Interest-Shift-Aware Logical Reasoning for Efficient Long-Sequence Recommendation

Fei Li[1*], **Qingyun Gao**[1*], **Enneng Yang**[1], **Jianzhe Zhao**[1 †], **Guibing Guo**[1†]
[1]Software College, Northeastern University, China
{neulifei,qingyungao,ennengyang}@stumail.neu.edu.cn, {zhaojz,guogb}@swc.neu.edu.cn

Abstract

Logical reasoning–based recommendation methods formulate logical expressions to characterize user–item interaction patterns, incorporating regularization constraints to ensure consistency with logical rules. However, these methods face two critical challenges: (1) As sequence length increases, they cannot effectively capture the dynamic transfer of user interests across subsequences (i.e., subsequence interest drift), thereby degenerating logical expressions to single-subsequence inference. (2) The time complexity of logical reasoning and rule learning scales quadratically with the sequence length, severely constraining computational efficiency in long-sequence recommendation. To address these challenges, we propose ELECTOR, an int**E**rest-shift-aware long-sequence **L**ogical reasoning for **E**ffi**C**ien**T** l**O**ng-sequence **R**ecommendation method. Specifically, we design a Subsequence Interest Learning Module (SIL) to model cross-subsequence interest drifts in long sequences. SIL employs a local attention mechanism to extract subsequence interests effectively and a global attention mechanism to capture the correlations among subsequence interests. Subsequently, we propose an Interest-aware Logical Reasoning (ILR) mechanism that performs logical reasoning using a limited set of subsequence and short-term interests, rather than reasoning over the entire sequence, significantly reducing time complexity. Additionally, ILR employs interest logical reasoning contrastive loss to ensure the model simultaneously considers multiple interests. Experiments on four real-world datasets demonstrate that our method significantly outperforms all baselines regarding computational efficiency and recommendation accuracy, confirming its effectiveness.

Code and Extended Version —
https://github.com/muzi1998/ELECTOR

Introduction

Recommendation systems require matching capabilities and cognitive reasoning abilities, as users' future behaviors depend not solely on matched historical interactions but also on the cognitive reasoning processes regarding their subsequent actions (Chen et al. 2021). Recommendation methods based on logical reasoning characterize user-item interaction patterns by constructing logical expressions. For example, if a user has purchased Tennis Racket and Tennis Ball, they will likely be interested in Tennis Apparel. This interaction pattern can be formalized as a logical expression: Tennis Racket $\wedge$ Tennis Ball $\rightarrow$ Tennis Apparel. Logical rules are incorporated into the model learning process to guide the construction of logical expressions consistent with predefined logical constraints (Shi et al. 2022; Zhang et al. 2022; Chen et al. 2022). Existing logical reasoning-based recommendation methods can be classified into two categories: those that derive embedding representations from logical rules and those that recast the recommendation problem as a logical reasoning task. The former leverages first-order logical rules to accurately derive embedding representations for users and items. For instance, SR-PLR (Yuan et al. 2023) incorporates probabilistic embeddings derived from first-order logic rules into sequence recommendation models, enhancing recommendation accuracy. In contrast, NLR (Shi et al. 2020) and NCR (Chen et al. 2021) reformulate recommendation as a logical reasoning task, rather than using logical rules solely as auxiliary constraints for embedding learning. These methods construct logical expressions based on user interaction histories and preferences, enabling more accurate modeling of user interest representations.

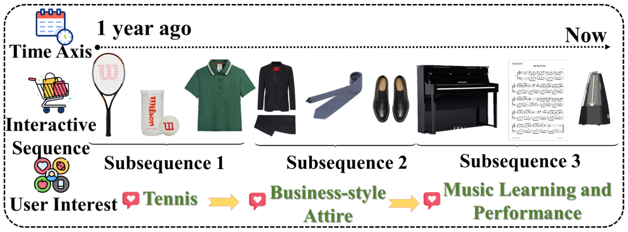


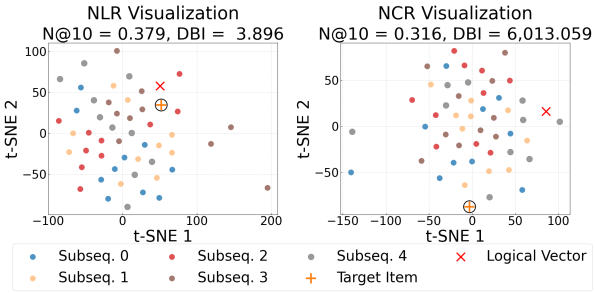
Figure 1: A toy example of subsequence interest drift.

However, logical reasoning-based recommendation methods typically treat user interests as static, overlooking their dynamic evolution driven by accumulating user–item interactions, primarily manifesting as drift across successive interaction subsequences. As shown in Fig. 1, Subsequence 1 (tennis racket, tennis ball, tennis apparel) reflects interest in tennis; Subsequence 2 (business suit, tie, leather shoes) indi-

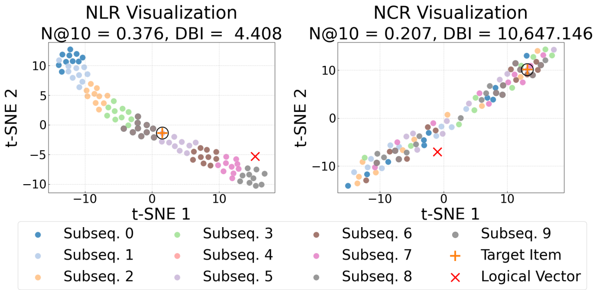
---

[*] These authors contributed equally.
[†] Guibing Guo and Jianzhe Zhao are corresponding authors.

cates business-style attire; and Subsequence 3 (piano, sheet music, metronome) demonstrates emerging interest in music learning and performance. Thus, user interests drift across successive subsequences. To quantify the impact of interest drift on logical reasoning-based recommendation methods, we conducted experiments on three public datasets to assess how the number of subsequences affects accuracy and computational efficiency for two representative methods: NLR and NCR. (i) As shown in Figs. 2a and 2b for the ML-1M dataset, when the sequence length increases from 50 to 100, the Davies-Bouldin Index (DBI) (Zhou et al. 2024) for NLR and NCR rises, while their Normalized Discounted Cumulative Gain@10 (N@10) declines. DBI quantifies a model's ability to separate item clusters within subsequences by calculating the average similarity between each cluster and its most similar counterpart. A lower DBI value indicates better subsequence separation. These results indicate that NLR and NCR struggle to distinguish interest differences across subsequences as the number of interactions grows. This limitation, subsequence interest drift, consequently degrades recommendation accuracy. (ii) As reported in Tab. 1, NLR and NCR training time increases significantly across three public datasets as the number of logical variables grows. This increase in training time is attributed to their quadratic time complexity for logical reasoning and rule learning.



(a) ML-1M dataset with a sequence length of 50.



(b) ML-1M dataset with a sequence length of 100.

Figure 2: Performance variation of NLR and NCR with increasing sequence length.

To address the above two issues, we propose the ELECTOR method, including the Subsequence Interest Learning (SIL) module and Interest-aware Logical Reasoning (ILR) mechanism. Specifically, (i) the SIL module designs a local attention mechanism with linear complexity to efficiently extract interest features from each subsequence while avoiding the quadratic time complexity of the self-attention mechanism. Furthermore, considering the correlation between users' future behaviors and historical interests - where subsequently purchased items are not only related to historical interactions but exhibit stronger associations with recent interactions - this module incorporates a global attention mechanism to model the correlations between subsequence interests and recent interactive items, thereby achieving more accurate characterization of subsequence interests. (ii) to reduce the high time complexity of logical reasoning and rule learning while enhancing recommendation accuracy, the ILR mechanism strategically utilizes a limited set of user interests (including subsequence interests and short-term interests learned from recent interactions) rather than all interactive items to construct logical expressions, which significantly reduces the number of logical variables in the expressions, lowering the computational complexity of logical reasoning and rule learning. Additionally, the ILR mechanism introduces an interest contrastive loss to ensure that the reasoning process comprehensively considers the user's multiple interests rather than relying solely on a single subsequence interest. This enhancement improves the completeness of logical expressions, thereby boosting the recommendation accuracy of our method.

The contributions can be summarized into the following:

- We identify key limitations of existing logical reasoning-based methods in long-sequence recommendations, including poor adaptability to interest drift and high computational cost due to quadratic complexity.
- We propose a SIL module that efficiently extracts subsequence interests and models the correlations between these interests and recent interactions, enabling accurate capture of dynamic interest shift.
- We propose an ILR mechanism that reduces reasoning complexity by reasoning over a limited set of user interests, while improving recommendation accuracy by considering multiple interests simultaneously.
- Extensive experiments demonstrate that ELECTOR achieves superior performance in both efficiency and accuracy on long-sequence recommendation tasks.

Assumption Validation

In this section, we investigate the effect of interaction sequence length on recommendation accuracy, subsequence recognition performance, and computational efficiency of the logical reasoning-based methods NLR and NCR across three public datasets. For the ML-1M and MoTV datasets, we set the subsequence length to 10, reflecting their relatively short average sequence lengths. In contrast, for the Amazon(1000) dataset, which features substantially longer user interaction sequences, we employ a subsequence length of 100. Recommendation accuracy is primarily evaluated using Normalized Discounted Cumulative Gain at rank 10 (N@10), while the Davies-Bouldin Index (DBI) is used to assess the quality of internal clustering.

The experimental results are shown in Tab. 1. We summarize the key findings: 1) As sequence length increases across all three datasets, DBI for NLR and NCR rises by 23.44% and 58.06% on average, respectively, indicating increased overlap among item embeddings across subsequences and hindering interest drift detection. Figs. 2a and 2b illustrate that failure to capture interest drift causes logical expressions (logic vectors) to be dominated by a single subsequence's embeddings. This degradation reduces N@10 by an average of 13.18% for NLR and 10.91% for NCR across the three datasets. 2) Computational costs rise sharply with sequence length: the number of logical variables required by NLR and NCR increases substantially, leading to significantly longer training times. For example, on the Amazon(1000) dataset, increasing the sequence length from 500 to 1000 yields training-time increases of 138.41% for NLR and 16.03% for NCR. These increases are primarily due to the quadratic time complexity ($\mathcal{O}({n_L}^2 d^2)$) of both logical reasoning and rule learning methods, which impairs computational efficiency in long-sequence settings.

Method		NLR			NCR		
Datasets	#Seq	N@10 ↑	DBI ↓	Time ↓	N@10 ↑	DBI ↓	Time ↓
ML-1M	50	**0.379**	**3.896**	**87s**	0.316	6,013.059	**48s**
ML-1M	100	0.376	4.408	148s	0.207	10,647.146	55s
MoTV	50	0.263	**4.226**	119s	0.341	4,653.998	**74s**
MoTV	100	0.221	4.267	204s	**0.359**	7,534.310	84s
Amazon(1000)	500	0.136	**5.891**	1,505s	**0.227**	29,447.442	**705s**
Amazon(1000)	1000	0.105	9.203	3,588s	0.219	39,821.155	818s

Table 1: Experimental results of two methods with increasing sequence length. #Seq denotes sequence length. ↑ indicates that higher values correspond to better performance; ↓ indicates that lower values correspond to better performance. Best results for each dataset are highlighted.

Methodology

We first introduce the preliminaries and formally define the logical reasoning-based recommendation problem. Then, we present the subsequence interest learning module and the interest-aware logical reasoning mechanism, detailing the training process and analyzing the computational complexity. Figure 3 illustrates the overall framework of ELECTOR.

Problem Definition

Preliminaries. Let $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$ and $\mathcal{I} = \{i_1, i_2, \dots, i_n\}$ denote the sets of users and items, respectively. Each user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$ is associated with an embedding vector $\boldsymbol{u}, \boldsymbol{i} \in \mathbb{R}^d$. The interaction history of user u is represented as a sequence $I_u = \{i_{n_1}, i_{n_2}, \dots, i_{n_L}\}$, where items are ordered chronologically based on their associated timestamps. We consider four logical operations: negation $\neg$, conjunction $\wedge$, disjunction $\vee$, and implication $\rightarrow$. These logical operations can cover most recommendation scenarios and satisfy some logical rules of propositional logic. A clause is a formula comprising literals (v), logical operations, and truth values (true or false, denoted T or F); for example, $v_i \wedge v_j$ or $v_i \vee v_j$. De Morgan's laws state: $\neg(v_i \wedge v_j) \iff \neg v_i \vee \neg v_j$ and $\neg(v_i \vee v_j) \iff \neg v_i \wedge \neg v_j$. Propositional formulas are constructed by combining clauses with logical operations; for instance, $\neg(v_i \wedge v_j) \vee v_g$ (a Horn clause) or $v_i \wedge v_j \rightarrow v_g$, noting that $v_i \wedge v_j \rightarrow v_g$ is equivalent to $\neg(v_i \wedge v_j) \vee v_g$.

Logical reasoning-based recommendation methods aim to predict the next item by formulating logical expressions over historical interactions. Specifically, for a candidate item $i^* \in \mathcal{I}$, we define a Horn clause as $E_{i^*} = i_{n_1} \wedge i_{n_2} \wedge \cdots \wedge i_{n_L} \rightarrow i^*$. The recommendation task can then be formalized as estimating the likelihood that the Horn clause E_{i^*} evaluates to true. Accordingly, the next item i_t is selected by maximizing the probability: $i_t = \arg\max_{i^* \in \mathcal{I}} P(T \mid E_{i^*})$.

Subsequence Interest Learning

The assumption validation section indicates that subsequence interest drift degrades recommendation accuracy. Moreover, KuaiFormer (Liu et al. 2024) indicates that users form stronger impressions of a few recent interactions, whereas their recall of numerous historical interactions remains relatively weak. Accordingly, we propose the Subsequence Interest Learning module to precisely learn subsequence interest embeddings and concurrently model correlations between subsequence interests and recent interactions. Given a user u with interaction sequence $I_u = \{i_{n_1}, \dots, i_{n_L}\}$, we define the most recent k interactions as the recent sequence $RS = \{i_{n_{L-k+1}}, \dots, i_{n_L}\}$ and the preceding $L-k$ interactions as the historical sequence $HS = \{i_{n_1}, \dots, i_{n_{L-k}}\}$. We then evenly partition HS into J subsequences $\{HS_j\}_{j=1}^{J}$, facilitating the learning of distinct interest representations for each subsequence.

Subsequence Interest Extraction. We propose a localized attention mechanism that captures the causal relationships among items within each subsequence, permitting later items to attend to earlier ones but not vice versa, while mitigating interest drift across subsequences. Specifically, we first apply a multilayer perceptron to each subsequence $\boldsymbol{HS}_j$ to produce the query matrix $\boldsymbol{Q}^j$, the key matrix $\boldsymbol{K}^j$, and the value matrix $\boldsymbol{V}^j$, where $t = \frac{n_L - k}{J}$ is the subsequence length. The context-mapping matrix is define as

$$\boldsymbol{C}^j = \begin{cases} \varphi\left(\boldsymbol{Q}^j_{pq}, \boldsymbol{K}^j_{pq}\right), & \text{if } (j-1)t \le p \le q \le jt \\ 0, & \text{else} \end{cases}. \quad (1)$$

The function $\varphi\left(\boldsymbol{Q}^j_{pq}, \boldsymbol{K}^j_{pq}\right)$ quantifies the causal relationship between the p-th and q-th items within subsequence HS_j, effectively isolating their interaction from external subsequence interference. The updated subsequence representation is formally defined as $\widehat{\boldsymbol{HS}}_j = \rho\left(\boldsymbol{C}^j, \boldsymbol{V}^j\right)$, where $\boldsymbol{C}^j$ aggregates relevant items through attention-based correlation. When computed using standard self-attention, this operation incurs a time complexity of $O\left(t^2\right)$. To alleviate this quadratic computational overhead, we implement two distinct optimizations: Low-Rank Matrix (LRM) Approximation (Wang et al. 2020) and Kernel Feature Maps (KFM) (Katharopoulos et al. 2020). For LRM, we reformulate the attention computation as $\rho\left(\boldsymbol{C}^j, \boldsymbol{V}^j\right) = \underbrace{\text{softmax}\left(\frac{\boldsymbol{Q}^j(\boldsymbol{E}\boldsymbol{K}^j)^T}{\sqrt{d}}\right)}_{t\times\kappa}\underbrace{\boldsymbol{F}\boldsymbol{V}^j}_{\kappa\times d}$, where low-rank projection

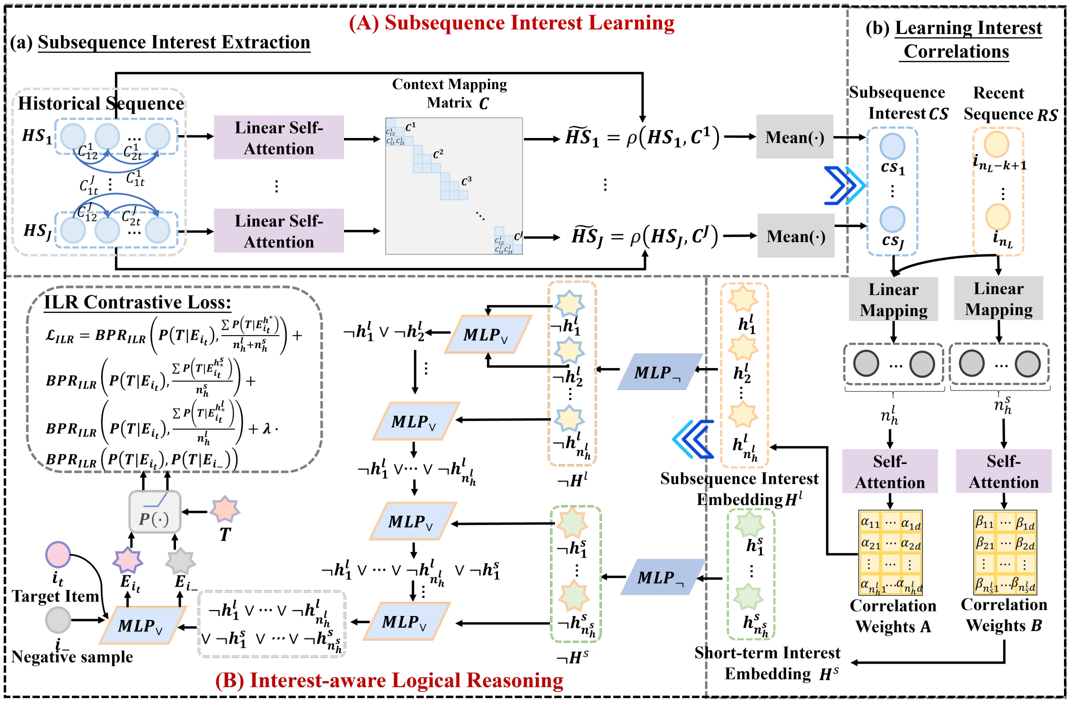


Figure 3: The overall framework diagram of ELECTOR.

matrices $\boldsymbol{E}, \boldsymbol{F} \in \mathbb{R}^{\kappa\times t}$ ($\kappa \ll t$) map $\boldsymbol{K}^j$ and $\boldsymbol{V}^j$ from $\mathbb{R}^{t\times d}$ to $\mathbb{R}^{\kappa\times d}$. LRM reduces the complexity from $O\left(t^2\right)$ to $O\left(t\right)$. For KFM, we compute attention through kernelized decomposition: $\rho\left(\boldsymbol{C}^j, \boldsymbol{V}^j\right)_{p:} = \frac{\phi(\boldsymbol{Q}^j_{p:})\boldsymbol{S}^j}{\phi(\boldsymbol{Q}^j_{p:})\boldsymbol{Z}^j}$, where $\boldsymbol{S}^j = \sum_{q=1}^{t}\phi(\boldsymbol{K}^j_{q:})^T\boldsymbol{V}^j_{q:}$ and $\boldsymbol{Z}^j = \sum_{q=1}^{t}\phi(\boldsymbol{K}^j_{q:})^T$ are precomputed once per subsequence. This kernelized decomposition achieves linear complexity in both time and memory.

We derive the subsequence interest embedding $\boldsymbol{cs}_j = \mathrm{Mean}(\widetilde{\boldsymbol{HS}}_j)$ through average pooling on over item embeddings in HS_j. The subsequence interests is defined as:

$$\boldsymbol{CS} = \left[\boldsymbol{cs}_1; \cdots; \boldsymbol{cs}_j; \cdots; \boldsymbol{cs}_J\right]. \tag{2}$$

Learning Interest Correlations. We utilize a self-attention mechanism (Lin et al. 2017) to model correlations between the subsequence interests and recently interacted items, as well as correlations among the recently interacted items. Specifically, the first correlation enables the model to distinguish between interests relevant to recent interactions and those that are not, thereby enhancing subsequence interest embeddings. It can be computed as $\boldsymbol{A} = \mathrm{softmax}(\boldsymbol{\mathcal{W}}_\alpha \tanh([\boldsymbol{CS} \parallel \boldsymbol{RS}]\boldsymbol{\mathcal{W}})^{\mathcal{T}})$, where $\boldsymbol{RS} \in \mathbb{R}^{k\times d}$ is the recent item embeddings in RS, $\boldsymbol{\mathcal{W}}_\alpha \in \mathbb{R}^{n_h^l\times d}$ is the correlation weights of n_h^l subsequence interests, and $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{d\times d}$ is a learned projection matrix. We then update subsequence interest embeddings via

$$\boldsymbol{H}^l = \mathrm{sigmoid}(\boldsymbol{A}[\boldsymbol{CS} \parallel \boldsymbol{RS}]). \tag{3}$$

The second correlation captures users' short-term interests directly. The correlation matrix of short-term interests is defined as $\boldsymbol{B} = \mathrm{softmax}(\boldsymbol{\mathcal{W}}_\beta \tanh((\boldsymbol{RS} + \boldsymbol{P}^s)\boldsymbol{\mathcal{W}})^{\mathcal{T}})$, where $\boldsymbol{\mathcal{W}}_\beta \in \mathbb{R}^{n_h^s\times d}$ is the correlation weights of n_h^s short-term interests, and $\boldsymbol{P}^s \in \mathbb{R}^{k\times d}$ is a learnable position encoding matrix. The short-term interest embeddings are obtained by

$$\boldsymbol{H}^s = \mathrm{sigmoid}(\boldsymbol{B}(\boldsymbol{RS} + \boldsymbol{P}^s)). \tag{4}$$

Interest-Aware Logical Reasoning and Training

NLR and NCR construct the logical expression E_{i^*} based on the sequence I_u to predict the next item, formalized as $i_t = \arg\max_{i^*\in\mathcal{I}} P(T|\ E_{i^*})$. However, for long-sequence recommendations, the clauses generated by NLR and NCR contain many logical variables, which significantly increases the time complexity of both logical reasoning and rule learning. To address this, our method constructs Horn clauses based on a limited set of interest embeddings, reducing the number of logical variables from n_L to $n_h^s + n_h^l$ (where $n_h^s + n_h^l \ll n_L$), thereby substantially decreasing the time complexity of logical reasoning and rule learning.

Specifically, the logical expression E_{i^*} comprises subsequence interests H^l, short-term interests H^s, and candidate items $i^* \in \mathcal{I}$. i.e., $E_{i^*} = h_1^l \wedge h_2^l \wedge \cdots \wedge h_{n_h^l}^l \wedge h_1^s \wedge h_2^s \wedge \cdots \wedge h_{n_h^s}^s \rightarrow i^*$. By De Morgan's laws, $E_{i^*} \iff \neg(h_1^l \wedge h_2^l \wedge \cdots \wedge h_{n_h^l}^l \wedge h_1^s \wedge h_2^s \wedge \cdots \wedge h_{n_h^s}^s) \vee i^*$.

We define the logical operations $\neg$, $\wedge$, and $\vee$ as logical operator networks: $MLP_\neg(\cdot)$, $MLP_\wedge(\cdot,\cdot)$, and $MLP_\vee(\cdot,\cdot)$ (Chen et al. 2021). Each logical operation is instantiated as a lightweight neural module, providing strong composability and scalability—an essential property for modeling complex user behaviors in long-sequence recommendations. These logical operator networks are capable of dynamically adapting to user interests. For example, consider $E_{i^*} = \neg(h_1^s \wedge h_2^s \wedge h_1^l) \vee i^*$. To compute its embedding, we employ the logical operator networks $MLP_{\neg}(\cdot)$ and $MLP_{\vee}(\cdot,\cdot)$. Specifically, we first compute the intermediate representations: $\boldsymbol{v_1} = MLP_{\vee}(MLP_{\neg}(\boldsymbol{h}_1^s), MLP_{\neg}(\boldsymbol{h}_2^s))$, which corresponds to the subexpression $\neg h_1^s \vee \neg h_2^s$, and then $\boldsymbol{v_2} = MLP_{\vee}(\boldsymbol{v_1}, MLP_{\neg}(\boldsymbol{h}_1^l))$ representing $(\neg h_1^s \vee \neg h_2^s) \vee \neg h_1^l$. Finally, the embedding of the entire expression is computed as: $\boldsymbol{E}_{i^*} = MLP_{\vee}(\boldsymbol{v_2}, \boldsymbol{i}^*)$. We use this expression embedding to select the next item i_t for recommendation by computing its similarity with the target embedding $\boldsymbol{T}$: $P(T|\ E_{i^*}) = \frac{\boldsymbol{E_{i^*}} \cdot \boldsymbol{T}}{\|\boldsymbol{E_{i^*}}\| \|\boldsymbol{T}\|}$, where $\boldsymbol{T}$ is the embedding of T, which is randomly initialized and kept fixed during training.

Interest Logical reasoning (ILR) Contrastive loss. This loss ensures that multiple user interests are jointly considered during logical reasoning. The ILR contrastive loss enforces that reasoning based on multiple interests yields scores closer to the true target than (i) reasoning based on a single interest and (ii) reasoning on negative items, as formalized by

$$\begin{aligned}\mathcal{L}_{ILR} =& \mathrm{BPR}_{\mathrm{ILR}}\left(P(T|E_{i_t}), \frac{\sum_{h_* \in H} P(T|E_{i_t}^{h_*})}{n_h^s + n_h^l}\right)\\ &+ \mathrm{BPR}_{\mathrm{ILR}}\left(P(T|E_{i_t}), \frac{\sum_{h_*^s \in H^s} P(T|E_{i_t}^{h_*^s})}{n_h^s}\right)\\ &+ \mathrm{BPR}_{\mathrm{ILR}}\left(P(T|E_{i_t}), \frac{\sum_{h_*^l \in H^l} P(T|E_{i_t}^{h_*^l})}{n_h^l}\right)\\ &+ \lambda \cdot \mathrm{BPR}_{\mathrm{ILR}}\left(P(T|E_{i_t}^{h_*^l}), P(T|E_{i\_})\right) \qquad (5)\end{aligned}$$

where $\mathrm{BPR}_{\mathrm{ILR}}(x_1, x_2) = -\log(\mathrm{sigmoid}(10(x_1 - x_2))$, and λ is a are preset hyperparameters. Each term in Eq. (5) is detailed in the extended version. By enforcing the constraints introduced in Eq.(5), the model is guided to jointly consider subsequence-level and short-term interests during the reasoning process. To further enhance the logical consistency of the model, we incorporate logical regularizers during training, which ensure that the three logical operator networks conform to predefined logical rules. These rules and their corresponding regularization terms, denoted as $\mathcal{L}_{reg} = \sum_k r_k$ Finally, we combine the logical regularization loss $\mathcal{L}$reg with the ILR contrastive loss $\mathcal{L}_{ILR}$, and include an L_2 regularization term to prevent overfitting. The overall training loss is defined as:

$$\mathcal{L}_{train} = \lambda_1 \mathcal{L}_{ILR} + \lambda_2 \mathcal{L}_{reg} + \lambda_{\Theta} \|\Theta\|_2^2, \qquad (6)$$

where Θ denotes the set of model parameters, and λ_1, λ_2, and λ_{Θ} are preset hyperparameters.

Complexity Analysis

Our method comprises four parts (Subsequence Interest Extraction, Learning Interest Correlations, ILR contrastive Loss, and Logical Rule Learning) and has an overall time complexity of $\mathcal{O}(n_L + n_h J d + n_h d^2 + 16 n_h^2 d^2)$, where $n_h, J \ll n_L$. In contrast, the time complexity of the logical rule learning component in our backbone method, NCR, is given by $\mathcal{O}(16 {n_L}^2 d^2)$. We can further simplify this time complexity expression. Assuming $n_L, J < d$, we obtain: $\mathcal{O}\big(n_L + n_h J d + n_h d^2 + 16 n_h^2 d^2\big) \approx \mathcal{O}\Big(16 n_h^2 d^2\Big)$. Therefore, our method's time complexity is substantially lower than that of NCR. Consequently, our method is more efficient than NCR for long-sequence recommendations.

Experiments

Experimental Settings

Description of Datasets and Baselines. We conduct extensive experiments on four publicly available datasets—ML-1M [1] (Harper and Konstan 2015), MoTV, Amazon-500, and Amazon-1000 [2] (Ni, Li, and McAuley 2019)—to evaluate the performance of fifteen recommendation methods. These baselines span four categories: traditional sequential methods (STAMP (Liu et al. 2018), GRU4Rec (Hidasi et al. 2016), NARM (Li et al. 2017), SASRec (Kang and McAuley 2018)), efficient sequential methods (LinRec (Liu et al. 2023), Mamba4Rec (Liu et al. 2024b), SIGMA (Liu et al. 2025), GLINT-RU (Zhang et al. 2025), RecBLR (Liu et al. 2024a)), multi-interest methods (MIND (Li et al. 2019), ComiRec (Cen et al. 2020)), and logical reasoning–based methods (NLR (Shi et al. 2020), NCR (Chen et al. 2021)).

Implementation Details. We use the leave-one-out strategy to divide the interaction data into training, validation, and test sets. We match a negative sample (i.e., an item the user has not interacted with) to conduct pairwise ranking training for each interactive item. We use the metrics N@5 and Hit Ratio@5 (H@5) to evaluate the recommendation performance of fifteen methods. The range for sequence length is set to $\{50, 100, 500, 1000\}$, and the range for the number of interests is $\{5, 7, 9\}$. We set the hyperparameters $\lambda = 10$, $\lambda_1 = 1$, $\lambda_2 = 10$, and $\lambda_{\Theta} = 0.0001$.

Performance Comparison

Recommendation Accuracy. We compared our methods (ELECTOR-LRM and ELECTOR-KFM)—which differ in using LRM versus KFM for subsequence interest extraction—against 13 baseline approaches across four datasets. The experimental results are summarized in Tab. 2. We draw the following key conclusions: (1) Our methods significantly outperform all baselines and are stable for every dataset. Notably, on the Amazon(1000) dataset, our method improves N@5 and H@5 by 288.02% and 238.80%, respectively, over the strongest baseline. This marked improvement demonstrates that, compared to the baselines, our methods can more clearly distinguish between different subsequence items in long sequences—thereby mitigating the negative impact of interest drift on user-interest modeling—and that considering multiple inter-interest correlations further enhances the accuracy of interest embedding learning. (2) The ILR comparative loss ensures that multiple user interests are considered simultaneously during logical reasoning, preventing the logical expressions from being dominated by a single subsequence and thus deviating from the user's true interests. As a result, the learned logical expressions can

---

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

[2] https://nijianmo.github.io/amazon/index.html

Dataset	ML1M				MoTV				Amazon(500)		Amazon(1000)	
Metric	N@5 ↑		H@5 ↑		N@5 ↑		H@5 ↑		N@5 ↑	H@5 ↑	N@5 ↑	H@5 ↑
#Seq	50	100	50	100	50	100	50	100	500	500	1000	1000
STAMP	0.357	0.354	0.501	0.501	0.221	0.218	0.314	0.294	0.250	0.339	0.052	0.095
GRU4Rec	0.395	0.395	0.550	0.555	0.263	0.266	0.319	0.358	0.323	0.436	0.173	<u>0.250</u>
NARM	<u>0.413</u>	<u>0.408</u>	<u>0.575</u>	<u>0.578</u>	0.258	0.256	0.344	0.359	0.351	0.473	0.115	0.170
SASRec	0.396	0.403	0.558	0.557	0.238	0.255	0.323	0.351	OOM	OOM	OOM	OOM
LinRec	0.232	0.201	0.353	0.315	0.198	0.166	0.268	0.216	0.230	0.305	0.097	0.125
Mamba4Rec	0.353	0.317	0.497	0.435	0.235	0.256	0.314	0.392	0.359	0.564	0.103	0.200
SIGMA	0.354	0.347	0.498	<u>0.578</u>	0.245	0.281	0.339	<u>0.440</u>	<u>0.377</u>	<u>0.594</u>	0.120	0.225
GLINT-RU	0.320	0.333	0.447	0.567	0.225	0.253	0.314	0.390	0.333	0.549	0.117	0.225
RecBLR	0.355	0.314	0.495	0.557	0.205	0.217	0.262	0.342	0.322	0.461	0.121	0.205
MIND	0.031	0.032	0.051	0.058	0.030	0.022	0.046	0.037	0.031	0.053	0.031	0.050
ComiRec-SA	0.371	0.143	0.508	0.224	0.181	0.181	0.253	0.252	0.171	0.255	0.031	0.070
NLR	0.321	0.188	0.451	0.282	0.263	0.186	0.322	0.358	0.251	0.314	0.095	0.140
NCR	0.189	0.129	0.198	0.153	<u>0.333</u>	<u>0.339</u>	<u>0.444</u>	0.348	0.332	0.371	<u>0.217</u>	0.233
ELECTOR-LRM	**0.788**†	**0.882**†	**0.799**†	**0.883**†	0.729†	0.837†	0.733†	0.839†	0.854†	0.854†	0.835†	0.837†
ELECTOR-KFM	0.729†	0.824†	0.732†	0.829†	**0.813**†	**0.846**†	**0.825**†	**0.852**†	**0.877**†	**0.879**†	**0.842**†	**0.847**†
Imp. (%)	90.80%	116.18%	38.96%	52.77%	144.14%	149.56%	85.81%	93.64%	132.63%	47.98%	288.02%	238.80%

Table 2: Experimental results of 15 methods on two evaluation metrics N@5 and H@5. The best overall results are highlighted in bold, while the best results among baselines are underlined. The symbol † indicates that our method significantly outperforms the optimal baseline at the 0.05 level based on the paired t-test. OOM stands for Out of Memory Error.

more accurately represent and entirely represent the user's interests, further boosting recommendation accuracy.

Computational Efficiency. As shown in Tab. 3, we conduct experiments on four datasets to compare the recommendation accuracy and training time of four methods for each epoch. From Tab. 3, we can draw the following conclusions: (1) Compared to NLR and NCR, which utilize a large number of interactive items for logical reasoning, Ours1 (ELECTOR-LRM) and Ours2 (ELECTOR-KFM) employ fewer subsequence compressed representations, significantly lowering the time complexity of logical reasoning and rule application, particularly demonstrating marked efficiency improvements in extremely long interaction sequences. For instance, for the Amazon(1000) dataset, the training times for Ours1 and Ours2 decreased by 70.54% and 81.30%, respectively, compared to NCR. (2) Unlike Ours1, which employs low-rank matrix mappings to mitigate the computational complexity of the attention matrix, Ours2 utilizes kernel feature maps to streamline attention computation by obviating redundant key–value operations and eliminating additional low-rank multiplications. Consequently, Ours2 is more efficient compared to Ours1. Furthermore, the limited representational capacity of low-rank matrices in Ours1 constrains its accuracy, resulting in inferior performance relative to Ours2.

Ablation Experiments

We conduct ablation experiments to investigate the effectiveness of each component in our method, including Subsequence Interest Extraction (SIE), Learning Interest Correlations for Subsequences (LIC-S), Learning Interest Correlations for Recent Items (LIC-RI), and Interest Logical Reasoning (ILR) contrastive loss. Tab. 4 uses "✔" means "used" and "✗" for "not used." We can draw four conclusions: We can draw four conclusions: (1) Compared to NCR, which performs logical reasoning on interacted items, our base model improves recommendation accuracy by extracting subsequence interests via SIE, mitigating the impact of interest drift. (2) Variant 2, by modeling LIC-S and correlating subsequence interests with recent interactions, enhances interest embeddings and outperforms the base model. (3) Variant 4, focused on short-term interests (LIC-RI) and logical reasoning based on recent interactions, surpasses item-based NCR, validating the effectiveness of interest-aware logical reasoning. (4) The ILR contrastive loss promotes multi-interest logical expressions, preventing dominance by a single subsequence, leading to improved performance in Variant 2 over Variant 1, and Variant 4 over Variant 3.

Dataset	ML-1M		MoTV		Amazon(1000)	
Metric	N@10 ↑	H@10 ↑	N@10 ↑	H@10 ↑	N@10 ↑	H@10 ↑
	Time ↓	Time ↓	Time ↓	Time ↓	Time ↓	Time ↓
#Seq	100	100	100	100	1000	1000
NLR	0.376 148s	0.437 148s	0.221 204s	0.372 204s	0.105 3588s	0.170 3588s
NCR	0.207 55s	0.248 55s	0.359 84s	0.390 84s	0.219 818s	0.238 818s
Ours1	**0.883**† 57s	**0.886**† 57s	<u>0.838</u> 80s	<u>0.842</u> 80s	<u>0.828</u> <u>241s</u>	<u>0.835</u> <u>241s</u>
Ours2	<u>0.824</u> **38s**	<u>0.829</u> **38s**	**0.846**† **58s**	**0.852**† **58s**	**0.842**† **153s**	**0.847**† **153s**

Table 3: Computational efficiency comparison of methods.

Sensitivity Analysis

Impact of Subsequence Length. From Fig. 4, we observe that as the subsequence length t increases, the time complexity of learning interest correlation decreases from $O(n_h J d)$, significantly reducing training time for both methods. Additionally, longer subsequences contain richer item-correlation information, which facilitates more accurate learning of subsequence interest embeddings and, in turn, improves recom-

Method	ST	Sub-component				ML-1M(100)		MoTV(100)		Amazon(500)		Amazon(1000)	
		SIE	LIC-S	LIC-RI	ILR	N@10 ↑	H@10 ↑	N@10 ↑	H@10 ↑	N@10 ↑	H@10 ↑	N@10 ↑	H@10 ↑
NCR (baseline)	I_u	✗	✗	✗	✗	0.207	0.248	0.359	0.390	0.354	0.427	0.219	0.238
Our Base	I_u	✔	✗	✗	✗	0.812	0.821	0.538	0.54	0.331	0.332	0.408	0.412
Our Variant 1	I_u	✔	✔	✗	✗	0.693	0.693	0.400	0.402	0.122	0.181	0.585	0.620
Our Variant 2	I_u	✔	✔	✗	✔	0.833	0.841	0.816	0.819	0.753	0.754	0.821	0.827
Our Variant 3	RS	✗	✗	✔	✗	0.204	0.433	0.157	0.245	0.235	0.237	0.325	0.325
Our Variant 4	RS	✗	✗	✔	✔	0.625	0.636	0.723	0.728	0.454	0.457	0.811	0.817
Our Method	I_u	✔	✔	✔	✔	**0.883**†	**0.886**†	**0.838**†	**0.842**†	**0.854**†	**0.855**†	**0.828**†	**0.835**†

Table 4: Performance Comparison of NCR, our method, and five variants. ST denotes the training sequence type.

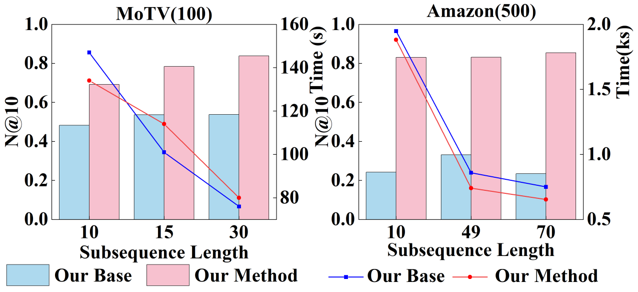


Figure 4: The impact of subsequence length.

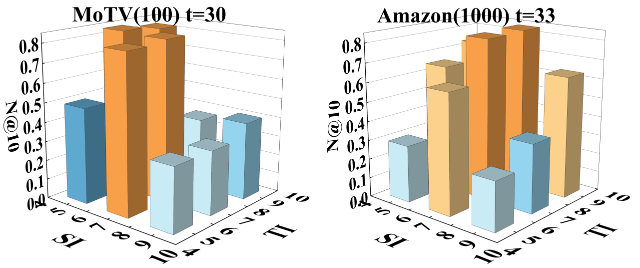


Figure 5: Effect of the number of subsequence interests (SI) and short-term interests (TI) on our method's accuracy.

mendation accuracy.

Impact of the number of subsequence interests and short-term interests. Fig. 5 shows that the recommendation accuracy of our method is relatively sensitive to the number of subsequence interests and short-term interests. Our experiments provide empirical settings for these two types of interests. When both the number of subsequence interests and the number of short-term interests are set to 7, our method achieves its highest accuracy on datasets with a sequence length of 100. When the number of subsequence interests is set to 7 and the number of short-term interests is set to 9, our method achieves its highest accuracy on datasets with sequence lengths greater than 100.

Related Work

Logical Reasoning-based Recommendation. Existing methods utilize first-order logical rules to learn the embedding representations of items and users accurately. for example, Tang et al. (Tang et al. 2023) proposed a recommendation method that integrates users' logical requirements, collaboratively exploring user preferences and logical constraints for personalized recommendations. To further enhance the accuracy of logical reasoning, KG-LRR (Wang et al. 2025) leverages an item knowledge graph to reinforce user and item representations. Spillo et al. (Spillo et al. 2022) introduced a knowledge-aware recommendation method that utilizes neuro-symbolic graph embeddings (Guo et al. 2016) and first-order logical rules (Smullyan 1995), learning user and item embeddings from knowledge graph triples and associated logical rules.

Sequential Recommendation. Sequential recommendation is a research hotspot in the field of recommendation (Dang et al. 2024; Li et al. 2024). However, these methods ignore the multiple interests of users, resulting in incomplete user representations. MIND designs a multi-interest extractor based on a dynamic routing mechanism (Sabour, Frosst, and Hinton 2017) to model and extract multiple interests from interaction sequences (Li et al. 2019). ComiRec employs a self-attention method (Lin et al. 2017) to capture users' multiple interests from interaction sequences while utilizing a controllable factor to balance the accuracy and diversity of recommendations (Cen et al. 2020). MIP generates users' multiple interests by clustering items in the interaction sequences (Shi et al. 2023). These methods are susceptible to being dominated by recent items, causing them to deviate from users' true interests.

To address the efficiency bottleneck caused by the quadratic complexity of the attention mechanism (Kang and McAuley 2018), LinRec (Liu et al. 2023)—drawing on linear-attention transforms the traditional dot-product attention matrix into a feature-mapping form, thus achieving linear time complexity. Inspired by the state-space model Mamba (Dao and Gu 2024), Mamba4Rec (Liu et al. 2024b) and SIGMA (Liu et al. 2025) replace conventional attention with efficient Mamba blocks to realize linear-time sequential modeling. RecBLR (Liu et al. 2024a) substitutes costly attention modules with Behavior-Dependent Linear Recurrent Units and a hardware-aware parallel acceleration algorithm. GLINT-RU (Zhang et al. 2025) employs a dense selective Gated Recurrent Unit (Chung et al. 2014) module instead of attention, capturing long- and short-term dependencies at linear computational cost. However, they ignore the adverse impact of interest drift on learning user representations.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant No. (62576083, 62032013, 62102074), and Natural Science Foundation of Liaoning Province (No. 2024-MSBA-49). We greatly appreciate the anonymous reviewers for their valuable comments and suggestions, which are important for improving and polishing our paper.

References

Cen, Y.; Zhang, J.; Zou, X.; Zhou, C.; Yang, H.; and Tang, J. 2020. Controllable multi-interest framework for recommendation. In *SIGACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2942–2951.

Chen, H.; Li, Y.; Shi, S.; Liu, S.; Zhu, H.; and Zhang, Y. 2022. Graph collaborative reasoning. In *ACM International Conference on Web Search and Data Mining*, 75–84.

Chen, H.; Shi, S.; Li, Y.; and Zhang, Y. 2021. Neural collaborative reasoning. In *International World Wide Web Conferences*, 1516–1527.

Chung, J.; Gülçehre, Ç.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Dang, Y.; Liu, Y.; Yang, E.; Guo, G.; Jiang, L.; Wang, X.; and Zhao, J. 2024. Repeated Padding as Data Augmentation for Sequential Recommendation. *arXiv preprint arXiv:2403.06372*.

Dao, T.; and Gu, A. 2024. Transformers are SSMs: generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*.

Guo, S.; Wang, Q.; Wang, L.; Wang, B.; and Guo, L. 2016. Jointly embedding knowledge graphs and logical rules. In *The annual Conference on Empirical Methods in Natural Language Processing*, 192–202.

Harper, F. M.; and Konstan, J. A. 2015. The MovieLens datasets: history and context. 5(4).

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations*.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining*, 197–206. IEEE.

Katharopoulos, A.; Vyas, A.; Pappas, N.; and Fleuret, F. 2020. Transformers are RNNs: fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*.

Li, C.; Liu, Z.; Wu, M.; Xu, Y.; Zhao, H.; Huang, P.; Kang, G.; Chen, Q.; Li, W.; and Lee, D. L. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *ACM International Conference on Information and Knowledge Management*, 2615–2623.

Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural attentive session-based recommendation. In *ACM International Conference on Information and Knowledge Management*, 1419–1428.

Li, Z.; Yang, C.; Chen, Y.; Wang, X.; Chen, H.; Xu, G.; Yao, L.; and Sheng, M. 2024. Graph and sequential neural networks in session-based recommendation: A Survey. *ACM Computing Surveys*, 57(2).

Lin, Z.; Feng, M.; dos Santos, C. N.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. In *International Conference on Learning Representations*.

Liu, C.; Cao, J.; Huang, R.; Zheng, K.; Luo, Q.; Gai, K.; and Zhou, G. 2024. KuaiFormer: Transformer-Based Retrieval at Kuaishou. *arXiv e-prints*, arXiv:2411.10057.

Liu, C.; Lin, J.; Liu, H.; Wang, J.; and Caverlee, J. 2024a. Behavior-dependent linear recurrent units for efficient sequential recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 1430–1440.

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

Liu, L.; Cai, L.; Zhang, C.; Zhao, X.; Gao, J.; Wang, W.; Lv, Y.; Fan, W.; Wang, Y.; He, M.; Liu, Z.; and Li, Q. 2023. LinRec: linear attention mechanism for long-term sequential recommender Systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 289–299.

Liu, Q.; Zeng, Y.; Mokhosi, R.; and Zhang, H. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1831–1839.

Liu, Z.; Liu, Q.; Wang, Y.; Wang, W.; Jia, P.; Wang, M.; Liu, Z.; Chang, Y.; and Zhao, X. 2025. SIGMA: selective gated mamba for sequential recommendation. In *The 39th Annual AAAI Conference on Artificial Intelligence*, 12264–12272.

Ni, J.; Li, J.; and McAuley, J. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*.

Sabour, S.; Frosst, N.; and Hinton, G. E. 2017. Dynamic routing between capsules. In *Conference on Neural Information Processing Systems*, 3856–3866.

Shi, H.; Gu, Y.; Zhou, Y.; Zhao, B.; Gao, S.; and Zhao, J. 2023. Everyone's preference changes differently: a weighted multi-interest model for retrieval. volume 202 of *International Conference on Machine Learning*, 31228–31242.

Shi, S.; Chen, H.; Ma, W.; Mao, J.; Zhang, M.; and Zhang, Y. 2020. Neural logic reasoning. In *ACM International Conference on Information and Knowledge Management*, 1365–1374.

Shi, S.; Xie, Y.; Wang, Z.; Ding, B.; Li, Y.; and Zhang, M. 2022. Explainable neural rule learning. In *International World Wide Web Conferences*, 3031–3041.

Smullyan, R. M. 1995. *First-Order Logic*. Dover Publications.

Spillo, G.; Musto, C.; De Gemmis, M.; Lops, P.; and Semeraro, G. 2022. Knowledge-aware recommendations based on neuro-symbolic graph embeddings and first-order logical rules. In *ACM Conference on Recommender Systems*, 616–621.

Tang, Z.; Floto, G.; Toroghi, A.; Pei, S.; Zhang, X.; and Sanner, S. 2023. LogicRec: recommendation with users' logical requirements. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, 2129–2133.

Wang, S.; Li, B. Z.; Khabsa, M.; Fang, H.; and Ma, H. 2020. Linformer: self-attention with linear complexity. *arXiv e-prints*, arXiv:2006.04768.

Wang, S.; Xie, B.; Ding, L.; Chen, J.; and Xiang, Y. 2025. Reinforced logical reasoning over KGs for interpretable recommendation system. *Machine Learning*, 114(95): 1–27.

Yuan, H.; Zhao, P.; Xian, X.; Liu, G.; Liu, Y.; Sheng, V. S.; and Zhao, L. 2023. Sequential recommendation with probabilistic logical reasoning. In *IJCAI*, 270.

Zhang, S.; Wang, M.; Wang, W.; Gao, J.; Zhao, X.; Yang, Y.; Wei, X.; Liu, Z.; and Xu, T. 2025. GLINT-RU: gated lightweight intelligent recurrent units for sequential recommender systems. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1948–1959.

Zhang, W.; Yan, J.; Wang, Z.; and Wang, J. 2022. Neuro-symbolic interpretable collaborative filtering for attribute-based recommendation. In *International World Wide Web Conferences*, 3229–3238.

Zhou, L.; Du, G.; Lü, K.; Wang, L.; and Du, J. 2024. A Survey and an Empirical Evaluation of Multi-View Clustering Approaches. *ACM Computing Surveys*, 56(7).