Towards Automatic Sampling of User Behaviors for Sequential Recommender Systems

Hao Zhang , **Mingyue Cheng*** , **Zhiding Liu** , **Junzhe Jiang**
State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China
{zh2001, zhiding, jzjiang}@mail.ustc.edu.cn, {mycheng}@ustc.edu.cn,

Abstract

Sequential recommender systems (SRS) have gained increasing popularity due to their remarkable proficiency in capturing dynamic user preferences. In the current setup of SRS, a common configuration is to uniformly consider each historical behavior as a positive interaction. However, this setting has the potential to yield sub-optimal performance as each individual item often have a different impact on shaping the user's interests. Hence, in this paper, we propose a novel automatic sampling framework for sequential recommendation, named AutoSAM, to non-uniformly treat historical behaviors. Specifically, AutoSAM extends the conventional SRS framework by integrating an extra sampler to intelligently discern the skew distribution of the raw input, and then sample informative sub-sets to build more generalizable SRS. To tackle the challenges posed by non-differentiable sampling actions and to introduce multiple decision factors for sampling, we further design a novel reinforcement learning based method to guide the training of the sampler. Furthermore, we theoretically devise multi-objective sampling rewards including *Future Prediction* and *Sequence Perplexity*, and then optimize the whole framework in an end-to-end manner by combining the policy gradient. We conduct extensive experiments on benchmark recommendation models and four real-world datasets. The experimental results demonstrate the effectiveness of the proposed AutoSAM.

1 Introduction

Recommender systems [Koren *et al.*, 2009; Liu *et al.*, 2018; Zhang *et al.*, 2025; Zhang *et al.*, 2024a; Cheng *et al.*, 2021] have become crucial tools for information filtering in various online applications, such as e-commerce, advertising, and online videos. Among them, sequential recommender systems (SRS) [Li *et al.*, 2017; Liu *et al.*, 2022] have become increasingly prevalent due to their ability to capture long- and short-term user interests.

*Corresponding author

So far, many efforts have been devoted to sequential recommendation, ranging from early matrix factorization and Markov chain based method [Rendle *et al.*, 2010] to state-of-the-art deep neural network models [Cheng *et al.*, 2022], including recurrent neural networks (RNNs) [Hidasi *et al.*, 2015; Hidasi *et al.*, 2016], convolutional neural networks (CNNs) [Tang and Wang, 2018; Yuan *et al.*, 2019], and self-attentive models [Kang and McAuley, 2018; Sun *et al.*, 2019; Li *et al.*, 2021]. Meanwhile, the effectiveness and efficiency of the generative loss (i.e., auto-regressive loss) [Yuan *et al.*, 2019] in SRS tasks have been widely demonstrated in these approaches. Despite their remarkable success, these methods often consider from a model perspective while treating all historical behaviors as uniformly positive. In fact, this may lead to sub-optimal performance as items in the sequences are usually of unequal importance on shaping user interests [Zhang *et al.*, 2013]. For instance, purchased items should hold more significance than merely clicked ones. Moreover, the motivations behind the same action (e.g. clicking) can also be very complex and multifaceted. Interactions on online platforms may sometimes be driven by a herd mentality or could even be the result of mis-clicks, while in other cases, they may accurately represent a user's genuine interests.

Recently, there have also been some works which attempt to improve the recommender systems from a data perspective [He *et al.*, 2023; Qin *et al.*, 2023]. SIM [Pi *et al.*, 2020] and UBR4CTR [Qin *et al.*, 2020] extracts user interests with search or retrieve units to capture the diverse user's long-term interest with target item. Similarly, SDIM [Cao *et al.*, 2022] samples from multiple hash functions to gather similar behavior items to the target for CTR prediction. Though the aforementioned methods have proven effective, they may suffer from the following limitations. Firstly, these methods often select items according to the target, which brings challenges to compute scores across all candidate items in parallel and only suitable to fine-ranking stage [Covington *et al.*, 2016]. And such setup may be too strict. Secondly, these sampling processes are mainly designed for CTR prediction tasks [Guo *et al.*, 2017; Wang *et al.*, 2017], which are often typically optimized as binary classification problems. As a result, these approaches often cannot be equipped with generative loss in SRS tasks, whose effectiveness have been demonstrated in many previous works [Kang and McAuley, 2018; Yuan *et al.*, 2019].

We hold that an ideal solution should be adaptive while maintaining the core training principles of SRS. Hence, to achieve this goal, we propose a general automatic sampling framework, named AutoSAM, to non-uniformly treat historical behaviors for sequential recommendation. To be concrete, an additional sampler layer is first employed to adaptively explore the skew distribution of raw input. Then, informative sub-sets are sampled from the distribution to build more generalizable sequential recommenders. In order to overcome the challenges of non-differentiable discrete sampling actions, and to introduce multiple decision factors for sampling, we further introduce a novel reinforcement learning based method to guide the training of the sampler due to its flexibility. We believe that a proper decision should not only focus on the target item in the future, but also consider the coherence with the previous context. Along this line, we incorporate the major factors into the reward estimation, including *Future Prediction* and *Sequence Perplexity*. Finally, both the SRS and the sampler can be jointly optimized in an end-to-end manner by combining the policy gradient. As a result, our method exhibits more accurate recommendations than previous approaches, and is generally effective for various sequential recommenders with different backbones. We summarize the contributions as follows:

- We propose to sample user behaviors from a non-uniform distribution for SRS task. We highlight that the main challenge is to sample historical behaviors dynamically while leveraging the benefits of generative loss.
- We propose a general automatic sampling framework for sequential recommendation, named AutoSAM, to build generalizable SRS with an additional sample layer. We further introduce a reinforcement learning based method to solve the challenges of non-differentiable actions and design multi-objective rewards to optimize the sampler.
- We conduct extensive experiments on public datasets to show superior recommendation results compared to previous competitive baselines. We also validate the generality of the proposed method and additionally analyze the effectiveness of the sampler.

2 Related Work

2.1 Sequential Recommendation

Sequential recommendation aims to predict users' future behaviors given their historical interaction data [Luo *et al.*, 2024; Cheng *et al.*, 2024; Zhang *et al.*, 2024b]. Early approaches mainly fuse Markov Chain and matrix factorization to capture both long- and short-term item-item transitions [Rendle *et al.*, 2010]. Latter, with the success of neural network, recurrent neural network (RNN) methods are widely conducted in sequential recommendation [Hidasi *et al.*, 2015; Hidasi *et al.*, 2016]. Besides, convolutional-based models [Tang and Wang, 2018; Yuan *et al.*, 2019] can also be very effective in modeling sequential behaviors. In addition, we notice that graph neural networks [Wu *et al.*, 2019; Xu *et al.*, 2019] have become increasingly prevalent by constructing graph structures from session sequences. And in recent years, self-attention models [Kang and McAuley, 2018; Sun *et al.*, 2019; de Souza Pereira Moreira *et al.*, 2021] have shown their promising strengths in the capacity of long-term dependence modeling. However, these methods focus on the model architectures and treat all behaviors as uniformly positive, may lead to the sub-optimal performance.

2.2 User Behavior Sampling

As a data-driven technology, recommendation systems have attracted a series of works that consider from the data perspective in recent years [Qin *et al.*, 2021]. To be specific, in SIM [Pi *et al.*, 2020], a two-stage method with a general search unit (GSU) and an exact search unit is proposed to model long-term user behaviors better. Similarly, UBR4CTR [Qin *et al.*, 2020] conducts retrieval-based method to achieve the goal. Besides, SDIM [Cao *et al.*, 2022] proposed a sampling-based approach which samples similar items to the target by multiple hash functions. And some other works [Wang *et al.*, 2021; Lin *et al.*, 2023] mainly formulate a denoising task with the aims of filtering irrelevant or noise items. Despite effectiveness, these methods are mainly designed for CTR prediction, which may be not suitable to conduct the left-to-right generative loss and sample items dynamically in the SRS task. We also notice some works which attempt to capture more informative patterns between user behaviors sequence by designing advance model architectures. RETR [Yao *et al.*, 2022] build recommender transformer for sampling the user behavior pathway. And an all-MLP based model is proposed [Zhou *et al.*, 2022] which adopts Fourier transform with learnable filters to alleviate the influence of the noise item. Differently, in this paper, we present a general framework to learn the distributions of the raw inputs adaptively with carefully designed multi-objectives, so as to enhance the SRS with more informative training data.

3 Preliminaries

3.1 Problem Definition

Assume that there are item set $\mathcal{I} = \{i_1, i_2, ..., i_{|\mathcal{I}|}\}$ and user set $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$, we denote each behavior sequence of user $u \in \mathcal{U}$ as $X^u = [x_1^u, x_2^u, ..., x_n^u]$, where $x_t^u \in \mathcal{I}$ is the item that u interacted at step t and n is the sequence length. Sequential recommender systems (SRS) aim to predict the item that users might interact with at the next time step. Different from traditional SRS, the main idea behind this work is to sample historical behaviors with a policy $\pi(A^u|X^u)$, where $A^u \in \{select = 1, discard = 0\}^n$, and then utilize $\hat{X^u} = [x_{k_1}^u, x_{k_2}^u, ..., x_{k_m}^u]$ to train more generalizable SRS, in which k is the set of time steps about sampled items, m is the length of $\hat{X^u}$.

3.2 Sequential Recommender System

As shown in Figure 1(a), modern SRS contain the components as following. First, each input item x_t^u is mapped into an embedding vector by an embedding lookup operation as: $E_t = \zeta_{x_t}$, where $\zeta \in \mathbb{R}^{|\mathcal{I}| \times d}$ is item embedding matrix. Then, the sequential embedding will be fed into stacked hidden layers to capture the long- and short-term dependence.

$$H^{(l)} = F^{(l)}(H^{(l-1)}), 1 \leq l \leq L, \tag{1}$$

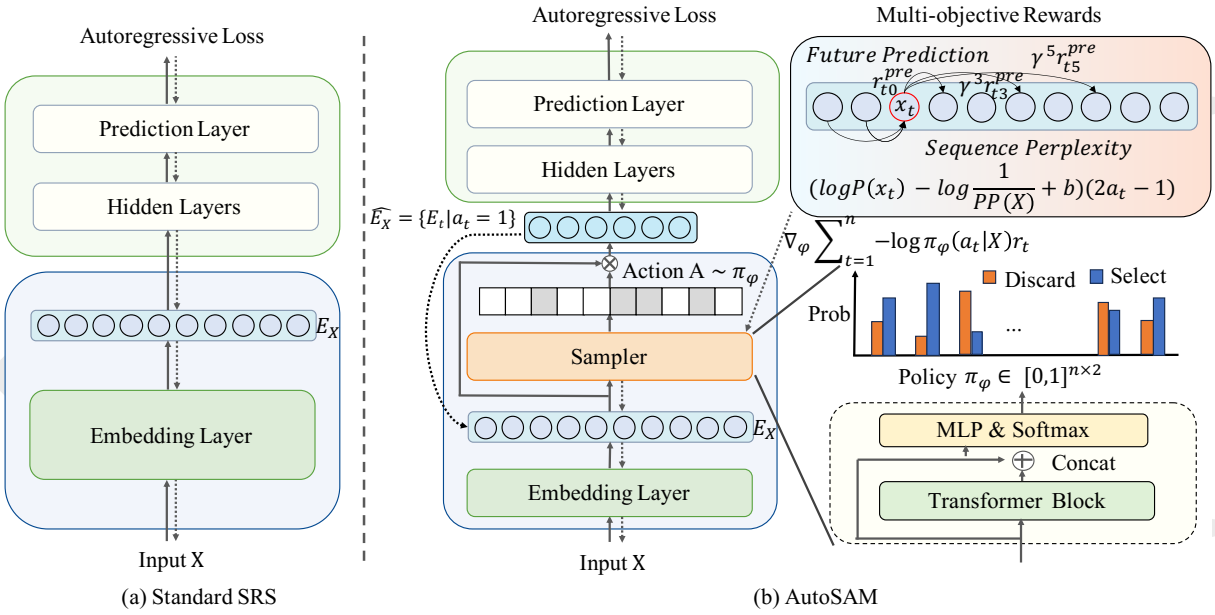


Figure 1: (a) is the architecture of the standard sequential recommender system (SRS). (b) is our proposed AutoSAM which augments the traditional sequential recommendation architecture with an additional sampler layer to adaptively explore the skew distribution of raw input. The dashed lines represent gradient backpropagation.

where L stands for the number of layers, $H^{(l)}$ is the output of the l-th hidden layer $F^{(l)}$ while $H^{(0)} = E$. Next, a prediction layer is adopted to generate the distribution P of the user's next preferences for each time step t:

$$P_t = Softmax(H_t^{(L)} W + b), \tag{2}$$

in which $W \in \mathbb{R}^{d \times |\mathcal{I}|}$, $b \in \mathbb{R}^{|\mathcal{I}|}$ is the learnable parameters. Finally, to train the model, the auto-regressive generative loss are employed as optimization objectives, i.e., left-to-right supervision signals. Combining the cross-entropy loss with such an objective, the loss function can be written as:

$$\mathcal{L}(X;\theta) = -\sum_{t=1}^{n-1} logP(y_t | X_{\leq t}; \theta) = -\sum_{t=1}^{n-1} logP_{t,y_t}, \tag{3}$$

where θ is the parameters of the model, $y_t = x_{t+1}$ is the item expect to be predicted at step t, and n is the input length.

4 AutoSAM: the Proposed Method

4.1 Overview of the Framework

The framework of our proposed AutoSAM is depicted in Figure 1(b). First, we employ a light-weighted sampler to adaptively learn the non-uniform distribution of the raw input. After that, the sequential recommender systems (SRS) are trained with informative sub-sets sampled from the whole sequence to gain stronger generalizations. Considering the challenges of non-differentiable sampling actions and the necessity to introduce multiple decision factors, we further introduce a novel reinforcement learning based method to optimize the sampler. Specifically, we treat the sampler as the agent. For each time step t of the user behavior sequence X, the sampler takes $X_{\leq t}$ as the current state s_t, and outputs an action $a_t \in \{0, 1\}$ of the t-th item. Then, it observes the carefully designed multi-objective rewards from the environment to update the model via the policy gradient. Finally, the sampler could make optimal decisions by continuously interacting with the environment. And it is worth mentioning that both the sampler and the SRS can be jointly optimized in an end-to-end manner in our proposed framework.

4.2 User Behavior Sampler

We first transform sequence X into embedding $E_X = \zeta_X + P_X$, where ζ and P are the embedding matrix and position embedding, respectively. We suggest that the sample decisions should be based on both local and global information. Hence, the sampler leverages a Transformer Block with triangular mask matrix to aggregate the global information $X_{\leq t}$ at each step t, and then concatenate it with the local item embedding to generate the sample policy π through an MLP layer. The above procedure can be formulated as:

$$S = ReLU([Transformer(E)||E]W_1 + b_1)W_2 + b_2, \tag{4}$$

where $||$ means concatenation, $W_1 \in \mathbb{R}^{2d \times d}, b_1 \in \mathbb{R}^d, W_2 \in \mathbb{R}^{d \times 2}, b_2 \in \mathbb{R}^2$ are learnable parameters. After that, binary decisions $A \in \{0, 1\}^n$ are sampled from the Bernoulli distribution as following:

$$\pi = Softmax(S/\tau) \in \mathbb{R}^{n \times 2}, \ A \sim \pi \in \{0, 1\}^n, \tag{5}$$

where τ is the temperature of the Softmax. Finally, we can build the sampled sub-set as $\hat{X} = \{x_t | a_t = 1\}$.

4.3 Multi-Objective Rewards

Now we discuss how to design our reward, which plays an important role in learning the optimal π. As mentioned before,

traditional SRS often follow the next item prediction task, which can neither update sampler network directly due to the non-differentiable challenge of discrete actions nor control the sample rate. Besides, such single objective may be too strict and one-sided. We argue that the proper decisions should not only focus on the target item but also consider the coherence with preceding context. Along this line, we incorporate the major factors into the reward estimation, including *Future Prediction* and *Sequence Perplexity*.

Future Prediction

Generally, items that contribute to predicting the future interactions should be of more opportunities to be selected. Accordingly, we theoretically propose our rewards with consideration of the recommendation loss in Eq. 3. Since we train the SRS with the sampled sub-set $\hat{X}$, the objective can be written as:

$$min\ J(\theta,\phi) = min\ \mathbb{E}_X \mathbb{E}_{A\sim\pi_\phi(A|X)}\left[\mathcal{L}(\hat{X};\theta)\right], \tag{6}$$

where θ and ϕ are the parameters of SRS and sampler, respectively. To optimize this objective, we first transform the prediction loss to an equivalent form:

$$\begin{aligned}\mathcal{L}(\hat{X};\theta) &= \sum_{t=1}^{m-1} -logP(\hat{y}_t|\hat{X}_{\le t};\theta) \\ &= \sum_{t=1}^{n-1} -a_{t+1} logP(y_t|\{x_{k_j}|1\le j<m; k_j\le t\};\theta) \\ &\triangleq \sum_{t=1}^{n-1} \mathcal{L}_t(X,A;\theta),\end{aligned} \tag{7}$$

where m, n is the length of $\hat{X}, X$ respectively. And then derive the gradients with respect to ϕ, we have:

$$\begin{aligned}\nabla_\phi J(\theta,\phi) &= \mathbb{E}_X \nabla_\phi \sum_A \pi_\phi(A|X)\mathcal{L}(\hat{X};\theta) \\ &= \mathbb{E}_X \mathbb{E}_{A\sim\pi_\phi(A|X)} \nabla_\phi \sum_{t=1}^{n}[log\pi_\phi(a_t|X) \sum_{t=1}^{n-1} \mathcal{L}_t(X,A;\theta)] \\ &= \mathbb{E}_X \mathbb{E}_{A\sim\pi_\phi(A|X)} \nabla_\phi \sum_{t=1}^{n} -log\pi_\phi(a_t|X) r^{pre},\end{aligned} \tag{8}$$

where $r^{pre} = \sum_{t=1}^{n-1} -\mathcal{L}_t(X,A;\theta)$ means the rewards associated to the sampler. In practice, we may make a few changes to it. First, we add a baseline $\mathcal{L}^b$ computed on raw input to $\mathcal{L}$ as it is always positive. Second, we discount the future rewards by a factor $\gamma \in [0,1]$ and ignore the targets whose sample probabilities are lower than ψ to increase stability. Then, the future prediction reward at each time step t can be defined as follows:

$$r_t^{pre} = \sum_{t'=t}^{n-1} -\gamma^{t'-t} I[\pi_\phi(a_{t'+1}|X) > \psi](\mathcal{L}_{t'}(X,A;\theta) - \mathcal{L}^b_{t'}), \tag{9}$$

in which $I[\cdot]$ is the indicator function, γ is set to 0.9 across all the experiments in this work.

Sequence Perplexity

In addition to sampling based on future predictions, it should also be attached great importance to considering the coherence with previous context. Inspired by perplexity (PP), which is widely used to evaluate the quality of sentences in Natural Language Processing (NLP) [Dathathri *et al.*, 2019] with the definition as:

$$PP(X) = P(x_1, x_2, ...x_n)^{-\frac{1}{n}} = \prod_{t=1}^{n} P(x_t|X_{<t})^{-\frac{1}{n}}. \tag{10}$$

As words causing high perplexity tend to stray from the context [Lin *et al.*, 2024], similarly, we insist that the items causing high perplexity may lack representativeness. A natural idea is to assign lower sampling probabilities to the items whose predicted probabilities according to historical behaviors are lower than $1/PP(X)$, so as to reduce the perplexity and obtain informative sub-sets in some way. To this end, the sampler should be encouraged to drop x_t ($a_t = 0$) while $P(x_t|X_{<t}) < 1/PP(X)$, and select x_t ($a_t = 1$) while $P(x_t|X_{<t}) > 1/PP(X)$.

In practice, we employ the SRS to compute the approximate objective at each time step t. Besides, a relax factor is further conducted to control the strictness of the sampler. Above all, the reward can be defined as follow:

$$r_t^{pp} = \left(logP(x_t|X_{<t}) - log\frac{1}{PP(X)} + b\right)(2a_t - 1), \tag{11}$$

where b is the relax factor, by which we can indirectly control the global sampling ratio. Generally, the smaller b will lead the sampler to be stricter.

4.4 Optimization

We summarize the rewards with trade-off parameter λ and scaling factors k as:

$$r = k(\lambda r^{pre} + (1-\lambda) r^{pp}). \tag{12}$$

To further learn the sequential recommendation task, we also derive the gradients according to Eq. 6 with respect to θ:

$$\nabla_\theta J(\theta,\phi) = \mathbb{E}_X \mathbb{E}_{A\sim\pi_\phi(A|X)} \nabla_\theta \mathcal{L}(\hat{X};\theta). \tag{13}$$

Finally, both the SRS and sampler can be optimized jointly in an end-to-end manner by combining the policy gradient:

$$\theta \leftarrow \theta - \alpha_1 \nabla_\theta \mathcal{L}(\hat{X};\theta),\ \phi \leftarrow \phi + \alpha_2 \nabla_\phi \sum_{t=1}^{n} log\pi_\phi(a_t|X) r_t, \tag{14}$$

where α_1, α_2 are the learning rates of SRS and sampler. Overall, we present our algorithm in Algorithm 1.

4.5 Time and Computation Complexity Analysis

To evaluate the efficiency of our proposed method for online services, we analyze the time complexity of AutoSAM during inference in this part. Denote L as the number of layers and N as the sequence length. Since we conduct Transformer-based SRS whose time complexity is $\mathcal{O}(LN^2)$, the sampling processing takes $\mathcal{O}(N^2)$ and the sequence modeling can be done within $\mathcal{O}(L(\mathbb{E}[(\sum_{i=1}^{N} a_i)^2])) = \mathcal{O}(L(\mu^2N^2 + N\sigma^2))$, in which $\mu \in \{0,1\}$ denotes the average sampling rate while σ is the variance. Thus, AutoSAM could reduce the time complexity of about $\mathcal{O}((L - \mu^2 L - 1)N^2)$ by shorten the behavior sequence.

Algorithm 1 Learning the AutoSAM framework

Initial SRS parameters θ and sampler parameters ϕ.
for $Epoch = 1, 2, ...T$ **do**
  for $u = 1, 2, ...|U|$ **do**
    $X \leftarrow [x_1^u, x_2^u, ..., x_n^u]$;
    $\hat{X} \leftarrow [\,]$;
    Compute Sampling policy $\pi_\phi(A|X) \in [0, 1]^{n\times 2}$;
    for $t = 1, 2, ...n$ **do**
      Sample a_t from π_t;
      if $a_t = 1$ **then** Append x_t to $\hat{X}$;
    end for
    Compute the recommendation loss $\mathcal{L}(\hat{X}; \theta) = -\sum_{t=1}^{m-1} logP(\hat{y}_t|\hat{X}_{\leq t}; \theta)$
    Compute reward $r = k(\lambda r^{pre} + (1 - \lambda)r^{pp})$
    $\theta \leftarrow \theta - \alpha_1 \nabla_\theta \mathcal{L}(\hat{X}; \theta)$
    $\phi \leftarrow \phi + \alpha_2 \nabla_\phi \sum_{t=1}^{n} log\pi_\phi(a_t|X) r_t$
  end for
end for

5 Experiments

5.1 Experimental Setup

Datasets

We conduct four real-world datasets from different online platforms as following: **Tmall**[1] mainly contains anonymized users' shopping logs of Tmall platform in the past 6 months before and on the "Double 11" day. **Alipay**[2] collects huge amount of user data of Alibaba Group and Koubei between July 1^{st}, 2015 and November 30^{th}, 2015. **Yelp**[3] is a public dataset where we obtain each user's sequential behaviors by ranking its rated items in time order. **Amazon Book**[4] (abbreviated as **Amazon**) is selected from Amazon review data. We construct sequential behaviors by using user's rating history. Similar to some previous works [Zhou *et al.*, 2022], we filter inactive users and items with fewer than c interactions, where c is set to 10 for Tmall, Alipay and Amazon while adjusted to 5 for Yelp due to the smaller data scale. The statics are summarized in Table 1.

Datasets	#Num. Users	#Num. Items	#Num. Actions	#Actions/Item	#Actions/User
Tmall	424,170	969,426	22,010,938	22.71	51.89
Alipay	520,064	2,076,041	20,976,085	10.10	40.33
Yelp	221,397	147,376	3,523,285	23.91	15.91
Amazon	724,012	1,822,885	18,216,875	9.99	25.16

Table 1: Statics of the used datasets in the experiments.

Hyper-parameter	Tuning Range	Tmall	Alipay	Yelp	Amazon
t	$[1.0, 3.0, 5.0, 7.0, 9.0]$	5.0	5.0	5.0	3.0
b	$[-0.5, 0.0, 0.5, 1.0, 1.5, 2.0]$	1.0	2.0	1.0	0.5
k	$[2e^{-1}, 2e^{-2}, 2e^{-3}, 2e^{-4}, 2e^{-5}]$	$2e^{-3}$	$2e^{-3}$	$2e^{-2}$	$2e^{-3}$
λ	$[0.25, 0.5, 0.75]$	0.5	0.5	0.5	0.5
ψ_0	$[0.2, 0.5, 0.8]$	0.8	0.8	0.8	0.5

Table 2: AutoSAM's hyper-parameter exploration.

Compared Methods

We first compare our method[5] from data perspective with following approaches: **(1) FullSAM** trains normally with full historical behaviors. **(2) FMLP-Rec** [Zhou *et al.*, 2022] conducts Fourier transform with learnable filters to alleviate the influence of noise. **(3) RETR** [Yao *et al.*, 2022] builds recommender transformer to sample the user behavior pathway for sequential recommendation. **(4) SDIM** [Cao *et al.*, 2022] replaces the attention mechanism with multiple hash functions to sample relevant historical items for modeling long-term preference. **(5) RanSAM**, **(6) LastSAM**, **(7) PopSAM** samples random, last or most popular behaviors respectively. Note that we adopt SASRec [Kang and McAuley, 2018] as the sequential recommender system for all sample-based methods due to its effectiveness. It is worth mentioning that some other sampling based methods e.g. UBR4CTR, SIM are mainly designed for CTR tasks and typically optimized as a binary classification problem. **While they sampling behaviors using target item**, **the sampled user behaviors often depend on what the candidate item is** [Cao *et al.*, 2022]. As a result, these methods are hard to compute the scores of large candidate set as the million-level setting in this paper.

Besides, We also compare the models with other architectures: **(8) PopRec** is a popularity-based method, in which each user is recommended according to the popularity. **(9) BPR-MF** [Rendle *et al.*, 2012] is a well-known matrix factorization-based method optimized by Bayesian personalized ranking loss. **(10) FPMC** [Rendle *et al.*, 2010] combines matrix factorization with the Markov chain to predict the next interaction. **(11) GRU4Rec** [Hidasi *et al.*, 2015] is a pioneering attempts by employing recurrent neural network for next item recommendation. **(12) NextitNet** [Yuan *et al.*, 2019] is a CNN-based method to capture long-term and short-term interests. **(13) BERT4Rec** [Sun *et al.*, 2019] utilizes a bidirectional self-attention network to model user sequence. **(14) SR-GNN** [Wu *et al.*, 2019] applies GNN with attention network to model each session. **(15) CL4SRec** [Xie *et al.*, 2022] utilizes contrastive learning to extract the discrimination information in sequential recommendation. **(16) DuoRec** [Qiu *et al.*, 2022] is proposed with a contrastive objective serving as the regularization over sequence representations.

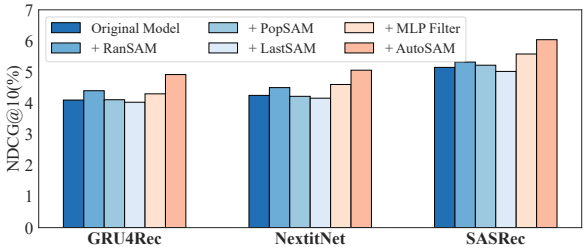
Hyper-parameter Settings

We set the batch size of 128 with 10, 000 random negative items per batch, and the embedding size is set to 128 for all methods. In all sampling based methods which employ two-layer SASRec as the backbone, we conduct 4 multi-head self-attention and 2 FFN layers, while the hidden size is set to 256. The sample rates of RanSAM, LastSAM and PopSAM are searched from $\{0.5, 0.6, 0.7, 0.8, 0.9\}$. We consistently employ Autom as the default optimizer for all recommenders, combined with a learning rate α_1 of $1 \times e^{-3}$. As for our sampler, we conduct SGD with a learning rate α_2 of $1 \times e^{-1}$. We use grid search to find the best group of AutoSAM's hyper-parameters as shown in Table 2.

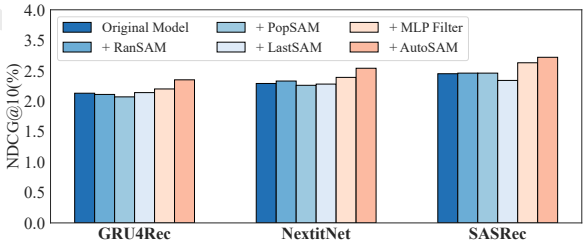
[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
[2]https://tianchi.aliyun.com/dataset/dataDetail?dataId=53
[3]https://www.yelp.com/dataset
[4]http://deepyeti.ucsd.edu/jianmo/amazon/index.html
[5]https://github.com/zh-ustc/AutoSAM

Method	Tmall				Alipay				Yelp				Amazon			
	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20
PopRec	0.0045	0.0053	0.0087	0.0118	0.0136	0.0143	0.0155	0.0182	0.0037	0.0048	0.0066	0.0111	0.0016	0.0018	0.0030	0.0038
BPR-MF	0.0217	0.0268	0.0402	0.0602	0.0129	0.0142	0.0181	0.0234	0.0173	0.0228	0.0343	0.0564	0.0116	0.0145	0.0212	0.0324
FPMC	0.0379	0.0452	0.0683	0.0971	0.0572	0.0632	0.0907	0.1144	0.0210	0.0273	0.0407	0.0657	0.0481	0.0520	0.0662	0.0817
GRU4Rec	0.0410	0.0503	0.0765	0.1135	0.0450	0.0518	0.0779	0.1050	0.0213	0.0276	0.0415	0.0666	0.0370	0.0421	0.0581	0.0783
NextitNet	0.0425	0.0510	0.0775	0.1115	0.0498	0.0568	0.0850	0.1124	0.0229	0.0295	0.0447	0.0711	0.0441	0.0493	0.0666	0.0863
BERT4Rec	0.0496	0.0595	0.0906	0.1301	0.0485	0.0551	0.0815	0.1078	0.0235	0.0305	0.0460	0.0739	0.0386	0.0441	0.0618	0.0840
SR-GNN	0.0359	0.0437	0.0664	0.0975	0.0401	0.0459	0.0681	0.0915	0.0199	0.0255	0.0383	0.0608	0.0312	0.0355	0.0485	0.0655
CL4SRec	0.0516	0.0616	0.0929	0.1322	0.0616	0.0694	0.1040	0.1351	0.0256	0.0322	0.0486	0.0766	0.0472	0.0528	0.0726	0.0950
DuoRec	0.0526	0.0627	0.0950	0.1350	0.0619	0.0698	0.1050	0.1365	0.0248	0.0317	0.0476	0.0752	0.0478	0.0534	0.0731	0.0954
FullSAM	0.0515	0.0614	0.0927	0.1320	0.0617	0.0695	0.1037	0.1346	0.0245	0.0314	0.0473	0.0746	0.0470	0.0524	0.0718	0.0932
FMLP-Rec	0.0558	0.0665	0.1007	0.1431	0.0630	0.0709	0.1052	0.1362	0.0263	0.0337	0.0506	0.0798	0.0480	0.0535	0.0729	0.0949
RETR	0.0526	0.0624	0.0942	0.1332	0.0608	0.0688	0.1027	0.1346	0.0250	0.0318	0.0478	0.0751	0.0471	0.0525	0.0711	0.0926
SDIM	0.0527	0.0626	0.0939	0.1333	0.0585	0.0685	0.0991	0.1332	0.0241	0.0308	0.0464	0.0734	0.0464	0.0513	0.0701	0.0898
RanSAM	0.0532	0.0633	0.0961	0.1354	0.0633	0.0713	0.1061	0.1380	0.0246	0.0315	0.0475	0.0749	0.0443	0.0500	0.0685	0.0914
LastSAM	0.0502	0.0600	0.0904	0.1291	0.0599	0.0667	0.0993	0.1297	0.0234	0.0297	0.0445	0.0698	0.0446	0.0514	0.0696	0.0893
PopSAM	0.0522	0.0623	0.0945	0.1347	0.0590	0.0665	0.0995	0.1292	0.0246	0.0316	0.0474	0.0752	0.0442	0.0496	0.0681	0.0893
AutoSAM	**0.0604**	**0.0717**	**0.1084**	**0.1528**	**0.0672**	**0.0753**	**0.1117**	**0.1437**	**0.0272**	**0.0347**	**0.0521**	**0.0817**	**0.0549**	**0.0610**	**0.0831**	**0.1074**
IMP	8.24%	7.82%	7.65%	6.78%	6.16%	5.61%	5.28%	4.13%	3.42%	2.97%	2.96%	2.38%	14.14%	14.02%	13.68%	12.58%

Table 3: Recommendation performance comparison of different models. The best and the second best performance methods are indicated by bold and underlined fonts. "IMP" denotes the improvements of AutoSAM compared to the best baseline.



(a) Tmall



(b) Yelp

Figure 2: Performance comparison using different backbones.

Evaluation Metrics

We evaluate the performance over all items with *Recall* and *Normalized Discounted Cumulative Gain (NDCG)*. The first one is an evaluation of unranked retrieval sets while the other reflects the order of ranked lists. We consider top-k items for recommendations where $k \in \{10, 20\}$. We split the dataset into training, validation and testing sets following the leave-one-out strategy [Cheng *et al.*, 2022; Zhao *et al.*, 2022].

5.2 Experimental Performance Analysis

Overall Performance

The result of different models on datasets are shown in Table 3. First of all, we can find that considering from the data perspective instead of treating all items as uniformly positive can benefit the performance. However, SDIM achieves limited improvement compared to FullSAM. A possible reason is that SDIM samples according to the last item maybe too strict to capture the user's rich interests. Besides, FMLP-Rec achieves much better performance than traditional SRS probably due to its stronger ability to filter noise information. And we surprisingly find that RanSAM outperforms FullSAM, this suggests that sampling can be regarded as a method of data augmentation in some way, which lead to significant enhancement in generalization capability by increasing the diversity of the training sequence. Unfortunately, RanSAM treats each interaction uniformly and may also drop lots of high-quality behaviors, making the improvement limited. Different from these baselines, the proposed AutoSAM adaptively explore the skew distribution of the raw input, and then enhance the sequntial recommender with informative samples. Consequently, our method performs best on all datasets, which gains an average improvement over the best baseline of **7.40%**, **7.99%** on Recall@10, NDCG@10, and **6.47%**, **7.60%** on Recall@20, NDCG@20 respectively.

Generality Analysis

To investigate the general applicability of AutoSAM to other models, we conduct a performance comparison between models using different sampling or filtering methods on various architectures, including RNN-based GRU4Rec [Hidasi *et al.*, 2015], CNN-based NextitNet [Yuan *et al.*, 2019] and Transformer-based SASRec [Kang and McAuley, 2018]. It should be noted that since RETR and SDIM are improvements based on attention mechanisms, they are not universally applicable to other architectures. Figure 2 presents the comparison results in terms of test NDCG@10 on Yelp and Tmall datasets. Note that "+ MLP Filter" corresponds to FMLP-Rec. It is evident that all the sequential recommenders demonstrate the most significant improvements through the integration of automatic sampling, highlighting the generality of our AutoSAM for different backbones.

Method	N@10	R@10	k	N@10	R@10	λ	N@10	R@10
AutoSAM	**0.0672**	**0.1117**	$2e^{-1}$	0.0663	0.1111	0.00	0.0663	0.1109
w/o r^{pre}	0.0663	0.1109	$2e^{-2}$	0.0666	0.1113	0.25	0.0668	0.1115
w/o r^{pp}	0.0603	0.1016	$2e^{-3}$	**0.0672**	**0.1117**	**0.50**	**0.0672**	**0.1117**
w/o r	0.0581	0.0984	$2e^{-4}$	0.0664	0.1109	0.75	0.0634	0.1057
			$2e^{-5}$	0.0596	0.0995	1.00	0.0603	0.1016

Table 4: Ablation study of each reward component, scaling factors k and trade-off parameter λ on Alipay dataset. "w/o r" means using the random initialized sampler throughout. We fix λ as 0.5 when tuning k, and fix k as $2e^{-3}$ when tuning λ.

Dataset	Sequence Length			
	0∼25	26∼50	51∼75	76∼100
Yelp	5.97	8.33	13.19	14.63
Tmall	13.45	14.58	16.56	17.40

Table 5: The Recall@20 improvement (%) compared to the original model of users grouped by varying lengths.

Ablation Study of Sampling Reward

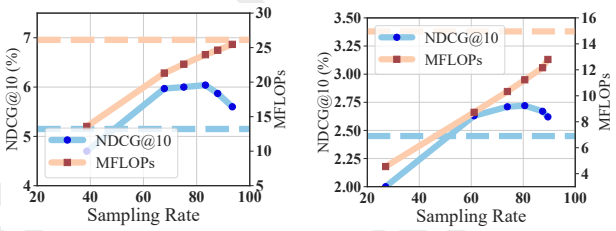
In the left column of Table 4, we analyze the efficacy of each component of the reward. From the results, we can find that removing either of the components decreases the performance. Besides, r^{pp} plays a more important role than r^{pre}. A possible reason is that r^{pp} could more directly reflect representativeness of a single item. We further show the impact of scaling factor k and trade-off parameter λ by performing a grid search in Table 4. The model achieves the best performance at $(k, \lambda) = (2e^{-3}, 0.5)$. The results demonstrate the effectiveness of the incorporation of these two aspects, and it also exhibits the stability of AutoSAM while the different settings of k within the appropriate range bring few influence.

Efficiency and Effectiveness of Different Relax Factors

In practice, it is necessary to balance efficiency and effectiveness when adapting to online services. Thus, we control the relax factors b in Eq. 11 to solve this problem. The performance and the computation cost of the well-trained model in inference w.r.t. average sample rate by setting relax factor from $\{-0.5, 0, ...2\}$ are shown in Figure 3. Note that the computation cost is measured with million floating-point operations (MFLOPs). We surprisingly observe that even preserving 50% interactions can achieve the comparable performance as baseline. However, while the sample rate reaches 80%, the performance becomes to drop which probably because the sampler leads to a uniform distribution. Besides, the shortened sequences save lots computations especially with small b, highlighting the value in practical applications.

Impact of Sequence Length

AutoSAM can model users' non-uniform interest over interacted items, so the method should compatible with different sequence length. To verify such effectiveness, we analyze the performance improvement of AutoSAM on sequences of different lengths. As shown in Table 5. We can find that **(1)** AutoSAM is also effective for users with short sequences. **(2)** Longer sequences do maximize the effectiveness of sampling, This is probably because longer sequences not only contain richer information but also tend to include noise, offering greater potential for optimization.



(a) Tmall (b) Yelp

Figure 3: Performance and computation cost w.r.t. sample rates by setting relax factor from $\{-0.5, 0, ..., 2\}$. The dashed line represents the baseline model, i.e., FullSAM.

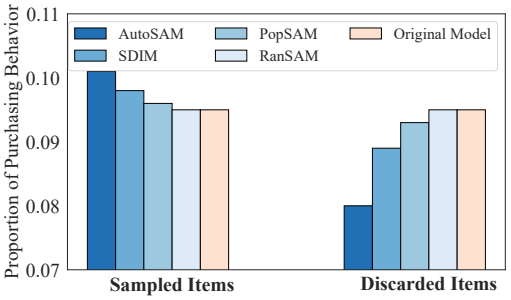


Figure 4: Portion of purchasing behaviors of sampled and discarded interactions on Tmall Dataset.

Sampling Quality Evaluation

Next, we assess the sample quality by comparing the proportion of purchasing behaviors on sampled and discarded items across different sampling strategies. Conventionally, items with purchase or favorite actions are deemed more critical than clicked ones, as they reflect stronger user intent. As illustrated in Figure 4, AutoSAM preserves the highest ratio of purchasing behaviors, indicating that the learned probability distributions better capture user preferences. Notably, AutoSAM retains a substantial number of clicked items. A plausible explanation is that clicked items can still significantly reflect user interests, though the underlying mechanisms governing such behaviors are often complex and multifaceted.

6 Conclusion

In this work, we proposed a general automatic sampling framework, named AutoSAM, to non-uniformly treat historical behaviors. Specifically, a light-weighted sampler was first leveraged to adaptively explore the distribution of raw input, so that the sequential recommender systems (SRS) could be trained with more informative and diverse samples. Considering the challenges of non-differentiable actions and the necessity to introduce multiple decision factors for sampling, we further introduced a novel reinforcement learning-based method to guide the training of the sampler in an end-to-end manner. We conducted extensive experiments on four public datasets. The experimental results showed that the AutoSAM could obtain a higher performance gain by adaptively sampling informative items. We hope this paper could inspire more works to be proposed from data perspective for SRS.

Acknowledgements

This research was supported by grants from the grants of Provincial Natural Science Foundation of Anhui Province (No.2408085QF193), USTC Research Funds of the Double First-Class Initiative (No. YD2150002501), the National Natural Science Foundation of China (62337001), the Key Technologies R & D Program of Anhui Province (No. 202423k09020039) and the Fundamental Research Funds for the Central Universities (No. WK2150110032).

References

[Cao *et al.*, 2022] Yue Cao, Xiaojiang Zhou, Jiaqi Feng, Peihao Huang, Yao Xiao, Dayao Chen, and Sheng Chen. Sampling is all you need on modeling long-term user behaviors for ctr prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2974–2983, 2022.

[Cheng *et al.*, 2021] Mingyue Cheng, Fajie Yuan, Qi Liu, Shenyang Ge, Zhi Li, Runlong Yu, Defu Lian, Senchao Yuan, and Enhong Chen. Learning recommender systems with implicit feedback via soft target enhancement. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 575–584, 2021.

[Cheng *et al.*, 2022] Mingyue Cheng, Zhiding Liu, Qi Liu, Shenyang Ge, and Enhong Chen. Towards automatic discovering of deep hybrid network architecture for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 1923–1932, 2022.

[Cheng *et al.*, 2024] Mingyue Cheng, Hao Zhang, Qi Liu, Fajie Yuan, Zhi Li, Zhenya Huang, Enhong Chen, Jun Zhou, and Longfei Li. Empowering sequential recommendation from collaborative signals and semantic relatedness. In *International Conference on Database Systems for Advanced Applications*, pages 196–211. Springer, 2024.

[Covington *et al.*, 2016] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[Dathathri *et al.*, 2019] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*, 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 *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 143–153, 2021.

[Guo *et al.*, 2017] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[He *et al.*, 2023] Zhicheng He, Weiwen Liu, Wei Guo, Jiarui Qin, Yingxue Zhang, Yaochen Hu, and Ruiming Tang. A survey on user behavior modeling in recommender systems. *arXiv preprint arXiv:2302.11087*, 2023.

[Hidasi *et al.*, 2015] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[Hidasi *et al.*, 2016] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 241–248, 2016.

[Kang and McAuley, 2018] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[Li *et al.*, 2017] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1419–1428, 2017.

[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*, pages 967–977, 2021.

[Lin *et al.*, 2023] Weilin Lin, Xiangyu Zhao, Yejing Wang, Yuanshao Zhu, and Wanyu Wang. Autodenoise: Automatic data instance denoising for recommendations. *arXiv preprint arXiv:2303.06611*, 2023.

[Lin *et al.*, 2024] Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, Weizhu Chen, et al. Not all tokens are what you need for pretraining. *Advances in Neural Information Processing Systems*, 37:29029–29063, 2024.

[Liu *et al.*, 2018] Qi Liu, Hong-Ke Zhao, Le Wu, Zhi Li, and En-Hong Chen. Illuminating recommendation by understanding the explicit item relations. *Journal of Computer Science and Technology*, 33:739–755, 2018.

[Liu *et al.*, 2022] Zhiding Liu, Mingyue Cheng, Qi Liu, Enhong Chen, et al. One person, one model–learning compound router for sequential recommendation. *arXiv preprint arXiv:2211.02824*, 2022.

[Luo *et al.*, 2024] Yucong Luo, Qitao Qin, Hao Zhang, Mingyue Cheng, Ruiran Yan, Kefan Wang, and Jie Ouyang. Molar: Multimodal llms with collaborative filtering alignment for enhanced sequential recommendation. *arXiv preprint arXiv:2412.18176*, 2024.

[Pi *et al.*, 2020] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun

Gai. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2685–2692, 2020.

[Qin *et al.*, 2020] Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. User behavior retrieval for click-through rate prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2347–2356, 2020.

[Qin *et al.*, 2021] Yuqi Qin, Pengfei Wang, and Chenliang Li. The world is binary: Contrastive learning for denoising next basket recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 859–868, 2021.

[Qin *et al.*, 2023] Jiarui Qin, Weinan Zhang, Rong Su, Zhirong Liu, Weiwen Liu, Guangpeng Zhao, Hao Li, Ruiming Tang, Xiuqiang He, and Yong Yu. Learning to retrieve user behaviors for click-through rate estimation. *ACM Transactions on Information Systems*, 2023.

[Qiu *et al.*, 2022] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 813–823, 2022.

[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*, pages 811–820, 2010.

[Rendle *et al.*, 2012] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[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*, pages 1441–1450, 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*, pages 565–573, 2018.

[Wang *et al.*, 2017] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pages 1–7. 2017.

[Wang *et al.*, 2021] 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*, pages 373–381, 2021.

[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*, volume 33, pages 346–353, 2019.

[Xie *et al.*, 2022] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 1259–1273. IEEE, 2022.

[Xu *et al.*, 2019] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. Graph contextualized self-attention network for session-based recommendation. In *IJCAI*, volume 19, pages 3940–3946, 2019.

[Yao *et al.*, 2022] Zhiyu Yao, Xinyang Chen, Sinan Wang, Qinyan Dai, Yumeng Li, Tanchao Zhu, and Mingsheng Long. Recommender transformers with behavior pathways. *arXiv preprint arXiv:2206.06804*, 2022.

[Yuan *et al.*, 2019] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 582–590, 2019.

[Zhang *et al.*, 2013] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 785–788, 2013.

[Zhang *et al.*, 2024a] Hao Zhang, Mingyue Cheng, Qi Liu, Yucong Luo, Rui Li, and Enhong Chen. Learning recommender systems with soft target: A decoupled perspective. In *International Conference on Database Systems for Advanced Applications*, pages 363–373. Springer, 2024.

[Zhang *et al.*, 2024b] Rujiao Zhang, Hao Zhang, Yucong Luo, Zhiding Liu, Mingyue Cheng, Qi Liu, and Enhong Chen. Learning the dynamics in sequential recommendation by exploiting real-time information. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4288–4292, 2024.

[Zhang *et al.*, 2025] Hao Zhang, Mingyue Cheng, Qi Liu, Junzhe Jiang, Xianquan Wang, Rujiao Zhang, Chenyi Lei, and Enhong Chen. A comprehensive survey on cross-domain recommendation: Taxonomy, progress, and prospects. *arXiv preprint arXiv:2503.14110*, 2025.

[Zhao *et al.*, 2022] Wayne Xin Zhao, Zihan Lin, Zhichao Feng, Pengfei Wang, and Ji-Rong Wen. A revisiting study of appropriate offline evaluation for top-n recommendation algorithms. *ACM Transactions on Information Systems*, 41(2):1–41, 2022.

[Zhou *et al.*, 2022] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-enhanced mlp is all you need for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2388–2399, 2022.