DTMFRec: Dynamic Temporal Weighted Multi-Path State Space Fusion for Sequential Recommendation

Jisheng Tian[1,3] Xiaoqiang Ren[1,3,*] Hongpei Ji[1,3] Wenpeng Lu[1,3] Xueying He[2]
[1] *Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China*
[2] *Shandong University of Traditional Chinese Medicine, Jinan, China*
[3] *Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing Shandong Fundamental Research Center for Computer Science, Jinan, China*
Email: jisheng.tjs@gmail.com, renxq@qlu.edu.cn*, 10431240008@stu.qlu.edu.cn, wenpeng.lu@qlu.edu.cn, hxy0104@163.com

***Abstract*—Sequential Recommendation Systems (SRS) aim to predict users' dynamic preferences from their historical interaction sequences. Although recent state space models such as Mamba exhibit strong potential with linear computational complexity, their inherently unidirectional architecture limits contextual comprehension and short-term interest modeling, especially in sparse data regimes. To address these challenges, we propose a Dynamic Temporal Weighted Multi-Path State Space Fusion (DTMFRec) Framework, a multi-path adaptive fusion framework. DTMFRec employs a multi-path architecture to simultaneously capture dependencies from both the raw and temporally enhanced sequences, thereby constructing more comprehensive contextual representations. Furthermore, an Adaptive Fusion Gating Layer is introduced to dynamically weight multiple information paths for optimal fusion. Experimental results on multiple real-world datasets demonstrate that DTMFRec significantly outperforms existing baselines and demonstrates superior performance in long-tail scenarios. The implementation code is available at https://github.com/Jisheng-Tian/DTMFRec.**

***Index Terms*—Sequential Recommendation, Mamba, TemporalWeight, Transformer, State Space Models**

I. Introduction

Sequential recommendation has long attracted research interest due to its wide applicability in domains such as e-commerce and social media. The key objective is to model users' evolving preferences based on their historical interactions to predict the next item of interest. Early approaches based on Convolutional Neural Networks (CNNs) [1] and Recurrent Neural Networks (RNNs) [2] pioneered the use of deep learning in this field, but suffered from catastrophic forgetting, which hindered their ability to capture long-term dependencies. Later, Transformer-based architectures [3], [4] achieved remarkable performance through self-attention mechanisms. However, their quadratic computational complexity limits scalability to long sequences, posing issues for real-time recommendation [5], [6].

In recent years, advances in State Space Models (SSMs) have offered an appealing alternative for long-sequence modeling. Representative methods such as Mamba [7] introduce input-dependent selectivity and achieve linear computational complexity, showing strong potential for efficient long-context modeling. Nevertheless, directly applying Mamba to sequential recommendation remains challenging. As illustrated in Figure 1, the upper bar chart reports the interaction-count distribution across different user groups, while the lower plot measures the corresponding recommendation performance via ΔHit@10. For the first three groups with relatively sparse interactions, Mamba4Rec performs substantially worse than SASRec. This observation suggests that, for long-tail users, a vanilla Mamba-style sequence encoder may fail to adequately capture recent preferences that often dominate user decisions, leading to performance degradation and further aggravating recommendation quality for under-represented users.

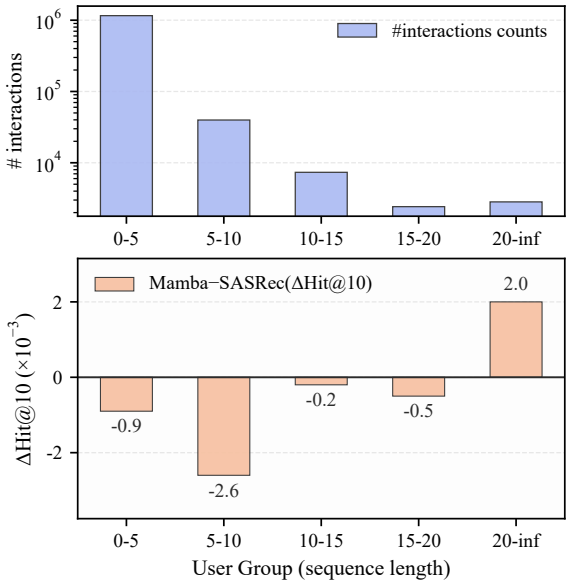

Fig. 1: The illustration for long-tail user problem.

To address the aforementioned challenges and more fully

* Corresponding authors

exploit the strengths of Mamba, we propose DTMFRec, a multi-path adaptive fusion framework that integrates complementary sequential signals through three parallel pathways. This framework effectively captures diverse types of sequential features through three parallel paths. Specifically, the Mamba path is designed to model global context; the Temporal Weighting path emphasizes recent user preferences; and the Residual Gating path incorporates long-term memory capabilities to ensure stability in long-sequence modeling [8]. A Length-Aware Adaptive Fusion Module further integrates these representations based on sequence characteristics. Residual connections, layer normalization, and position-wise feed-forward networks are also applied to enhance nonlinearity and training stability.

Extensive experiments conducted on five public datasets demonstrate that DTMFRec consistently outperforms existing baselines while maintaining high computational efficiency. The main contributions of this work are summarized as follows:

(i) We propose DTMFRec, which incorporates three parallel paths: a dynamic temporal weighting path that enhances the modeling of recent user preferences; the other paths preserve long-term interest capture through stable long-sequence memory mechanisms.

(ii) We design a length-aware adaptive fusion gating layer that dynamically fuses multi-path information based on sequence length and contextual features, achieving an optimal balance between global and local sequence representations;

(iii)We evaluate DTMFRec on five public benchmark datasets, and the results consistently demonstrate its superior effectiveness over existing baseline models.

II. METHOD

The overall architecture of DTMFRec is illustrated in Figure 2. The model is composed of three key components: Standard Mamba, Temporal Mamba, and Residual Gate. The core idea is to dynamically enhance the modeling of recent interactions in order to better capture users' recent preferences, while simultaneously integrating long-term interests. In the following sections, we provide a detailed analysis of each component.

A. Problem Definition and Embedding Layer

The objective of sequential recommendation is to predict the next item a user may interact with, given their historical interaction sequence [9]. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ denote the set of users and $\mathcal{V} = \{v_1, v_2, \ldots, v_{|\mathcal{V}|}\}$ denote the set of items. Each user $u \in \mathcal{U}$ is associated with an interaction sequence ordered by time: $\mathcal{S}_u = [s_1, s_2, \ldots, s_{L_u}]$, where L_u represents the length of the sequence for user [10].

To facilitate effective modeling in sequential recommender systems, it is common to map discrete item IDs into continuous high-dimensional representations [11]. For notational simplicity, we omit the user index (u) in the remainder of this paper. Given an interaction sequence as the input tensor, we employ a learnable item embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d}$ to project each item s_i into its embedding vector $\mathbf{e}_i \in \mathbb{R}^d$.

Accordingly, the embedded representation of an interaction sequence batch is defined as

$$\mathbf{E}_0 = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_L] \in \mathbb{R}^{B \times L \times d} \tag{1}$$

where B denote the batch size,L denotes the length of the interaction sequence and d is the dimensionality of the item embeddings.

B. Multi-Path Dynamic Temporal Weighting (MDTW) Module

1) Standard Mamba Path: To capture the global dependencies within user interaction sequences, we adopt the Mamba state space model as the primary sequential encoder. Given the embedded input sequence $\boldsymbol{E}_0 = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_L]$, the Standard Mamba Path processes the sequence without introducing any temporal bias, and leverages the linear-complexity state space mechanism to effectively model long-range item interactions. Formally, the Mamba encoder can be expressed as a function $\boldsymbol{f}_{\text{mamba}}(\cdot)$ parameterized by convolutional and recurrent operators, and its output is defined as:

$$\boldsymbol{H_{std}} = \boldsymbol{f}_{mamba}(\boldsymbol{E}_0) \in \mathbb{R}^{B \times L \times d} \tag{2}$$

This path is mainly designed to capture standard sequential features. Unlike attention-based models with quadratic complexity, the Mamba layer provides efficient sequence modeling with linear complexity while preserving strong expressiveness [7], [12].

2) Temporal Mamba Path: To mitigate the limitation of Mamba's unidirectional architecture in capturing short-term user interests [7], we introduce a Temporal Mamba module. The key idea is to impose a monotonically increasing, position-dependent temporal weighting over the input sequence so that more recent interactions contribute more prominently to representation learning [4], [13].Given the embedded interaction sequence $\boldsymbol{E}_0 \in \mathbb{R}^{B \times L \times d}$,we define an exponential temporal weight for each position $t \in \{1, \ldots, L\}$ as

$$w_t = \exp\left(\alpha \cdot \frac{t-1}{L-1}\right), \quad t = 1, 2, \ldots, L \tag{3}$$

where $\alpha > 0$ controls the growth rate, the method is insensitive to α, so we fix $\alpha = 3$ in all experiments. The temporally reweighted sequence is computed by element-wise scaling along the temporal dimension:

$$\boldsymbol{E_{weighted}} = \boldsymbol{E}_0 \odot w_t \tag{4}$$

where $\odot$ denotes element-wise multiplication with broadcasting across the feature dimension. Intuitively, since w_t increases with t, the module amplifies recent behaviors while retaining the original chronological order. The weighted sequence is then fed into the Mamba encoder to obtain temporally enhanced hidden representations:

$$\boldsymbol{H_{temporal}} = \boldsymbol{f}_{mamba}(E_{weighted}) \tag{5}$$

Compared with heuristic strategies such as sequence reversal [14], Temporal Mamba preserves the forward temporal structure of user interactions, avoiding disruption of chronological dependencies. This design provides an order-consistent

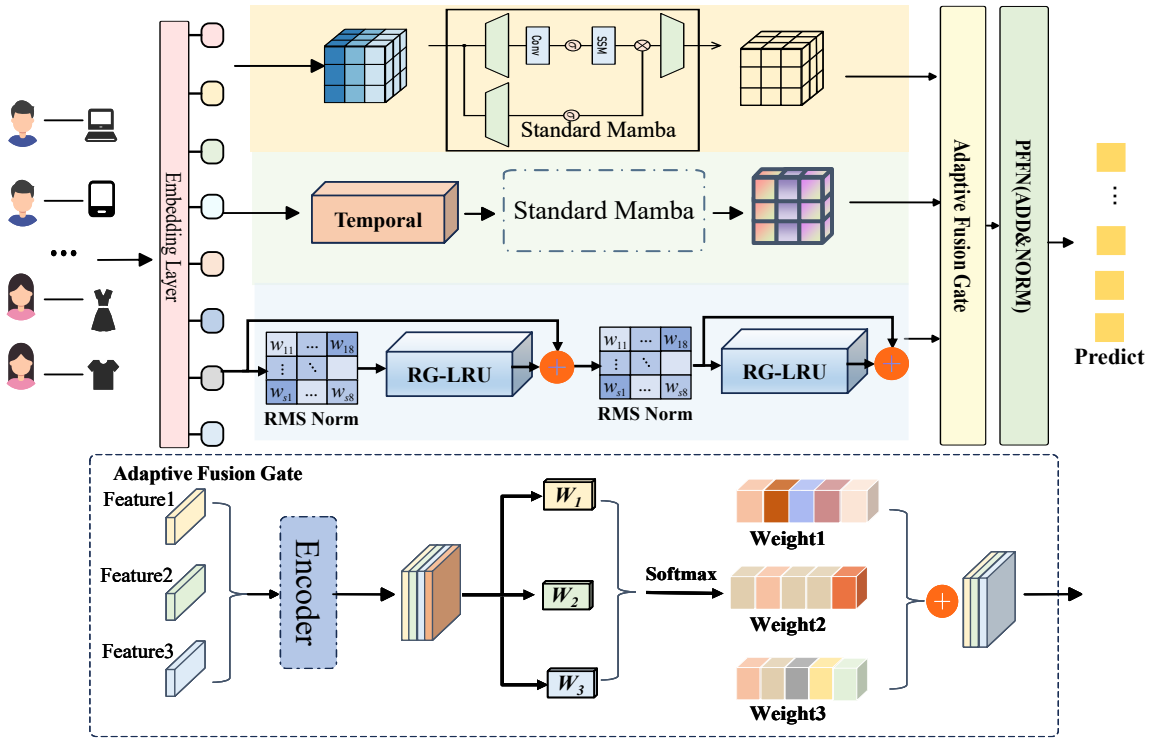

Fig. 2: Architecture of the proposed DTMFRec.

way to emphasize recency, thereby improving the model's ability to capture evolving user intent .

3) Residual Gating Path: The Residual Gating Path is designed to jointly model long-term dependencies and rapidly evolving short-term preferences in user behavior sequences. This path consists of an RMSNorm layer, an RG-LRU–based gated module, and a dual-residual structure, and operates in parallel with the backbone Mamba network to enhance sequence representation capacity.

Specifically, inspired by the Griffin architecture [8], we incorporate an RG-LRU module that adaptively regulates information flow across time steps via a gating mechanism. The input sequence $\boldsymbol{E}_0$ is processed by the RG-LRU module together with the dual-residual structure, enabling dynamic fusion of newly updated knowledge and original representations. Formally, for each time step t, the hidden state $\boldsymbol{h}_t$ is updated as follows [8]:

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_a \boldsymbol{x}_t + \boldsymbol{b}_a), \tag{6}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{b}_x), \tag{7}$$

$$\boldsymbol{a}_t = a^{c\boldsymbol{r}_t}, \tag{8}$$

$$\boldsymbol{h}_t = \boldsymbol{a}_t \odot \boldsymbol{h}_{t-1} + \sqrt{1 - \boldsymbol{a}_t^2} \odot (\boldsymbol{i}_t \odot \boldsymbol{x}_t), \tag{9}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\odot$ represents element-wise multiplication. The parameter $a \in (0, 1)$ is a learnable recurrent weight, which is constrained via the parameterization $a = \sigma(\Lambda)$ to ensure training stability and well-behaved temporal dynamics.

Through this design, the Residual Gating Path preserves stable long-term memory while remaining responsive to recent inputs, thereby achieving an effective balance between long-term preference modeling and short-term adaptability. Notably, this path attains strong representation capacity without introducing self-attention mechanisms and their associated computational overhead, resulting in a favorable trade-off between inference efficiency and recommendation performance.

C. Length-Aware Adaptive Fusion Module

In our framework, we establish three distinct pathways to effectively capture user interaction characteristics: the Standard Mamba focuses on global contextual dependencies [7], the Temporal Mamba emphasizes recent behaviors through temporal weighting, and the Residual Gating Path [15] preserves sequential dependencies while modeling both long-term and recent preferences. However, relying solely on concatenation or static weighting fails to achieve a balanced integration across these representations, often leading to redundancy or bias that limits recommendation accuracy. To address this issue, we propose the **Length-Aware Adaptive Fusion Module (AFM)**, which introduces a length-aware embedding and a gating network to dynamically allocate fusion weights. Specifically, **AFM** first takes three input representations: the Standard Mamba representation $\boldsymbol{H}_{\text{std}}$, the Temporal Mamba

representation $\boldsymbol{H}_{\text{temporal}}$, and the Residual Gating representation $\boldsymbol{H}_{\text{trg}}$. we incorporate a length embedding $\boldsymbol{E}_\ell$, wherein the original sequence length ℓ is normalized and subsequently projected into the corresponding embedding space $\boldsymbol{E}_\ell$:

$$\boldsymbol{E}_\ell = \text{ReLU}\left(\boldsymbol{W}_\ell \left(\frac{\ell}{L_{\text{norm}}}\right) + \boldsymbol{b}_\ell\right) \in \mathbb{R}^{B\times d} \tag{10}$$

where B is the batch size, d is the hidden dimension, and L_{norm} is set to the maximum sequence length used in training, to keep the normalized length in a stable range. $\boldsymbol{E}_\ell$ is concatenated with the three representations to form the input $\boldsymbol{Z}$ for the gating network:

$$\boldsymbol{Z} = [H_{std}; H_{temporal}; H_{trg}; \boldsymbol{E}_\ell] \in \mathbb{R}^{B\times L\times 4d} \tag{11}$$

The gating network, implemented as a lightweight feed-forward structure, computes normalized weights $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = \text{Softmax}\left(W_2 \cdot \text{GELU}(W_1 Z + b_1) + b_2\right) \in \mathbb{R}^{B\times L\times 3} \tag{12}$$

Finally, AFM performs weighted fusion as:

$$\boldsymbol{H_{fused}} = \alpha_1 \odot H_{std} + \alpha_2 \odot H_{temporal} + \alpha_3 \odot H_{trg} \tag{13}$$

where $\odot$ denotes element-wise multiplication. In essence, the strength of AFM lies in its adaptivity: by leveraging length-aware embeddings and gating mechanisms, it effectively overcomes the limitations of static fusion and produces more robust user embeddings, thereby improving recommendation performance across diverse scenarios.

D. Training

The DTMFRec model is trained to minimize the Cross-Entropy (CE) loss [16] for next-item prediction. Given an input sequence of item embeddings $\boldsymbol{E}_0 = e_1, e_2, \ldots, e_L$, where each item embedding $e_i \in \mathbb{R}^d$,the multi-path encoder jointly with the adaptive fusion module produces a fused hidden representation sequence, denoted as H_{fused}.We define the sequence-level representation h as the hidden state corresponding to the last valid position in the sequence.This representation aggregates global contextual information, temporally weighted signals, and residual-gated long-term memory through the proposed multi-path fusion architecture, thereby capturing the user's most recent preference state.

To obtain prediction scores (logits) over the item vocabulary, the model computes a scaled dot product between the sequence representation $\boldsymbol{h}$ and the item embedding matrix $\mathbf{E} \in \mathbb{R}^{K\times d}$ [17], where $K = |V|$ denotes the total number of items in the vocabulary V:

$$\text{logits} = \frac{\mathbf{h}\cdot\mathbf{E}^\top}{\tau}. \tag{14}$$

Here, $\tau > 0$ is a temperature parameter that modulates the sharpness of the output distribution. Although it does not alter the relative ranking of items, temperature scaling facilitates numerical stability and improves optimization by smoothing the softmax probabilities.

The Cross-Entropy loss [16] is then defined as:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{B}\sum_{i=1}^{B} \log\left(\frac{\exp(\text{logits}_{i,y_i})}{\sum_{j=1}^{K}\exp(\text{logits}_{i,j})}\right) \tag{15}$$

where B denotes the batch size, K is the number of items in the vocabulary, and y_i represents the index of the ground-truth next item for the i-th training instance. Minimizing this objective encourages the model to assign higher probabilities to relevant future items, thereby improving next-item prediction performance.

TABLE I: The statistics of datasets

Dataset	# Users	# Items	Sparsity	Avg. Length
Yelp	82,900	64,210	99.98%	9.68
Sports	75,185	48,567	99.98%	8.07
Beauty	22,364	12,102	99.93%	8.88
ML-1M	6,041	3,417	95.53%	165.60
Games	55,145	17,287	99.94%	9.01

III. EXPERIMENTS

A. Datasets and Metrics

In this paper, we conduct extensive experiments on five representative real-world datasets, including Yelp, the Amazon product review datasets (Beauty, Sports and Games), and MovieLens-1M. All datasets are preprocessed using the RecBole framework, and the detailed statistics after preprocessing are summarized in Table I.

For performance evaluation, we adopt three widely used metrics in recommender systems: Hit Rate at Top-10 (HR@10), Normalized Discounted Cumulative Gain at Top-10 (NDCG@10), and Mean Reciprocal Rank at Top-10 (MRR@10) [6]. These metrics are standard in sequential recommendation and collectively provide a comprehensive assessment of ranking accuracy, position sensitivity, and recommendation quality.

B. Implementation Details

All experiments are conducted using PyTorch 2.4.0, RecBole 1.2.1 [18], Mamba-SSM 2.2.3, and Causal-Conv1D 1.5.0.post4. We employ the Adam optimizer for model training. All experiments are performed on a single NVIDIA RTX 4090 GPU. To ensure a fair comparison, the embedding dimension of all evaluated models is uniformly set to 64.

C. Baselines

We compare DTMFRec with representative sequential recommenders from three families. (i) RNN-based: GRU4Rec. (ii) Transformer-based: BERT4Rec [17] and SASRec [4] as canonical attention-based sequential models, together with recent variants LinRec [19] and FEARec [20]. (iii) SSM-based: Mamba4Rec [12], ECHOMamba4Rec [14] and SIGMA [6]. All baselines are evaluated under the same data split and metrics, and we align the embedding dimension to $d = 64$ for fair comparison.

TABLE II: Performance comparison of different models across five datasets using HR@10, NDCG@10, and MRR@10 metrics. Best results are highlighted in bold.

Datasets	Eval Metrics	GRU4Rec	BERT4Rec	SASRec	LinRec	FEARec	Mamba	ECHO	SIGMA	DTMFRec
Yelp	HR@10	0.0441	0.0489	0.0551	0.0579	0.0554	0.0552	0.0578	0.0629	**0.0680**
	NDCG@10	0.0296	0.0317	0.0354	0.0382	0.0391	0.0344	0.0389	0.0412	**0.0427**
	MRR@10	0.0218	0.0243	0.0297	0.0322	0.0321	0.0290	0.0302	0.0346	**0.0350**
Sports	HR@10	0.0523	0.0579	0.0721	0.0709	**0.0746**	0.0676	0.0689	0.0735	0.0743
	NDCG@10	0.0486	0.0501	0.0546	0.0541	0.0575	0.0563	0.0569	0.0590	**0.0608**
	MRR@10	0.0453	0.0477	0.0513	0.0501	0.0521	0.0527	0.0534	0.0556	**0.0566**
Beauty	HR@10	0.0612	0.0764	0.0852	0.0837	0.0967	0.0880	0.0903	0.0986	**0.1022**
	NDCG@10	0.0334	0.0395	0.0532	0.0519	0.0530	0.0540	0.0567	0.0604	**0.0625**
	MRR@10	0.0242	0.0285	0.0392	0.0371	0.0410	0.0436	0.0447	0.0488	**0.0504**
ML-1M	HR@10	0.2944	0.2977	0.2998	0.3102	0.3283	0.3253	0.3239	0.3308	**0.3374**
	NDCG@10	0.1652	0.1687	0.1692	0.1764	0.1843	0.1868	0.1848	0.1906	**0.1947**
	MRR@10	0.1252	0.1294	0.1279	0.1357	0.1459	0.1413	0.1429	0.1479	**0.1512**
Games	HR@10	0.1484	0.1502	0.1592	0.1604	0.1616	0.1564	0.1578	0.1627	**0.1698**
	NDCG@10	0.0964	0.0978	0.1002	0.1021	0.1032	0.1050	0.1044	0.1088	**0.1130**
	MRR@10	0.0735	0.0728	0.0794	0.0824	0.0843	0.0894	0.0887	0.0924	**0.0957**

IV. RESULTS

A. Main Results

Table II summarizes the performance of all compared methods on the recommendation task. As shown in the table, our proposed DTMFRec framework consistently outperforms most baseline methods across various evaluation metrics, clearly demonstrating the effectiveness of deeply integrating the Mamba architecture [7] into sequential recommendation.

SSM-based models often struggle to capture short-term user interests.Attempts to enhance short-term modeling by reversing the input sequence may disrupt the temporal order, thereby limiting their practical applicability.In contrast, our proposed Multi-Path Dynamic Temporal Weighting module introduces a temporal weighting path that dynamically assigns varying importance to different timesteps within the sequence, enabling more accurate modeling of recent preferences.Furthermore, we design a length-aware adaptive fusion module that dynamically integrates information from multiple paths,allowing the model to more precisely identify user preferences and deliver superior recommendation performance.

B. Efficiency Comparison

In this subsection, we evaluate the inference efficiency of **DTMFRec** in comparison with representative Transformer-based and SSM-based sequential recommendation models, with results reported in Table III. Transformer-based methods, due to their quadratic time complexity, typically incur higher GPU memory consumption and longer inference latency. In contrast, Mamba-based models benefit from linear-time sequence modeling, offering superior efficiency. Although **DTMFRec** introduces additional computational overhead through its multi-path architecture and length-aware adaptive fusion module, it remains significantly more efficient than Transformer-based approaches, achieving a better balance between efficiency and effectiveness.

TABLE III: Efficiency analysis on the Beauty dataset

Dataset	Model	GPU Mem	Inf. Time	NDCG@10
Beauty	SASRec	7.58G	123ms	0.0532
	FEARec	8.11G	129ms	0.0530
	Mamba	2.58G	72ms	0.0540
	SIGMA	2.89G	68ms	0.0604
	Ours	**3.11G**	**76ms**	**0.0625**
Games	SASRec	7.23G	189ms	0.1002
	FEARec	7.98G	260ms	0.1032
	Mamba	3.40G	174ms	0.1050
	SIGMA	3.11G	171ms	0.1088
	Ours	**4.23G**	**182ms**	**0.1130**

TABLE IV: Ablation analysis on the Beauty dataset

Model Components	HR@10	NDCG@10	MRR@10
Default	**0.1022**	**0.0625**	**0.0504**
w/o Temporal Mamba	0.0993	0.0612	0.0496
w/o Residual Path	0.0932	0.0569	0.0457
w/o Adaptive Fusion	0.1008	0.0612	0.0491
w/o length embedding	0.0955	0.0575	0.0462

C. Grouped Users Analysis

This section analyzes recommendation performance across user groups with varying interaction history lengths, with the goal of gaining deeper insights into the effectiveness of DTMFRec in improving the experience of long-tail users. Figure 3 reports the results on the Beauty dataset, from

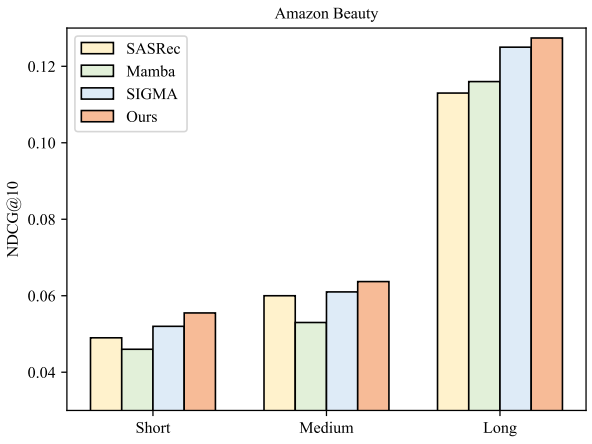

Fig. 3: User group analysis on Beauty.

which several observations can be drawn. Mamba4Rec, which directly applies the original Mamba architecture to sequential recommendation, exhibits inferior performance for users with short and medium interaction histories. In contrast, SIGMA, by incorporating partial sequence reversal and bidirectional modeling, achieves consistently superior results across all user groups. Notably, DTMFRec outperforms all baseline methods for every user group, demonstrating a favorable efficiency–performance trade-off.

D. Ablation Study

The ablation study results reported in Table IV confirm the importance of each component in our framework. Removing any module results in a significant performance drop, highlighting the critical and complementary roles these components play in modeling dynamic user preferences.

V. CONCLUSION

This paper introduces Dynamic Temporal Weighted Multi-Path State Space Fusion (DTMFRec), a framework for sequential recommendation that effectively models users' evolving interests.It employs a multi-path architecture to process both the original and temporally weighted sequences and a Length-Aware Adaptive Fusion module to adaptively integrate these paths. Extensive experiments on five real-world datasets demonstrate the significant performance gains of DTMFRec and validate the effectiveness of each module.

ACKNOWLEDGMENT

This work is supported by the Program of Innovation Improvement for Small and Medium-sized Enterprises of Shandong (No.2025TSGCCZZB0116), the Program of New Twenty Policies for Universities of Jinan (No.202333008), the Pilot Project for Integrated Innovation of Science, Education, Industry of Qilu University of Technology (Shandong Academy of Sciences) (No.2025ZDZX01), and Shandong Talent Introduction Program (No.WSR2025005).

REFERENCES

[1] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the eleventh ACM international conference on web search and data mining*, 2018, pp. 565–573.

[2] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1419–1428.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[4] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018, pp. 197–206.

[5] F. D. Keles, P. M. Wijewardena, and C. Hegde, "On the computational complexity of self-attention," 2022. [Online]. Available: https://arxiv.org/abs/2209.04881

[6] Z. Liu, Q. Liu, Y. Wang, W. Wang, P. Jia, M. Wang, Z. Liu, Y. Chang, and X. Zhao, "Sigma: Selective gated mamba for sequential recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 12, 2025, pp. 12 264–12 272.

[7] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[8] S. De, S. L. Smith, A. Fernando, A. Botev, G. Cristian-Muraru, A. Gu, R. Haroun, L. Berrada, Y. Chen, S. Srinivasan *et al.*, "Griffin: Mixing gated linear recurrences with local attention for efficient language models," *arXiv preprint arXiv:2402.19427*, 2024.

[9] H. Fang, D. Zhang, Y. Shu, and G. Guo, "Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations," *ACM Transactions on Information Systems (TOIS)*, vol. 39, no. 1, pp. 1–42, 2020.

[10] Q. Liu, X. Wu, Y. Wang, Z. Zhang, F. Tian, Y. Zheng, and X. Zhao, "Llm-esr: Large language models enhancement for long-tailed sequential recommendation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 26 701–26 727, 2024.

[11] X. Zhao, M. Wang, X. Zhao, J. Li, S. Zhou, D. Yin, Q. Li, J. Tang, and R. Guo, "Embedding in recommender systems: A survey," *arXiv preprint arXiv:2310.18608*, 2023.

[12] C. Liu, J. Lin, J. Wang, H. Liu, and J. Caverlee, "Mamba4rec: Towards efficient sequential recommendation with selective state space models," *arXiv preprint arXiv:2403.03900*, 2024.

[13] M. Rahmani, J. Caverlee, and F. Wang, "Incorporating time in sequential recommendation models," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 784–790.

[14] Y. Wang, X. He, and S. Zhu, "Echomamba4rec: Harmonizing bidirectional state space models with spectral filtering for advanced sequential recommendation," 2024. [Online]. Available: https://arxiv.org/abs/2406.02638

[15] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[16] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[17] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.

[18] L. Xu, Z. Tian, G. Zhang, J. Zhang, L. Wang, B. Zheng, Y. Li, J. Tang, Z. Zhang, Y. Hou, X. Pan, W. X. Zhao, X. Chen, and J. Wen, "Towards a more user-friendly and easy-to-use benchmark library for recommender systems," in *SIGIR*. ACM, 2023, pp. 2837–2847.

[19] L. Liu, L. Cai, C. Zhang, X. Zhao, J. Gao, W. Wang, Y. Lv, W. Fan, Y. Wang, M. He *et al.*, "Linrec: Linear attention mechanism for long-term sequential recommender systems," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 289–299.

[20] X. Du, H. Yuan, P. Zhao, J. Qu, F. Zhuang, G. Liu, Y. Liu, and V. S. Sheng, "Frequency enhanced hybrid attention network for sequential recommendation," in *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, 2023, pp. 78–88.