

Article

Not peer-reviewed version

MSD: Multi-order Semantic Denoising Model for Session-Based Recommendation

[Shulin Cheng](#)^{*}, Wentao Huang, Zhenqiang Yu, [Jianxing Zheng](#)

Posted Date: 16 July 2024

doi: 10.20944/preprints202407.1337.v1

Keywords: Recommender system; session-based recommendation; graph neural network; attention mechanism



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

MSD: Multi-Order Semantic Denoising Model for Session-Based Recommendation

Shulin Cheng^{1,2,*}, Wentao Huang¹, Zhenqiang Yu¹ and Jianxing Zheng³

¹ School of Computer and Information, Anqing Normal University, Anqing, China

² The University Key Laboratory of Intelligent Perception and Computing of Anhui, Anqing, China

³ School of Computer and Information Technology (School of Big Data), Shanxi University

* Correspondence: ChengShL@aqnu.edu.cn

Abstract: Session-based recommendations which aim to predict subsequent user-item interaction based on historical user behaviour during anonymous sessions can be challenging tasks. Two main challenges need to be addressed and improved: (1) How to analyze these sessions to accurately and completely capture users' preferences, and (2) how to identify and eliminate any interference caused by noisy behavior. Existing methods have not adequately addressed these issues since they either neglected the valuable insight that can be gained from analyzing consecutive groups of items or failed to take these noisy data in sessions seriously and handled them properly, which can jointly impede recommendation systems from capturing users' real intentions. To address these two problems, we designed a multi-order semantic denoising (MSD) model for session-based recommendations. Specifically, we grouped items of different lengths as varying multi-order semantic units to mine the user's primary intention from multiple dimensions. Meanwhile, a novel denoising network was designed to alleviate the interference of noisy behavior and provide a more precise session representation. The result of extensive experiments on three real-world datasets demonstrated that the proposed MSD model exhibited improved performance compared with existing state-of-the-art methods in session-based recommendation.

Keywords: recommender system; session-based recommendation; graph neural network; attention mechanism

1. Introduction

A recommendation system is designed to help people filter information and make efficient choices. Collaborative filtering is the main method used in traditional recommendation systems, providing content for users with specific interests by analyzing their historical behavior. However, in many scenarios, user profiles are unavailable because users are not logged into the system or because of privacy concerns. Moreover, user preferences can change over time, making short-term user behavior more reflective of current user preferences [1]. In such cases, session-based recommendations have garnered growing attention owing to its considerable practical value. Session-based recommendation is one of the most important methods in modern Recommendation System, which capitalize on the analysis of users' immediate interaction behaviors within a given session, such as the sequence of item clicked or viewed. This approach excel in cases where user profiles are either unavailable or incomplete, by dynamically adapting recommendations based on real-time data observed during a users' active session, making it particularly valuable for delivering personalized content and suggestions that are closely aligned with the users' immediate preference and needs.

Considering the sequential nature of sessions, various methods have been proposed for session-based recommendation, such as methods based on Markov chain [2,3], RNN-based models [4–6], and GNN-based models [7,8]. Despite the advancement brought about by these approaches in session-based recommendation, there are still two challenges that have not garnered enough attention. On the one hand, the existing methods always consider each individual item in isolation without taking into account the semantic information that arises from combining multiple items. On the other hand, noisy behavior similar to that in Figure 1 is an inevitable problem in recommendation systems, which can lead to deviations from the primary users' intentions.



Figure 1. Instances of items clicked by mistake or out of curiosity, which negatively impact the generation of reliable recommendation.

Sadly, current methods have not tackled these challenges effectively. For the higher-order semantic information, recent work mainly utilized hypergraph [9,10] to represent relationships between items. Although methods based on hypergraph can facilitate the propagation of information from higher-order connections, the underlying node structure remains unchanged. We argue that nodes in session graph should not be limited to representing single item. For the noisy behavior, reinforcement learning [11] and attention mechanism [12] were the most frequently used methods. Regarding the uncertainty in user behavior, attention mechanism was employed to mitigate the influence of irrelevant items by assigning them lower weights. While it had alleviated the noise issue to some extent, there is still a main limitation that the item serving as the reference for the attention mechanism could itself be a noisy data.

To tackle these two challenges, we designed an innovative model to accurately capture user intentions from two perspectives. Firstly, we proposed a new kind of multi-order semantic unit for analyzing user intentions. In previous methods, researchers used each individual item as a basic unit for this purpose. While in our method, we combined consecutive items of different lengths as varying multi-order semantic units. Meanwhile, we improved the method of generating item embeddings, making it possible for information to propagate between semantic units of different lengths. Secondly, we developed a novel denoising module to explicitly differentiate and handle noisy items. Specifically, we proposed to mine the users' main intentions in the session as the reference to calculate the importance of every semantic unit, and then assigned corresponding weights to each semantic unit based on their importance. Finally, we constructed a pure session embedding using the denoised session sequences.

The contributions of this study can be summarized as follows:

- We proposed a new kind of multi-order semantic unit with varying node lengths to mine user intentions from multiple dimensions and improve the information propagation strategy to fuse information from different semantic units.
- We introduced a novel denoising module based on the varying multi-order semantic units to filter out noisy items and constructed pure session embedding by adjusting the weight of each semantic unit.
- Experiments analysis conducted on three real-world benchmark datasets revealed that our proposed multi-order semantic denoising (MSD) model achieved better performance than other state-of-the-art models in terms of the two evaluation metrics.

2. Related Work

In this section, we review related work on session-based recommendation systems from two perspectives, namely, traditional methods and deep-learning methods.

2.1. Traditional Methods

Conventional methods used in recommendation systems have widely adopted matrix factorization [13,14] to learn the characteristic vectors of both users and items from the user-item interaction matrix to generate recommendation results based on vector similarity. To take advantage of the implicit feedback in user behavior, Bayesian personalized ranking matrix factorization methods (BPR-MF) [15] made use of a generic optimization criterion for personalized ranking, providing a generic learning algorithm for optimizing models. Inspired by the idea of collaborative filtering, that is, similar users tend to have similar interests in similar items, some methods based on the nearest neighbor algorithm [16–18] were proposed to improve the performance of recommendation systems. Item-KNN [16] could be used to calculate the item similarity in a session and provide recommendations based on the last item in the current session. Owing to the lack of knowledge of sequential information, Markov-chain-based methods [3,19] were proposed to capture sequential dependencies in a session. For example, factorizing personalized Markov chains (FPMC) [3] could capture long-term user interests and sequential user behavior to provide more accurate recommendations. The hierarchical representation model (HRM) [19] comprised a two-level hierarchical structure to fuse user preferences and sequence information. However, these traditional methods could not deeply explore the complex transformation relationships between items within a session, that is, these methods assumed that all items in a session were equally important for the analysis of user preferences, which was inconsistent with the actual situation. Given the aforementioned problems associated with traditional methods, researchers have begun to introduce deep-learning methods to session-based recommendation systems.

2.2. Deep-Learning Methods

In recent years, deep learning has been widely used in various research fields as well as in recommendation systems. The accuracy, precision, and impact of personalized recommendations based on deep learning have greatly improved, especially in session-based recommendation systems. Neural-based methods such as RNNs [20] have been widely adopted to capture sequential information in sessions. GRU4Rec [4] introduced an RNN into the field of session-based recommendations, using gated recurrent units [21] to model sequential user behavior. Subsequently, several methods [6,22] have been proposed to compensate for the deficiencies of GRU4Rec. Some methods improved the RNN model through data enhancement [6] while another method [22] proposed a hierarchical RNN model that used both intra- and cross-session information to describe the changes in user interests to generate personalized recommendations. The neural attentive recommendation machine (NARM) [5] introduced an attention mechanism for GRU4Rec that could capture the users' main intentions. Inspired by the NARM, other methods have applied attention mechanisms to enhance models [9,23,24]. The short-term attention/memory priority model (STAMP) [23] focused on strengthening the impact of users' recent behavior when modeling user click behavior over a long time series. BERT4Rec [25] used a bidirectional self-attention network to improve the representation ability of the model. Recently, researchers have found that graph structures can contribute to exploring the relationships between items, providing supplemental information that is not available in sequential structures. Consequently, methods based on session graphs have been widely adopted for session-based recommendation systems.

The GNN [26,27] was introduced to model complex dependencies between items by building a session graph. Session-based recommendation with graph neural network (SR-GNN) [7] was the first to encode sequential session data into a graph structure and apply a gated graph neural network to extract transition information. To solve the problem of information loss in GNNs, the lossless edge-order preserving aggregation and shortcut graph attention for session-based recommendation method (LESSR) [28] adjusted the network structure to mine the users' consumption habits during a certain period of time. Star graph neural networks with highway networks (SGNN-HN) [24] used star graph to propagate information from items without direct connections. Although these studies achieved promising results, there was still a problem to be solved, that is, there were always noisy

clicks in the sessions, which meant that not all consecutive items had dependencies with one another. Consequently, several methods have been proposed to handle the problem of noisy clicks [11,12,29]. Reinforcement learning-based denoising network (RED) [11] transformed the sequential denoising problem into a Markov decision process (MDP), applying reinforcement learning to separate session sequences. The Graph neural network with global noise filtering (GNN-GNF) [29] filtered noisy clicks at an item-level and at a session-level. The dynamic intent-aware iterative denoising network (DIDN) [12] included an iterative denoising module by learning dynamic item embedding. These methods reduced noisy-click interference on the model to some extent, but none of them could filter noisy clicks from multiple dimensions. In this study, our proposed model adopted semantic units as the basic nodes to construct a novel denoising network which could discriminate and remove noisy clicks based on both individual and combined expressions of items, providing more accurate and comprehensive recommendation results.

3. Methodology

3.1. Problem Definition

Session-based recommendations aim to predict a user's next click based on the sequence of items with which the user has interacted. In contrast to sequential recommendations, users are anonymous (without profile information) in session-based recommendations. Here, we set $I = \{v_1, v_2, v_3, \dots, v_n\}$ to denote all items in the dataset and $V = \{v_{s,1}, v_{s,2}, v_{s,3}, \dots, v_{s,l}\}$ to denote the current anonymous session, where n and l denote the number of items in the dataset and session, respectively. The proposed MSD model is applied to produce a set of candidate items $C = \{v_{r,1}, v_{r,2}, v_{r,3}, \dots, v_{r,k}\}$, in which users are more likely to click on items with higher ranks. In the proposed model, we select the top N items from set C as the recommendation results.

3.2. Model Framework

Our idea of designing models is specifically tailored to address the two challenges presented in the introduction. Firstly, the products users browse on an e-commerce platform are not isolated data points. They collectively form the user's browsing path, which can reveal the user's underlying purchase intention and preferences. Combining the browsed products into semantic units of varying lengths is an attempt to capture and utilize this supplemental intention information. Semantic units can be constructed based on the similarity of products (such as category, brand, price, etc.) or defined based on the sequence and time intervals of user browsing. Semantic units of different lengths can reflect different aspects of user behavior, where short sequences may represent immediate interests, while long sequences may reveal long-term preferences or shopping habits. Secondly, Users navigating through an e-commerce platform often explore various products out of curiosity, without a clear intention to purchase, which are not align with their primary interests or needs at that moment. Attention mechanisms can help identify noisy items in user sessions. By arranging the focus of user browsing behaviors, the model can concentrate attention on products that are relevant to the user's actual interests, reducing the impact of noisy items on recommendation results.

To solve these challenges and improve the performance of session-based recommendation, we design the method framework of our proposed MSD model, which contains three key modules, namely multi-order semantic construction module, denoising module and prediction module. As shown in Figure 2, the upper part is the multi-order semantic construction module. In this module, we combine items of different lengths as semantic units and convert them to a heterogeneous graph. Then we use two types of fusion functions to generate the embedding of each semantic units. For example, in Figure 2 we first divide and merge the items in the session into 3-order semantic units. Following that, we employ the set-aware fusion function and sequence-aware fusion function to generate the embedding. The bottom left part is the denoising module. In this module, we employ an attention mechanism to judge the similarity between different semantic units and the main intentions to screen out noisy nodes. The biggest difference from traditional attention mechanisms, which use the last

item of a session as the reference, is that we first mine the user's main intentions from multi-order semantic units. We argue that the last item in the session could also be a noise data. Moreover, using only the last item may not capture the user's evolving preferences and interests over the course of the session, leading to a bias towards recent items in the recommendation. By using the main intention as a reference standard, these two issues can be effectively avoided. The bottom right part of Figure 2 is prediction module. In this module, we employ a prediction layer to compare the correlation between the session embedding and the embedding of each item, selecting the top 20 items with the highest correlation as our recommendation results. By integrating results from different orders, we can take into account semantic information across various dimensions to generate more diverse recommendation outcomes.

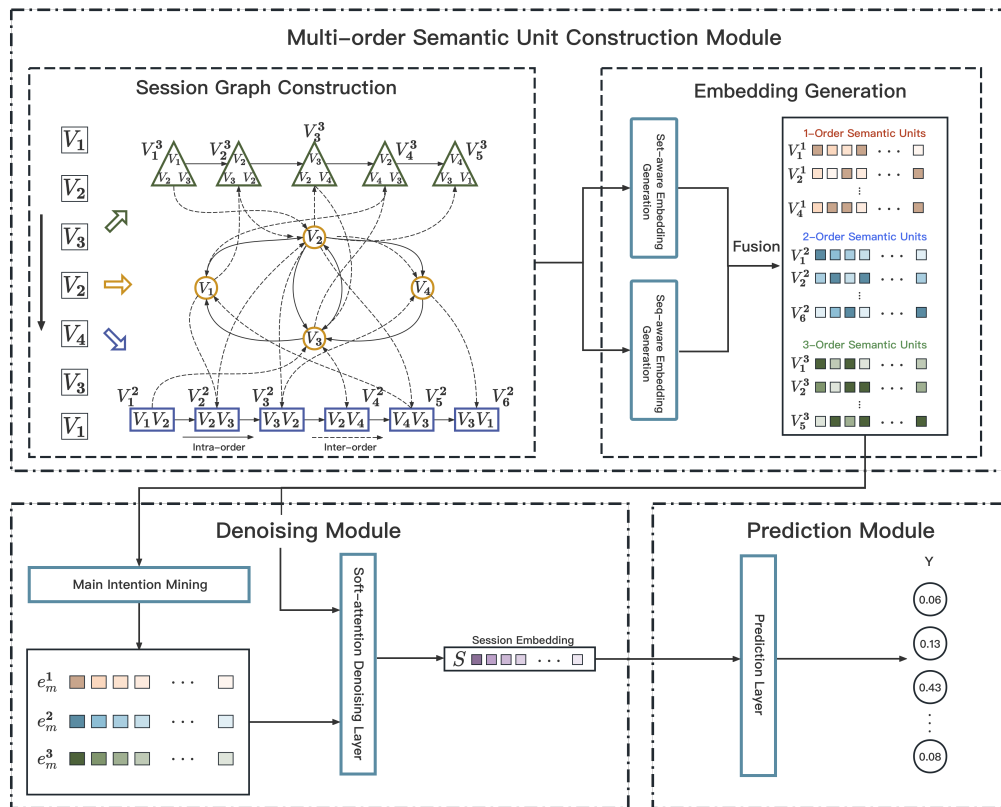


Figure 2. The framework of the proposed MSD model.

Next, we will elaborate on the technical details of the three modules mentioned above in Section 3.3, Section 3.4 and Section 3.5.

3.3. Multi-Order Semantic Unit Construction Module

In this section, we demonstrate how to convert item nodes into semantic units and how to initialize and update their representations.

3.3.1. Multi-Order Semantic Unit Construction

Currently, most session-based recommendation methods consider each item separately. However, high-level semantic information [30] is ignored. In our model, we not only consider the correlation between individual items but also focus on the combined information revealed by several consecutive items. We define v_j^k to be a semantic unit, in which k represents its order and j represents its position. For example, in Figure 2 the session is $S_1 = \{v_1, v_2, v_3, v_2, v_4, v_3, v_1\}$. Here, we can obtain the first-order semantic units, that is, $v_1^1 = \{v_1\}, v_2^1 = \{v_2\}, v_3^1 = \{v_3\}, \dots, v_6^1 = \{v_1\}$. The combined session fragments

$\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_3, v_1\}$ are second-order semantic units denoted by $v_1^2, v_2^2, \dots, v_3^2$, the third-order semantic units being $v_1^3 = \{v_1, v_2, v_3\}, v_2^3 = \{v_2, v_3, v_1\}, v_3^3 = \{v_3, v_1, v_2\}, \dots, v_5^3 = \{v_4, v_3, v_1\}$.

During node initialization, we assign the learnable embedding vector (e_j^1) to represent each first-order semantic unit. A high-order semantic unit is initialized by fusing the first-order semantic units. Specifically, we designed a two-part fusion function to aggregate the related information revealed by the included items. Considering that the items in each high-order semantic unit are sequential, that is, although two high-level semantic units may contain the same items, the positions of items in two high-order units may differ, we use two ways to integrate item information, named the *Set-aware* function and *Sequence-aware* function. In the *Set-aware* function, we use the *MEAN* functions to integrate the dimension information, whereas in the *Sequence-aware* function, we use a gated recurrent unit (GRU) to extract the position information. For a higher-order semantic unit, the final intention can be calculated as follows:

$$e_j^k = F_{set}(e_j^1, e_{j+1}^1, e_{j+2}^1, \dots, e_{j+k-1}^1) + F_{seq}(e_j^1, e_{j+1}^1, e_{j+2}^1, \dots, e_{j+k-1}^1) \quad (1)$$

where e_j^k denotes the learnable embedding vector of the k-order semantic units, and F_{set} and F_{seq} denote the *Set-aware* and *Sequence-aware* functions, respectively.

3.3.2. Semantic Unit Embedding Learning

Before learning the session representation, we construct a heterogeneous session graph for each session. Each session can be modeled as a directed graph $\mathcal{G} = (\mathcal{V}_s, \mathcal{E}_s)$, where \mathcal{V}_s and \mathcal{E}_s denote the semantic units and the edges between them, respectively. An example of a heterogeneous session graph is shown in Figure 2, there being two edge types in this graph, namely, the intra-order edge and the inter-order edge. Intra-order edges connect semantic units of the same order, whereas inter-order edges provide a channel for intention propagation between high-order semantic units and their adjacent first-order semantic units.

With the evolution of the network, the model constantly encounters unseen nodes. Inspired by the GraphSAGE framework [31], we employ a heterogeneous graph convolution network to learn the representation of each semantic unit. Node embeddings are generated by aggregating related information from the node and its neighbors. Specifically, for node V , we can update its representation as follows: first, we aggregate the representation of the nodes in V 's immediate neighbors into a single-vector $A_{N(v)}^{k-1}$. After aggregating the neighbors' feature vectors, we concatenate the node's current representation h_v^{k-1} with the aggregated neighborhood vector $A_{N(v)}^{k-1}$ before feeding it through a fully connected layer with nonlinear activation function σ . Given a neighbor set $N(v)$, the aggregation mechanism for updating node V can be expressed as follows:

$$A_{N(v)}^{k-1} = \text{aggregate}(\{h_j^{k-1}, \forall j \in N(v)\}) \quad (2)$$

$$h_v^k = \sigma(W_K \cdot \text{concat}(A_{N(v)}^{k-1}, h_v^{k-1})) \quad (3)$$

$$h_v^k = \text{Norm}(h_v^k) \quad (4)$$

where $W_k \in R^{d \times d}$ denotes the projection weights which are learned and layer-specific across different layers, and $h_v^0 = e_v^0$ denotes the initial representation of semantic unit V .

3.4. Denoising Module

The denoising process was completed in three steps. First, we extracted the user's main intention from the session as a criterion to determine which items tended to be noisy clicks. We then used the

soft-attention mechanism [32] to measure the correlation between each semantic unit and the main intention. Finally, we performed noise reduction based on the attention coefficients.

3.4.1. Main Intention

The first step of denoising module is to find the approximate direction in which users are searching for, which we call the main intention. Considering that user intentions may change in a session, much like the initialization of higher-order semantic units, both the items contained in the session and their relative positions need to be handled carefully. We use two methods to extract these two pieces of information separately. Given session S , the main intention can be expressed as follows:

$$e_m^k = C_1 \cdot F_{set}(e_1^k, e_2^k, e_3^k, \dots, e_n^k) + C_2 \cdot F_{seq}(e_1^k, e_2^k, e_3^k, \dots, e_n^k) \quad (5)$$

where $C_1 + C_2 = 1$ denotes the weight coefficient learned by the network, which determines the importance of the two types of information, n denotes the number of semantic units in each order, e_m^k denotes the primary purpose of the k -order semantic units, and F_{set} and F_{seq} denote the *set-aware* and *sequence-aware* functions, respectively.

3.4.2. Attention Coefficient

To measure the importance of each semantic unit in a session, we can employ an attention mechanism to calculate the relevance between each semantic unit and the main intention. For the k^{th} -order semantic unit, the attention coefficient for each semantic unit v_j^k can be expressed as follows:

$$a_j^k = W_0^k \sigma(W_1^k h_j^k + W_2^k e_m^k + b^k) \quad (6)$$

where $W_0^K \in R^d$, $W_1^K \in R^{d \times d}$, and $W_2^K \in R^{d \times d}$ denote learnable parameters, $b^k \in R^d$ denotes the bias, and $\sigma(\cdot)$ denotes the sigmoid function, which is not shared between different layers.

3.4.3. Denoising Coefficient

Based on the attention coefficient calculated using Eq. (6), we can assign a normalized weight coefficient to each semantic unit to generate the final session representation. Before normalization, we first set a threshold to remove semantic units with an attention coefficient much lower than the average level for the interference elimination of items that obviously do not represent user interest, as follows:

$$a_j^k = \begin{cases} a_j^k, & a_{avg}^k - a_j^k < \delta \\ 0, & a_{avg}^k - a_j^k \geq \delta \end{cases} \quad (7)$$

where δ denotes a parameter controlling the denoising depth, the average attention coefficient being calculated using $a_{avg}^k = (\frac{1}{n} \sum_{j=1}^n a_j^k)$.

3.5. Prediction Module

This section comprises two parts, that is, we demonstrate how to generate session embedding which describes user behavior preferences throughout the entire session, and how to train our model.

3.5.1. Session Embedding Learning

In Generally, Session-based recommendation system divide the session embedding into two parts: local preference and global preference. The local preference focuses on the immediate action within the session, reflecting short-term behaviors and specific contextual influence. On the other hand, the global preference captures broader and more enduring interests that are consistent across the whole

session. Similarly, we also adopt this approach in our model. we consider the last item v_n^k of each order to be the local embedding, as follows:

$$S_{local}^k = h_n^k \quad (8)$$

We can use denoised semantic units to represent the global embedding, calculated as follows:

$$S_{global}^k = \sum_{j=1}^n Softmax(a_j^k) h_j^k \quad (9)$$

Finally, We can combine the local and global embeddings to generate session embeddings for each order of the semantic units, as follows:

$$S^k = W_3^k [S_{local}^k \cdot S_{global}^k] \quad (10)$$

where $[\cdot]$ denotes the concatenation operation and W_3^k denotes the projection matrix that transforms S^k from R^{2d} into R^d .

3.5.2. Training and Optimizing

After generating session embeddings from all orders, that is $S^1, S^2, S^3, \dots, S^k$, we can adopt the inner product to calculate the similarity between the candidate item and the k -order session embedding. Specifically, given a candidate set $C = \{v_1, v_2, v_3, \dots, v_n\}$, we can calculate the score for each item $v_i \in C$, as follows:

$$z_i^k = \langle S^k \cdot e_i^1 \rangle \quad (11)$$

where $\langle \cdot \rangle$ denotes the inner product operation. We can then apply a *SoftMax* function to calculate the probabilities of items becoming the next-click item, as follows:

$$y^k = Softmax(z_1^k, z_2^k, z_3^k, \dots, z_n^k) \quad (12)$$

The final score for each item v_i in the candidate set C can be calculated as follows:

$$Score_i = \sum_{k=1}^k y_i^k \quad (13)$$

Finally, we can train the proposed model using the cross-entropy loss, the loss function being as follows:

$$L(p, score) = - \sum_{i=1}^{|C|} p \cdot \log(Score_i) + (1 - p) \cdot \log(1 - Score_i) \quad (14)$$

where p denotes the ground truth (that is, whether the user will click on item v_i).

4. Experimental Results

In this section, we describe the details of our experimental settings, including the datasets, baselines, evaluation metrics and experimental results with comparisons. Among these, we focus on analyzing the experimental results.

4.1. Datasets

We evaluated the performance of MSD on three real-world representative datasets commonly used in the field of session-based recommendations, that is, the Diginetica, Yoochoose 1/64, and Yoochoose 1/4 datasets.

- Diginetica is a personalized e-commerce dataset released at the CIKM Cup 2016. It contains user interaction records from an online store that we used the sessions in the previous week for testing.
- Yoochoose is a public dataset obtained from the RecSys Challenge 2015. It contains a stream of user clicks on an e-commerce website within six months, which we used as the 1/64 and 1/4 subsamples of all training sessions for testing.

Following Wu et al. [7] and Chen et al. [28], we filtered out sessions with lengths less than two and items that appeared less than five times. We also filtered items from the test dataset that did not appear in the training dataset. Additionally, the data augmentation methods described by Li et al. [5] and Liu et al. [23] were used to process the data in both datasets. The statistics of the preprocessed data are presented in Table 1.

Table 1. Statistics of the preprocessed data.

	Diginetica	Yoochoose1/64	Yoochoose1/4
#clicks	982,961	557,248	8,326,407
#train sessions	719,470	369,859	5,917,746
#test sessions	60,858	55,898	55,898
#unique items	43,097	16,766	29,618
Average length	5.12	6.16	5.71

4.2. Baselines

We carefully selected three kinds of methods, namely, the nearest neighbor methods, RNN-based methods, and GNN-based methods, as our baselines, their descriptions being as follows:

- The Item-KNN model [16] recommends items that are similar to the previously clicked items in the current session where cosine similarity is adopted to calculate the similarity between the items.
- The GRU4Rec model [4] adopts RNNs to model the sequential behavior of items in current session.
- The NARM model [5] improves the GRU4Rec model by adding an attention mechanism to the RNN to capture the main purposes of users.
- The STAMP [23] captures the general interests and current interests of users by replacing the RNN encoder with an attention layer.
- The SR-GNN model [7] encodes session sequences into a graph structure and employs a GNN to capture the complex item transitions.
- The SGNN-HN model [24] applies a SGNN to propagate information from items without direct connections and uses a HN to tackle overfitting problems.
- The LESSR model [28] tackles the information loss and long-range dependency problems of GNN-based models by introducing two kinds of session graphs.
- The GNN-GNF [29] model leverages graph neural networks and global noise filtering to enhance accuracy and personalization in session-based recommendation tasks.
- The DIDN [12] model utilizes dynamic intention awareness and iterative denoising to filter noisy items based on user intention, continuously refining recommendation results during the process to enhance personalized recommendations.

4.3. Metrics

To compare the performance of the proposed MSD method with the baseline models, we adopted two widely used metrics, namely, the $MRR@k$ and $P@K$ metrics. To maintain the same setup as the previous baselines, we chose to use the top 20 items to evaluate the proposed model, that is, the $MRR@20$ and $P@20$ metrics.

$MRR@20$ (mean reciprocal rank) was used to measure the average reciprocal ranks of correct items, representing whether the correct item was in the top 20 items on the recommendation list. $P@20$ (precision) was used to measure the prediction accuracy, representing the ratio of the number of correct items in the top 20 items on the recommendation list.

4.4. Experimental Setup

For the proposed MSD model and all baselines, we optimized the model hyperparameters via a grid search on the validation dataset, which was randomly split from the training dataset (using 10% of it). Specifically, we set the number of embedding dimensions to 256 and the batch size to 512. We used the Adam optimizer [33] to optimize our model, where the initial learning rate was set to 0.001 and decayed every three epochs at a rate of 0.1. It should be noted that we searched for the best semantic unit order from 1 to 6 and the best denoising depth from 0.0005 to 0.005. We compared our MSD model with the baseline models using the $P@20$ and $MRR@20$ metrics.

4.5. Performance Comparison

To evaluate the overall performance of the proposed model, we compared it with the other representative session-based recommendation methods. A summary of the experimental results for the two datasets is presented in Table 2.

Table 2. Summary of the experimental results for the three datasets.

Methods	Diginetica		Yoochoose1/64		Yoochoose1/4	
	$P@20$	$MRR@20$	$P@20$	$MRR@20$	$P@20$	$MRR@20$
Item-KNN	35.75	11.57	51.60	21.81	52.31	21.70
GRU4Rec	29.45	8.33	60.64	22.89	59.53	22.60
NARM	49.70	16.17	68.32	28.63	69.73	29.23
STAMP	45.64	14.32	68.74	29.67	70.44	30.00
SR-GNN	50.73	17.59	70.57	30.94	71.36	31.89
SGNN-HN	55.67	19.45	72.06	32.61	72.85	32.55
LESSR	52.17	18.13	70.94	31.16	71.40	31.56
GNN-GNF	51.61	17.77	71.50	31.35	72.11	31.67
DIDN	56.22	20.30	72.12	31.69	72.65	32.59
MSD	56.93	19.87	73.61	33.75	74.55	34.21

From the results, we can make the following important observations: First, conventional methods typically generate recommendations based on co-occurring or successive items. These methods mostly use simple models and are insufficient for extracting and representing complex information. Deep-learning methods greatly improve the representation ability of models through the use of neural networks, thus achieving better performance than conventional methods. Second, the GRU4Rec model achieves relatively poor results, whereas the NARM achieves competitive results. The reason is that the GRU4Rec model applies only sequential information, while the NARM employs both sequential information and global preferences. Moreover, the SR-GNN model outperforms all previous methods, demonstrating the effectiveness of the graph structure in representing the transition relationship of items in the session. The SGNN-HN model further improves the ability to propagate information and addresses the overfitting problem, indicating that there is still room for improving the ability to extract information. Third, the excellent performance of DIDN model demonstrates the importance of session denoising. Finally, by extracting information from multiple dimensions and tackling the noise issue, the proposed MSD model achieves the best performance in terms of the $P@20$ and $MRR@20$ metrics over the three datasets.

We can elaborate on the improvement in the MSD model from two perspectives. Firstly, the MSD model has the capability to leverage user intention from multiple dimensions. Introducing multi-order semantic units as fundamental nodes within the session allows the MSD model to propagate and

extract information more comprehensively. This means that the model can capture nuances and intricate patterns in user behavior, leading to a richer representation of user preferences and interests.

Secondly, the model includes a denoising module following the information propagation step. This denoising module plays a crucial role in enhancing the model’s performance. It discriminates noisy items from the session data and removes them effectively. By doing so, the model is able to generate a cleaner and more accurate representation of the items, focusing on the most relevant and meaningful interactions. This process improves the quality of the learned embeddings and ultimately leads to more precise and personalized recommendations for users.

4.6. Ablation Study

In this section, we evaluate each key component of our model through ablation experiments. We successively removed the denoising and multi-order semantic modules in each experiment and recorded the change in performance, which indicated the effectiveness of the removed component.

Table 3. Recommendation performance comparisons of ablation models on the three datasets.

Methods	Diginetica		Yoochoose1/64		Yoochoose1/4	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
<i>MSD</i>	56.93	19.87	73.61	33.75	74.55	34.21
<i>MSD_{-D}</i>	56.68	19.41	73.18	33.27	73.87	33.75
<i>MSD_{-D&M}</i>	55.54	18.89	72.63	32.86	73.52	33.48

4.6.1. Impact of the Denoising Module

To tackle the challenges posed by natural noise in session data, we introduced a novel denoising module designed to filter out noisy user behaviors in the session following the propagation of information. In order to demonstrate the effectiveness of this module, we conducted experiments by setting the denoising depth to 1 and removing the attention coefficient calculation. This configuration meant that the model could not differentiate between semantic units and all units were treated as equally important in the session. The results presented in Table 3 indicate a significant decrease in both *P@20* and *MRR@20* metrics.

These findings underscore the presence of natural noise in session data and emphasize the importance of identifying and mitigating its impact. Removing the denoising module leads to the proposed system being misled by noisy items in the session, resulting in a recommendation list that failed to align with users’ interests. Additionally, treating all items as equally important hindered the system’s ability to discern the main intentions of users, thereby further compromising the accuracy of the recommendation results.

This analysis highlights the critical role played by the denoising module in improving the quality of the recommendation system by effectively filtering out noisy data and enhancing the model’s ability to capture users’ true preferences and interests.

4.6.2. Impact of the Multi-Order Semantic Module

To address the issue of information loss in previous methods, we adopted semantic units as fundamental nodes to explore user preferences more comprehensively. To assess the effectiveness of the semantic unit approach, we conducted experiments with the order limited to 1. This configuration meant that the model could not combine items of different lengths into semantic units, instead treating each individual item as a basic node.

The results of our ablation study clearly indicate that removing the multi-order semantic module has a detrimental effect on both *P@20* and *MRR@20* metrics. This is because higher-order semantic units contain rich user intention information, and mining this from various orders of semantic units helps mitigate the problem of information loss. Without the multi-order semantic module, the model loses the ability to capture the intricate relationships and dependencies between items across different

lengths of semantic units. This limitation restricts the model's context awareness and propagation path for long dependencies, ultimately leading to a decrease in recommendation accuracy.

In summary, the multi-order semantic module plays a crucial role in capturing and leveraging the diverse user intention information present in different semantic unit orders. Its removal not only harms the model's ability to address information loss but also limits its capability to understand the complex patterns and contexts within user sessions, resulting in a decrease in the quality of recommendation results.

4.6.3. In-Depth Analysis

Additionally, we conducted experiments to evaluate the contributions of the intra-/inter-edge and main intention. Specifically, we trained two models without intra- or inter-edges. In the third model, we replaced the main intention with a representation of the last semantic unit to validate its impact. As shown in Table 4, all three models perform worse than the original models.

Table 4. Impact of two kinds of edges and main intention.

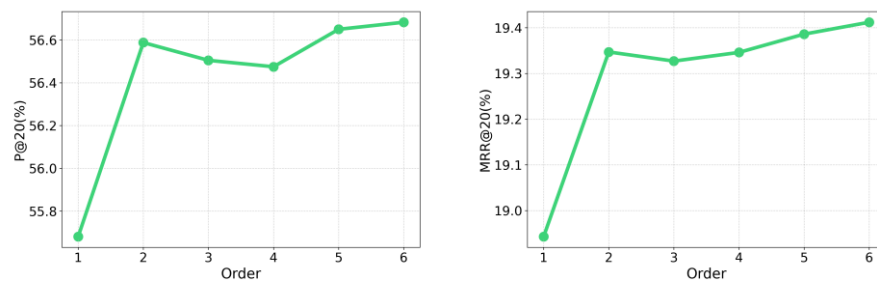
Methods	Yoochoose1/64		Diginetica	
	P@20	MRR@20	P@20	MRR@20
MSD	73.61	33.75	56.93	19.87
w/o Inter edge	73.39	33.25	56.77	19.76
w/o Intra edge	73.26	33.21	56.71	19.58
w/o Main intention	73.40	33.58	56.81	19.73

The model's performance suffers when intra-edges are removed, as these edges are essential for propagating information between same-order semantic units, providing a crucial foundation for analyzing user preferences. Additionally, the inter-edges also benefit from this propagation of information, further enhancing the model's understanding of user behavior and improving recommendation accuracy. Unlike previous methods that relied on the last item as a criterion for judging the importance of other items, we calculated the main intention of the session to identify noisy clicks. The results clearly demonstrate that the last item may not always represent the user's current interest or could even be a noisy item. By calculating the main intention of the session, we can better filter out noisy clicks and focus on the items that truly reflect the user's interests at a given moment. This approach results in more accurate and personalized recommendations compared to relying solely on the last item of the session, which may not always be indicative of the user's true preferences.

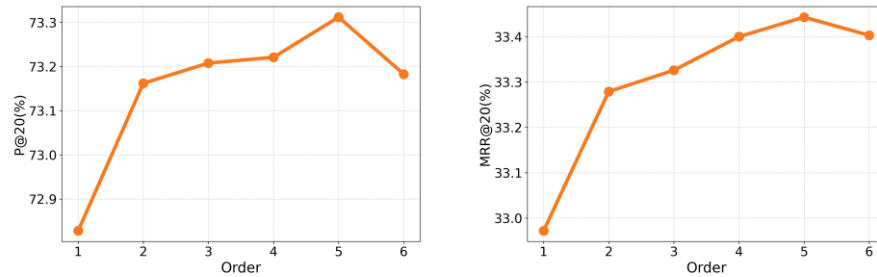
4.7. Parameter Analysis

The two hyperparameters of the order number and denoising depth are important in the proposed mode. Here, we evaluated their roles in the proposed MSD model in terms of the two metrics on the Diginetica and Yoochoose1/64 datasets. The effects of these two hyperparameters are shown in Figures 3 and 4, respectively.

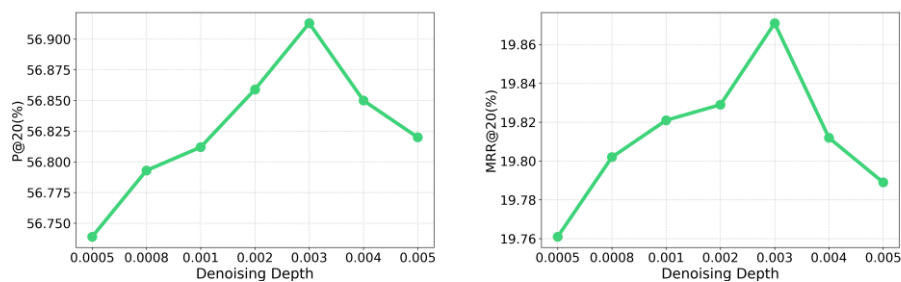
As depicted in Figure 3, we present the experimental results across different orders ranging from 1, 2, 3, 4, 5, 6. Generally, as the order increases, the performance of the proposed model demonstrates stable improvement. Notably, for both the Diginetica and Yoochoose 1/64 datasets, significant enhancements in model performance were observed when transitioning from order 1 to 2. This indicates that the proposed high-order semantic unit, compared to previous methods that analyze user preference based on individual items, possesses superior expressive capability.



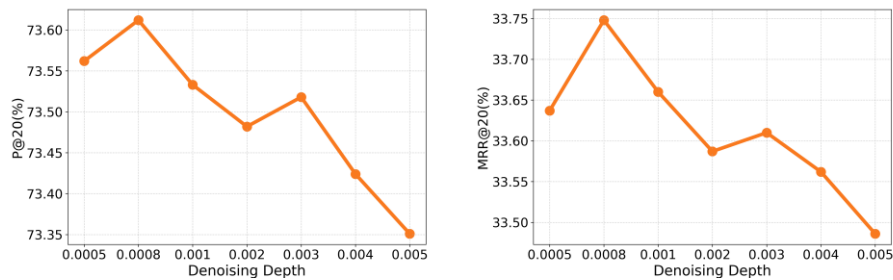
(a) P@20 and MRR@20 on the Diginetica dataset



(b) P@20 and MRR@20 on the Yoochoose 1/64 dataset

Figure 3. The performance of different orders.

(a) P@20 and MRR@20 on the Diginetica dataset



(b) P@20 and MRR@20 on the Yoochoose 1/64 dataset

Figure 4. The performance of denoising depths.

However, the rate of improvement becomes relatively modest as the order surpasses two. This phenomenon can be attributed to the fact that sessions in both datasets are typically not very long. Continuously increasing the order may lead to overfitting, resulting in diminishing returns in performance improvement.

The primary distinction between the two datasets lies in the magnitude of improvement observed when utilizing the Yoochoose 1/64 dataset compared to the Diginetica dataset. This discrepancy can be attributed to the average session length; the Yoochoose 1/64 dataset exhibits longer average session lengths than the Diginetica dataset. Consequently, a dataset with lengthier sessions necessi-

tates a higher order to effectively capture and model user preferences, leading to more substantial improvements in performance.

We conducted experiments with varying denoising depths to assess their impact on the proposed model. Overall, when integrating the denoising module with the multi-order semantic module, improvements were observed in both the $P@20$ and $MRR@20$ metrics. As illustrated in Figures 4, it was evident that excessively large or too low denoising depths did not yield optimal performance. The optimal performance points differed notably between the two datasets. This indicates that each dataset contains a certain level of noisy data, with the degree of noise varying between them.

The results demonstrate the importance of appropriately tuning the denoising depth to achieve optimal performance for a specific dataset. Too high of a denoising depth may lead to excessive filtering, potentially removing relevant information and hindering the model's ability to capture important patterns in the data. Conversely, too low of a denoising depth may fail to effectively filter out noise, resulting in suboptimal performance. The optimal denoising depth strikes a balance between reducing noise interference and preserving valuable information, leading to improved recommendation accuracy.

5. Conclusions

In this study, we introduced a novel Multi-Order Semantic Denoising (MSD) model aimed at addressing two long-standing challenges in session-based recommendation systems. In contrast to previous methods, our approach involves mining a user's main intention from a higher dimension by combining items of different lengths as multi-order semantic units. These semantic units are then encoded into a heterogeneous graph, and a Graph Neural Network (GNN) is employed to propagate the information effectively.

Additionally, we developed a denoising module utilizing a soft-attention mechanism to assess the relevance between different semantic units. This module helps recognize and mitigate the impact of noisy behavior in the session data. An ablation study was conducted using three real-world datasets to assess the effectiveness of each component of our model. The results highlighted the importance of capturing user preferences from multiple dimensions and the significance of filtering out noisy data to accurately analyze user intentions.

In future work, we aim to streamline the item representation and optimize the information propagation mechanism to make our model lighter and more efficient. This will not only improve the model's computational efficiency but also its scalability and applicability in large-scale recommendation systems.

Author Contributions: Shulin Cheng: Conceptualization, Methodology, Formal analysis, Writing - review & editing, Supervision, Funding acquisition. Wentao Huang: Methodology, Writing - original draft, Visualization. Zhenqiang Yu: Data curation, Validation. Jianxing Zheng: Writing - Review & Editing.

Funding: The work was supported by grants from the Nature Science Foundation of Anhui Province in China, No.2008085MF193, the Outstanding Young Talents Program of Anhui Province in China, No.gxyqZD2018060.

Data Availability Statement: The source codes of all the experiments in this paper are available on github https://github.com/shulincheng/MSD_Recsys.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Wang, S.; Cao, L.; Wang, Y.; Sheng, Q.Z.; Orgun, M.A.; Lian, D. A survey on session-based recommender systems. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–38.
2. Shani, G.; Heckerman, D.; Brafman, R.I.; Boutilier, C. An MDP-based recommender system. *J. Mach. Learn. Res.* **2005**, *6*.
3. Rendle, S.; Freudenthaler, C.; Schmidt-Thieme, L. Factorizing personalized markov chains for next-basket recommendation. Proceedings of the 19th international conference on World wide web, 2010, pp. 811–820.

4. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. *arXiv* **2015**, arXiv:1511.06939.
5. Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; Ma, J. Neural attentive session-based recommendation. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1419–1428.
6. Tan, Y.K.; Xu, X.; Liu, Y. Improved recurrent neural networks for session-based recommendations. Proceedings of the 1st workshop on deep learning for recommender systems, 2016, pp. 17–22.
7. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-based recommendation with graph neural networks. Proceedings of the AAAI conference on artificial intelligence, 2019, pp. 346–353.
8. Wang, X.; He, X.; Chua, T.S. Learning and reasoning on graph for recommendation. Proceedings of the 13th international conference on web search and data mining, 2020, pp. 890–893.
9. Wang, J.; Ding, K.; Zhu, Z.; Caverlee, J. Session-based recommendation with hypergraph attention networks. Proceedings of the 2021 SIAM international conference on data mining (SDM), 2021, pp. 82–90.
10. Xia, X.; Yin, H.; Yu, J.; Wang, Q.; Cui, L.; Zhang, X. Self-supervised hypergraph convolutional networks for session-based recommendation. Proceedings of the AAAI conference on artificial intelligence, 2021, pp. 4503–4511.
11. Tong, X.; Wang, P.; Niu, S. Reinforcement learning-based denoising network for sequential recommendation. *Appl. Intell.* **2023**, *53*, 1324–1335.
12. Zhang, X.; Lin, H.; Xu, B.; Li, C.; Lin, Y.; Liu, H.; Ma, F. Dynamic intent-aware iterative denoising network for session-based recommendation. *Inf. Process. Manag.* **2022**, *59*, 102936.
13. Mnih, A.; Salakhutdinov, R.R. Probabilistic matrix factorization. *Adv. Neural Inf. Process. Syst.* **2007**, *20*.
14. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37.
15. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
16. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web, 2001, pp. 285–295.
17. Davidson, J.; Liebald, B.; Liu, J.; Nandy, P.; Van Vleet, T.; Gargi, U.; Gupta, S.; He, Y.; Lambert, M.; Livingston, B.; others. The YouTube video recommendation system. Proceedings of the fourth ACM conference on Recommender systems, 2010, pp. 293–296.
18. Dias, R.; Fonseca, M.J. Improving music recommendation in session-based collaborative filtering by using temporal context. 2013 IEEE 25th international conference on tools with artificial intelligence, 2013, pp. 783–788.
19. Wang, P.; Guo, J.; Lan, Y.; Xu, J.; Wan, S.; Cheng, X. Learning hierarchical representation model for nextbasket recommendation. Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval, 2015, pp. 403–412.
20. Duan, J.; Zhang, P.F.; Qiu, R.; Huang, Z. Long short-term enhanced memory for sequential recommendation. *World Wide Web* **2023**, *26*, 561–583.
21. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* **2014**.
22. Quadrana, M.; Karatzoglou, A.; Hidasi, B.; Cremonesi, P. Personalizing session-based recommendations with hierarchical recurrent neural networks. proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 130–137.
23. Liu, Q.; Zeng, Y.; Mokhosi, R.; Zhang, H. STAMP: short-term attention/memory priority model for session-based recommendation. Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 1831–1839.
24. Pan, Z.; Cai, F.; Chen, W.; Chen, H.; De Rijke, M. Star graph neural networks for session-based recommendation. Proceedings of the 29th ACM international conference on information & knowledge management, 2020, pp. 1195–1204.
25. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 1441–1450.
26. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24.

27. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI open* **2020**, *1*, 57–81.
28. Chen, T.; Wong, R.C.W. Handling information loss of graph neural networks for session-based recommendation. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1172–1180.
29. Feng, L.; Cai, Y.; Wei, E.; Li, J. Graph neural networks with global noise filtering for session-based recommendation. *Neurocomputing* **2022**, *472*, 113–123.
30. Guo, J.; Yang, Y.; Song, X.; Zhang, Y.; Wang, Y.; Bai, J.; Zhang, Y. Learning multi-granularity consecutive user intent unit for session-based recommendation. *Proceedings of the fifteenth ACM International conference on web search and data mining*, 2022, pp. 343–352.
31. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
32. Yuan, J.; Song, Z.; Sun, M.; Wang, X.; Zhao, W.X. Dual sparse attention network for session-based recommendation. *Proceedings of the AAAI conference on artificial intelligence*, 2021, pp. 4635–4643.
33. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.