TriMLP: Revenge of a MLP-like Architecture in Sequential Recommendation

Yiheng Jiang\textsuperscript{1,2}, Yuanbo Xu\textsuperscript{1,2,\textsuperscript{go}}, Yongjian Yang\textsuperscript{1,2}, Funing Yang\textsuperscript{1,2}, Pengyang Wang\textsuperscript{3}, Hui Xiong\textsuperscript{4,5}

\textsuperscript{1}Lab of Mobile Intelligence (MIC), Jilin University, Changchun, China
\textsuperscript{2}College of Computer Science and Technology, Jilin University, Changchun, China
\textsuperscript{3}Department of Computer and Information Science, University of Macau, Macao SAR, China
\textsuperscript{4}Thrust of Artificial Intelligence, Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China
\textsuperscript{5}Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong SAR, China

\textsuperscript{go}jiangyh22@mails.jlu.edu.cn;\textsuperscript{yyj, yfn}@jlu.edu.cn;\textsuperscript{pywang@um.edu.mo;\textsuperscript{xionghui@ust.hk}}

Abstract—Sequential recommenders concentrate on modeling the transmission patterns shrouded in sequences of historical user-item interactive behaviors (or referred as token) and inferring dynamic preferences over candidate items. Fueled by diverse advanced neural network architectures like RNN, CNN and Transformer, existing methods have enjoyed rapid performance boost in the past years. Recent progress on MLP lights on an efficient method, token-mixing MLP, to establish contact among tokens. However, due to the unrestricted cross-token communications, i.e., information leakage issue, caused by the inherently fully-connection structure, we find that directly migrating these modern MLPs in recommendation task would neglect the chronological order of historical sequences and lead to subpar performances. In this paper, we present a MLP-like architecture for sequential recommendation, namely TriMLP, with a novel Triangular Mixer for cross-token communications. In designing Triangular Mixer, we simplify the cross-token operation in MLP as the basic matrix multiplication, and drop the lower-triangle neurons of the weight matrix to block the anti-chronological order connections from future tokens. Accordingly, the information leakage issue can be remedied and the prediction capability of MLP can be fully excavated under the standard auto-regressive mode. Take a step further, the mixer significantly alternates two delicate MLPs with triangular shape, tagged as global and local mixing, to separately capture the long range dependencies and local patterns on fine-grained level, i.e., long and short-term preferences. Empirical study on 12 datasets of different scales (50K\textasciitilde10M user-item interactions) from 4 benchmarks (Amazon, MovieLens, Tenrec and LBSN) show that TriMLP consistently attains promising accuracy/efficiency trade-off, where the average performance boost against several state-of-the-art baselines achieves up to 14.88\% with 8.65\% less inference cost. Our code is available at https://github.com/jiangyiheng1/TriMLP.

Index Terms—sequential recommendation, data mining, multi-layer perceptron

I. INTRODUCTION

Sequential recommendation processes sequences of historical user-item interactive behaviors (or referred as tokens in this paper), concentrates on mining dependencies among tokens and inferring preferences over time, and provides pleasing suggestions [1]. The performances of sequential recommenders are closely tied with the reliant neural network architectures, which serve as essential components for establishing contact among tokens and capturing transformation patterns.

![Fig. 1. Accuracy/Efficiency traded-off on QB-Video. Along the vertical axis, the higher, the better recommendation performance; along the horizontal axis, the more left, the less inference cost.](image-url)

Innovations in neural network architectures have consistently played a major role in sequential recommendation. Recurrent neural network (RNN)-based sequential recommenders, represented by [2], [3], transmits the information in tokens step-by-step. Methods like [4], [5], adopting convolutional neural network (CNN), aggregate the local spatial features with sliding filters. Credited to the superb adaptability with sequential tasks, Transformer architecture [6] that dynamically re-weights and integrates tokens through the self-attention mechanism has become the de-facto backbone in modern sequential recommenders [7]–[9].

In most recent, “retrospective” research on purely multi-layer perceptron-based (MLP) models, pioneered by MLP-Mixer [10] and ResMLP [11], investigates an conceptually simple and computationally efficiency idea to realize the cross-token communication, namely token-mixing MLP, where tokens interact with each other independently and identically across dimensions [12]. In terms of structural properties, token-mixing MLP holds the global reception field as Transformer while reserves the learned sequential dependency as...
In this paper, we propose to build a MLP-like architecture for sequential recommendation, with the aim of persisting the computationally efficiency advantage, gearing to the auto-regressive training fashion and catching up to the performances obtained using advanced neural network architectures.

In designing the MLP-based token mixer, we present Triangular Mixer to remedy the issues brought by fully-connection. It is inspired by the use of masking strategy in Transformer-based methods [9]. In principle, since the cross-token interactions endowed by MLP can be simplified as the matrix multiplication, undesirable interactions can be forbidden by disabling specific neurons in MLP. In practice, we drop the lower-triangle elements in the weight matrix of MLP to block the connections from future tokens and ensure that each token can only attend to itself and previous ones. Naturally, the information leakage issue is avoided, and the potential of MLP can be fully excavated under the auto-regressive training.

Take a step further, since MLP with global reception field excels in modeling the long-range relations among tokens while fails in capturing local patterns [24], we derive two mixing layers based on the above delicate MLP with triangular shape, tagged as global mixing and local mixing. The global mixing follows the vanilla triangular shape and attaches importance to all tokens in sequences for inferring long-term preference. The local one further drops specific upper-triangle neurons of weight matrix and treats the input sequence as multiple non-overlapped independent sessions with equal length. Specifically, the shape of active neurons is converted as several isosceles right sub-triangles arranged along the main diagonal whose sides are equal to the session length. Each sub-triangle is responsible for capturing the short-term preference contained in the corresponding session. Triangular Mixer serially alternates global mixing and local mixing for the fine-grained sequential dependency modeling.

To this end, we present a MLP-like sequential recommender TriMLP based on the proposed Triangular Mixer. In summary, our major contributions can be listed as follows:

- We refer and explore the idea of all-MLP architecture in sequential recommendation. To the best of our knowledge, we are the first to empirically point out that the fully-connection in MLP is not compatible with the standard auto-regressive training mode.
- We present a MLP-like sequential recommender, namely TriMLP, with a novel Triangular Mixer which endows the chronological interactions among tokens.
- We put forward Triangular Mixer with the global mixing and local mixing, to reconcile the long-rang dependencies and local patterns in sequences.
- We evaluate TriMLP with 12 datasets of different scales from 4 benchmarks (MovieLens, Amazon, Tenrec and LBSN) which contain 50K~10M user-item interactive behaviors. The experimental results demonstrate that TriMLP attains stable and promising accuracy/efficiency trade-off over all validated datasets, i.e., averagely surpasses the performance of several state-of-the-art baselines by 14.88% with 8.65% less inference cost.
II. RELATED WORK

A. Sequential Recommendation

Sequential recommendation aims at capturing dynamic preferences from sequences of historical user-item interactive behaviors and providing pleasant suggestions [1]. Building upon technological breakthroughs in the past decade [25]–[28], this field has ushered a new era of deep learning. Hidasi et al. [2] leveraged RNN to model the sequential dependency which transmits the information contained in token step-by-step. The spatial local information aggregation in CNN also benefits sequential recommenders [4], [5]. SASRec [7] and Bert4Rec [8] separately employed unidirectional and bidirectional Transformer-base encoder [6], [23] to dynamically extract the relationship between target and relevant items. Towards all-MLP methods, FMLP4Rec [29] referred the learnable filter-based denoising manner and encoded sequential patterns with Fourier transform, and MLP4Rec [30] incorporated contextual information (e.g., item characteristics) into the MLPMixer architecture [10]. In separate lines of research, [31], [32] utilized graph neural network (GNN) to enhance item representations, [33] adopted hierarchical structures, [34], [35] introduced data augmentation and [36]–[38] exploited pre-training techniques.

TriMLP architecture focuses on improving the primary sequential modeling capacity of MLP under the essential auto-regressive training mode without assistance of any auxiliary information. Credited to the triangular design, TriMLP successfully merges the performance gap between MLP and other advanced neural network-based sequential recommenders.

B. Toke-mixing MLP

Since the pioneering MLPMixer [10] and ResMLP [11] have been proposed in the early 2020s, all-MLP models are staging a comeback in vision community. These models rely on the novel MLP-based token mixer where tokens interact independently and identically across the channel dimension. Due to the simple concept with less intensive computation, such deep MLP models have stirred up a lot of interest [12] and derive a surge of variants. According to the dimensions of mixing tokens, these variants can be divided into three categories: (i) employing both the axial direction and the channel dimension [14], [16], [17], [39], [40] which proposes to orthogonally decompose the cross-token mixing, maintain long-range dependency and encode cross-token relation along axial directions, (ii) considering only the channel dimension [18]–[22], [41], [42] which aligns features at different spatial locations to the same channel by axial shifting, and then interacting with spatial information through channel projection. (iii) reserving the entire spatial and channel dimensions [10], [11], [15], [17], [24], [43] which retains the global reception field and channel projection.

In the pilot experiments, we empirically investigate that the inherent fully-connection design in MLP is incompatible with sequential tasks especially under the auto-regressive training fashion. In contrast, the proposed Triangular Mixer provides a simple, effective and efficient alternative to remedy the issue.

III. PRELIMINARY

A. Basic Definition

Let $U$ and $I$ denote the user and item set, respectively. Accordingly, we have the following basic definitions.

**Definition 1:** (User-Item Interactive Behavior) A user-item interactive behavior, or referred as token in this paper, is represented as a triplet $x = (u, i, t)$, which denotes that the user $u \in U$ interacted with the item $i \in I$ at time $t$.

**Definition 2:** (Historical Sequence) A historical sequence, tagged as $X^u = x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_{|X^u|}$, chronologically records $|X^u|$ user-item interactive behaviors of the user $u$.

B. Problem Statement

**Sequential Recommendation** Given the specific user $u$ and his/her historical sequence $X^u$. Sequential recommendation problem infers the dynamic preferences and provides the top-K recommendation list, which contains $K$ items that the user might be most likely to interact in the next time step. It can be formulated as the following equation,

$$F(X^u) \rightarrow \text{TopK}^u,$$

where $\text{TopK}^u$ denotes the top-$K$ recommendation list and $F(\cdot)$ is the abstract symbol of any sequential recommender.

IV. METHODOLOGY

A. Architecture Overview

The macro overview of TriMLP architecture is depicted in the Figure 3 (a). TriMLP takes a historical sequence of $n$ user-item interactive behaviors (or tokens) as input, where $n$ is the maximum sequence length. The tokens are independently pass through the Embedding layer to form the $d$-dimension sequence representation matrix. The resulting embedding is then fed into the Triangular Mixer to produce cross-token interactions. The Classifier takes these encoded representations as input, and predicts the probabilities over all candidate items.

B. Embedding

Considering that the historical sequences of different users are inconsistent in length, we set the maximum sequence length to $n$. Following the operation in [7], we split the longer sequences into several non-overlapping sub-sequences of length $n$. For the shorter ones, we repeatedly add the “padding” token in the head until their lengths grow to $n$. For the clarity and conciseness, we omit the superscript that denotes the specific user $u$, and the embedding layer can be formulated with the following equation,

$$\text{Embed}(X) \rightarrow X \in \mathbb{R}^{n \times d},$$

where $X$ is the sequence representation matrix. Note that the padding tokens are encoded with constant zero vectors [9] and excluded from the gradient update.
Fig. 3. (a) depicts the proposed MLP-like architecture TriMLP. (b) reveals an illustrative example with sequence length \( n = 4 \) and session length \( l = 2 \) to explain the cross-token interactions in global and local mixing.

### C. Triangular Mixer

Triangular Mixer endows the cross-token communication in strict compliance with chronological order. As shown in Figure 3 (a), the mixer takes as input the sequence representations \( X \), and encodes the sequential dependency through the global mixing layer and local mixing layer, successively. Formulically, it can be expressed as,

\[
Y = \text{TriMix}(X) = \text{Mix}_L\left(\text{Mix}_G\left(X^T\right)^T\right),
\]

where “\(^T\)” is the matrix transposition and \( Y \in \mathbb{R}^{n \times d} \) is the encoded sequence representations. The global mixing \( \text{Mix}_G(\cdot) \) injects the long-range sequential dependency and the local mixing \( \text{Mix}_L(\cdot) \) further captures the local patterns. These two mixing layers share the identical structure (as Figure 3 (b)), and can be expressed as,

\[
Y_{(G,L)} = \text{Mix}_L\left(\text{Mix}_G\left(X_{(G,L)}\right)\right) = \text{Act}(X_{(G,L)} \cdot \text{Softmax}(M_{(G,L)}))
\]

where \( M_{(G,L)} \in \mathbb{R}^{n \times n} \) are the mixing kernels in global and local mixing, respectively. i.e., linear combination, we utilize activation function \( \text{Act}(\cdot) \) to inject the non-linearity. We also adopt \( \text{Softmax}(\cdot) \) to convert the parameters in MLP as the probabilities over tokens. Note that we employ the unified notations in Eq. 4 for simplicity. Specifically, the input for global mixing \( X_G \in \mathbb{R}^{d \times n} \) is the transposed sequence embedding, and the corresponding output \( Y_G \in \mathbb{R}^{d \times n} \) also serves as the input for local mixing \( X_L \in \mathbb{R}^{d \times n} \). The output of local mixing \( Y_L \in \mathbb{R}^{d \times n} \) is transposed to form the final output of the mixer \( Y \).

Next, we devote into the details of the delicate \( M_{(G,L)} \) in global and local mixing, respectively.

1) **Global Mixing:** The triangular design in \( M_G \) gets insight from the utilization of mask strategy in Transformer-based sequential recommenders [7], [9], which masks the lower-triangular elements in the attentive map to prevent the information leakage. Similarly, we drop the lower-triangular neurons in \( M_G \) to cut off the contact from future tokens. The lower part of Figure 3 (c) provides an illustrative example with the input sequence length \( n = 4 \) to explain the cross-token communication, where “1, 2, 3, 4” denote the chronological order. \( \text{Mix}_G(\cdot) \) compels that the 2 nd token can only attend to the previous 1 st token and itself. Mathematically, the \( i \) th token in the global mixing interact with each other as

\[
Y_{G,i,j} = \sum_{j=1}^{i} X_{G,j} M_{G,i,j}, \quad \text{for} \quad i = 1, \ldots, n,
\]

where “\( \sum \)” denotes any dimension in \( d \) and the upper bound \( i \) of cumulative sum blocks the information from future tokens. In the premise of avoiding information leakage, global mixing \( \text{Mix}_G(\cdot) \) reserves the global reception field for the long-range sequential dependency.

2) **Local Mixing:** Based on the aforementioned global triangular design, local mixing \( \text{Mix}_L(\cdot) \) further selectively drops specific upper-triangle neurons to capture the local patterns, calling for the short-term preferences. In principle, it treats the input sequence as \( s \) non-overlapping sessions of length \( l \)
where \( n = s \times l \) and forbids the cross-session communications. In practice, the shape of active neurons in \( M_L \) converts into \( s \) isosceles right triangles of equal side length \( l \), which arrange along the main diagonal. The resulting sub-triangles are responsible for capturing the local patterns contained in corresponding sessions. As the example in Figure 3 (c) where the session length \( l = 2 \), the 4 th token attaches importance to the 3 rd token and itself, while the information from the 1 st and 2 nd token is ignored. Generally, the \( i \) th token in the local mixing interact with each other as

\[
Y_{L,i} = \sum_{j=1}^{i} X_{L,j} M_{L,i,j}, \quad \text{for } i = 1, \cdots, n, \quad (6)
\]

where the lower bound of cumulative sum, i.e., the round-up operation \( j = \lceil i/l \rceil \in [1, s] \), further cuts off the connections from previous sessions.

![Fig. 4. Cross-token in another variant of local mixing that endows the cross-session connections.](image)

We exploit another variant of local mixing which endows the cross-session communications\(^2\). It treats the input historical sequence of length \( n \) into \( s = n - l + 1 \) sessions with length \( l \). Accordingly, the active neurons in local mixing are reshaped as an isosceles trapezoid with the waist length of \( l \). The cross-session connections are built upon the overlapped tokens, i.e., for adjacent sessions, there are \( l - 1 \) over lapped tokens. Along with the example in Figure 4, the input sequence of length \( n = 4 \) is split into \( s = 3 \) sessions \( s_{0,1,2} \) of length \( l = 2 \), and the session \( s_0, s_1 \) are connected by the overlapped 2 nd token, et cetera. This variant interacts tokens as

\[
Y_{L,i} = \sum_{j=\max(1,i-l+1)}^{i} X_{L,j} M_{L,i,j}, \quad \text{for } i = 1, \cdots, n. \quad (7)
\]

The PyTorch-like pseudo-code of Triangular Mixer is presented in Algorithm 1. Since we convert the weights in \( M_{(G,L)} \) to the probabilities over tokens by \( \text{Softmax}(\cdot) \), the dropping operation can be easily implemented by padding these neurons with “\(-\infty\)”. Moreover, we find that initializing the active neurons with \( 1 \) which enforces each token contributes equally to the target during the early stage of training procedure\(^3\).

3) Discussion: Triangular Mixer shares some similarities with the self-attention mechanism [6], including the global reception field and parallel processing capability. Notably, our mixer departs from the self-attention mechanism with the following peculiarities:

- **Positional Sensitive**: Since Triangular Mixer compels the cross-token interactions in strict line with the chronological order, the extra positional information (e.g. Positional Encoding [6] or Embedding [23]) is no more necessary.

- **Independent and Static Weights**: Triangular Mixer reserves the sequential dependency as static weights in MLP which is agnostic to the input, rather than the attention map which is dynamically generated by the scale-dot product of query and key matrices.

- **Fewer Parameters and Higher Efficiency**: Triangular Mixer shrinks the parameter-scale by removing the query, key, value matrices mapping and Feed-Forward Network. Accordingly, our method is less computationally intensive than the self-attention mechanism.

### D. Classifier

Recall that the output of Triangular Mixer is denoted as \( Y \), the Classifier, i.e., implemented with the plain linear layer and Softmax function, converts the \( d \)-dimension representation vector of each token to the probabilities over all candidate items at each time step. As follows,

\[
P = \text{Softmax}(Y \cdot W + b), \quad (8)
\]

where \( W \in \mathbb{R}^{d \times |I|} \) and \( b \in \mathbb{R}^{|I|} \) are learnable parameters. \( P \in \mathbb{R}^{n \times |I|} \) is the calculated probability matrix where \( p_{i,c} \in [0, 1] \) is the probability over candidate item \( c \) at time step \( i \).

### E. Model Training and Recommendation

During the training processing, we apply the standard auto-regressive fashion. Specifically, TriMLP takes the historical sequence excluded the \( i \) th interacted item. We use the following cross entropy loss to optimize TriMLP,

\[
L = - \sum_{X \in X^u} \sum_{i=1}^{n} \log(p_{i,t_i}), \quad (9)
\]

where \( X^u \) is a training set of all users’ historical sequences , \( t_i \) is the target item at step \( i \) and \( p_{i,t_i} \) is the probability.

During the recommendation stage, TriMLP first extracts the last row \( p_n \in \mathbb{R}^{|I|} \) from \( P \) which contains the information of all interacted items in the historical sequence. Then, it ranks all candidate items according to the probabilities and retrieves \( K \) items as the top-K recommendation list.
V. EXPERIMENT AND DISCUSSION

In this section, we start from introducing the datasets, metrics, baselines and the implement details. Then, we analyze the experimental results, including the overall recommendation performance and ablation study. Take a step further, we explore various characteristics of our Triangular Mixer. In summary, we conduct a large amount of experiments to answer the following four research questions:

- **RQ 1** How is the recommendation performance and interference cost of TriMLP compared to other neural network-based state-of-the-art sequential recommenders?
- **RQ 2** How is the effectiveness of Triangular Mixer under the TriMLP architecture? Can global and local mixing both contribute to model the sequential dependency?
- **RQ 3** How is the property of Triangular Mixer with respect to various internal structures and micro designs?
- **RQ 4** How do the different settings of session number (or session length \( l \)) influence the performance of TriMLP and token-mixing manner in Triangular Mixer?

### A. Datasets

We evaluate our method on 12 publicly available datasets from 4 benchmarks. Specifically, we select **Beauty**, **Sports**, **ML-100K**, **ML-1M**, **ML-10M** from MovieLens\(^4\), **QB-Article**, **QB-Video** from Tenrec\(^6\) [13], and **NYC\(^7\)**, **TKY\(^7\)**, **Brightkite\(^8\)**, **Yelp\(^9\)**, **Gowalla\(^10\)** from the scenario of Location-based Social Network (LBSN).

In accordance with [9], we remove the “inactive” users who interact with fewer than 20 items and the “unpopular” items which are interacted by less than 10 times. According to the number of interactions, we categorize the 12 datasets into 4 different scales: Tiny, Small, Base and Large which separately contain 50K<100K, 150K<500K, 1M<2M and 2M<10M interactions.

Table I summarizes the statistics.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Scale</th>
<th># Users</th>
<th># Items</th>
<th># Interactions</th>
<th>Avg. Seq. Length</th>
<th>Max Seq. Length</th>
<th>Sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beauty</td>
<td>Tiny</td>
<td>1,664</td>
<td>38,918</td>
<td>56,558</td>
<td>33.99</td>
<td>32</td>
<td>99.91%</td>
</tr>
<tr>
<td>Sports</td>
<td>Tiny</td>
<td>1,498</td>
<td>59,084</td>
<td>56,044</td>
<td>50.05</td>
<td>32</td>
<td>99.90%</td>
</tr>
<tr>
<td>ML-100K</td>
<td>Tiny</td>
<td>952</td>
<td>1,182</td>
<td>97,746</td>
<td>104.18</td>
<td>128</td>
<td>90.90%</td>
</tr>
<tr>
<td>NYC(^4)</td>
<td>Small</td>
<td>1,031</td>
<td>5,135</td>
<td>142,237</td>
<td>137.06</td>
<td>128</td>
<td>97.31%</td>
</tr>
<tr>
<td>QB-Article</td>
<td>Small</td>
<td>4,671</td>
<td>1,844</td>
<td>184,919</td>
<td>159.53</td>
<td>32</td>
<td>98.99%</td>
</tr>
<tr>
<td>TKY(^7)</td>
<td>Small</td>
<td>2,267</td>
<td>7,973</td>
<td>444,183</td>
<td>195.93</td>
<td>128</td>
<td>97.51%</td>
</tr>
<tr>
<td>ML-1M</td>
<td>Tiny</td>
<td>6,034</td>
<td>2,361</td>
<td>988,428</td>
<td>165.47</td>
<td>128</td>
<td>94.92%</td>
</tr>
<tr>
<td>QB-Video</td>
<td>Tiny</td>
<td>10,047</td>
<td>15,608</td>
<td>1,370,177</td>
<td>71.96</td>
<td>64</td>
<td>99.54%</td>
</tr>
<tr>
<td>Brightkite</td>
<td>Tiny</td>
<td>5,714</td>
<td>48,181</td>
<td>1,765,247</td>
<td>308.93</td>
<td>256</td>
<td>99.50%</td>
</tr>
<tr>
<td>Yelp(^9)</td>
<td>Large</td>
<td>32,439</td>
<td>131,329</td>
<td>2,990,783</td>
<td>92.20</td>
<td>64</td>
<td>99.93%</td>
</tr>
<tr>
<td>Gowalla(^10)</td>
<td>Large</td>
<td>8,714</td>
<td>48,181</td>
<td>1,765,247</td>
<td>98.53</td>
<td>64</td>
<td>99.93%</td>
</tr>
<tr>
<td>ML-10M</td>
<td>Large</td>
<td>932</td>
<td>1,152</td>
<td>97,746</td>
<td>104.88</td>
<td>128</td>
<td>90.90%</td>
</tr>
<tr>
<td>NYCTK</td>
<td>Large</td>
<td>1,958</td>
<td>55,688</td>
<td>58,844</td>
<td>30.05</td>
<td>32</td>
<td>99.95%</td>
</tr>
</tbody>
</table>

### B. Metric

We introduce the following 3 metrics to measure the efficiency and accuracy of sequential recommenders.

- **Inference Time (Infer. Time)** calculates the average time cost of finishing 100 rounds recommendation.
- **Hit Rate (HR@K)** [46] counts the fraction of times that target item is among the top-K recommendation list.
- **Normalized Discounted Cumulative Gain (NDCG@K)** [46] rewards the method that ranks the positive items in first few positions of the top-K recommendation list.

The smaller Infer. Time stands for the better efficiency, and the recommendation performance is positively correlated with the values of HR and NDCG. We report \( K = \{5, 10\} \) in our experiments. To avoid the bias brought by different negative sampling strategies [47], we compare the probability of the target item with all other items in the dataset, and compute the HR, NDCG based on the ranking of all items.

### C. Baselines

Since TriMLP concentrates on improving the primary ability of MLP in encoding cross-token interactions, we compare it with the following four representative sequential recommenders developed from different neural networks:

- **GRU4Rec** [2] utilizes RNN to model historical sequences and dynamic preferences for sequential recommendation.
- **NextItNet** [4] is a state-of-the-art CNN-based generative model for recommendation, which learns high-level representation from both short and long-range dependencies.
- **SASRec** [7] employs Transformer-based encoder for recommendation where the self-attention mechanism dynamically models the sequential dependency.
- **FMLP4Rec** [29] is a state-of-the-art all-MLP sequential recommender, which follows the denoising manner and establishes cross-token contact by Fourier Transform.

### D. Implementation Details

For the rigorous comparison, we uniform the width and depth of TriMLP and baselines, and ensure that all the compared method differentiate only in the neural network architecture. Specifically, we set the embedding dimension \( d \) to 128, and the intermediate dimension in Triangular Mixer to \( n \). Since TriMLP contains 2 MLP layers for cross-token interactions, we stack 2 token-mixing encoders in each baseline.

During the training stage, we perform the standard auto-regressive manner, and adopt the identical gradient-updating strategy for all compared methods where dropout rate is 0.5 and the optimizer is Adam [48] with the learning rate of 0.001. The parameters keep updating until the performance no longer increases for consecutive 10 epochs. All experiments are conducted on a single server with 64GB RAM, AMD Ryzen 5900X CPU and NVIDIA RTX 3090 GPU.

### E. Overall Recommendation Performance (RQ 1)

The experimental results of all compared methods on 12 datasets are summarized in Table II. From the table, we have the following observations.

---

\(^{4}\)http://snap.stanford.edu/data/web-Amazon-links.html
\(^{5}\)https://grouplens.org/datasets/movielens/
\(^{6}\)https://static.qbiv.gq/com/qbiv/h/s algo-frontend/tenrec_dataset.html
\(^{7}\)https://sites.google.com/site/yangdingqi/home/foursquare-dataset
\(^{8}\)https://snap.stanford.edu/data/loc-brightkite.html
\(^{9}\)https://www.yelp.com/dataset
\(^{10}\)https://snap.stanford.edu/data/loc-Gowalla.html
Throughout all 12 validated datasets, TriMLP shows superior results across the board, consistently outperforming its competitors in both HR and NDCG. It demonstrates that our method equips MLP with the ample sequential modeling ability under the same training manner, which is competitive to RNN, CNN, Transformer and Fourier transform.

Observation 2: Incremental ascendancy in efficiency. According to the metric of Infer. Time, TriMLP saves the inference cost by 2.08%, 2.99%, 8.80% and 20.72% respectively on Tiny, Small, Base and Large datasets compared to the fastest competitor. The reduction increases with the scale of datasets which is in line with our expectation. Since the number of interactions are minor in Tiny and Small datasets, the computation is more intensive on the Embedding part and Classifier. Along with the number of interactions increases, i.e., mixing token occupies the main part of the computation, the efficiency advantages of TriMLP show up. It is credited to the plain structure in Triangular Mixer, which only contains 2 matrix transposition and multiplications. We provide a more detailed case study on the largest ML-10M in Section V-F to reveal the advantage of TriMLP in computational complexity.

Observation 3: Surprisingly good accuracy/efficiency tradeoff. Throughout all 12 validated datasets, TriMLP can averagely provide 14.88% higher recommendation performance against SOTA and reduce 8.64% inference time. The proposed TriMLP architecture reveals the promising potential to be served as an alternative for sequential recommenders.

Table II: Overall Recommendation Performance. The arrow "↑" (or "↓") denotes that the higher (or lower) value, the better metric. We use boldface and underline to indicate the best and second results in each column, respectively. The "Comparison" row reports the relative improvement or decline of TriMLP against the strongest baseline.

Table III: Computational Complexity Comparison on ML-10M.

F. Computational Complexity Analysis on ML-10M

We count the Multiply–Accumulate Operations (MACs), Parameter Scale (Para. Scale) and GPU Memory Occupation (GPU Mem.) on the largest dataset ML-10M. Since all the compared methods share the common Embedding part and Classifier, we only calculate MACs and Para. Scale of the encoder in each model, i.e., the RNN (or CNN) layer in GRU4Rec (or NextItNet) and the self-attention (or denoised Fourier) layer in SASRec (or FMLP4Rec).

The experimental results are summarized in Table III. Albeit the computational complexity is quadratic correlated to the input sequence length $n$, TriMLP possesses the higher efficiency due to parallel and minimalist matrix multiplication. Notably, TriMLP shrinks 83.51% MACs, 84.21% Para. Scale, 23.15% GPU Mem. and 23.73% Infer. Time.

G. Macro-Design of Triangular Mixer (RQ 2)

Recall our vanilla implementation of TriMLP contains the complete Triangular Mixer, including both global and local mixing (without cross-session interactions), we derive the following 4 variants to carry on the ablation study:

- **EyeMLP** replaces Triangular Mixer with the identity matrix where tokens no longer interact with each other.
remarkably boost the performance against Brightkite

<table>
<thead>
<tr>
<th>Model</th>
<th>HR@5</th>
<th>NDCG@5</th>
<th>HR@10</th>
<th>NDCG@10</th>
<th>Avg. Impv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EyeMLP</td>
<td>0.08135</td>
<td>0.09942</td>
<td>0.09769</td>
<td>0.06616</td>
<td>-</td>
</tr>
<tr>
<td>SqrMLP</td>
<td>0.04211</td>
<td>0.05621</td>
<td>0.07851</td>
<td>0.04573</td>
<td>-93.51%</td>
</tr>
<tr>
<td>TriMLP-G</td>
<td>0.08233</td>
<td>0.06177</td>
<td>0.11599</td>
<td>0.07281</td>
<td>-8.28%</td>
</tr>
<tr>
<td>TriMLP-L</td>
<td>0.0975</td>
<td>0.06218</td>
<td>0.09164</td>
<td>0.06574</td>
<td>7.12%</td>
</tr>
<tr>
<td>TriMLP</td>
<td>0.09615</td>
<td>0.07070</td>
<td>0.12560</td>
<td>0.08003</td>
<td>+21.57%</td>
</tr>
</tbody>
</table>

Evaluating the performance of serial-connected mixing layers jointly utilizing these two mixing layers is productive to the sole employing either global or local mixing. It shows that under the auto-regressive training mode. Therefore, the bidirectional particularity of MLP is sufficiently evokes the sequential modeling potential in MLP.

Finding 1: Fully-connection profoundly impairs performance. Compared to EyeMLP, SqrMLP erodes performances on all validated 12 datasets averagely by 77.99%. It unveils that the incompatibility between the fully-connection structure and the auto-regressive training fashion. The resulting information leakage is a serious and non-negligible issue which originally motivates this paper. We also explore the feasibility of adopting the auto-encoding manner to inegrate the bidirectional particularity of MLP in section V-K.

Finding 2: Triangular design does matter. Both TriMLP-G and TriMLP-L remarkably boost the performance against EyeMLP, where the leading margins achieve up to 27.55% and 26.05%, singly. It demonstrates that our triangular design sufficiently evokes the sequential modeling potential in MLP under the auto-regressive training mode.

Finding 3: Two mixing layers complement each other. Triangular Mixer constantly attains superior performance than solely employing either global or local mixing. It shows that jointly utilizing these two mixing layers is productive to the fine-grained modeling of long and short-term preferences. We compare the performance of serial-connected mixing layers with other internal structures in Triangular Mixer in Section V-H, and visualize how is the mutual influence among these two mixing branches in Section V-I2.

**H. Micro-Design of Triangular Mixer (RQ 3)**

This section analyzes the intrinsic properties of Triangular Mixer by decomposing it into various internal structures and operation components.

1) Sensitivity w.r.t. Different Internal Structures: Recall that the vanilla Triangular Mixer is denoted as Mix-G → Mix_L, we consider the following 3 variants:

- Mix_L → Mix_G follows the serial-connection. It first encodes the local patterns by local mixing, then models the long-range dependency with global mixing.
- Mix_G + Mix_L employs the parallel structure that combines the results of independent global and local mixing branches with element-wise addition.
- Mix_G || Mix_L employs the parallel structure that concatenates the results of independent global and local mixing branches and merges them with linear layer.

As depicted in Figure 5, the serial combinations generally perform better than the parallel ones on most datasets, Mix_G → Mix_L achieves more stable scores. Note that Mix_L → Mix_G has extremely poor performances on Tiny datasets. It might be caused by the split of historical sequences, where the inappropriate truncation leads to the non-uniform distribution of time intervals among user-item interactions. Thus, the short and long-term preferences might vary greatly, and the encoded local patterns would mislead the global one.

2) Sensitivity w.r.t. Various Operation Components: Recall that the vanilla Triangular Mixer drops the positional information and Feed-Forward Network, and utilizes the 1- initialization and Softmax normalization, we consider the following 4 alternatives:

- **w. PE** injects the absolute order information into the embedding [6] after the Embedding layer.
1) Influence on Performance: We mainly verify the influence brought by setting different session number \( s \) (or session length \( l \)) in local mixing, which decides the short-term preference modeling.

2) Influence on Reception Field: As shown in Figure 6, we find that independently modeling the short-term preferences in shorter sessions improves the performance on most datasets. On QB-Article, Brightkite and Yelp, we observe that modeling the dependency from the global perspective \( s = 1 \) is more suitable, while Gowalla and TKY reveal the contrast situation \( s = n \). The possible reason also lies in the split of historical sequences (as explained in Section V-H1). Generally, TriMLP achieves the best performance when setting \( s = 2 \) on NYC, QB-Video, ML-10M, \( s = 8 \) on ML-1M, \( s = 16 \) on Beauty, Sports and \( s = 32 \) on ML-100K.

2) Influence on Reception Field: We visualize the weights of global and local kernels on ML-1M \((n = 128)\) to explore how these two mixing layers complement each other. The corresponding 8 heat maps with different session numbers \( s \) (or session lengths \( l \)) are plotted in Figure 8. Accordingly, we observe the following characteristics:

Characteristic 1: Local kernels sustain more attention on the tokens around the current time step. We observe that the weights in all local kernels share the similar distribution, that the elements nearing the diagonal have greater absolute values than others, i.e., more active. When setting \( s = n = 128 \)
Table V
Sensitivity w.r.t. Various Operation Components in Global and Local Mixing. The best performance is boldfaced.
The vanilla implementation is marked with purple shading.

<table>
<thead>
<tr>
<th>Dataset-Tiny</th>
<th>Alternative</th>
<th>Beauty</th>
<th>Sports</th>
<th>ML-100K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer</td>
<td>w. PE</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00001</td>
</tr>
<tr>
<td></td>
<td>w. FFN</td>
<td>0.00102</td>
<td>0.00077</td>
<td>0.00664</td>
</tr>
<tr>
<td>Normalization</td>
<td>w.o. Softmax</td>
<td>0.01634</td>
<td>0.01507</td>
<td>0.01839</td>
</tr>
<tr>
<td>Triangular Mixer</td>
<td></td>
<td>0.00253</td>
<td>0.01492</td>
<td>0.06414</td>
</tr>
<tr>
<td></td>
<td>w.o. 1-0 Init</td>
<td>0.00609</td>
<td>0.03815</td>
<td>0.11695</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset-Small</th>
<th>Alternative</th>
<th>Beauty</th>
<th>Sports</th>
<th>NYC</th>
<th>QB-Article</th>
<th>TKY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer</td>
<td>w. PE</td>
<td>0.02619</td>
<td>0.02469</td>
<td>0.05723</td>
<td>0.02445</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w. FFN</td>
<td>0.03977</td>
<td>0.02226</td>
<td>0.05529</td>
<td>0.02750</td>
<td></td>
</tr>
<tr>
<td>Normalization</td>
<td>w.o. Softmax</td>
<td>0.04643</td>
<td>0.03975</td>
<td>0.05723</td>
<td>0.02990</td>
<td></td>
</tr>
<tr>
<td>Triangular Mixer</td>
<td></td>
<td>0.03956</td>
<td>0.02990</td>
<td>0.04524</td>
<td>0.03113</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w.o. 1-0 Init</td>
<td>0.04194</td>
<td>0.04433</td>
<td>0.13766</td>
<td>0.06523</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset-Base</th>
<th>Alternative</th>
<th>Beauty</th>
<th>Sports</th>
<th>QB-Video</th>
<th>Brightkite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer</td>
<td>w. PE</td>
<td>0.00819</td>
<td>0.00785</td>
<td>0.00451</td>
<td>0.00123</td>
</tr>
<tr>
<td></td>
<td>w. FFN</td>
<td>0.01950</td>
<td>0.01899</td>
<td>0.02068</td>
<td>0.01157</td>
</tr>
<tr>
<td>Normalization</td>
<td>w.o. Softmax</td>
<td>0.04194</td>
<td>0.04433</td>
<td>0.13766</td>
<td>0.06523</td>
</tr>
<tr>
<td>Triangular Mixer</td>
<td></td>
<td>0.07621</td>
<td>0.07193</td>
<td>0.04433</td>
<td>0.03700</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset-Large</th>
<th>Alternative</th>
<th>Beauty</th>
<th>Sports</th>
<th>Yelp</th>
<th>Gowalla</th>
<th>ML-10M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer</td>
<td>w. PE</td>
<td>0.00102</td>
<td>0.00077</td>
<td>0.00664</td>
<td>0.00527</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w. FFN</td>
<td>0.00112</td>
<td>0.00112</td>
<td>0.00112</td>
<td>0.00112</td>
<td></td>
</tr>
<tr>
<td>Normalization</td>
<td>w.o. Softmax</td>
<td>0.00570</td>
<td>0.00279</td>
<td>0.00319</td>
<td>0.00319</td>
<td></td>
</tr>
<tr>
<td>Triangular Mixer</td>
<td></td>
<td>0.00154</td>
<td>0.00963</td>
<td>0.02784</td>
<td>0.01357</td>
<td></td>
</tr>
</tbody>
</table>

(oriented Figure 8 (h)), the local kernel completely degrades into the identity matrix where all remaining diagonal elements are 1.

Figure 7. NDCG@5 comparison between w/o. cross-session interactions. The axes of all sub-figures stand for the variable session number s. Characteristic 2: Shorter sessions encourage the global kernel attaching more importance to previous tokens. Compared to Figure 8 (a), the upper-right elements in global kernels have greater absolute values (as Figure 8 (d–g)), which are responsible for the long-term user-item interactions. It indicates that the shorter sessions call for the larger reception field in global kernels to support the sufficient sequential dependency modeling. Unfortunately, such pattern seems to lose efficacy with s = {2, 4, 128} (as Figure 8 (b, c, h)), i.e.,
global kernels are no longer sharp to long-range tokens, which accord to the inferior performances in Figure 6 (g).

Characteristic 3: Suitable session settings produce superior performances. As the experimental results in Figure 6 (g), TriMLP achieves comparable and preferred scores when setting s = {8, 32} on ML–1M. Combined with the corresponding heatmaps in Figure 8 (d, f), besides the adequate short-term patterns offered by the local kernels, both of the global kernels in these two cases are more perceptive to the previous tokens. It proves the effectiveness of the serial structure in Triangular Mixer, that the global mixing layer and local mixing layer mutually assist each other indeed to realize the fine-grained modeling of sequential dependency.

J. Cross-session Communications in Local Mixing
Recall another variant of local mixing that endows the cross-session communications (Eq. 7), denoted as TriMLP_CS, we verify the performance on 4 datasets Sports, QB-Video, ML-100K and Brightkite, where the maximum sequence length n ranges from 32 to 256. TriMLP and TriMLP_CS share the same local mixing layer when s = {1, n}.

Figure 7 reports the NDCG@5 scores correlated with different session number s. It shows that connection sessions decreases the performance. The possible reasons are two-fold. On the one hand, cross-session connections bring more previous information to the current overlapped token after
multiple iterations, and degrade the local mixing as the global scheme, which disables the short-term preference modeling. On the other hand, since such connections are build solely upon the overlapped tokens, the corresponding neurons in the mixing kernel might accumulate more errors especially when the static weights are agnostic to the input sequence, which leads to the biased local pattern.

K. Auto-regressive V.S. Auto-encoding

Auto-encoding is the other popular training fashion, represented by [8], [23], that utilizes the past and future tokens to predict the current one. To verify whether or not the auto-encoding is compatible with MLP, we derive another variant BiMLP which stacks 2 MLP layers as the basic architecture, and train it on 4 datasets of different scales Beauty, NYC, ML-1M and Yelp under the auto-encoding manner. Specifically, we randomly mask the tokens in historical sequences with setting mask ratio \( r = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\} \). Among these masked tokens, 80% are padded with mask token, 10% are replaced with other tokens and 10% reserve the original ones.

Figure 9 reports the HR@5 scores correlated with different mask ratio. We find that the recommendation performance of BiMLP stands far behind TriMLP. The possible reason lies in that the auto-encoding training mode is more inclined to suffer from the data-hungry issue. It is more compatible with dense datasets for learning better mask representations, while sequential recommendation datasets are always extremely sparse (sparsity is usually larger than 99%). Moreover, this observation is also in line with [13], [29], [53], [54], that unidirectional models offer better results than bidirectional ones. Thus, we exploit the triangular design and put forward TriMLP under the unidirectional auto-regressive training scheme.
Fig. 8. Visualizing the weights of global and local mixing kernels with different session number $s$ on ML-1M. The global and local kernel are separately plotted in the upper and lower part of each sub-figure. Black indicates that the weight is 0, and the brighter, the greater the weight’s absolute value.

VI. Conclusion

In this paper, we make our exploration to study the capacity of MLP in sequential recommendation. We present the MLP-like sequential recommender TriMLP with a novel Triangular Mixer. Credited to the chronological cross-token communication and the serial mixing structure in Triangular Mixer, TriMLP successfully realizes the fine-grained modeling of sequential dependency. The experimental results on 12 datasets demonstrate that TriMLP attains stable, competitive and even better performance than several state-of-the-art baselines under the essential auto-regressive training mode with prominent less inference time, which well performs the “Revenge of MLP in Sequential Recommendation”.

In the future, we will further improve TriMLP by introducing auxiliary information like temporal factors and item attributes, data augmentation and pre-training techniques. Moreover, it is intriguing to deliberate how to decouple the strong correlation between the sequence length and MLP shape, which will enable MLP to flexibly handle sequences of different lengths.

REFERENCES
