Probabilistic Masked Attention Networks for Explainable Sequential Recommendation

Huiyuan Chen[1] , **Kaixiong Zhou**[2] , **Zhimeng Jiang**[3] , **Chin-Chia Michael Yeh**[1] , **Xiaoting Li**[1] , **Menghai Pan**[1] , **Yan Zheng**[1] , **Xia Hu**[2] , **Hao Yang**[1]
[1]Visa Research
[2]Rice University
[3]Texas A&M University
{hchen, miyeh, xiaotili, menpan, yazheng, haoyang}@visa.com, {Kaixiong.Zhou, xia.hu}@rice.edu, zhimengj@tamu.edu

Abstract

Transformer-based models are powerful for modeling temporal dynamics of user preference in sequential recommendation. Most of the variants adopt the Softmax transformation in the self-attention layers to generate dense attention probabilities. However, real-world item sequences are often noisy, containing a mixture of true-positive and false-positive interactions. Such dense attentions inevitably assign probability mass to noisy or irrelevant items, leading to sub-optimal performance and poor explainability. Here we propose a Probabilistic Masked Attention Network (PMAN) to identify the sparse pattern of attentions, which is more desirable for pruning noisy items in sequential recommendation. Specifically, we employ a probabilistic mask to achieve sparse attentions under a constrained optimization framework. As such, PMAN allows to select which information is critical to be retained or dropped in a data-driven fashion. Experimental studies on real-world benchmark datasets show that PMAN is able to improve the performance of Transformers significantly.

1 Introduction

Transformer [Vaswani *et al.*, 2017] and its variants have become the dominant architectures for language modeling tasks, due to their efficient parallel training and good ability of modeling long-range dependencies within sequences [Chen *et al.*, 2022b; Yeh *et al.*, 2022]. In light of this, many researchers have applied Transformers to understand the item-item dependencies within users' sequential actions, which has recently achieved remarkable performance in sequential recommendation. Some popular Transformer-based sequential models include SASRec [Kang and McAuley, 2018], BERT4Rec [Sun *et al.*, 2019], TiSASRec [Li *et al.*, 2020], Transformers4Rec [de Souza Pereira Moreira *et al.*, 2021], and STOSA [Fan *et al.*, 2022].

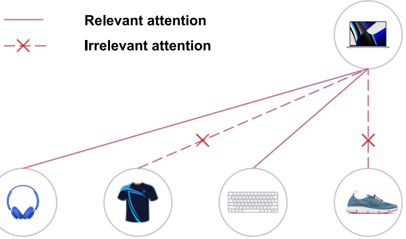

Figure 1: An example of a sequential data (*i.e.*, [*Headphones*, *T-shirts*, *Keyboards*, *Shoes*] → *Macbooks*) that contains both relevant and irrelevant item-item dependencies in the attention maps.

At the heart of the Transformer is the self-attention mechanism [Vaswani *et al.*, 2017], which offers insights of how a user makes decisions by inspecting the attention distribution to determine the amount of influence of each item in the decision-making process. However, the existing Transformer-based models often rely on *dense* and *fully-connected* attention maps: the attention probabilities are computed by the Softmax function, which always returns positive values [Kang and McAuley, 2018; Sun *et al.*, 2019; Li *et al.*, 2020]. As a result, dense attention maps inevitably assign certain credits to every item in the sequence for next-item recommendation, causing misleading explainability.

Recent studies show that the dense attentions are not necessary for real-world scenarios, where users' logged data is often noisy, containing a mixture of true-positive and false-positive interactions [Wang *et al.*, 2021b; Chen and Li, 2019a; Wang *et al.*, 2021c; Chen and Li, 2019b; Chen *et al.*, 2022a; Wang *et al.*, 2022]. For instance, as shown in Figure 1, a user's click logs may contain *Headphones*, *T-shirts*, *Keyboards*, *Shoes*, and *Macbooks* in a chronological order. Nevertheless, *T-shirts* and *Shoes* are irrelevant to *Macbooks*, leading to poor explanation in sequential recommendation. To address this issue, ADT [Wang *et al.*, 2021a] discards the large-loss samples with a dynamic threshold to identify and prune the noisy interactions. Rec-denoiser [Chen *et al.*, 2022a] learns a sparse attention to remove the redundant attentions. Nevertheless, Those frameworks suffers from large computational bottleneck, i.e, reweighting loss, multi-head attention, and double forward computation.

In this work, we propose a Probabilistic Masked Attention Network (PMAN) that is more effective for pruning noisy items in sequential recommendation. Our idea is rooted in the principle that explicitly sparsifying attention maps is able to improve the concentration of attention on the global context through an explicit selection of the most relevant segments [Child *et al.*, 2019; Correia *et al.*, 2019; Zaheer *et al.*, 2020; Beltagy *et al.*, 2020]. This phenomenon can be also verified by the recent *Lottery Ticket Hypothesis* [Frankle and Carbin, 2018; Chen *et al.*, 2020], showing that a sparse sub-network is enough to achieve good performance without training a dense network. To identify the sparse pattern of attentions, PMAN adopts a probabilistic mask to select task-specific items in a data-driven fashion, which provides better accuracy. In addition, PMAN uses the squared ReLU rather than Softmax as the activation function to compute attention distributions, abandoning the probabilistic constraints. This allows to generate exactly zero attention scores for irrelevant items, mitigating the impact of false-positive interactions.

Our PMAN can be formulated as a constrained optimization problem, which can be efficiently solved by standard projected gradient descent. More importantly, our PMAN is an easy-to-implement drop-in replacement for existing self-attention layers with no specialized operations. Remarkably, the sparse attentions are enough to obtain better accuracy than the standard Transformer with a fewer number of model parameters. Experimental studies on real-world benchmark datasets show that PMAN can significantly improve the performance of Transformers, and the performance gain becomes larger for more noisy sequences.

Overall, our contributions are summarized as follows:

- We propose a novel Probabilistic Masked Attention Network (PMAN), which simplifies the design of Transformers and is able to reduce the negative impacts of noisy items in sequential recommendation.
- We introduce a probabilistic masked mechanism to achieve sparse attention distributions, resulting in better performance. Our probabilistic mask can be learned automatically via a constrained optimization framework.
- We empirically demonstrate the effectiveness and efficiency of PMAN in real-world datasets. Besides the superior performance, our PMAN could provide a certain level of explainability, *i.e.*, eliminating the redundant item-item dependencies.

2 Related Work

2.1 Sequential Recommendation

Sequential recommendation aims to predict the next item based on a historical sequence of users' actions. Earlier studies mainly adopt Markov Chain models to learn local item-item transition patterns [Rendle *et al.*, 2010]. Deep sequential models have received much attention owing to their efficiency. Current efficient architectures include Recurrent Neural Networks [Hidasi *et al.*, 2016], Convolutional Neural Networks [Tang and Wang, 2018], and Graph Neural Networks [Wu *et al.*, 2019]. Recently, Transformer-based models have shown promising potential by using the self-attention mechanism to learn pairwise item-item relationships in the sequence. For example, one can adopt either left-to-right unidirectional attentions (*e.g.*, SASRec [Kang and McAuley, 2018] and TiSASRec [Li *et al.*, 2020]) or bidirectional attentions (*e.g.*, BERT4Rec [Sun *et al.*, 2019]) to predict the next item. SSE-PT [Wu *et al.*, 2020] is a personalized Transformer model that further incorporates user embeddings to enhance performance. Recently, LSAN [Li *et al.*, 2021] proposes a twin-attention network to learn both long- and short-term user preference via a dedicated self-attention operation.

However, vanilla self-attention mechanism uses the Softmax function to compute dense attention distributions, which is not Lipschitz continuous [Kim *et al.*, 2021], and is thus sensitive to noisy sequences. Rec-denoiser [Chen *et al.*, 2022a] learns a sparse attention to remove the redundant attentions but with expensive computational efforts due to the Softmax operator. Inspired by recent Softmax-free attention mechanisms [Hua *et al.*, 2022; Shazeer, 2020], we put forward a probabilistic attention network that consists of a probabilistic mask and the ReLU normalization, to prune the noisy items within a sequence. As such, the redundancy item-item dependencies can be removed efficiently, providing clean and sparse attention maps with better explainability.

2.2 Sparse Transformer

Learning sparse and efficient attention mechanisms has recently garnered considerable interest in language modeling [Child *et al.*, 2019; Correia *et al.*, 2019; Zaheer *et al.*, 2020; Beltagy *et al.*, 2020]. The key idea of these sparse Transformers is to sparsify the attention maps by fixing the field of view with pre-defined patterns. For example, Sparse Transformer [Child *et al.*, 2019] uses a fixed attention patterns, where specific cells summarize previous locations and propagate the information to all future cells. Longformer [Beltagy *et al.*, 2020] further increases the receptive field by employing dilated sliding window.

However, these sparse models highly depend on pre-defined attention schemes, which require domain-specific knowledge and lack flexibility [Yun *et al.*, 2020]. Also, the original purpose of these models is to capture long-range dependencies for long sequences, not to prune noise within sequences. Despite the extension is conceptually straightforward, this direction is less-explored for sequential recommendation. It is thus unclear whether these fixed sparse patterns could generalize well for noisy item sequences [Child *et al.*, 2019; Beltagy *et al.*, 2020]. In this paper, we follow a different route with the aim of learning sparse attentions via a probabilistic mask. As such, the mask can be simultaneously optimized with the downstream objective, which is able to remove the task-irrelevant items in sequential recommendation.

3 Background

3.1 Problem Setup

In the sequential recommendation tasks, let $\mathcal{U}$ be a set of users, $\mathcal{V}$ be a set of items, and $\mathcal{S} = \{\mathcal{S}^1, \mathcal{S}^2, \dots, \mathcal{S}^{|\mathcal{U}|}\}$ a collection of users' actions. Each user $u \in \mathcal{U}$ is associated with a sequence of items $\mathcal{S}^u = (S_1^u, S_2^u, \dots, S_{|\mathcal{S}^u|}^u)$ in a chronological order, where $|\mathcal{S}^u|$ is the length of the sequence, and $S^u_t \in \mathcal{V}$ is the item that user u has interacted with at time t. The sequential recommendation is commonly evaluated as next-item prediction [Hidasi *et al.*, 2016; Kang and McAuley, 2018]. For each user u, we seek to predict the next item $S^u_{|S^u|+1}$ at time step $|S^u|+1$ based on the interaction history $\mathcal{S}^u$.

3.2 Self-attention Network

Owing to the efficient parallel training, Transformers have been widely used in sequential recommendation [Kang and McAuley, 2018; Sun *et al.*, 2019; Li *et al.*, 2020; de Souza Pereira Moreira *et al.*, 2021]. Here we briefly introduce the design of self-attention mechanism.

Embedding Layer. Transformer-based recommenders maintain an item embedding table $\mathbf{T} \in \mathbb{R}^{|\mathcal{V}|\times d}$, where d is the size of embedding vector. For each sequence $\mathcal{S}^u$, it can be converted into a fixed-length sequence $(S^u_{|S^u|-n+1}, \ldots, S^u_{|S^u|})$, where n is the maximum length, such as keeping the most recent n items by truncating or padding items as needed [Kang and McAuley, 2018]). The embedding for $(S^u_{|S^u|-n+1}, \ldots, S^u_{|S^u|})$ is denoted as $\mathbf{E} \in \mathbb{R}^{n\times d}$, which can be retrieved from the embedding table $\mathbf{T}$. To preserve the time information, a learnable positional embedding $\mathbf{P} \in \mathbb{R}^{n\times d}$ is further constructed. Usually, the item embedding and the positional embedding are added up:

$$\mathbf{X} = \mathbf{E} + \mathbf{P}, \tag{1}$$

where the composited embedding $\mathbf{X} \in \mathbb{R}^{n\times d}$ can be directly fed to any sequential recommenders.

Self-Attention Layer. The self-attention layer is critical to learn long-range dependencies within a sequence [Vaswani *et al.*, 2017]. The scaled dot-product attention is widely used as:

$$\text{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \tag{2}$$

where $\mathbf{Q} = \mathbf{X}\mathbf{W}^Q$, $\mathbf{K} = \mathbf{X}\mathbf{W}^K$, and $\mathbf{V} = \mathbf{X}\mathbf{W}^V$ are the queries, keys and values, respectively; $\{\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V\} \in \mathbb{R}^{d\times d}$ are weights. Also, one can include other components like point-wise feed-forward layer, residual connection, and layer normalization. We skip the details as these parts are the same as the original Transformer [Vaswani *et al.*, 2017].

Limitations. The scaled dot-product attention in Eq. (2) always generates dense attention distributions, which complicates the item-item dependencies, and even degrades the performance. Moreover, recent studies show that Eq. (2) is not Lipschitz continuous [Kim *et al.*, 2021], implying that small perturbations (*e.g.*, noise) on input sequences are likely to cause large variances of attention distributions. This usually increases the training difficulty.

4 Our Proposed Framework

In this section, we present our Probabilistic Masked Attention Network (PMAN), a simpler yet more desirable for detecting noisy sequences in sequential recommendation.

4.1 Probabilistic Masked Attention Network

Sparse Attention. Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ (from Eq. (1)) be the representations over n items, we make the use of a much simpler attention mechanism to compute the attention distribution [Hua *et al.*, 2022]:

$$\mathbf{A} = \mathcal{Q}(\mathbf{Z})\mathcal{K}(\mathbf{Z})^T, \text{where } \mathbf{Z} = \phi_z(\mathbf{X}\mathbf{W}_z), \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ contains dense item-item attentions, $\mathbf{Z} \in \mathbb{R}^{n\times d}$ is a shared representation for both query $\mathcal{Q}(\mathbf{Z})$ and key $\mathcal{Q}(\mathbf{Z})$, where $\mathcal{Q}(\cdot)$ and $\mathcal{K}(\cdot)$ are two simple affine functions that apply per-dim scalars and offsets to $\mathbf{Z}$, $\mathbf{W}_z \in \mathbb{R}^{d\times d}$ is the weight, and $\phi_z(\cdot)$ is a nonlinear function.

To address the issue of noise, PMAN introduces masked mechanism that allows to control what information should be retained/dropped. Specifically, we have:

$$\begin{gathered}\mathbf{H} = \frac{1}{nd}\text{ReLU}^2(\mathbf{A} \odot \mathbf{M})\mathbf{V}, \\ \text{where } \mathbf{M} \in \{0,1\}^{n\times n}, \text{ and } \mathbf{V} = \phi_v(\mathbf{X}\mathbf{W}_v),\end{gathered} \tag{4}$$

where $\mathbf{H} \in \mathbb{R}^{n\times d}$ is the output item representations; $\odot$ stands for element-wise product; $\mathbf{V}$ is the value with weight $\mathbf{W}_v \in \mathbb{R}^{d\times d}$ and nonlinear function $\phi_v(\cdot)$; $\mathbf{M} \in \mathbb{R}^{n\times n}$ is a trainable binary mask that sparsifies the attention maps, *i.e.*, $\mathbf{M}_{ij} = 0$ means that the attention $\mathbf{A}_{ij}$ is pruned, otherwise $\mathbf{A}_{ij}$ is kept. Empirically, we choose the squared ReLU rather than Softmax as activation function. As such, the combination of the binary mask with squared ReLU normalization is able to yield exactly zero probabilities for irrelevant items. We will show how to jointly optimize the discrete variable $\mathbf{M}$ with model parameters later.

Loss Function. One can predict the next item (given the first t items) based on $\mathbf{H}_t$ (Eq. (4)). An inner product is used to predict users' preference score of item i as:

$$r_{i,t} = \langle \mathbf{H}_t, \mathbf{T}_i \rangle, \tag{5}$$

where $\mathbf{T}_i \in \mathbb{R}^d$ is the embedding of item i. Typically, Transformer feeds a sequence $(S^u_{|S^u|-n}, \ldots, S^u_{|S^u|-1})$ and its desired output is a shifted version of the same sequence $\mathbf{o} = (S^u_{|S^u|-n+1}, \ldots, S^u_{|S^u|})$. Finally, one can adopt the binary cross-entropy loss as objective function:

$$\mathcal{L}(\boldsymbol{\Theta}) = -\sum_{\mathcal{S}^u \in \mathcal{S}} \sum_{t=1}^{n} \left[\log(\sigma(r_{o_t,t})) + \log(1 - \sigma(r_{o'_t,t}))\right], \tag{6}$$

where $\boldsymbol{\Theta}$ is the model parameters, $o'_t \notin \mathcal{S}^u$ is a negative sample corresponding to o_t, and $\sigma(\cdot)$ is the sigmoid function.

4.2 Probabilistic Sparsification Framework

Constrained Optimization. We jointly train the model parameters $\boldsymbol{\Theta}$ and the binary mask $\mathbf{M}$ in a unified optimization framework. The empirical risk minimization is:

$$\begin{gathered}\min_{\boldsymbol{\Theta},\mathbf{M}} \quad \mathcal{L}(\boldsymbol{\Theta}, \mathbf{M}), \\ \text{s.t. } \|\mathbf{M}\|_1 \leq B \text{ and } \mathbf{M} \in \{0,1\}^{n\times n},\end{gathered} \tag{7}$$

where L_1 norm is applied to the mask to control the sparse degree of the attentions, with the upper bound B. Nevertheless, the objective is discrete with respect to $\mathbf{M}$, which is

Algorithm 1 PMAN

```
Input: The training sequence set $\mathcal{S}$, attention capacity $B$, embedding size $d$.
Initialize model parameters $\mathbf{\Theta}$, mask parameters $\mathbf{s}$;
for epoch $t = 1, 2, \ldots, T$ do
    Schedule the temperature parameter by $\tau = 0.97(1-t/T)+0.03$;
    for each mini-batch do
        Compute the input embedding $\mathbf{X}$ in Eq. (2);
        Compute dense attentions $\mathbf{A}$ in Eq. (3);
        Compute sparse attentions $\mathbf{H}$ in Eq. (4);
        Compute the entropy loss $\mathcal{L}$ in Problem (8);
        Draw samples $\{\boldsymbol{g}_\mathbf{1}^{(i)}, \boldsymbol{g}_\mathbf{0}^{(i)}\}_{i=1}^{I} \sim$ Gumbel$(0,1)$;
        Compute gradient $\nabla_{\boldsymbol{s}}\mathbb{E}_{p(\boldsymbol{m}|\boldsymbol{s})}\mathcal{L}(\mathbf{\Theta}, \boldsymbol{m})$ in Eq. (9);
        Update the mask variables $\mathbf{s}$ via PGD in Eq. (12);
        Update the model parameters $\mathbf{\Theta}$ via SGD;
    end for
end for
return 1) The model parameters $\mathbf{\Theta}$;
       2) The trained mask $\mathbf{m}$ that can be sampled from
       the distribution $p(\mathbf{m}|\mathbf{s})$.
```

computationally intractable. To overcome this issue, we view each element of mask $\mathbf{M}$ as a binary random variable, then Problem (7) can be relaxed into an excepted loss minimization problem over the probability spaces, which is continuous and differentiable for gradient computation.

Formally, we flatten the mask $\mathbf{M}$ as a vector $\boldsymbol{m} \in \mathbb{R}^{n^2}$, and view each element m_i as a Bernoulli random variable with probability s_i to be 1 and $(1-s_i)$ to be 0, that is $m_i \sim \mathrm{Bern}(s_i)$, where $s_i \in [0,1]$. Assuming the elements of variable $\boldsymbol{m}$ are independent, its distribution becomes $p(\boldsymbol{m}|\boldsymbol{s}) = \Pi_i (s_i)^{m_i}(1-s_i)^{(1-m_i)}$. Then we have $\mathbb{E}_{\boldsymbol{m}\sim p(\boldsymbol{m}|\boldsymbol{s})}\|\boldsymbol{m}\|_1 = \sum_{i=1}^{n^2} s_i$. As such, Problem (7) can be relaxed into the following excepted loss minimization:

$$\begin{aligned} \min_{\mathbf{\Theta}, \boldsymbol{s}} \quad & \mathbb{E}_{p(\boldsymbol{m}|\boldsymbol{s})}\mathcal{L}(\mathbf{\Theta}, \boldsymbol{m}), \\ \text{s.t. } & \mathbf{1}^\top \boldsymbol{s} \le B \text{ and } s_i \in [0,1]. \end{aligned} \tag{8}$$

where $\mathbf{1}$ is the all-one vector.

For above problem, we use an alternating optimization schema to iteratively update $\mathbf{\Theta}$ and $\boldsymbol{s}$. For model parameters $\mathbf{\Theta}$, it can be learned via standard Stochastic Gradient Descent. For variable $\boldsymbol{s}$, it involves a constrained optimization problem. Next we show how to efficiently update $\boldsymbol{s}$ by using Projected Gradient Descent (PGD).

Projected Gradient Descent. We adopt Gumbel Softmax trick [Jang *et al.*, 2017] to calculate the gradients of the binary variable $\boldsymbol{s}$ as:

$$\begin{aligned} & \nabla_{\boldsymbol{s}}\mathbb{E}_{p(\boldsymbol{m}|\boldsymbol{s})}\mathcal{L}(\mathbf{\Theta}, \boldsymbol{m}) \\ & = \mathbb{E}_{\boldsymbol{g}_\mathbf{0}, \boldsymbol{g}_\mathbf{1}} \nabla_{\boldsymbol{s}}\mathcal{L}\left(\mathbf{\Theta}, \mathbb{I}\left[\log(\frac{\boldsymbol{s}}{\mathbf{1}-\boldsymbol{s}}) + \boldsymbol{g}_\mathbf{1} - \boldsymbol{g}_\mathbf{0} \ge 0\right]\right) \\ & \approx \mathbb{E}_{\boldsymbol{g}_\mathbf{0}, \boldsymbol{g}_\mathbf{1}} \nabla_{\boldsymbol{s}}\mathcal{L}\left(\mathbf{\Theta}, \sigma\left(\frac{\log\left(\frac{\boldsymbol{s}}{1-\boldsymbol{s}}\right) + \boldsymbol{g}_\mathbf{1} - \boldsymbol{g}_\mathbf{0}}{\tau}\right)\right) \\ & \approx \frac{1}{I}\sum_{i=1}^{I} \nabla_{\boldsymbol{s}}\mathcal{L}\left(\mathbf{\Theta}, \sigma\left(\frac{\log\left(\frac{\boldsymbol{s}}{1-\boldsymbol{s}}\right) + \boldsymbol{g}_\mathbf{1}^{(i)} - \boldsymbol{g}_\mathbf{0}^{(i)}}{\tau}\right)\right), \end{aligned} \tag{9}$$

Dataset	#users	#items	#interactions	sparsity
Beauty	22.4k	12.1k	198.5k	0.07%
Sports	25.6k	18.4k	296.3k	0.05%
Yelp	30.4k	20.0k	316.4k	0.05%
MovieLens1M	6.0k	3.4k	987.0k	4.78%
Steam	334.7k	13.0k	3686.1k	0.08%

Table 1: Statistics of the benchmark dataset.

where $\mathbb{I}[\cdot]$ is the indicator function; $\boldsymbol{g}_0$ and $\boldsymbol{g}_1$ are two random variables, with each element being i.i.d sampled from Gumbel $(0,1)$ distribution. We use I pairs of Monte Carlo samples $(\boldsymbol{g}_1^{(i)}, \boldsymbol{g}_\mathbf{0}^{(i)})$ to approximate the expectation; $\sigma(\cdot)$ is the Sigmoid function; τ is the temperature that can be decreased linearly during training [Jang *et al.*, 2017].

Besides, we denote $C = \{\boldsymbol{s}|\mathbf{1}^\top\boldsymbol{s} \le B \text{ and } s_i \in [0,1]\}$ as the feasible region of Problem (8). For any vector $\boldsymbol{y}$, Proposition 1 shows that its projection on the set C has a closed-form solution. More details can be found in Appendix[1].

Proposition 1. *Given any vector* $\mathbf{y}$*, its projection on the set* $C = \{\boldsymbol{s}|\mathbf{1}^\top\boldsymbol{s} \le B, s_i \in [0,1]\}$ *is computed as:*

$$\mathit{proj}_C[\boldsymbol{y}] = \min(1, \max(0, \boldsymbol{y} - \alpha\mathbf{1})), \tag{10}$$

where $\alpha = \max(0, \beta)$*, and* β *is the solution of the equation:*

$$\mathbf{1}^\top[\min(1, \max(0, \boldsymbol{y} - \beta\mathbf{1}))] - B = 0. \tag{11}$$

In addition, let $f(\beta) = \mathbf{1}^\top[\min(1, \max(0, \boldsymbol{y} - \beta\mathbf{1}))] - B$, and it can be verified that $f(\beta)$ is a monotone decreasing function with respect to β. Thus, the equation $f(\beta) = 0$ can be efficiently solved by using the bisection method that converges in the logarithmic rate. After obtaining β^*, we can get $\alpha^* = \max(0, \beta^*)$, which can be then used to compute the projection in Eq. (10).

To this end, we can apply PGD to update $\boldsymbol{s}$ for Problem (8) by jointly considering Eq. (9) and Eq. (10):

$$\boldsymbol{s} \leftarrow \mathrm{proj}_C[\boldsymbol{s} - \eta \cdot \nabla_{\boldsymbol{s}}\mathbb{E}_{p(\boldsymbol{m}|\boldsymbol{s})}\mathcal{L}(\mathbf{\Theta}, \boldsymbol{m})] \tag{12}$$

where η denotes the learning rate. We briefly summarize the overall training procedure of PMAN in Algorithm 1.

Complexity Analysis. From Eq. (3) and Eq. (4), PMAN has quadratic complexity over the sequence length as the Transformers, but with a cheaper architecture. More importantly, we empirically observe that PMAN only requires one head, without the need of point-wise feed-forward layer, and layer normalization as in the original Transformer, which enhances computing efficiency. The running time of different Transformer-based models can be found in Appendix. Additionally, it is worth noting that our PMAN can easily achieve linear complexity by using low-rank matrix decomposition [Chen *et al.*, 2021] or Nyström approximation [Xiong *et al.*, 2021]. We leave the extension of linearized attentions in the future.

Model	Beauty		Sports		Yelp		MovieLens1M		Steam	
	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10
BPRMF	0.0536	0.0320	0.0289	0.0154	0.0240	0.0124	0.0547	0.0361	0.0143	0.0118
FPMC	0.0534	0.0311	0.0341	0.0178	0.0261	0.0134	0.0532	0.0340	0.0148	0.0112
GRU4Rec	0.0543	0.0318	0.0352	0.0180	0.0272	0.0138	0.0553	0.0368	0.0152	0.0122
Caser	0.0541	0.0323	0.0348	0.0183	0.0276	0.0141	0.0560	0.0357	0.0158	0.0129
SR-GNN	0.0563	0.0337	0.0350	0.0181	0.0281	0.0152	0.0567	0.0371	0.0164	0.0133
S^3-Rec	0.0595	0.0348	0.0362	0.0187	0.0303	0.0163	0.0574	0.0379	0.0171	0.0137
DSAN	0.0570	0.0327	0.0351	0.0179	0.0292	0.0155	0.0562	0.0369	0.0167	0.0130
STOSA	0.0638	0.0360	0.0381	0.0198	0.0297	0.0160	0.0577	0.0384	0.0177	0.0144
SASRec	0.0601	0.0324	0.0361	0.0183	0.0285	0.0148	0.0570	0.0372	0.0168	0.0131
+PMAN	0.0645	0.0357	0.0387	0.0201	0.0306	0.0157	0.0615	0.0398	0.0185	0.0147
%Improv	+7.32%	+10.2%	+7.20%	+9.84%	+7.37%	+6.08%	+7.89%	+6.98%	+10.1%	+12.2%
TiSASRec	0.0623	0.0346	0.0370	0.0190	0.0296	0.0159	0.0584	0.0393	0.0181	0.0153
+PMAN	**0.0656**	**0.0377**	**0.0403**	**0.0214**	**0.0314**	**0.0167**	**0.0635**	**0.0418**	**0.0199**	**0.0173**
%Improv	+5.30%	+8.96%	+8.92%	+12.6%	+6.08%	+5.03%	+8.73%	+6.36%	+9.94%	+12.07%
BERT4Rec	0.0617	0.0328	0.0353	0.0189	0.0287	0.0143	0.0576	0.0381	0.0160	0.0127
+PMAN	0.0649	0.0349	0.0376	0.0207	0.0305	0.0152	0.0627	0.0407	0.0175	0.0142
%Improv	+5.19%	+6.40%	+6.52%	+9.52%	+6.27%	+6.29%	+8.85%	+6.82%	+9.38%	+11.8%

Table 2: Overall Performance of different models ("%Improv" denotes the relative improvements of PMANs over their backbones). The best performing results are boldfaced and the second best ones are underlined.

5 Experiment

5.1 Experimental Setup

Dataset. We consider five benchmark datasets: Amazon-Beauty, Amazon-Sports[2], Yelp[3], MovieLens1M[4], and Steam[5]. For each dataset, we group the interactions by users, and sort their items by the timestamps ascendingly. Following [Fan *et al.*, 2022], we adopt 5-core setting to filter out unpopular items and inactive users with fewer than five interaction records. Their statistics are listed in Table 1.

Baseline. We compare our PMAN with the following methods: 1) **BPRMF** [Rendle *et al.*, 2009] is a matrix factorization model with Bayesian personalized ranking loss; 2) **FPMC** [Rendle *et al.*, 2010] utilizes Markov Chains to learn item transitions; 3) **GRU4Rec** [Hidasi *et al.*, 2016] adopts GRU to learn the item sequences; 4) **Caser** [Tang and Wang, 2018] is a CNN-based sequential model; 5) **SR-GNN** [Wu *et al.*, 2019] employs graph neural network to capture complex item transitions. 6) **SASRec** [Kang and McAuley, 2018], **TiSASRec** [Li *et al.*, 2020], **BERT4Rec** [Sun *et al.*, 2019] are all Transformer-based sequential models; 7) **S^3-Rec** [Zhou *et al.*, 2020] applies self-supervised learning for sequential recommendation; 8) **DSAN** [Yuan *et al.*, 2021] is a sparse attention network by replacing softmax with α-entmax. 9) **STOSA** [Fan *et al.*, 2022] is a recent Transformer model that uses Wasserstein distance mechanism. Our proposed probabilistic mask can generally plug in the Transformer-based models during training process. For our PMAN, we choose SASRec, TiSASRec, and BERT4Rec as its backbones by simply replacing the Transformer blocks (*e.g.*, self-attention layer, point-wise feed-forward layer, and layer normalization *etc.*) with our sparse masked attention block. Note that we do not apply our mask mechanism on STOSA since it adopts Wasserstein distance attention, rather than the standard scaled dot-product attention.

Parameter Settings. The parameters for the baselines are initialized as their original settings and are then carefully tuned to obtain optimal performance. We adopt Adam as optimizer and search embedding dimension d in Eq. (2) within $\{32, 64, 128\}$, the length of item sequence n within $\{25, 50\}$. For the attention capacity B in Problem (8), we vary the ratio r in $\{0.3, 0.5, 0.7, 0.9\}$, such that $B = r \cdot n^2$. Moreover, all of our PMANs only use single-head attention in the experiments. The impact of different number of heads will be discussed later.

Evaluation. Following the procedure [Kang and McAuley, 2018; Li *et al.*, 2020; Fan *et al.*, 2022], we use the last item of each user's sequence for testing, the second-to-last for validation, and the remaining items for training. We adopt the widely-used Hit@k and NDCG@k as the evaluation metrics ($k = 10$ by default). Instead of the biased sampling evaluation, we compute Hit@10 and NDCG@10 by the all-ranking protocol in the experiments [Krichene and Rendle, 2020].

5.2 Overall Performance

Table 2 presents the overall recommendation performance of all methods on the five datasets. All the simulated experiments are repeated five times independently and the average results are reported in the table.

From Table 2, we have the following observations:

- All Transformer-based models (*e.g.*, S^3-Rec, STOSA, SASRec, BERT4Rec, TiSASRec, and PMAN) generally outperform BPRMF, FPMC, GRU4Rec, Caser, and SR-GNN with a large margin, implying that the attention

[1] https://www.dropbox.com/s/7namvdjm0jy4cmj/Appendix.pdf?dl=0

[2] https://jmcauley.ucsd.edu/data/amazon/

[3] https://www.yelp.com/dataset

[4] https://grouplens.org/datasets/movielens/1m/

[5] https://cseweb.ucsd.edu/ jmcauley/datasets.html#steam_data

38::It Takes Two::Comedy	49::When Night Is Falling::Drama|Romance	53::Lamerica::Drama	11::American President::Comedy|Drama|Romance
25::Leaving Las Vegas::Drama|Romance	19::Ace Ventura: When Nature Calls::Comedy	2::Jumanji::Adventure|Children's|Fantasy	17::Sense and Sensibility::Drama|Romance
37::Across the Sea of Time::Documentary	12::Dracula: Dead and Loving It::Comedy|Horror	14::Nixon::Drama	30::Shanghai Triad::Drama
4::Waiting to Exhale::Comedy|Drama	15::Cutthroat Island::Action|Adventure|Romance	50::Usual Suspects::Crime|Thriller	34::Babe::Children's|Comedy|Drama
8::Tom and Huck::Adventure|Children's	43::Restoration::Drama	51::Guardian Angel::Action|Drama|Thriller	41::Richard III::Drama|War
48::Pocahontas::Animation|Children's|Musical|Romance	18::Four Rooms::Thriller	16::Casino::Drama|Thriller	5::Father of the Bride Part II::Comedy
1::Toy Story::Animation|Children's|Comedy	40::Cry, the Beloved Country::Drama	21::Get Shorty::Action|Comedy|Drama	31::Dangerous Minds::Drama
22::Copycat::Crime|Drama|Thriller	46::How to Make an American Quilt::Drama|Romance	47::Seven (Se7en)::Crime|Thriller	36::Dead Man Walking::Drama
45::To Die For::Comedy|Drama	27::Now and Then::Drama	6::Heat::Action|Crime|Thriller	33::Wings of Courage::Adventure|Romance
10::GoldenEye::Action|Adventure|Thriller	3::Grumpier Old Men::Comedy|Romance	9::Sudden Death::Action	35::Carrington::Drama|Romance
52::Mighty Aphrodite::Comedy	7::Sabrina::Comedy|Romance	13::Balto::Animation|Children's	Predict next
44::Mortal Kombat::Action|Adventure	20::Money Train::Action	29::City of Lost Children::Adventure|Sci-Fi	**26::Othello::Drama**
42::Dead Presidents::Action|Crime|Drama	39::Clueless::Comedy|Romance	24::Powder::Drama|Sci-Fi	

Table 3: A random user's historical behaviors in the MovieLens1M dataset, where we aim to visualize its attentions.

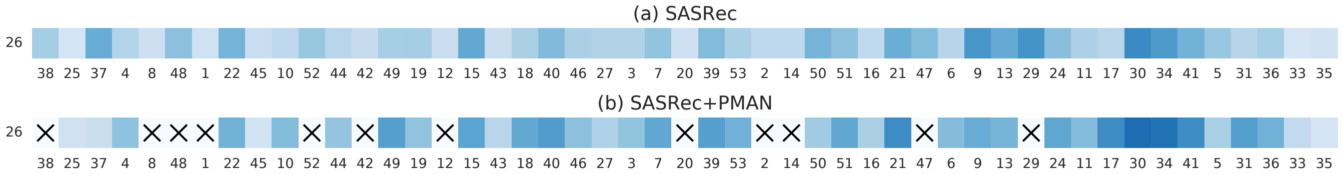

Figure 2: (a) Dense attentions for SASRec, where each attention score is non-zero, and (b) Sparse attentions for SASRec+PMAN, where × means that the corresponding attention scores are zeros. The color saturation indicates attention distribution.

mechanism is able to capture long-range item dependencies for sequential recommendation.

- The relative improvements of PMANs over their corresponding backbones are significant for all datasets. For example, SASRec+PMAN outperforms the vanilla SASRec by 7.97% and 9.06% on average in terms of Hit@10 and NDCG@10, respectively. This is mainly attributed to the ability of pruning noisy items via learnable mask in PMANs.
- STOSA and PMANs generally perform better than other Transformer-based models. The reason is that the vanilla self-attention mechanism is not Lipschitz continuous and is vulnerable to small perturbations in real-world applications. STOSA and PMANs address this issue by modeling uncertain noise within sequence.
- TiSASRec+PMAN consistently obtains the best performance for all datasets, which indicates the benefit of considering both temporal information and sparse attentions simultaneously. STOSA models dynamic uncertainty by using stochastic Gaussian distribution, but still generating dense attention maps, which cannot exactly remove the negative impacts of noisy items. Moreover, DSAN performs worse than our PMANs since its α-entmax function is relatively sensitive to the hyperparameters, which may cause overfitting issues.

In terms of running time, we empirically observe that the training time of the PMANs is around 0.81 times that of their backbones with the same hardware. That is because PMANs have a much simpler attention module with fewer model parameters. For example, we only use per-dim scalars and offsets to compute the query and key. Also, our single-head attention is sufficient to achieve the best performance without the requirements of point-wise feed-forward layer and layer normalization as in the original Transformer.

Overall, the experimental results demonstrate the superiority of our PMANs. Specifically, the proposed methods outperform all baselines with less time complexity.

5.3 Explainability

In addition to the good performance of our PMANs, we also visualize the attention maps to provide model's interpretability. In Table 3, a random user's engagement history in the Movielens1M dataset is given in a chronological order (column-wise). One can observe that the user tends to watch various types of movies, including *Comedy*, *Drama*, *Animation*, *Action*, *Romance*, etc. Specially, the drama and romance are the two types that the user likes to watch most. In this work, we regard the last item (*e.g.*, *26::Othello::Drama*) as test set, and aim to investigate the relationships between the last item and the previous engagement history. This can be achieved by inspecting the attention distribution to determine the amount of influence of each item in the sequence.

From humans, it is natural to reason that not all of historical movies are related to the target movie. For example, there is not obvious relationship between *38::It Takes Two::Comedy* and *26::Othello::Drama*. To see the models' reasoning, we provide attention distribution for both SASRec and SASRec+PMAN in Figure 2. For SASRec, all context movies have non-zero attention weights due to the Softmax attention mechanism. As a result, the dense attention maps cause misleading explainability. In contrast, our SASRec+PMAN can yield exactly zero probabilities for irrelevant movies, such as *38::It Takes Two::Comedy* and *29::City*

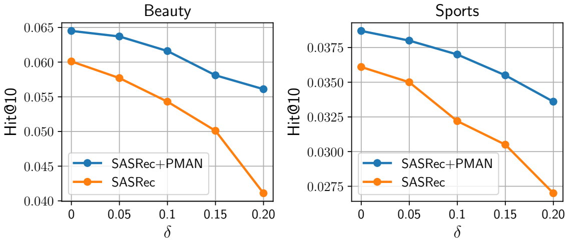

Figure 3: The impact of different noisy ratios δ.

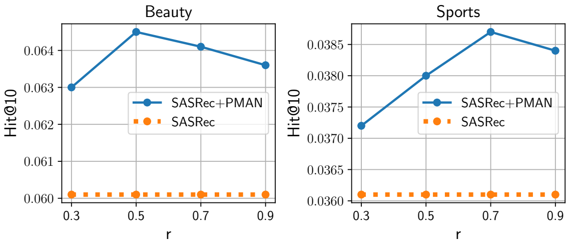

Figure 4: The impact of the degree of attention capacity r.

of Lost Children::Adventure|*Sci-Fi*. This is attributed to our probabilistic mask's ability of pruning task-irrelevant items in sequential recommendation. As such, our SASRec+PMAN can pay more attentions on the relevant drama movies, such as *31::Dangerous Minds::Drama* and *36::Dead Man Walking::Drama*. We observe the similar phenomenon for both TiSASRec+PMAN and BERT4Rec+PMAN, and their attention distribution for both can be found in Appendix.

Concisely, the visualization of spare attentions strongly shows that our PMANs can reduce the negative impacts of noisy items, which highly improves model interpretability.

5.4 Further Probe

Robustness. To further evaluate the robustness of our models, we follow the strategy [Ma *et al.*, 2020] to corrupt the training sequences by randomly replacing a portion of the observed items with uniformly sampled items that are not in the valid/test set. We vary the corrupted ratio δ from 0% to 20%. We report the results of SASRec and SASRec+PMAN in terms of Hit@10 on Beauty and Sports datasets, and we observe similar trends in other datasets. Figure 3 shows that the performance gain becomes larger for more noisy sequences, *i.e.*, the relative improvement ranges from 8.57% to 36.49% for different values of δ. This demonstrates the better robustness of our PMAN than its backbone in the task of sequential recommendation.

Sparsity. One important hyperparameter of our PMANs is the attention capacity B in Problem (8). A small B may lead to aggressive pruning, *i.e.*, $B = 0$ will mask out all of attentions. Here we set $B = r \cdot n^2$, and vary the sparsity ratio r within $\{0.3, 0.5, 0.7, 0.9\}$. Figure 4 shows the impact of the attention capacity B for Beauty and Sports datasets. We observe that the performance of SASRec+PMAN is consistently better than SASRec, indicating the benefits of pruning techniques. In practice, it is reasonable to set the sparsity ratio within $[0.5, 0.7]$ in our experiments.

	Beauty		Sports	
SASRec	Hit@10	NDCG@10	Hit@10	NDCG@10
Dropout (0.3)	0.0595	0.0316	0.0358	0.0175
Dropout (0.5)	0.0601	0.0324	0.0361	0.0183
Dropout (0.7)	0.0589	0.0319	0.0353	0.0170
PMAN	**0.0645**	**0.0357**	**0.0387**	**0.0201**

Table 4: The results of SASRec with different Dropout rates.

	Beauty		Sports	
	Hit@10	NDCG@10	Hit@10	NDCG@10
SASRec	0.0601	0.0324	0.0361	0.0183
PMAN (h=1)	0.0645	**0.0357**	**0.0387**	0.0201
PMAN (h=2)	**0.0647**	0.0353	0.0385	**0.0204**
PMAN (h=4)	0.0643	0.0354	0.0381	0.0181

Table 5: The results of PMAN with different numbers of heads.

Learnable Mask vs. Dropout. Our probabilistic mask in Eq. (4) is an elegant extension to Binary Dropout [Srivastava *et al.*, 2014]. Dropout drops neurons randomly, whereas our probabilistic mask is trainable with the model parameters. We compare our PMAN with Dropout in Table 4. We observe that our model consistently performs better than Dropout across different dropping ratios. Dropout is known to be susceptible to bias: the fact that attentions can be dropped randomly does not mean that the model allows them to be dropped. In contrast, our probabilistic mask would become close to a deterministic sparse mask as the optimizer goes on. Thus, a full trained mask would have much lower variance with better interpretability.

Single Head vs. Multi-head. For Transformer [Vaswani *et al.*, 2017], it is often useful to use multi-head mechanism, which applies self-attention operator in h subspaces, where h denotes the number of heads. Our Eq. (4) can be easily extended to multi-head attentions by following the similar procedure as in Transformer (*e.g.*, projecting $\mathbf{X}$ in more subspaces.). As shown in Table 5, the performance of single head roughly achieves the similar results as two heads. Nevertheless, the performance slightly drops with four heads, which may owe to overfitting issue. The appealing feature of our single-head method allows to greatly enhance computing efficiency for large-scale datasets.

6 Conclusion

Here we propose a Probabilistic Masked Attention Networks (PMAN) to filter out irrelevant item-item dependencies to enhance the robustness of Transformer-based recommender systems. We design a probabilistic masked mechanism to sparsify the attention distribution, and jointly train the mask with model parameters. PMAN is compatible with various Transformers, such as SASRec, TiSASRec, and BERT4Rec. Our experiments demonstrate the effectiveness of the proposed PMANs on benchmark datasets. Also, our models are able to provide a certain level of explainability by pruning irrelevant items in the sequences.

References

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

[Chen and Li, 2019a] Huiyuan Chen and Jing Li. Adversarial tensor factorization for context-aware recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019.

[Chen and Li, 2019b] Huiyuan Chen and Jing Li. Data poisoning attacks on cross-domain recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

[Chen *et al.*, 2020] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained bert networks. In *Advances in neural information processing systems*, 2020.

[Chen *et al.*, 2021] Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention. In *Advances in Neural Information Processing Systems*, 2021.

[Chen *et al.*, 2022a] Huiyuan Chen, Yusan Lin, Menghai Pan, Lan Wang, Chin-Chia Michael Yeh, Xiaoting Li, Yan Zheng, Fei Wang, and Hao Yang. Denoising self-attentive sequential recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*, 2022.

[Chen *et al.*, 2022b] Huiyuan Chen, Chin-Chia Michael Yeh, Fei Wang, and Hao Yang. Graph neural transport networks with non-local attentions for recommender systems. In *Proceedings of the ACM Web Conference 2022*, 2022.

[Child *et al.*, 2019] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[Correia *et al.*, 2019] Gonçalo M Correia, Vlad Niculae, and André FT Martins. Adaptively sparse transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[de Souza Pereira Moreira *et al.*, 2021] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation. In *Fifteenth ACM Conference on Recommender Systems*, 2021.

[Fan *et al.*, 2022] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S. Yu. Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM Web Conference 2022*, 2022.

[Frankle and Carbin, 2018] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

[Hidasi *et al.*, 2016] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations*, 2016.

[Hua *et al.*, 2022] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc V Le. Transformer quality in linear time. In *International Conference on Machine Learning*, 2022.

[Jang *et al.*, 2017] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[Kang and McAuley, 2018] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining*, 2018.

[Kim *et al.*, 2021] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *International Conference on Machine Learning*, 2021.

[Krichene and Rendle, 2020] Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[Li *et al.*, 2020] Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020.

[Li *et al.*, 2021] Yang Li, Tong Chen, Peng-Fei Zhang, and Hongzhi Yin. Lightweight self-attentive sequential recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021.

[Ma *et al.*, 2020] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009.

[Rendle *et al.*, 2010] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, 2010.

[Shazeer, 2020] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 2014.

[Sun *et al.*, 2019] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM*

International Conference on Information and Knowledge Management, 2019.

[Tang and Wang, 2018] Jiaxi Tang and Ke 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.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.

[Wang *et al.*, 2021a] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, 2021.

[Wang *et al.*, 2021b] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[Wang *et al.*, 2021c] Zitai Wang, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Implicit feedbacks are not always favorable: Iterative relabeled one-class collaborative filtering against noisy interactions. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.

[Wang *et al.*, 2022] Yu Wang, Yuying Zhao, Yushun Dong, Huiyuan Chen, Jundong Li, and Tyler Derr. Improving fairness in graph neural networks via mitigating sensitive attribute leakage. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

[Wu *et al.*, 2019] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[Wu *et al.*, 2020] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. Sse-pt: Sequential recommendation via personalized transformer. In *Fourteenth ACM Conference on Recommender Systems*, 2020.

[Xiong *et al.*, 2021] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[Yeh *et al.*, 2022] Chin-Chia Michael Yeh, Mengting Gu, Yan Zheng, Huiyuan Chen, Javid Ebrahimi, Zhongfang Zhuang, Junpeng Wang, Liang Wang, and Wei Zhang. Embedding compression with hashing for efficient representation learning in large-scale graph. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

[Yuan *et al.*, 2021] Jiahao Yuan, Zihan Song, Mingyou Sun, Xiaoling Wang, and Wayne Xin Zhao. Dual sparse attention network for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4635–4643, 2021.

[Yun *et al.*, 2020] Chulhee Yun, Yin-Wen Chang, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. O (n) connections are expressive enough: Universal approximability of sparse transformers. In *Advances in Neural Information Processing Systems*, 2020.

[Zaheer *et al.*, 2020] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, 2020.

[Zhou *et al.*, 2020] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1893–1902, 2020.