applied sciences

MDPI

Article

CDF-LS: Contrastive Network for Emphasizing Feature Differences with Fusing Long- and Short-Term Interest Features

Kejian Liu [1,2,*], Wei Wang [1], Rongju Wang [1,†], Xuran Cui [1,†], Liying Zhang [1,†], Xianzhi Yuan [1,†] and Xianyong Li [1,†]

[1] School of Computer and Software Engineering, Xihua University, Chengdu 610039, China; weiwang@stu.xhu.edu.cn (W.W.); wangrongju@stu.xhu.edu.cn (R.W.); 212021085400056@stu.xhu.edu.cn (X.C.); zhangliying@stu.xhu.edu.cn (L.Z.); yuanxianzhi@stu.xhu.edu.cn (X.Y.); lixy@mail.xhu.edu.cn (X.L.)
[2] Lab of Security Insurance of Cyberspace, Chengdu 610039, China
* Correspondence: liukejian@gmail.com
† These authors contributed equally to this work.

Abstract: Modelling both long- and short-term user interests from historical data is crucial for generating accurate recommendations. However, unifying these metrics across multiple application domains can be challenging, and existing approaches often rely on complex, intertwined models which can be difficult to interpret. To address this issue, we propose a lightweight, plug-and-play interest enhancement module that fuses interest vectors from two independent models. After analyzing the dataset, we identify deviations in the recommendation performance of long- and short-term interest models. To compensate for these differences, we use feature enhancement and loss correction during training. In the fusion process, we explicitly split long-term interest features with longer duration into multiple local features. We then use a shared attention mechanism to fuse multiple local features with short-term interest features to obtain interaction features. To correct for bias between models, we introduce a comparison learning task that monitors the similarity between local features, short-term features, and interaction features. This adaptively reduces the distance between similar features. Our proposed module combines and compares multiple independent long-term and short-term interest models on multiple domain datasets. As a result, it not only accelerates the convergence of the models but also achieves outstanding performance in challenging recommendation scenarios.

Keywords: recommendation system; contrast learning; deep learning

Citation: Liu, K.; Wang, W.; Wang, R.; Cui, X.; Zhang, L.; Yuan, X.; Li, X. CDF-LS: Contrastive Network for Emphasizing Feature Differences with Fusing Long- and Short-Term Interest Features. *Appl. Sci.* **2023**, *13*, 7627. https://doi.org/10.3390/app13137627

Academic Editor: Giacomo Fiumara

Received: 17 May 2023
Revised: 20 June 2023
Accepted: 26 June 2023
Published: 28 June 2023

1. Introduction

Recommendation systems play a critical role in accurately recommending items or content that match users' preferences in various fields, such as news [1], e-commerce [2,3], video [4], online advertising [2], and so on. Traditional recommendation methods, such as collaborative filtering [5], KNN [6], and matrix factorization [7], use user-item interaction information, including clicks, follows, ratings, and purchase history, to find similar users or items for recommendation. However, these methods use static information that is difficult to capture the dynamic interests of users. Although matrix factorization technology was later proposed to mine users' potential interests, its performance in recommendation is limited.

In recent years, deep learning technology has been widely used in various fields, such as anomaly detection [8,9], data enhancement [9–11], and so on. In the recommendation system field, based on the consideration of users' long- and short-term interests [1–3,12–14] over time, the core principle is to model user interests based on the order of interacting items over a period of time. We can train long- and short-term interests separately based on

the length of the sequence data and combine them to make recommendations that balance between personalization and diversity.

Long-term interests: extracting stable interests from sequential data based on long-term interests has always been a research focus. A common solution is to learn longer user behavior sequences as much as possible and store user interest features offline. DIN [2] considers that each user's attention to the target item should be different and proposes a model based on attention mechanism, which uses target objects and historical sequences to calculate attention scores to update sequence information. MIMN [15] decouples user interest memory storage units from recommendation modules. Long-term user interests do not need to change in a short time, and offline storage can remember longer user sequence data. SIM [16] proposes a method for quickly retrieving user behavior memory sequences throughout the user's life cycle, and improves the memory sequence to tens of thousands. SURGE [17] integrates different types of preferences in long-term user behaviors into clusters in the graph. Although longer sequence data can extract more stable user interests, there are also problems with difficult updates, difficult training convergence, and high data requirements.

Short-term interests: deep learning models based on time-series analysis have been found to be effective for modeling short-term user interests [3,12,14,18,19]. Several notable models, including GRU4REC [14] and DIEN [3], have integrated recurrent neural networks and attention mechanisms to improve recommendation accuracy. More recent models such as SDM [12] and CGNN-MHSA-AR [20] have further enhanced these approaches by incorporating multi-dimensional information. However, short-term interest models have limitations in dealing with noisy and incomplete data, which can lead to bias in recommendation results when coordinated with long-term interest models.

LS-term interests: typically, recommendation systems rely on either long-term or short-term benefits to generate recommendations. However, considering only one side interest will lose the platform and user experience [21]. Recent methods [1,12,18,21–24] have proposed solutions to this problem by dividing the model into two parts that separately model the user's long-term and short-term interests. The final recommendation is then based on a fusion of these two models. The advantage of hybrid recommendation lies in its ability to combine the strengths and weaknesses of different recommendation algorithms and applicable scenarios, and then choose the most suitable ratio to fuse multiple algorithm recommendations based on real-time data. Current feature combination methods can be divided into three categories: summation [1], concatenation [24,25], and weighted fusion [12,26]. However, these methods mainly consider the overall relationship between independent features, even the latest practical industrial methods. When it comes to the long-short interest model, it ignores the relationship between aspect-level interest and short-term interest in long-term interest features. Therefore, we explicitly partition the long-term interest features and use similarity and contrast loss to compensate for the shortcomings of previous methods. In practical applications, more side information is often introduced to enhance user and item features and reduce the bias of fusion features [27–29]. However, less research has been conducted on the enhancement of interest features. To address this gap, we specifically designed a novel fusion enhancement module, and the final experimental results demonstrate the effectiveness of our method.

In the following text, we will answer the following three main questions:

Q1. Why train the long and short interest models separately?
Q2. How does the integration module correct for interest bias?
Q3. What is the versatility of our approach?

In conclusion, the starting point of this paper is to connect the independent long and short interest models and correct the deviation between each other to improve the performance of the models. The main contributions of this paper are summarized as follows:

1. A plug-and-play user long and short interest fusion module is designed, which can effectively and quickly fuse long and short interests using the shared attention mechanism, thus improving the model accuracy;

2. The sources of interest bias are analyzed experimentally, and an improved ternary contrast loss function is introduced to accelerate the convergence of the model by using the bias between features as the index of the loss function;
3. The effectiveness and generality of our proposed method is demonstrated by combining and experimenting several different long and short interest models on several different domain datasets with data of different sequence lengths as input.

2. Related Work

2.1. Entanglement Training

The long- and short-term user interest model is a crucial component of recommender systems. It involves two stages: entangling long- and short-term interests for joint training and decoupling them to enhance interpretability and reduce confusion. Various models have been proposed for both stages, each with distinctive characteristics. For the entangling stage, SASrec [30], LUTUR [1], RCNN [18], and DIEN [3] are some of the popular models that employ complex neural network architectures to model the joint distribution of long- and short-term interests. SASrec uses the Markov chain assumption to pass sequence data through an attention network layer and a feed-forward network layer to predict the action probability of the last click. LUTUR uses user ID to capture long-term interests and long-term data to initialize short-term interest models. RCNN models long-term interest preferences using RNN and short-term interests using CNN on the hidden state. DIEN adds an improved GRU network layer with better temporal sensitivity to the lower layer of the attention mechanism network.

2.2. Disentanglement Training

To address the problem of poor model interpretability and confusion of captured user interests, researchers have proposed decoupling methods to separate long- and short-term user interests. PLASTIC [19], GNewsRec [24], SDM [12], MA-GNN [31], and CLSR [21] are some of the models that have employed different approaches for this stage. PLASTIC uses a combination of multi-factorization machine and RNN to capture long- and short-term interests separately and uses reinforcement learning to dynamically train fusion weight coefficients. GNewsRec constructs a relationship graph based on user-item interaction data to mine long-term interests and trains short-term interest using LSTM and attention mechanism. SDM trains long-term interests using item labels and user attributes and uses LSTM and multi-headed attention mechanism to obtain short-term interests. MA-GNN uses GNN to model short-term interests and a multi-headed attention mechanism to model long-term interests. CLSR proposes to use self-supervised signals to separate long- and short-term interests to reduce the biased expression of user interests.

2.3. Feature Fusion

Feature fusion refers to the fusion of information from different feature sources into a unified feature vector to improve the performance and robustness of a model in various tasks. It includes feature-level fusion, modality-level fusion and feature correction. For example, in image recognition tasks, features such as different scales and colors can be extracted from the original image and then fused to form a new feature vector. Kaiming He et al. [32] proposed a deep residual structure to fuse different levels of features across scales, and Nitish Srivastava et al. [11] proposed a model using a deep Boltzmann machine for training features from visual, text and speech data from different modalities for cross-modal learning and feature fusion, and Albert Gordo et al. [33] proposed a convolutional neural network based image retrieval method that integrates feature correction mechanisms such as local voting and pooling in the network structure, which is used to improve retrieval robustness and accuracy. Regardless, feature enhancement is a not an indispensable part of feature fusion, which refers to the reconstruction or expansion of existing features, such as data enhancement [10], transformation of features from different sources [11], gradient selection [9] and feature reconstruction [34]. We borrowed the feature fusion ideas from the

above papers and proposed the use of attention mechanism and global average residual structure to fuse long and short interest features, and proved the effectiveness of our method in the final feature fusion comparison experiments.

3. Our Approach

The recommendation system operates by forecasting a user's future actions based on their past behavior. We denote the set of all users using the mathematical symbol U, and the set of all items using the symbol I. The user interaction sequence has a fixed length of N, denoted by $X_u = x_1^u, x_2^u, \ldots, x_N^u$, which represents the sequence of interaction items ordered in time by user u. The time series model utilized in this study learns both the user's long-term and short-term interests from the first $N-1$ items of the interaction sequence, as well as the user's ID data, to predict the probability of clicking on the Nth item in the prediction model. To incorporate the latest sequence information, we draw inspiration from the paper [23] and extract the most recent consecutive segment of sequence, $\tilde{N}$, as input to the short-term interest model. Embedding techniques have been widely utilized in data processing phases of recommender systems. In our study, we apply the embedding technique to represent user discrete data and item discrete data, where each user dimension is represented by u and each item dimension is represented by i. This process can be expressed as Equation (1).

$$p^{u \in U}, q^{i \in X} = Embed(U), Embed(X) \tag{1}$$

$p^{u \in U}$ is the embedding vector of user u, $q^{i \in X}$ is the embedding vector of item i

3.1. LS Interest Modeling

The behavioral motivation of users is often complex. Even users with stable long-term interests may occasionally click on popular items unrelated to their interests. As a result, short-term user interests are considered unstable and contain a significant amount of noise data. Zheng et al. [21] have emphasized that accurately predicting a user's long-term interests can increase click rates, while accurate short-term interest prediction is critical for platform profitability and user experience. In sequence modeling, RNN structures are frequently employed because they take into account the time factor of sequence data. Although a user's interests may differ significantly in a short period of time, they are still linked, and the RNN structure captures the weight of the relationship between items before and after.

Table 1 displays the prediction results of our DIN model on different types of users with varying sequence lengths, trained on Amazon Electric data. The findings indicate that when users have few click sequences, the model recommends more diverse item types and popular items to explore their interests, resulting in fluctuating hit rates. As the number of user interactions increases, the proportion of recommended item types matching the user's click history increases, and the proportion of popular items decreases, indicating that more data can extract stable user interests. For users with diverse interests, the long-sequence model has minimal impact on the diversity of recommended items, while users with narrower interests experience the opposite effect, with longer models resulting in greater bias. A long time span of item interaction can be fatal to a short-term sequence model, leading to recommended popular items to compensate for the deficiency. However, the long-term interest model has minimal effect on short-term interest prediction and should be used judiciously to avoid impairing the prediction accuracy of the model.

In summary, we can answer the first question (Q1): long sequence models are more suitable for users with diverse interests, while short sequence models are better for users with less diverse interests. Long sequence models prioritize diversity over accuracy, while short sequence models prioritize accuracy over diversity. To overcome these challenges, we propose decoupling the short-length interest model, redesigning the feature enhancement module, and introducing a loss function to account for differences in interests.

Table 1. Table of prediction results of different length sequence models for different users. "Class" represents the number of categories, "popu" represents the proportion of popular items, and "cross" represents the proportion of recommended results crossed with historical items categories.

DIN Model	User Histories	Sequences		Categories		Time Span	
Length	**Thread**	**≤5**	**≥50**	**≤2**	**≥20**	**≤1**	**≥3300**
10	class	31	34	2	67	41	39
	popu	0.016	0.005	0.000	0.008	0.010	0.005
	cross	0.027	0.333	0.384	0.176	0.015	0.034
20	class	67	87	52	41	47	64
	popu	0.010	0.008	0.004	0.016	0.032	0.012
	cross	0.027	0.276	0.014	0.208	0.040	0.120
40	class	145	165	164	144	201	193
	popu	0.014	0.005	0.007	0.010	0.003	0.005
	cross	0.064	0.202	0.014	0.096	0.026	0.082
50	class	189	98	202	89	141	133
	popu	0.018	0.001	0.008	0.010	0.009	0.012
	cross	0.025	0.330	0.011	0.173	0.028	0.085

3.2. Fusion Module

The fusion module workflow is detailed in Figure 1. Firstly, the user interest features from the long-term interest model are divided into four equal parts. This was inspired by Feng et al. [35], who suggested that user behavior consists of short sessions with varying interest points. After segmentation, multiple local features are cross-fused with short-term interest features through a shared attention mechanism, capturing the correlation between short-term features and local features. The aim is to identify the correlation between short-term feature fusion and local features to account for differences between long-term interests and short-term interests.

$$a_i = \frac{exp(f(p_s^u, p_l^{u,i}))}{\sum_{j=1}^{n} exp(f(p_s^u, p_l^{u,j}))} \tag{2}$$

$$att(p_l^{u,i}, p_s^u) = a_i p_l^{u,i} \tag{3}$$

$p_l^{u,i}$ refers to the ith feature part equally divided into n parts. f is the fusion enhancement function. a_i calculates the attention fraction between the short-term and local features. The att is obtained as a result of fusion of individual local features with short-term features.

The short-and long-term interest model incorporates an attention mechanism that calculates weights between historical serial goods and target goods, providing a direct integration of global information while ignoring one-sided interest. Meanwhile, the attention module in the fusion module leverages shared parameters to calculate weights between each local feature and short-term features, thereby avoiding an overly large number of parameters. While this approach can effectively integrate local features with short-term features and obtain part of the global long-term interest features, it still lacks the ability to fully integrate global information. To address this limitation, we propose the use of an MLP and sigmoid to calculate the weights of each interaction feature, and then stitch all interaction features together as user interest features.

Multiple output feature vectors contain the evolution direction of user interest, and effectively integrating them is a crucial problem. Inspired by the channel attention mechanism in SE-Net for images, we propose a lightweight fusion unit that treats each feature as a channel. This unit dynamically adjusts the weights using the attention mechanism. Unlike the SDM model, our fusion unit is computationally efficient and adheres to the plug-and-play principle. The fusion process can be expressed as follows:

$$p_{out} = o_u + o_u \cdot \sigma(f(maxpool(p^u) + avgpool(p^u))) \tag{4}$$

maxpool and *avgpool* represent maximum pooling and average pooling, respectively, f is the multilayer perceptron, σ is the sigmoid function, and o_u is the output result set interaction feature of the attention mechanism. The output p^u is obtained by splicing the output of a multi-head attention mechanism. The fully connected layer f is followed by a sigmoid function.

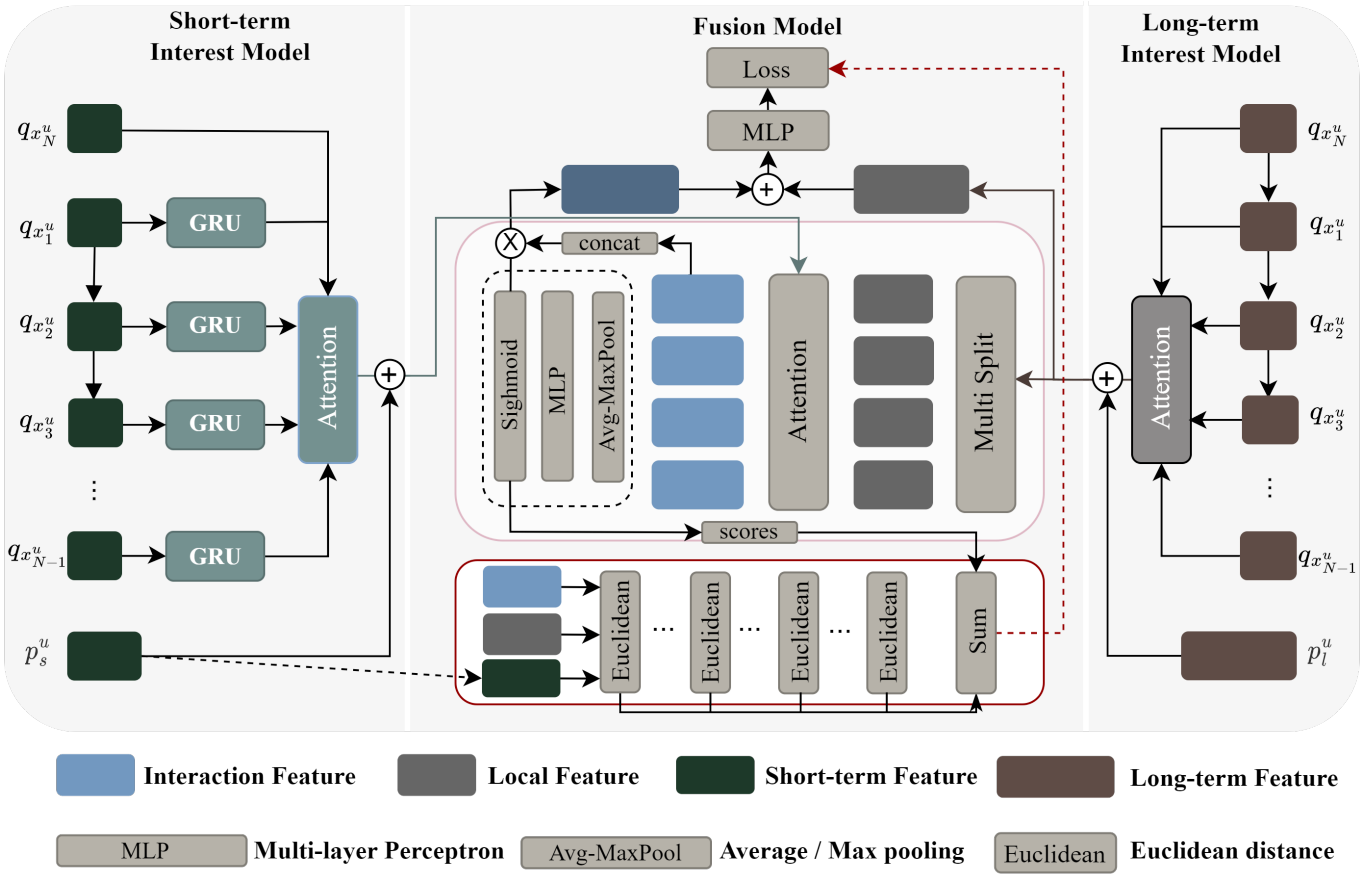

Figure 1. The fusion module is the main component of our model, which requires both long-term and short-term user interest features as input, and the user interest model can be replaced according to actual needs. GRU is a variant of recurrent neural networks, but with fewer parameters, which alleviates the gradient problem in long-term memory and back propagation. attention is the attention mechanism. Multi split splits longer long-term interest features into multiple copies.

Based on the fusion module's content, we will address the second question (Q2): although p_l^u and p_s^u are explicitly separated, the disentanglement between them cannot be fully guaranteed, and there is no corresponding label to supervise the learning of their differences. To address this limitation, we propose a new contrast learning task that dynamically adjusts the similarity between the interactive feature and the original feature. The recommendation for different types of users varies in the ratio of long-term and short-term interest. For example, in the case of cold startup, local features should be more similar to short-term features, while regular users require the opposite. The goal of this task is to minimize the similarity between features and maximize features with large differences.

3.3. Dynamic Decouple Loss

The fusion module incorporates an attention mechanism to integrate local features and short-term features. The binary loss function adjusts the difference between the output and input. However, the model may face challenges with slow convergence and instability due to large model parameters and diverse user types. To overcome these issues, we use the attention scores obtained from the attention mechanism to represent the contribution

of each local feature in the interaction feature of the output. We introduce a ternary comparison loss function that uses the feature similarity differences as an index to sum to the final loss value. The final loss function is reformulated as follows:

$$Loss = L_{BCE} + \sum_{i=1}^{n} a_i L_{tri,i} \tag{5}$$

The loss function used in this study is a combination of binary cross entropy loss (L_{BCE}) and triplet loss (L_{tri}), where a represents the attention fraction. The traditional triplet loss function is defined as follows:

$$L = \sum_{i=1}^{n} \max(d(o_u^i, p_l^{u,i}) - d(o_u^i, p_s) + margin, 0) \tag{6}$$

$d(o_u^i, p_l^{u,i})$ computes the Euclidean distance between interaction features and local features, $d(o_u^i, p_s)$ counts the Euclidean distance between interaction features and short-term features, and margin is a custom constant. However, the default setting is dominated by local features, which may reduce the weight of short-term features and lead to biased results. This shortcoming can be attributed to the traditional triplet loss function, which calculates the difference between the anchor sample and the positive and negative samples, and the result is mandatory. To account for varying user preferences in prediction results, constraints are incorporated into the loss function and dynamic parameters are introduced through attention scores. The formula for this is as follows:

$$sim_{min}, sim_{max} = sorted(s(o^{u,i}, p_l^{u,i}), s(o_u^i, p_s^u)) \tag{7}$$

$$L_{tri} = max(\frac{1}{n} \sum_{i=1}^{n} a_i(1 - sim_{max}) + sim_{min} + margin, 0)) \tag{8}$$

For each set of triplet features, we sort the similarity between the remaining interaction features and the two features. The similarity is calculated using the formula s, which is the reciprocal of the Euclidean distance between the features. The group of features with the highest similarity is considered dominant. n is the number of interaction features and a_i represents the attention weights of each interaction feature. $1 - sim_{max}$ and sim_{min} will gradually increase the distance between dissimilar features during the iterative learning process, which can serve to regulate the ratio of long-term interest to short-term interest contribution to the final result.

To summarize, we propose an enhanced triplet loss function that incorporates an attention score to regulate the influence of local interests, a similarity term to measure the difference between local and short-term interests, and an indexing mechanism to incorporate these factors into the overall loss function. In our comparative experiments, we demonstrate that this loss function efficiently accelerates model convergence.

4. Experiments

In this section, we present a comprehensive overview of our experimental process and results. First, we describe the multiple field datasets used in our experiments and the user interest model employed for comparison. Next, we explore and analyze the performance of our fusion module by addressing three main research questions:

- **RQ1:** How do our modules perform in practice?
- **RQ2:** What is the individual contribution of each component in our model?
- **RQ3:** Can our model effectively handle the complexity of sequence data with varying lengths across different scenarios?

We provide detailed answers to these questions and present our experimental findings in a structured and rigorous manner.

4.1. Datasets and Experimental Setup

To simulate various recommendation scenarios, we have selected four publicly available datasets in the domain. Table 2 provides detailed information about each dataset. For instance, the Amazon and Douban datasets have opposite user interest breadth, the Taobao dataset has longer user click sequences and richer data, and the Yelp dataset has fewer user clicks, which can effectively simulate the cold-start situation. In order to measure the effect of the model, three metrics are selected, namely Accuracy (ACC), Area Under Curve (AUC) and F1-score (F1). The higher the number of the three, the better the effect.

Table 2. The number of users and items in the four datasets. Average click sequence indicates the average number of items clicked by users; average click categories indicates the average number of items clicked by users; average click time span indicates the average time span clicked by users.

Datasets	Users	Items	Average Click Sequence	Average Click Categories	Average Click Time Span
Amazon	19,240	63,001	8.780	8.780	1.050
Taobao	104,693	1,592,919	102.170	24.385	0.009
Douban	52,539	140,502	6.321	2.726	0.418
Yelp	1,542,656	209,393	3.956	3.025	0.324

4.2. Competitors

In the experiment, the long and short interest model and the mixed training model were compared, respectively, including DIN [2], DIEN [3], NARM [13], PACA [36], RCNN [18], SLiRec [37], FMLP [38], CLSR [21], GRU4REC [14], LUTUR [1], CASER [39]. Among them, LUTUR, SLiRec and CLSR are hybrid models, which also share our model architecture.

4.3. Experimental Metrics

The experimental metrics include Accuracy, AUC and F1. The CTR task is a binary classification task, which represents the proportion of the total number of correct model predictions, and the accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{N} \tag{9}$$

TP, TN denotes the number of positive samples predicted correctly and the number of negative samples predicted correctly, respectively, and N represents the total number of samples.

AUC is a measure of the ranking performance of a recommender system. The ROC curve is a curve with FPR (False Positive Rate) as the horizontal axis and TPR (True Positive Rate) as the vertical axis, and AUC is the area under the ROC curve. The AUC is calculated as follows:

$$AUC = \frac{\sum I(p_{positive}, p_{negative})}{P \times N} \tag{10}$$

$$I(p_{positive}, p_{negitive}) = \begin{cases} 1, & p_{positive} > p_{negitive} \\ 0.5, & p_{positive} = p_{negitive} \\ 0, & p_{positive} < p_{negitive} \end{cases} \tag{11}$$

P is the number of positive samples, N is the number of negative samples, $p_{positive}$ is the positive sample prediction score, and $p_{negative}$ is the negative sample prediction score. The F1 score is a composite of accuracy and recall and can be calculated by the following formula:

$$F1 = 2 * \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

Precision and recall denote accuracy and recall, respectively. The F1 score is a combination of precision and recall, while the AUC score is a combination of TPR and FPR.

4.4. Overall Performance Comparison (RQ1)

Table 2 details the overall performance of all models on the four datasets, from which the following four results can be observed:

- **The overall performance of the short-term interest model is better.** From the overall results, the short-term interest model is better overall than the long-term interest model because it can capture the actual by-sequence information of user interaction well. The results from DIN, PACA, and RCNN show that the length interest features are more informative, and if more effective methods can be used to fit the long-term interest features, the improvement of the models' effectiveness is significant.
- **The long-term interest model has the advantage of playing in two situations: a large variety of products and a long time span of user clicks.** Although the short-term interest model is generally better, the short-term model does not necessarily outperform the long-term model for the two cases of large variety of items and long click time span. The RCNN and CASER models also use CNN networks, but CASER is slightly less effective than the RCNN model, but there is a large gap between them and FMLP, which indicates that the user's pre-click data helps the model to capture long-term interest, but the short-term interest weight is generally larger than the long-term interest weight. The cold start problem is a difficult problem for both models, and the best results for the Yelp dataset are lower than the remaining three datasets, and there is a large gap between the long-term and short-term models, which verifies that fitting to the length of the sequence data and effective extraction of the sequence data are the keys to improve the performance.
- **Joint modeling of long and short interests is a generally effective approach.** Joint modeling of models somewhat alleviates the poor performance of independent models in cold starts, large span of user clicks, and many types of user clicks, but it is not always effective. NARM, LUTUTR, and SLIREC are trained by entangling long and short interests with each other, which somewhat increases the redundancy of the models, and the performance of SLIREC in ACC and F1 which are inadequate. In contrast, CLSR decouples the calculation of long and short interests, and the adaptive weighting fuses the long and short interest weights, which alleviates the training volume of the model, which makes up for the lack of realizations of SLIREC on Taobao.
- **Contrast learning and feature enhancement can effectively improve model performance.** Our model differs slightly from the best comparison model CLSR on the data-rich dataset Taobao, but improves AUC by almost 0.01 on Douban, which has a much smaller variety of products, and for the cold-start case, our model leads CLSR in all metrics, and has into 1.3% improvement on the long time span Taobao dataset. The results from Table 3 also show that CDF-LS achieves a better balance in terms of computational cost. These all validate the effectiveness of contrast loss and feature enhancement.

Table 3. Comparison of computational cost of LS-term model on Taobao and Yelp datasets. Params represents the number of parameters of the model, FLOPs represents the floating point computation of the model, and the unit G = 1,000,000 for the above two metrics. Throughput represents the throughput of the model, and the unit Samples/s represents the maximum number of samples processed per second.

Model	Taobao			Yelp		
	Params (G)	FLOPs (G)	Throughput (Samples/s)	Params (G)	FLOPs (G)	Throughput (Samples/s)
LUTUR	29.02	1.28	250,246.81	52.79	0.64	248,976.63
SLIREC	28.98	2.37	64,913.81	52.75	2.54	70,198.13
CLSR	29.0	1.62	98,586.71	52.77	1.62	102,487.96
CDF-LS	31.57	1.20	79,161.06	54.84	1.21	84,611.67

Figure 2 depicts the decline diagram of training loss value of four mixed models on all datasets. The zigzag curve is due to mild overfitting. CDF-LS can rapidly decline iteration according to the difference index between features in the early stage of training, and it is also applicable in difficult recommendation scenarios. This verifies that the introduction of contrast loss between short and long features is effective.

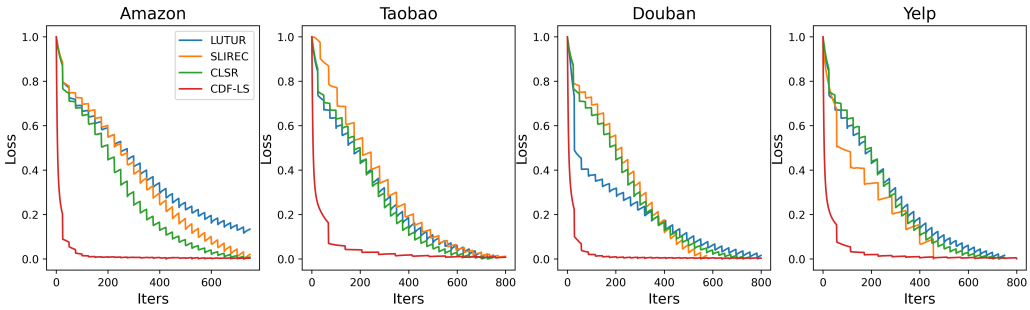

Figure 2. LUTUR, SLIREC, CLSR and CDF-LS on four datasets with loss descent plots.

4.5. Results of Ablation Experiments(RQ2)

4.5.1. Contrast Loss

Contrast learning facilitates model fitting and interpretability by learning the similarity between long and short features, combined with dynamic weight assignment. We conducted ablation experiments on contrast loss, we compared the performance of two models LUTUR, SLIREC with and without contrast loss, and replaced the contrast loss function of CLSR. The experimental effects of dynamic assignment of weights were also compared, and Table 4 shows all the comparison results in detail.

Table 4. The red font indicates the best results, and the LS-term models all use the RCNN and FMLP user vector weights for the best results.

Category		Long-Term				Short-Term				LS-Term			
Dataset	**Model**	**DIN**	**PACA**	**NARM**	**RCNN**	**CASER**	**GRU4REC**	**DIEN**	**FMLP**	**LUTUR**	**SLIREC**	**CLSR**	**CDF-LS**
Amazon	ACC	0.7148	0.7057	0.7364	0.7698	0.7665	0.7747	0.7805	0.7881	0.7924	0.8002	0.8046	0.8014
	AUC	0.8095	0.8154	0.8340	0.8465	0.8415	0.8574	0.8636	0.8716	0.8786	0.8773	0.8857	0.8824
	F1	0.7167	0.7096	0.7310	0.7668	0.7633	0.7789	0.7774	0.7830	0.7911	0.7973	0.8077	0.8096
Taobao	ACC	0.6895	0.7033	0.7021	0.7174	0.7122	0.7189	0.7296	0.7374	0.7561	0.7543	0.7607	0.7724
	AUC	0.7624	0.7761	0.7723	0.8084	0.7096	0.8087	0.8390	0.8389	0.8391	0.8318	0.8388	0.8392
	F1	0.6941	0.7097	0.7029	0.7218	0.7145	0.7187	0.7279	0.7448	0.7719	0.7693	0.7691	0.7710
Douban	ACC	0.8549	0.8440	0.8699	0.8740	0.8710	0.8811	0.8951	0.8941	0.9066	0.9018	0.9132	0.9134
	AUC	0.8974	0.8838	0.9174	0.9281	0.9204	0.9168	0.9286	0.9378	0.9454	0.9463	0.9481	0.9567
	F1	0.8577	0.8369	0.8787	0.8817	0.8763	0.8837	0.8938	0.8869	0.9010	0.9086	0.9047	0.9079
Yelp	ACC	0.6566	0.6610	0.6834	0.7093	0.7307	0.7399	0.7524	0.7580	0.7630	0.7719	0.7818	0.7907
	AUC	0.7035	0.7271	0.7319	0.7504	0.7809	0.7807	0.8036	0.8073	0.8098	0.8122	0.8164	0.8204
	F1	0.6589	0.6683	0.6803	0.7017	0.7393	0.7271	0.7513	0.7515	0.7641	0.7783	0.7730	0.7943

As can be seen from Table 5, the contrast loss adaptation can effectively improve the performance of the LS-term model and is applicable to difficult recommendation scenarios. The assignment of dynamic weights can serve to effectively assign the weights of long and short features. We replaced the contrast loss in the CLSR model, which outperformed our model by 0.35% on the Amazon dataset and by 0.13% in the cold start case. This indicates that the decoupling framework in CLSR is applicable to most domains and can effectively improve model performance, and that self-supervised decoupling on how to do so effectively is a future direction for improvement.

Table 5. Contrast stands for using the comparison loss function, Weights stands for using dynamic weights to update the length of interest, and CLSR itself has dynamic weights, so only the new loss function is compared. The gray numbers indicate the increase relative to the original model results.

Model	Contrast	Weights	Amazon		Yelp	
			AUC	F1	AUC	F1
LUTUR	✔		0.8804 + 0.0018	0.8013 + 0.0102	0.8149 + 0.0051	0.7736 + 0.0095
	✔	✔	0.8820 + 0.0034	0.8049 + 0.0136	0.8157 + 0.0059	0.7746 + 0.0105
SLIREC	✔		0.8853 + 0.0080	0.8051 + 0.0078	0.8166 + 0.0044	0.7804 + 0.0021
	✔	✔	0.8861 + 0.0088	0.8064 + 0.0091	0.8171 + 0.0049	0.7847 + 0.0063
CLSR	✔	✔	0.8859 + 0.0002	0.8101 + 0.0024	0.8217 + 0.0053	0.7881 + 0.0151

4.5.2. Feature Fusion

To illustrate that our model is suitable for most short and long interest models and easy to debug and add, we combined four sets of short and long interest models and conducted exploratory experiments on more difficult prediction datasets. Table 6 shows the results of the comparison experiments in detail, from which we can find that for DIN and CASER, the worst-performing group of individual models, the performance improvement is obvious, especially for DIN, which improves the AUC metric on Yelp by 15.4%. This is attributed to the high-weighted short-term features and the effective fusion method. As for the best performance combination RCNN and FMLP, its performance approaches CLSR on Amazon and even surpasses CLSR by 0.02% on Yelp, which strongly validates that there is still room for progress in the extraction of user interest, the enhancement of features with long and short term interest can effectively play the performance of the respective models, and the effectiveness of our module for feature fusion.

Table 6. Four combinatorial models comparing the simple splicing method with our fusion method. The gray numbers represent the relative increase compared to the results of the original model. The maximum increase reached 0.0314.

Model	Fusion	Amazon		Yelp	
		AUC	F1	AUC	F1
DIN + CASER	✗	0.8640 + 0.0225	0.7811 + 0.0178	0.7988 + 0.0179	0.7439 + 0.0046
	✔	0.8715 + 0.0300	0.7866 + 0.0233	0.8123 + 0.0314	0.7501 + 0.0118
NARM + DIEN	✗	0.8695 + 0.0059	0.7854 + 0.0008	0.8101 + 0.0065	0.7613 + 0.0100
	✔	0.8763 + 0.0127	0.7878 + 0.0104	0.8160 + 0.0124	0.7729 + 0.0216
PACA + GRU4REC	✗	0.8641 + 0.0067	0.7811 + 0.0022	0.8010 + 0.0203	0.7684 + 0.0413
	✔	0.8720 + 0.0146	0.7869 + 0.0080	0.8103 + 0.0296	0.7700 + 0.0429
RCNN + FMLP	✗	0.8740 + 0.0024	0.7846 + 0.0016	0.8074 + 0.0001	0.7517 + 0.0002
	✔	**0.8809** + 0.0093	**0.7903** + 0.0073	**0.8166** + 0.0093	**0.7624** + 0.0109

Based on the findings presented in Tables 1 and 3, we can address the third research question on universality (Q3) raised in the introduction. Our results indicate that the CBF-LS approach can effectively be applied across multiple datasets and models in various fields. This is achieved by dynamically adjusting the contribution ratio of short and long-term interest features based on the attention weight score and leveraging the differences between features in the process of reverse optimization. The clear theoretical foundation of our approach can be easily extended to other domains, and the faster iterations result in reduced time and costs. Our results demonstrate that the wide applicability and training efficiency of our approach support its universality.

4.6. Robustness Test Experimental Results (RQ3)

To demonstrate the versatility of our designed module for various models and scenarios, we conducted experiments on two challenging recommendation datasets using different models with varying sequence lengths for training the long and short interest models. The fusion module was utilized for pairing the models. Table 7 shows that the combination of the longest sequence model and the shortest sequence model outperforms

the baseline model. However, when using similar models, the results were poorer. This is mainly due to the lack of compensating measures in similar models, which prevents the optimization of contrast loss based on differences between the two models. This can even result in interference, as evidenced by smaller results for 20 and 30 sequence lengths compared to 40 and 50 sequence lengths. Our tabular results demonstrate that our module significantly improves model performance in cases where there are large differences between long and short interest models. However, it is less effective for similar models, highlighting the limitations of contrast learning.

Table 7. Underline represents the baseline model for the combination, bolded represents the best result. The longer sequences in each row are used to train the long-term interest model, and the shorter sequences are used to train the short-term interest model.

Model	Sequences				Amazon		Yelp	
	20	30	40	50	AUC	F1	AUC	F1
DIN + CASER	✔	✔			0.8427	0.7696	0.7909	0.7415
	✔		✔		0.8715	0.7866	0.8123	0.7501
	✔			✔	**0.8753**	**0.7917**	**0.8149**	**0.7553**
		✔	✔		0.8631	0.7898	0.8077	0.7430
		✔		✔	0.8649	0.7910	0.8083	0.7431
			✔	✔	0.8548	0.7704	0.7905	0.7257
FMLP + RCNN	✔	✔			0.8701	0.7792	0.8075	0.7547
	✔		✔		0.8809	0.7903	0.8166	0.7724
	✔			✔	**0.8847**	**0.7953**	**0.8189**	**0.7764**
		✔	✔		0.8703	0.7765	0.8086	0.7561
		✔		✔	0.8761	0.7846	0.8137	0.7704
			✔	✔	0.8602	0.7739	0.8060	0.7433

5. Conclusions

In this paper, we present an experimental analysis of the differences in recommendation outcomes between long and short user interest models. To address this gap, we propose a novel interest fusion module that assimilates both long and short user interests. Our module is designed as a plug-and-play feature that employs a shared attention mechanism to fuse the long and short interest features. Additionally, we incorporate a comparison task to assess the similarity between the two interest representations and the fused interest representations. Finally, we evaluated the effectiveness of our proposed module for recommendation results on multiple datasets.

Author Contributions: Conceptualization, K.L. and W.W.; methodology, K.L. and W.W.; software, W.W.; validation, K.L. and W.W.; formal analysis, R.W.; investigation, X.C.; resources, L.Z.; data curation, X.Y.; writing—original draft preparation, W.W.; writing—review and editing, K.L. and X.L.; visualization, R.W., X.C. and X.L.; supervision, L.Z. and X.Y.; project administration, W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by National Natural Science Foundation of China (No. 62202390), Science and Technology Fund of Sichuan Province (No. 2022NSFSC0556), and the Opening Project of Lab of Security Insurance of Cyberspace, Sichuan Province.

Data Availability Statement: Amazon: https://nijianmo.github.io/amazon/index.html; Taobao: https://tianchi.aliyun.com/dataset/649; Douban: https://pan.baidu.com/s/1BYEBk1nkX53SucGcldgFnw?pwd=1234; Yelp: https://www.yelp.com/dataset.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. An, M.; Wu, F.; Wu, C.; Zhang, K.; Liu, Z.; Xie, X. Neural news recommendation with long-and short-term user representations. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 336–345.
2. Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; Gai, K. Deep interest network for click-through rate prediction. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1059–1068.
3. Zhou, G.; Mou, N.; Fan, Y.; Pi, Q.; Bian, W.; Zhou, C.; Zhu, X.; Gai, K. Deep interest evolution network for click-through rate prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 5941–5948.
4. Covington, P.; Adams, J.; Sargin, E. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 191–198.
5. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
6. Xu, G.; Wu, Z.; Zhang, Y.; Cao, J. Social networking meets recommender systems: Survey. *Int. J. Soc. Netw. Min.* **2015**, *2*, 64–100. [CrossRef]
7. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
8. Chen, P.; Liu, H.; Xin, R.; Carval, T.; Zhao, J.; Xia, Y.; Zhao, Z. Effectively detecting operational anomalies in large-scale IoT data infrastructures by using a gan-based predictive model. *Comput. J.* **2022**, *65*, 2909–2925. [CrossRef]
9. Liu, M.; Deng, J.; Yang, M.; Cheng, X.; Liu, N.; Liu, M.; Wang, X. Cost Ensemble with Gradient Selecting for GANs. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI, Vienna, Austria, 23–29 July 2022; pp. 1194–1200. [CrossRef]
10. Xie, T.; Cheng, X.; Wang, X.; Liu, M.; Deng, J.; Zhou, T.; Liu, M. Cut-thumbnail: A novel data augmentation for convolutional neural network. In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021; pp. 1627–1635.
11. Li, N.; Liu, Y.; Wu, Y.; Liu, S.; Zhao, S.; Liu, M. Robutrans: A robust transformer-based text-to-speech model. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8228–8235.
12. Lv, F.; Jin, T.; Yu, C.; Sun, F.; Lin, Q.; Yang, K.; Ng, W. SDM: Sequential deep matching model for online large-scale recommender system. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2635–2643.
13. Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; Ma, J. Neural attentive session-based recommendation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1419–1428.
14. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. *arXiv* **2015**, arXiv:1511.06939.
15. Pi, Q.; Bian, W.; Zhou, G.; Zhu, X.; Gai, K. Practice on long sequential user behavior modeling for click-through rate prediction. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2671–2679.
16. Pi, Q.; Zhou, G.; Zhang, Y.; Wang, Z.; Ren, L.; Fan, Y.; Zhu, X.; Gai, K. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 2685–2692.
17. Chang, J.; Gao, C.; Zheng, Y.; Hui, Y.; Niu, Y.; Song, Y.; Jin, D.; Li, Y. Sequential recommendation with graph neural networks. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 11–15 July 2021; pp. 378–387.
18. Xu, C.; Zhao, P.; Liu, Y.; Xu, J.; Sheng, V.S.S.; Cui, Z.; Zhou, X.; Xiong, H. Recurrent convolutional neural network for sequential recommendation. In Proceedings of the The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3398–3404.
19. Zhao, W.; Wang, B.; Ye, J.; Gao, Y.; Yang, M.; Chen, X. PLASTIC: Prioritize Long and Short-term Information in Top-n Recommendation using Adversarial Training. *IJCAI* **2018**, 3676–3682. [CrossRef]
20. Song, Y.; Xin, R.; Chen, P.; Zhang, R.; Chen, J.; Zhao, Z. Identifying performance anomalies in fluctuating cloud environments: A robust correlative-GNN-based explainable approach. *Future Gener. Comput. Syst.* **2023**, *145*, 77–86. [CrossRef]
21. Zheng, Y.; Gao, C.; Li, X.; He, X.; Li, Y.; Jin, D. Disentangling user interest and conformity for recommendation with causal embedding. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 2980–2991.
22. Dong, D.; Zheng, X.; Zhang, R.; Wang, Y. Recurrent Collaborative Filtering for Unifying General and Sequential Recommender. *IJCAI* **2018**, 3350–3356. [CrossRef]
23. Bai, T.; Du, P.; Zhao, W.X.; Wen, J.R.; Nie, J.Y. A long-short demands-aware model for next-item recommendation. *arXiv* **2019**, arXiv:1903.00066.
24. Hu, L.; Li, C.; Shi, C.; Yang, C.; Shao, C. Graph neural news recommendation with long-term and short-term interest modeling. *Inf. Process. Manag.* **2020**, *57*, 102142. [CrossRef]

25. Ma, M.; Wang, G.; Fan, T. Improved DeepFM Recommendation Algorithm Incorporating Deep Feature Extraction. *Appl. Sci.* **2022**, *12*, 1992. [CrossRef]
26. Shao, J.; Qin, J.; Zeng, W.; Zheng, J. Multipointer Coattention Recommendation with Gated Neural Fusion between ID Embedding and Reviews. *Appl. Sci.* **2022**, *12*, 594. [CrossRef]
27. Ho, T.L.; Le, A.C.; Vu, D.H. Multiview Fusion Using Transformer Model for Recommender Systems: Integrating the Utility Matrix and Textual Sources. *Appl. Sci.* **2023**, *13*, 6324. [CrossRef]
28. Zuo, Y.; Liu, S.; Zhou, Y.; Liu, H. TRAL: A Tag-Aware Recommendation Algorithm Based on Attention Learning. *Appl. Sci.* **2023**, *13*, 814. [CrossRef]
29. Liang, N.; Zheng, H.T.; Chen, J.Y.; Sangaiah, A.K.; Zhao, C.Z. TRSDL: Tag-Aware Recommender System Based on Deep Learning–Intelligent Computing Systems. *Appl. Sci.* **2018**, *8*, 799. [CrossRef]
30. Kang, W.C.; McAuley, J. Self-attentive sequential recommendation. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 197–206.
31. Ma, C.; Ma, L.; Zhang, Y.; Sun, J.; Liu, X.; Coates, M. Memory augmented graph neural networks for sequential recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 5045–5052.
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
33. Srivastava, N.; Salakhutdinov, R.R. Multimodal learning with deep boltzmann machines. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 2222–2230.
34. Volpi, R.; Morerio, P.; Savarese, S.; Murino, V. Adversarial feature augmentation for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5495–5504.
35. Feng, Y.; Lv, F.; Shen, W.; Wang, M.; Sun, F.; Zhu, Y.; Yang, K. Deep session interest network for click-through rate prediction. *arXiv* **2019**, arXiv:1905.06482.
36. Zhang, J.; Ma, C.; Mu, X.; Zhao, P.; Zhong, C.; Ruhan, A. Recurrent convolutional neural network for session-based recommendation. *Neurocomputing* **2021**, *437*, 157–167. [CrossRef]
37. Yu, Z.; Lian, J.; Mahmoody, A.; Liu, G.; Xie, X. Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation. *IJCAI* **2019**, *7*, 4213–4219.
38. Zhou, K.; Yu, H.; Zhao, W.X.; Wen, J.R. Filter-enhanced MLP is all you need for sequential recommendation. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 2388–2399.
39. Tang, J.; Wang, K. Personalized top-n sequential recommendation via convolutional sequence embedding. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Los Angeles, CA, USA, 5–9 February 2018; pp. 565–573.