historical information rather than complex grouping structures. Across all tasks, k = 4 consistently achieves the highest scores,

Table 3
Comparison of GroupMamba with Mamba and transformer across various task sets.

Models / Tasks	COPA	PIQA	Winograd	WinoGrande	StoryCloze	OpenBookQA	Avg
Transformer	60.00	64.36	62.02	49.09	57.64	29.60	53.79
Mamba	62.00	64.50	62.92	52.88	56.59	29.20	54.68
GroupMambe$_{\text{default}}$	64.00	65.24	63.82	49.10	57.04	29.41	54.76
GroupMambe$_{k=4}$	63.00	64.15	64.25	49.99	57.38	30.20	54.82
GroupMambe$_{k=8}$	61.00	64.48	63.11	51.58	58.16	30.22	54.75
GroupMambe$_{k=1}$	60.00	64.28	62.01	53.71	58.10	30.57	54.78

Table 4
Ablation study on the effects of module grouping and cross-attention on benchmark datasets.

k	CrossAttention	COPA	PIQA	Winograd
1 (mamba)	-	62	64.5	62.92
2	×	60	63.11	63.66
2	✓	64	65.24	63.82
4	×	62	62.17	62.33
4	✓	63	64.15	64.25
8	×	60	61.99	61.00
8	✓	61	64.48	63.11
1	×	57	62.36	59.21
1	✓	60	64.28	62.01

suggesting that an intermediate *k* value offers better generalization across various datasets. These findings imply that choosing an appropriate *k* value can significantly impact model performance, providing valuable guidance for future tasks where the optimal *k* value may be uncertain.

4.4 Ablation study

The ablation study offers valuable insights into the effects of the number of modules per group (*k*) and the inclusion of cross-attention on model performance. Notably, for the COPA and PIQA datasets, the configuration with k = 2 and the inclusion of cross-attention achieves the highest scores of 64 and 65.24, respectively, as shown in Table 4. This indicates that these datasets benefit from a setup with fewer modules, enhanced by cross-attention, which likely helps in capturing essential interdependences within the data. In contrast, for the Winograd dataset, the optimal performance of 64.25 is observed with k = 4 and the inclusion of cross-attention. This suggests that a moderate number of modules combined with cross-attention strikes an effective balance, providing sufficient historical context without overwhelming the model. These results emphasize the importance of tuning both the number of modules and the use of cross-attention to achieve optimal performance across different datasets, tailoring the model's structure to the specific demands of the task. Interestingly, larger *k* values, such as 8, do not yield the best results, suggesting that an excessive number of modules may lead to redundancy or overfitting. In contrast, the inclusion of cross-attention consistently enhances performance across different configurations, underscoring its role in improving the model's ability to focus on relevant information. These findings indicate that selecting an optimal *k* value, in conjunction with cross-attention, significantly boosts the model's generalization capabilities. In practical scenarios where the ideal k is uncertain, adopting a balanced approach with a moderate k value and incorporating cross-attention could be a promising strategy for achieving robust performance across a range of tasks.

5. Conclusions

GroupMamba marks a significant advancement in the field of SSMs by synthesizing the strengths of both Mamba and transformer architectures. This novel model introduces a groundbreaking grouping strategy that dynamically adjusts hidden states through a cross-attention mechanism. This approach addresses the inherent limitations of traditional SSMs, particularly their struggle to effectively retain and leverage historical information across long sequences. The flexibility of GroupMamba is a key feature, allowing for a customizable number of groups tailored to the specific requirements of different tasks. This adaptability not only optimizes performance but also demonstrates superior versatility across a wide range of applications. The model's innovative design enhances computational efficiency while achieving a noteworthy 2% improvement in accuracy on public benchmark tests, underscoring its effectiveness. Moreover, GroupMamba's integration of dynamic grouping and cross-attention mechanisms bridges the gap between the simplicity of conventional SSMs and the comprehensive contextual understanding offered by transformers. This hybrid approach provides a robust and efficient solution for complex NLP tasks, positioning GroupMamba as a promising model that combines the best aspects of both architectural paradigms. The enhanced accuracy and adaptability showcased by GroupMamba highlight its potential to advance the capabilities of NLP systems, offering a sophisticated tool for tackling a broad spectrum of language understanding and generation challenges.

Acknowledgments

This research was supported by the Hainan Provincial Natural Science Foundation of China (No. 621MS038), the Research Project on Higher Education Reform of Jiangxi Province (No. JXJG-22-46-1), and the projects under Nos. NSTC 113-2622-E-390-001 and NSTC 113-2221-E-390-011.

References

1 A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and L. Polosukhin: Attention is all you need, in: Neural Information Processing Systems. https://doi.org/10.48550/arXiv.1706.03762 (accessed Sep. 2017).

2 N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher: CTRL: A Conditional Transformer Language Model for Controllable Generation (2019). https://doi.org/10.48550/arXiv.1909.05858

3 J. Zhang, H. Luan, M. Sun, F. Zhai, J. Xu, M. Zhang, and Y. Liu: Proc. 2018 Conf. Empirical Methods in Natural Language Processing (EMNLP 2018, Brussels, Belgium) 533–542.

4 M. I. Jordan: Adv. Psychology **121** (1997) 471.

5 S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski: 2018 32nd Conf. Neural Information Processing Systems (NIPS 2018, Montreal, Canada).

6 A. Gu and T. Dao: Mamba: Linear-Time Sequence Modeling with Selective State Spaces (2023). https://doi.org/10.48550/arXiv.2312.00752

7 A. Manikas, V. Sridhar, and Y. I. Kamil: 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2016, Rio de Janeiro, Brazil) 1–5.

8 W. Liu and T. Yairi: IEEE Access **12** (2024) 5920.

9 F. Thalmayr and G. Fischerauer: Sens. Actuators, B **144** (2010) 27.

10 S. Zhao, Y. Ma, and B. Huang: IEEE Trans. Ind. Electron. **66** (2019) 2154.

11 A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin: Proc. 31st Inter. Con. Neural Information Processing Systems (NIPS 2017) 6000–6010.

12 A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever: https://paperswithcode.com/paper/language-models-are-unsupervised-multitask (accessed March 2019).

13 T. Brown, B. Mann, N. Ryder, M. Subbiah, *et al*: Proc. 34th Int. Conf. Neural Information Processing Systems 159 (NIPS 2020) 1877–1901.

14 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin: 31st Con. Neural Information Processing Systems (NIPS 2017) (Long Beach, CA, USA).

15 J. Devlin, M. W. Chang, K. Lee, and K. Toutanova: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/arXiv.1810.04805 (accessed May 2019).

16 G. Welch and G. Bishop: An Introduction to the Kalman Filter, University of North Carolina at Chapel Hill Chapel Hill (1995) NC USA.

17 L. Gao, J. Tow, B. Stella, B. Sid, D. Anthony, F. Charles, G. Laurence, H. Jeffrey, M. Kyle, M. Niklas, P. Jason, R. Laria, T. Eric, T. Anish, W. Ben, W. Kevin, and Z. Andy: https://zenodo.org/records/5371629 (accessed Sep. 2021).