Flow-based Time-aware Causal Structure Learning for Sequential Recommendation

Hangtong Xu[1] , **Yuanbo Xu**[1*] , **Huayuan Liu**[1] and **En Wang**[1*]
[1]MIC Lab, College of Computer Science and Technology, Jilin University
{xuht24, liuhy5522}@mails.jlu.edu.cn, {yuanbox, wangen}@jlu.edu.cn,

Abstract

Sequential models aim to predict future interactions based on users' historical interaction sequences. Traditional sequential methods primarily focus on capturing intra-historical sequence dependencies, overlooking the influence of unobserved confounders in recommendation scenarios. Recent studies incorporate time as additional information helps the model capture dynamic user preferences. However, time is just the external manifestation of the influence of confounders but not the actual cause of the dynamic of user preference. Additionally, improperly integrating time with item embeddings can obstruct the model's ability to capture sequence dependencies. To address these challenges, we first revisit the sequential recommendation problem from a causal perspective and incorporate confounders as a new task. We propose a new framework—Flow-based Time-aware Causal Structure for Sequential Recommendation (FCSRec)—explicitly incorporating unobserved confounders' influence in the recommendation process. Specifically, we use Normalizing Flows to learn the causal graph of confounders and incorporate time information as conditional info to capture confounders' time-sensitive representations. To balance the influence of confounders and sequence dependencies, we introduce a classifier-free training paradigm by randomly masking the influence of confounders during training to encourage the model to learn both sequence dependencies and confounders' influence equally. We validate FCSRec on manifold real-world datasets, and experimental results show that FCSRec outperforms several state-of-the-art methods in recommendation performance. Our code is available at Code-link.

1 Introduction

Sequential recommendation systems play a crucial role in filtering and personalizing content for users on digital platforms. Sequential recommendation models provide efficient real-time recommendations by predicting future interactions based on users' historical interaction sequences, item attributes, and various contextual features.

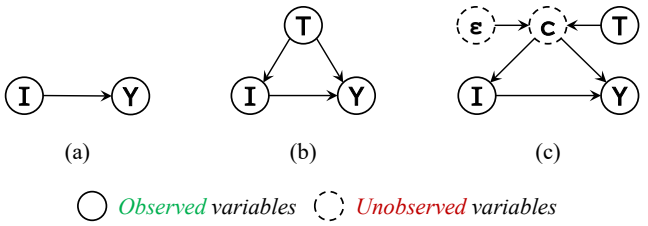


Figure 1: Conventional modeling versus a more rational modeling approach. (a) is the conventional modeling, (b) uses time as additional information where time acts as the confounder, and (c) is the rational modeling proposed in this work. **I**→Item; **Y**→User feedback; **T**→Timestamp; $\boldsymbol{\epsilon}$→exogenous variables of confounders; **c**→confounders.

Recent advancements in sequential recommendation techniques have led to the development of various methods that leverage the correlations in users' historical interactions. These approaches aim to effectively learn the sequence dependencies within users' interaction history to improve the recommendation performance. Recurrent Neural Networks (RNNs) [Hidasi and Karatzoglou, 2018; Kang and McAuley, 2018] are frequently used to capture different aspects of user engagement and sequential dependencies. Meanwhile, Graph Neural Networks (GNNs) [Xu *et al.*, 2019] are employed to identify complex co-occurrence relationships and higher-order structural dependencies within sequential data. As illustrated in Figure 1 (a), traditional sequential recommendation methods primarily focus on modeling the sequential dependencies in historical interaction sequences ($\mathbf{I} \rightarrow \mathbf{Y}$). As a result, these methods often produce highly similar recommendations for users with identical interaction sequences.

Several methods have been proposed that incorporate time as additional information to improve performance in modeling user-item interactions [Wang *et al.*, 2022a; Jiang *et al.*, 2023]. As illustrated in Figure 1 (b), time simultaneously influences both items and interactions: items may exist in different states at different times ($\mathbf{T} \rightarrow \mathbf{I}$), and user preferences can change over time ($\mathbf{T} \rightarrow \mathbf{Y}$), where time serves as a con-

* Corresponding author.
https://github.com/MICLab-Rec/FCSRec

founder. By incorporating time information, models can better capture time-sensitive item embeddings. However, time is just one of many observable confounders that affect user-item interactions, and numerous unobserved confounders are yet to be considered. Furthermore, directly combining time and item embeddings for joint learning risks pushing the model toward the local optimum, potentially failing to capture item sequential dependencies and underutilizing the representation space, thus harming the model's performance. This approach risks inadequately capturing the dependencies within item sequences while underutilizing the representation space for both items and time, which could negatively impact the model's performance.

The unobservability of confounders limits our access to observable features, such as items or user attributes related to them. Moreover, the causal graph between confounders is also unknown, posing significant challenges in modeling their influences. As shown in Figure 1 (c), confounders primarily depend on their external variables and experience periodic or irregular changes over time, finally influencing the interaction between users and items, resulting in shifts in item states as well as users' long-term and short-term preferences. We argue that time does not directly affect the state of items or user preferences. Instead, it is the impact of confounders that evolves over time. In other words, time information serves as a reflection of the changing influence of confounders. Specifically, time information serves as an external manifestation of the influence of confounders.

To address these challenges, we reformulate the sequential recommendation task to incorporate the influence of the unobserved confounders. We demonstrate that we can transform any acyclic causal graph into a topological causal order to model confounders and learn the causal relations between them, allowing us to learn the weights of the causal order without knowing the actual causal graph, thereby obtaining a usable causal graph. Specifically, we proposed a framework based on Normalizing Flows, where we sample the exogenous variables of confounders from a distribution and obtain the final confounder representation using the given causal order. Meanwhile, to capture the temporal dynamics of confounders, we incorporate time as conditional information to the sampler, thereby obtaining time-dependent representations of the confounders. To the best of our knowledge, we are the first to explicitly consider the influence of unobserved confounders on the performance of sequential recommendation models.

Furthermore, we propose a classifier-free training paradigm to better balance the importance of confounder influence and sequence dependencies. By randomly masking the influence of confounders, the model is encouraged to treat sequence dependencies and confounders' influence as two equally important tasks during the learning process. Additionally, the mask conceals the confounders' influence on items, ensuring the model better uncovers the sequence dependencies within the historical sequence. Finally, we propose a new framework named **F**low-based Time-aware **C**ausal **S**tructure for Sequential **Rec**ommendation (FCSRec) to learn confounders and sequence dependencies jointly. We validate the model's performance on eight datasets of varying sizes and types, and experimental results show that FCSRec attains superior recommendation performance to several state-of-the-art methods. The contributions of our work can be summarized as follows:

- We have reformulated the sequential recommendation task by incorporating the influence of restocking unobserved confounders as one of the key objectives.
- We propose a confounder modeling method based on Normalizing Flows, which can obtain a usable causal graph without knowing the exact causal graph of the confounders. Additionally, we incorporate time as conditional information to capture the sensitivity of confounders to temporal changes.
- We propose a classifier-free training paradigm that effectively balances the contributions between sequence dependencies and confounders. During training, it simultaneously models the sequence dependencies and captures the influence of confounders.
- We validate the proposed FCSRec on eight real-world datasets, and experimental results demonstrate that FCSRec outperforms several state-of-the-art methods in recommendation performance.

2 Preliminaries

We clarify the conventional sequential recommendation problem and the sequential recommendation problem influenced by confounders as proposed in this paper as follows:

Conventional Sequential Recommendation

Given the specific user u and his/her historical sequence $\mathbf{X}$. The sequential recommendation problem infers the dynamic preferences and provides the top K recommendation list, which contains K items that the user might be most likely to interact with in the next time step. It can be formulated as the following equation:

$$P(x_{t+1}|\mathbf{X}_{1:t}) = f(\mathbf{X}_{1:t}), \tag{1}$$

where f is the abstract symbol of any sequential recommender, $\mathbf{X}_{1:t}$ is the sequence data.

Sequential Recommendation under Confounders

Given the specific user u and his/her historical sequence $\mathbf{X}$. The sequential recommendation problem infers the dynamic preferences and models the influence of unobserved confounders, provides the top K recommendation list under the given timestamp t, which contains K items that the user might be most likely to interact with in the next time step. It can be formulated as the following equation:

$$\begin{aligned} &P(x_{t+1}|\mathbf{X}_{1:t}) = f(\mathbf{X}_{1:t}, \mathbf{c}), \\ &where \quad \mathbf{c} = \mathcal{F}(\mathbf{X}_{1:t}, \mathbf{T}_{1:t}), \end{aligned} \tag{2}$$

where $\mathcal{F}$ is the causal model to learn the causal graph and capture the influence of confounders, $\mathbf{c}$ is the representation of confounders for sequential recommendation.

3 Method

3.1 Structure Causal Model

To model the influence of confounders in sequential recommendation, we first utilize a Structure Causal Model (SCM) to learn the causal relationships between confounders. The SCM refers to a tuple $\mathcal{M} = (\tilde{f}, P_{\boldsymbol{\epsilon}})$ describing the data-generation process that transforms a set of k exogenous variables, $\boldsymbol{\epsilon} \sim P_{\boldsymbol{\epsilon}}$, into a set of k endogenous variables (confounders), $\mathbf{c}$, according to $\tilde{f}$, we can formulated the process as follows:

$$\boldsymbol{\epsilon} := (\epsilon_1, \epsilon_2, \ldots, \epsilon_k) \sim P_{\boldsymbol{\epsilon}}, \tag{3}$$

where $P_{\boldsymbol{\epsilon}}$ represents the distribution of exogenous variables $\boldsymbol{\epsilon}$, we use the normal Gaussian distribution in this paper. Specifically, the exogenous variables $\boldsymbol{\epsilon}$ are mutually independent:

$$p(\boldsymbol{\epsilon}) = \prod_{i=0}^{k} p(\epsilon_i). \tag{4}$$

Given the exogenous variable ϵ_i, each i-th component of $\tilde{f}$ maps the i-th ϵ_i to the i-th confounder c_i:

$$\begin{aligned} &\mathbf{c} := (c_1, c_2, \ldots, c_k), \\ &c_i = \tilde{f}_i(c_{pa_i}, \epsilon_i) \quad for \;\; i = 1, 2, \ldots, k. \end{aligned} \tag{5}$$

where the c_{pa_i} is the directly cause (causal parents) of c_i represented by a adjacency matrix $\mathbf{A}$ of the causal graph, and the value of a_{ij} in $\mathbf{A}$ can be viewed as an indicator vector, where $a_{ij} = 1$ signifies that node i is the parent node of node j, indicating that node j is influenced by node i. In contrast, $a_{ij} = 0$ implies that node j and node i are unrelated.

Unfortunately, in real-world scenarios, causal graphs are rarely directly available. The causal relationships between nodes need to be learned through causal discovery methods. Therefore, the main objective of Structural Causal Models (SCM) is to learn the adjacency matrix $\mathbf{A}$ of the causal relationships behind the nodes. Due to the unobservability of confounders, learning the causal relationships between confounders becomes more challenging.

To address this challenge, we shift our perspective to learning the strength of the causal paths in a given causal graph. Based on the acyclic property of $\mathbf{A}$, we can pick a causal order $\boldsymbol{\pi}$ to describe the causal relationships between confounders, the permutation $\boldsymbol{\pi}$ to be the causal ordering if the SCM $\mathcal{M}$ of the confounders if and only if for every c_i, that directly cause c_j, we have the $\pi_i \leq \pi_j$. We formalize the above content as the following theorem:

Theorem 1. *For any given DAG $G = (V, E)$, there always exists a topological order $\boldsymbol{\pi}$ that satisfies both the monotonicity and triangular increasing structure required by a Triangular Monotonic Increasing (TMI) map.*

Theorem 1 implies that, for any number of confounders, if an acyclic adjacency matrix can represent the causal relationships between the confounders, then this adjacency matrix can be formalized as a lower triangular causal order $\boldsymbol{\pi}$. By utilizing the theorem, we successfully transform the problem into the task of learning the edge weight of the given causal order $\boldsymbol{\pi}$, where the $\pi_i = 1$ if the row index i is greater than or equal to the column index j, and zero otherwise, the causal model need learns the edge weights of this causal order $\boldsymbol{\pi}$. The proof of Theorem 1 can be found in the Appendix A.2.

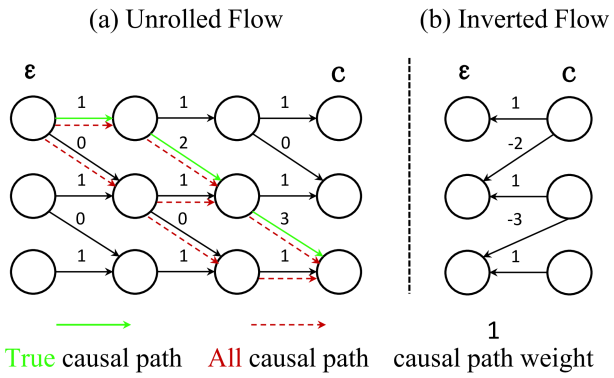


Figure 2: Example of the linear **SCM** $\{c_1 := \epsilon_1;\ c_2 := 2c_1 + \epsilon_2;\ c_3 := 3c_2 + \epsilon_3\}$ written (a) without recursions with each step made explicit; and (b) writing $\boldsymbol{\epsilon}$ as a function of $\mathbf{c}$. The green arrows show the true causal influence path of ϵ_1 on c_3 for all equations from $\boldsymbol{\epsilon}$ to $\mathbf{c}$, and the red dashed arrows show the total causal influence path of ϵ_1 on c_3 for all equations from $\boldsymbol{\epsilon}$ to $\mathbf{c}$.

3.2 Time guidance Causal Normalizing Flows

Causal Normalizing Flows

Given the causal order $\boldsymbol{\pi}$, we reformulate the task as learning the weights of causal paths between confounders. Existing research on causal discovery has proposed various methods, including DNNs [Nasr-Esfahany *et al.*, 2023], GANs [Xia *et al.*, 2022], and DDPMs [Chao *et al.*, 2023]. However, the high complexity of these methods makes seamless integration with recommendation system algorithms impossible. We aim to learn the complete causal-generating process using a neural network that is as simple as possible. Normalizing flows (NFs) are a natural choice approximating a broad class of causal data-generating processes [Balgi *et al.*, 2022; Javaloy *et al.*, 2024], and we use Autoregressive Normalizing flows (ANFs) in practice.

Given the observed user feedback data $\mathbf{X}$ and the number of confounders k, an autoregressive normalizing flow model $T(\cdot)$ is a neural network with parameters θ that takes $\boldsymbol{\epsilon}$ and $\boldsymbol{\pi}$ as input and outputs the representation of the confounders:

$$\mathbf{c} := T_\theta(\boldsymbol{\epsilon}, \boldsymbol{\pi}). \tag{6}$$

In ANFs the i-th output of each layer l of the network, denoted by z_i^l, is computed as:

$$\begin{aligned} z_i^{l-1} &= \mathcal{T}_i^l(z_i^{l-1}; \mathbf{h}_i^{l-1}), \\ \mathbf{h}_i^{l-1} &= \mathcal{F}_i(z_{pa_i}^l), \end{aligned} \tag{7}$$

where $\mathcal{T}_i$ and $\mathcal{F}_i$ termed the transformer and the conditioner. The transformer is a strictly monotonic function of z_i^{l-1},

We use the "transformer" to indicate the function for establishing a mapping between the source distribution (usually a simple base distribution, such as a Gaussian distribution) and the target distribution (the more complex data distribution), and "Transformer" for the model Transformer.

while the conditioner only takes the variables preceding z_i as input. For simplicity, we provide an example of using ANFs to learn the confounder weights under a linear SCM, as shown in Figure 2. The architecture shown in Figure 2 (a) defined the ANFs as a function from $\boldsymbol{\epsilon} \rightarrow \mathbf{c}$, ANFs will learn spurious correlations due to the fully connected nature of MLPs, which will harmful the model performance, if and only if ANFs have extra information such as true causal graph $\mathbf{A}$ to learn necessary zeroes to fulfill the causal consistency, but we only have the causal order $\boldsymbol{\pi}$.

To mitigate this issue and enhance the stability of our model, we adopt an alternative architecture as shown in Figure 2 (b), building a causal ANFs from $\mathbf{c} \rightarrow \boldsymbol{\epsilon}$, this architecture is capable of capturing all indirect dependencies of $\mathbf{c}$ on $\boldsymbol{\epsilon}$, because ANFs compute the inverse sequentially enhanced the indirect influence of ϵ_1 on c_3 via c_2 has to generate c_2 first necessarily. Based on the architecture discussion above, we rewrite the computed of ANFs as follows:

$$\begin{aligned} z_i^l &= \mathcal{T}_i^l(z_i^l; \mathbf{h}_i^l), \\ \mathbf{h}_i^l &= \mathcal{F}_i(z_{pa_i}^{l-1}). \end{aligned} \tag{8}$$

Time guidance

In the sequential recommendation scenario, the user's interaction environment continuously evolves over time, meaning that the influence of confounders should not remain the same at different timestamps. Additionally, the intrinsic characteristics of the confounders influence their time-varying nature, such as the variations in temperature at different timestamps.

To model the varying influence of confounders over time, we incorporate time as conditional information to guide the generation of confounders. We categorize the timestamp t into three levels: month, day, and hour, represented as a triple tuple $t = (t_{\text{month}}, t_{\text{day}}, t_{\text{hour}})$ to capture changes in confounders over different periods effectively. We fed time-based information into a multi-layer perceptron (MLP). By inputting different levels of temporal information into an MLP, we can integrate the influences of various time granularities, thus obtaining time-conditioned information of the confounders. Specifically, we formalize the generation of time-conditioned information as follows:

$$\mathsf{Info_t} = \text{MLP}(t_{\text{month}} \parallel t_{\text{day}} \parallel t_{\text{hour}}). \tag{9}$$

In this formulation, $\mathsf{Info_t}$ represents the time-conditioned information of confounders. Based on the obtained time-conditioned information, we have rewritten the conditioner in the formula 8, incorporating the temporal information as a prior condition for the generation of confounders:

$$\mathbf{h}_i^l = \mathcal{F}_i(z_{pa_i}^{l-1}, \mathsf{Info_t}). \tag{10}$$

This approach enables the model to capture the confounders' time-sensitive representation under any specific timestamp t, which helps the model learn item representation under the influence of confounders at the timestamp t more accurately.

3.3 Item-specific causal strength

We can drive the mixed item representation of feedback data at a specific timestamp with the reconstructed causal time-sensitive representation of confounders $\mathbf{c}$. For k confounders,

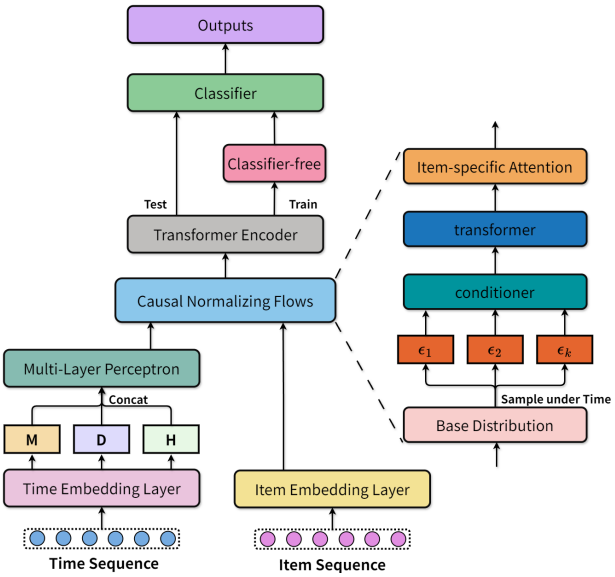


Figure 3: The architecture of our framework FCSRec.

an item may be influenced by some rather than all. To maintain this characteristic, we utilize the attention mechanism to capture how different confounders influence various items. Specifically, we calculate the attention score as follows:

$$\begin{aligned} Q &= f(\mathsf{Norm}(\mathbf{I})), & K &= g(\mathsf{Norm}(\mathbf{c^t})), \\ V &= q(\mathbf{c^t}), & score &= \mathsf{Softmax}(\frac{QK^\top}{\sqrt{d}}), \\ \mathbf{c_I^t} &= score \cdot V, \end{aligned} \tag{11}$$

where f, g and q are learnable linear layers, $\mathsf{Norm}(\cdot)$ means normalization, d is the latent embedding size, $\mathbf{c^t}$ is the time-sensitive representation of confounders, $\mathbf{I}$ is the embedding of items and $\mathbf{c_I^t}$ is the representation of confounders various influence on items.

We use a Transformer encoder to capture the sequential dependencies of items, which can be formalized as follows:

$$s^I = encoder(\mathbf{X}, \mathbf{I}). \tag{12}$$

Then we mix the s^I and $\mathbf{c_I^t}$ to get the reconstructed mixed item representation of feedback data as follows:

$$z = \mathsf{FFN}(s^I + \mathbf{c_I^t}), \tag{13}$$

where $\mathsf{FFN}(\cdot)$ denotes the feed-forward layer, but in practice, we found that simply adding s^I and $\mathbf{c_I^t}$ is enough. z integrates item embeddings with causal representations of confounders, producing an item representation that reflects the influence of confounders at a specific timestamp t. This item representation is then input into the decoder or classifier to calculate the log-probability of the user u interacting with the item:

$$y_{i,t} = decoder(z). \tag{14}$$

Identification

4 Training and Prediction

4.1 Training process

Classifier-free Guidance Paradigm

We propose a classifier-free variant method for recommendation systems to better balance the importance of confounders and item sequential dependencies, inspired by work in the image domain [Ho, 2022]. Specifically, we introduce a control factor, α, to regulate the strength relationship between confounders and sequential dependencies. The paradigm can be formalized as follows:

$$\begin{aligned} \hat{z} &= (1-\alpha)\cdot \mathsf{FFN}(s^I) + \alpha \cdot \mathsf{FFN}(s^I + \mathbf{c_I^t}), \\ \alpha &= \mathbb{I}(\cdot), \end{aligned} \tag{15}$$

where $\mathbb{I}$ is an indicator function, which equals 1 if a value randomly sampled from a normal distribution is greater than 0.5; otherwise, it equals 0. When $\alpha = 0$, the model considers only the sequential dependencies of the items, while when $\alpha = 1$, the model equally considers both the confounders and the sequential dependencies. It is important to emphasize that the same operation is applied to each item in the sequence during the forward process. The randomness in the missing confounders brings the following benefits. During the forward process, the model learns both pure sequential dependencies and those influenced by confounders. The relevant proof can be found in Appendix A.3.

Objective function

During the training processing, we employ the standard autoregressive fashion. Specifically, FCSRec takes the historical sequence that excludes the last token as the source, and the sequence excludes the first token as a target. At each time step i, FCSRec aims at predicting the i+1 th token, i.e., maximizing the probability of the i + 1 th interacted item.

$$\mathcal{L}_{CE} = -\sum_{i=1}^{N} \log(y_{i,t}), \tag{16}$$

where $y_{i,t}$ is the probability of target item at step i in time t.

4.2 Prediction process

During the recommendation stage, FCSRec first extracts the last row $y_{n,t} \in \mathbb{R}^{|\mathbf{I}|}$ from $\mathbf{P}$ which contains the information of all interacted items in the historical sequence. Then, it ranks all candidate items according to the probabilities and retrieves K items as the top-K recommendation list.

5 Experiments and Discussions

5.1 Experimental Settings

Datasets

To comprehensively and fairly evaluate the models' effectiveness, we conducted experiments using nine publicly available datasets encompassing a variety of recommendation scenarios (such as movies and pois) and different densities. We select five datasets of varying sizes ranging from 100k to 10M: `Beauty`, `ML-100K`, `NYC`, `TKY`, `ML-1M`, `Gowalla` and `ML-10M` to evaluate the robustness of the model to the dataset size. Following prior works [Jiang *et al.*, 2024; Xu *et al.*, 2024], we remove the "inactive" users who interact with fewer than 20 items and the "unpopular" items who have interacted with users less than 10 times. We set the maximum sequence length l of each dataset according to the average one. Towards the data partition, we select each user's last previously un-interacted item as the target during the recommendation procedure and all the prior items for training.

Dataset	Scale	# Users	# Items	# Interactioons	Sparsity
ML-100K	Tiny	932	1,152	97,746	90.90%
Beauty		1,664	36,938	56,558	99.91%
NYC	Small	1,031	5,135	142,237	97.31%
TKY		2,267	7,873	444,183	97.51%
ML-1M	Base	6,034	3,260	998,428	94.92%
Brightkite		5,714	48,181	1,765,247	99.36%
Gowalla	Large	42,461	101,269	2,199,786	99.95%
ML-10M		69,865	9,708	9,995,230	98.53%

Table 1: Data Statistics (after pre-processed). The eight datasets are categorized into 4 scales Tiny, Small, Base, and Large which contain 50K 100K, 150K 500K, 1M 2M, and 2M 10M user-item interactions, respectively.

Baselines

A range of advanced models have been proposed to enhance sequential recommendation by capturing temporal patterns and user preferences. **GRU4Rec** [Hidasi, 2015] utilizes RNNs to model dynamic user behavior, while **NextItNet** [Yuan *et al.*, 2019] adopts a CNN-based architecture to capture both short- and long-range dependencies. Transformer-based methods like **SASRec** [Kang and McAuley, 2018] and its time-aware variants **TiSASRec** [Li *et al.*, 2020] and **TiCoSeRec** [Dang *et al.*, 2023] model sequential dependencies with attention mechanisms and incorporate temporal dynamics. **CLS4Rec** [Xie *et al.*, 2022] leverages contrastive learning to better distinguish positive and negative interactions. **CD-SASRec** [Chen and Li, 2024] introduces a causality-driven framework to improve user modeling from a causal inference perspective.

Setups

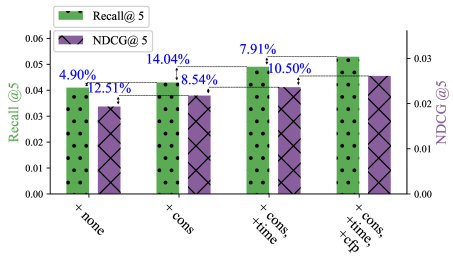
We implement FCSRec and baselines in PyTorch. All models are trained with the Adam optimizer with early stopping at patience = 10. We set the learning rate to 1e-3 and the l_2-regularization weight to 1e-6. For FCSRec, we tune the hyper-parameter concepts k in the range of $[1, 8]$ for different datasets. To detect significant differences in FCSRec and the best baseline on each dataset, we repeated their experiments five times by varying the random seeds. We choose the average performance to report. All ranking metrics are computed at cutoffs K=[10,20] for the Top-K recommendation. Our implementation of the baselines is based on the original papers or the open-source codebase Recbole [Zhao *et al.*, 2021].

5.2 Overall Performance Comparison

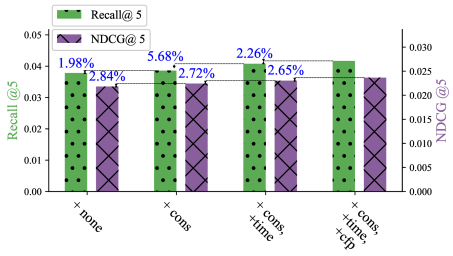
The comparison between FCSRec and various baselines is shown in Table 2. The best results (compared across two

Datasets	Scale	Model	GRU4Rec	NextItNet	SASRec	TiSASRec	CLS4Rec	TiCoSeRec	CD-SASRec	FCSRec
Beauty	Tiny	R@10 ↑	0.00919	0.00919	0.00854	0.00948	0.00826	0.00853	0.00824	**0.01138**
		N@10 ↑	0.00592	0.00584	0.00546	0.00589	0.00595	0.00531	0.00533	**0.00688**
ML-100K		R@10 ↑	0.06178	0.06821	0.04753	0.05339	0.05143	0.04805	0.04572	**0.09752**
		N@10 ↑	0.03161	0.02420	0.02093	0.02694	0.02414	0.02435	0.02047	**0.04137**
NYC	Small	R@10 ↑	0.03124	0.03940	0.04774	0.04838	0.04904	0.04354	0.04562	**0.05273**
		N@10 ↑	0.01682	0.02136	0.02416	0.02300	0.02471	0.02105	0.02374	**0.02565**
TKY		R@10 ↑	0.04603	0.04237	0.04961	0.04787	0.05063	0.04307	0.04783	**0.05087**
		N@10 ↑	0.02314	0.02186	0.02513	0.02411	0.02526	0.02209	0.02462	**0.02542**
Brightkite	Base	R@10 ↑	0.04431	0.04622	0.07189	0.07094	0.07268	0.02654	0.06943	**0.07551**
		N@10 ↑	0.02826	0.03288	0.05737	0.05308	0.05667	0.04813	0.05591	**0.05935**
ML-1M		R@10 ↑	0.13061	0.13582	0.14137	0.12240	0.14122	0.11343	0.13430	**0.14967**
		N@10 ↑	0.06095	0.06848	0.06520	0.05595	0.06002	0.05316	0.06113	**0.07678**
Gowalla	Large	R@10 ↑	0.05966	0.07689	0.07330	0.05262	0.05871	0.04774	0.07146	**0.08066**
		N@10 ↑	0.02494	0.03370	0.03306	0.02323	0.02918	0.02109	0.03224	**0.03707**
ML-10M		R@10 ↑	0.09126	0.09034	0.09281	0.09269	0.09269	0.09033	0.09064	**0.09818**
		N@10 ↑	0.03492	0.03643	0.03525	0.03567	0.03608	0.03646	0.03686	**0.03881**

Table 2: The overall performance comparison results of applying our model and baselines on eight real-world datasets. We evaluated the recommendation performance as a ranking task, underlined the best baseline result in each line, and put the best result in each line in bold; Higher Recall and NDCG mean better model performance. The arrow '↑' (or '↓') denotes that the higher (or lower) value means better performance on the metric. The result is calculated based on the mean of five repetitions with different random seeds for all models on each metric.



(a) ML-100K



(a) Beauty

Figure 4: Ablation study of FCSRec on ML-100K and Beauty: $none$ (without the confounders and classifier guidance paradigm, equivalent to SASRec); + $cons$ (add the influence of confounders but without time guidance); + $cons, time$ (without the classifier-free guidance paradigm); and + $cons, time, cfp$ (full version of FCSRec).

classes) are shown in bold, and the runner-ups are underlined. In summary, we have the following observations:

- The result demonstrates that the FCSRec model consistently outperforms the baselines regarding Recall and NDCG across various datasets and evaluation metrics, indicating its superior ability to recommend relevant next items to users. Remarkably, FCSRec substantially improves Recall and NDCG compared to the baselines.
- Traditional sequential recommendation algorithms, such as SASRec, achieve competitive results across all datasets by capturing the sequential dependencies between items in the historical interaction sequences. However, time-based methods like TiSASRec are sometimes hindered by the interference of time information when modeling item sequential dependencies, leading to worse performance on specific datasets than methods that do not incorporate time information.
- FCSRec ensures the model can learn both sequence dependencies and confounders' influence equally through the Classifier-free Guidance Paradigm and leverages time information by modeling confounders. As a result, it achieves superior recommendation performance compared to traditional sequence-based and time-aware recommendation methods.

5.3 Ablation Study

The Figure 4 presents results for different variants of FCSRec: $none$ (without the confounders and classifier-free guidance paradigm, equivalent to SASRec); + $cons$ (add the influence of confounders but without time guidance); + $cons, time$ (without the classifier guidance paradigm); and + $cons, time, cfp$ (full version of FCSRec). we have the following observations:

• Modeling confounders is essential for the performance of the model. After incorporating the influence of confounders, the model's performance improved, demonstrating that the influence of confounders is indeed significant. Furthermore, by utilizing time information to explore the sensitivity of con-

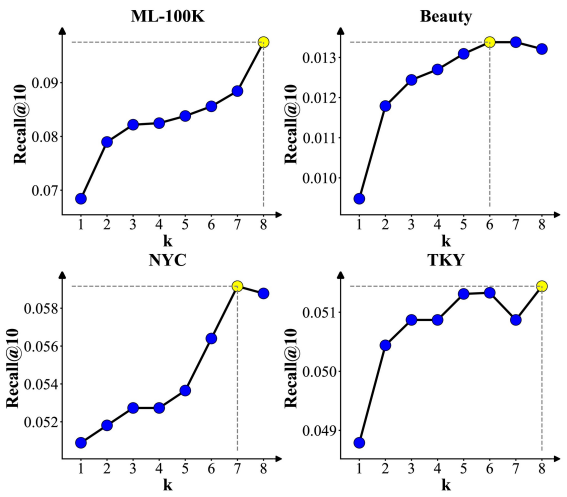


Figure 5: Sensitivity of FCSRec with different confounders number k on ML-100K, Beauty, NYC and TKY. The horizontal axes of all sub-figures are the variable k.

founders to time further, the model becomes more suitable for sequential recommendation scenarios, leading to an additional performance enhancement.

• The classifier-free guidance paradigm is essential for the model's joint modeling of sequential dependencies and confounders. Through this approach, the model can complete two tasks during training: pure sequence modeling and sequence modeling under the influence of confounders, ensuring that the model can better capture item relationships, mitigate the negative impact of confounders, and ultimately lead to further performance improvement.

5.4 Effect of different number of Confounders

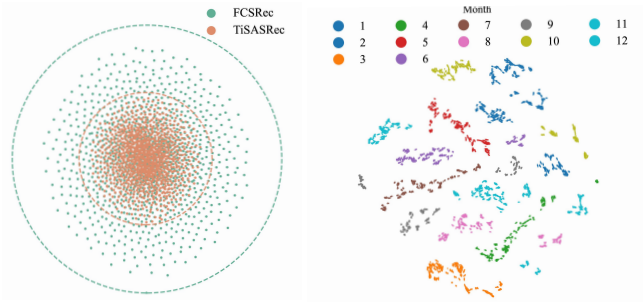
We experimented with various values of k on the ML-100k, Beauty, NYC, and TKY datasets to verify the influence of the number of confounders. As shown in Figure 5, we made the following observations:

• **Performance improvement with more confounders:** As the number of confounders increases, the model's recommendation performance improves. A larger number of confounders allows the model to capture more influences from these confounders. Additionally, the increase in confounders leads to a finer granularity in the captured influences, which enhances model performance.

• **Diminishing returns after a certain point:** When the number of confounders exceeds a certain threshold, the improvement in model performance becomes less pronounced. The increase in confounders introduces challenges in learning the underlying causal graph. Specifically, the number of new edges between confounders grows exponentially, which limits further performance gains.

5.5 Visualizing item representation on different Time

We use T-SNE to visualize the items' embedding before and after fusion of the influence of confounders on ML-100k in the month level, with $k = 4$. From Figure 6, We have the following observations:



(a) Items without confounders (b) Items with confounders

Figure 6: Visualization of the items' embedding pure and after fusion influence of confounders on ML-100k in the month level.

• FCSRec learns item representations more comprehensively through training. As shown in Figure 6 (a), the item representations are not concentrated in a single center but are spread throughout the entire representation space, indicating that FCSRec can fully explore the item representation space, resulting in more comprehensive and effective item representations.

• FCSRec successfully captures the sensitivity of confounders to time. As shown in Figure 6 (b), items are clustered into 12 distinct groups based on different months after incorporating the mixed influence of confounders, indicating that the model effectively captures the time-sensitivity of confounders. The figure also validates the sensitivity of items to time, further demonstrating the superiority of FCSRec in capturing both confounder and time effects.

6 Conclusions and Future Work

In this work, we proposed FCSRec, a novel framework incorporating unobserved confounders and their temporal dynamics into sequential recommendation systems. Our model, based on Normalizing Flows and a classifier-free training paradigm, demonstrates significant improvements in recommendation performance compared to existing state-of-the-art methods. In future work, we aim to enhance the interpretability of confounder modeling by incorporating more advanced techniques for causal inference and providing more precise explanations of the learned causal structures. Additionally, we will explore methods to improve the model's capability to infer complex causal relationships, which could further refine the recommendations and offer deeper insights into the underlying processes driving user behavior.

7 Acknowledgement

This work is supported by the Natural Science Foundation of China No. 62472196, Jilin Science and Technology Research Project 20230101067JC, National Key R&D Program of China under Grant No. 2021ZD0112501 and 2021ZD0112502, National Natural Science Foundation of China under Grant No. 62272193, National Key R&D Program of China under Grant Nos. 2022YFB3103700 and 2022YFB3103702.

References

[Balgi *et al.*, 2022] Sourabh Balgi, Jose M Pena, and Adel Daoud. Personalized public policy analysis in social sciences using causal-graphical normalizing flows. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11810–11818, 2022.

[Chao *et al.*, 2023] Patrick Chao, Patrick Blöbaum, and Shiva Prasad Kasiviswanathan. Interventional and counterfactual inference with diffusion models. *arXiv preprint arXiv:2302.00860*, 4:16, 2023.

[Chen and Li, 2024] Xingming Chen and Qing Li. Causality-driven user modeling for sequential recommendations over time. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 1400–1406, New York, NY, USA, 2024. Association for Computing Machinery.

[Dang *et al.*, 2023] Yizhou Dang, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, Xiaoxiao Xu, Qinghui Sun, and Hong Liu. Uniform sequence better: time interval aware data augmentation for sequential recommendation. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023.

[Hidasi and Karatzoglou, 2018] Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 843–852, 2018.

[Hidasi, 2015] B Hidasi. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[Ho, 2022] Jonathan Ho. Classifier-free diffusion guidance. *ArXiv*, abs/2207.12598, 2022.

[Javaloy *et al.*, 2024] Adrián Javaloy, Pablo Sánchez-Martín, and Isabel Valera. Causal normalizing flows: from theory to practice. *Advances in Neural Information Processing Systems*, 36, 2024.

[Jiang *et al.*, 2023] Yiheng Jiang, Yongjian Yang, Yuanbo Xu, and En Wang. Spatial-temporal interval aware individual future trajectory prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Jiang *et al.*, 2024] Yiheng Jiang, Yuanbo Xu, Yongjian Yang, Funing Yang, Pengyang Wang, Chaozhuo Li, Fuzhen Zhuang, and Hui Xiong. Trimlp: A foundational mlp-like architecture for sequential recommendation. *ACM Trans. Inf. Syst.*, 42(6), October 2024.

[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.

[Kocaoglu *et al.*, 2017] Murat Kocaoglu, Christopher Snyder, Alexandros G. Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training, 2017.

[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*, WSDM '20, page 322–330, New York, NY, USA, 2020. Association for Computing Machinery.

[Nasr-Esfahany *et al.*, 2023] Arash Nasr-Esfahany, Mohammad Alizadeh, and Devavrat Shah. Counterfactual identifiability of bijective causal models. In *International Conference on Machine Learning*, pages 25733–25754. PMLR, 2023.

[Tillman and Spirtes, 2011] Robert Tillman and Peter Spirtes. Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 3–15, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[Wang *et al.*, 2022a] En Wang, Yiheng Jiang, Yuanbo Xu, Liang Wang, and Yongjian Yang. Spatial-temporal interval aware sequential poi recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 2086–2098. IEEE, 2022.

[Wang *et al.*, 2022b] En Wang, Yiheng Jiang, Yuanbo Xu, Liang Wang, and Yongjian Yang. Spatial-temporal interval aware sequential poi recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 2086–2098. IEEE, 2022.

[Xia *et al.*, 2022] Kevin Xia, Yushu Pan, and Elias Bareinboim. Neural causal models for counterfactual identification and estimation. *arXiv preprint arXiv:2210.00035*, 2022.

[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, 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.

[Xu *et al.*, 2024] Hangtong Xu, Yuanbo Xu, and Yongjian Yang. Separating and learning latent confounders to enhancing user preferences modeling. In *International Conference on Database Systems for Advanced Applications*, pages 67–82. Springer, 2024.

[Xu *et al.*, 2025] Hangtong Xu, Yuanbo Xu, Chaozhuo Li, and Fuzhen Zhuang. Causal structure representation learning of unobserved confounders in latent space for recommendation. *ACM Trans. Inf. Syst.*, April 2025. Just Accepted.

[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*. ACM, 2019.

[Zhao *et al.*, 2021] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *CIKM*, pages 4653–4664. ACM, 2021.

[Zheng *et al.*, 2018] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning, 2018.

A Appendix

A.1 Background

Sequential Recommendation

Sequential recommendation methods leverage users' historical interactions to model sequence dependencies, often using RNNs [Hidasi and Karatzoglou, 2018; Kang and McAuley, 2018] for sequential patterns and GNNs [Xu *et al.*, 2019] for structural relationships. Traditional approaches (Figure 1 (a)) focus on direct dependencies, leading to overly similar recommendations for users with identical histories. Time-aware models incorporate temporal information to enhance performance [Wang *et al.*, 2022a; Jiang *et al.*, 2023], but often overlook unobserved confounders. Moreover, directly merging time and item embeddings risks underutilizing representation spaces and missing key dependencies. There are also some works that improve the performance of sequence recommendation models by introducing information such as POI [Wang *et al.*, 2022b], . Existing methods often overlook unobserved confounders and their temporal dynamics, leading to suboptimal modeling of dependencies and underutilization of representation spaces.

Causal Structure Learning

We refer to causal representations constructed by causal graphs as causal representations. Over the past few decades, discovering causal graphs from purely observational data has garnered significant attention. [Zheng *et al.*, 2018] proposed NOTEARs with a fully differentiable DAG constraint for causal structure learning, [Tillman and Spirtes, 2011; Xu *et al.*, 2025]show the identifiability of learned causal structure from interventional data. The community has raised interest in combining causality and disentangled representation, and [Kocaoglu *et al.*, 2017] proposed a method called CausalGAN, which supports "do-operation" on images, but it requires the causal graph given as a prior. We draw on key ideas from causal structure learning to enhance the application of latent structure learning in recommendations and successfully deployed in the sequential recommendation scenario.

A.2 Proof of Theorem 1

Proof. A Directed Acyclic Graph (DAG) is a directed graph with no cycles.

A TMI map (Triangular Monotonic Increasing map) refers to a node ordering that satisfies two conditions: The ordering is monotonically increasing, i.e., nodes are arranged in non-decreasing order of their values; The ordering respects the triangular structure of dependencies, meaning that for any directed path $u \to v \to w$, we have $\sigma(u) < \sigma(v) < \sigma(w)$.

Let $G = (V, E)$ be a DAG, where V is the set of nodes and E is the set of directed edges. We aim to show that there exists a topological ordering σ of G such that σ is a TMI map.

A topological sort of a DAG is a linear ordering of the nodes such that for every directed edge $(u, v) \in E$, u appears before v in the ordering. This can be formalized as:

$$\forall (u, v) \in E, \quad \sigma(u) < \sigma(v),$$

where $\sigma(u)$ and $\sigma(v)$ represent the positions of nodes u and v in the ordering σ. A topological sort inherently respects the monotonicity condition. Since for every edge (u, v), we have $\sigma(u) < \sigma(v)$, the order is monotonically increasing with respect to the directed edges. Therefore, the first condition of a TMI map (monotonicity) is satisfied. Consider any directed path $u \to v \to w$ in G. In a topological sort, the ordering satisfies:

$$\sigma(u) < \sigma(v) < \sigma(w).$$

This is because the topological sort respects all dependencies, including transitive dependencies, ensuring that if $u \to v \to w$, then u appears before v, and v appears before w.

Thus, the second condition of a TMI map (triangular structure) is also satisfied. □

A.3 Proof of Classifier-free Guidance Paradigm

The Classifier-free Guidance Paradigm is defined as:

$$\hat{z}_i = (1 - \alpha) \cdot \mathsf{FFN}(s_i^I) + \alpha \cdot \mathsf{FFN}(s_i^I + c_I^t),$$

where s_i^I is the sequential information for item i, and c_I^t is the confounder at time t. α is a binary random factor defined as:

$$\alpha = \mathbb{I}(x > 0.5),$$

where $x \sim \mathcal{N}(0, 1)$ is a standard normal variable. Hence, α is randomly chosen between 0 and 1, controlling the model's reliance on sequential information ($\alpha = 0$) versus both sequential and confounding information ($\alpha = 1$).

Proof. Let L_0 be the loss when only sequential dependencies are considered ($\alpha = 0$) and L_1 be the loss when both sequential dependencies and confounders are considered ($\alpha = 1$). The expected loss function is:

$$\mathbb{E}[L(\theta, \alpha)] = (1 - \mathbb{E}[\alpha])L_0 + \mathbb{E}[\alpha]L_1.$$

Since $\mathbb{E}[\alpha] = 0.5$, we have:

$$\mathbb{E}[L(\theta, \alpha)] = 0.5L_0 + 0.5L_1.$$

Thus, the model's training objective is to minimize the combined expected loss over both sequential dependencies and confounders. This forces the model to learn a balance between the two, preventing it from overfitting to just one dependency type. □