applied sciences

MDPI

Article

DACSR: Decoupled-Aggregated End-to-End Calibrated Sequential Recommendation

Jiayi Chen [1], Wen Wu [1,2,*], Liye Shi [1], Yu Ji [1], Wenxin Hu [3], Xi Chen [2], Wei Zheng [4] and Liang He [1]

[1] School of Computer Science and Technology, East China Normal University, Shanghai 200062, China
[2] Shanghai Key Laboratory of Mental Health and Psychological Crisis Intervention, School of Psychology and Cognitive Science, East China Normal University, Shanghai 200062, China
[3] School of Data Science and Engineering, East China Normal University, Shanghai 200062, China
[4] Information Technology Services, East China Normal University, Shanghai 200062, China
* Correspondence: wwu@cc.ecnu.edu.cn

Abstract: Sequential recommendations have made great strides in accurately predicting the future behavior of users. However, seeking accuracy alone may bring side effects such as unfair and overspecialized recommendation results. In this work, we focus on the calibrated recommendations for sequential recommendation, which is connected to both fairness and diversity. On the one hand, it aims to provide fairer recommendations whose preference distributions are consistent with users' historical behaviors. On the other hand, it can improve the diversity of recommendations to a certain degree. But existing methods for calibration have mainly relied on the post-processing on the candidate lists, which require more computation time in generating recommendations. In addition, they fail to establish the relationship between accuracy and calibration, leading to the limitation of accuracy. To handle these problems, we propose an end-to-end framework to provide both accurate and calibrated recommendations for sequential recommendation. We design an objective function to calibrate the interests between recommendation lists and historical behaviors. We also provide distribution modification approaches to improve the diversity and mitigate the effect of imbalanced interests. In addition, we design a decoupled-aggregated model to improve the recommendation. The framework assigns two objectives to two individual sequence encoders, and aggregates the outputs by extracting useful information. Experiments on benchmark datasets validate the effectiveness of our proposed model.

Keywords: sequential recommendation; calibrated recommendation; fairness; diversity

Citation: Chen, J.; Wu, W.; Shi, L.; Ji, Y.; Hu, W.; Chen, X.; Zheng, W.; He, L. DACSR: Decoupled-Aggregated End-to-End Calibrated Sequential Recommendation. *Appl. Sci.* **2022**, *12*, 11765. https://doi.org/10.3390/app122211765

Academic Editor: Keun Ho Ryu

Received: 25 October 2022
Accepted: 16 November 2022
Published: 19 November 2022

1. Introduction

Recommender systems aim to help users find their interests among large-scale items. In recent years, sequential recommendation has achieved great attention, which predicts users' future behaviors according to sequences of historical behaviors. Existing studies focus on modeling sequences, learning item transitions and obtaining accurate recommendations. The deep learning-based architectures, such as Recurrent Neural Networks and Graph Neural Networks, have progressed in sequential recommendation [1–3]. Until now, most studies have focused on obtaining high accuracy of recommendation lists. However, previous studies argue that recommendation algorithms should consider more than accuracy. For example, diversity [4–8], coverage [9–11], unexpectedness [12–14] and fairness [15,16] are also important concepts in measuring a recommender system. Among these concepts, diversity requires the recommender system to generate item lists that contain more item attributes (e.g., genres of movies), and a fair recommender can provide unbiased recommendation lists for consumers or providers. From these two perspectives, we focus on the calibration of sequential recommendation, which is related to diversity and fairness. The calibrated recommendation was first proposed by Steck [17]. It aims to provide the recommendation list

which reflects the user's historical behaviors [17]. For example, if a user has watched 70% action movies and 30% comedies, a fully calibrated recommendation list should also contain action and comedy movies with this ratio. Compared to diversity, it can also provide diversified recommendation lists to a certain degree [18]. The difference between calibration and diversity is that calibration limits the covered item attributes of recommendations in the range of attributes interacted in the user's historical behaviors. Furthermore, if a user has homogeneous interests, the recommendations will become skewed under the constraint of calibration. From the perspective of fairness, it is a type of C-fairness according to the taxonomy proposed by Burke et al. [19], which is the fairness from the consumer's perspective. For users with similar historical interests, the calibrated recommendation model is able to provide recommendation lists with similar interest distributions. This somehow avoids bias and reflects the fairness of the recommendation system.

Existing studies in calibrated recommendations adopt a post-processing paradigm, which ranks items from a candidate list that has been generated by a basic recommendation model such as neural collaborative filtering [17,20,21]. This is different from the end-to-end recommendation paradigm of sequential recommendation. As illustrated in Figure 1, both end-to-end and post-processing-based models require predicted scores of all items. The difference is that the end-to-end models directly select the top-K items as final recommendations, while post-processing-based models apply a re-ranking stage on the top-Z ($Z > K$) items. For example, Kaya et al. [18] and Silva et al. [21] generated calibrated recommendations based on the top-100 items provided by the basic algorithms. The advantage of post-processing-based approaches is that they can be applied to almost any recommendation model, because they only require the items and scores of the candidate list generated by the base model. However, the post-processing-based models have two limitations. On the one hand, these models do not consider the relation between accuracy and calibration. The post-processing-based models make calibrated predictions based on the trained basic models which are optimized by the accuracy objective. In this scenario, accuracy and calibration are separate objectives, so that recommendation accuracy is limited by the trained basic model and considering calibration may sacrifice the accuracy. On the other hand, these models need more response time in generating recommendation lists because they have an extra ranking stage than the end-to-end models. During the ranking stage, the items of the final recommendation list are decided step by step, where the model needs to compute the gains of all candidates. Though the number of candidates can be much less than the size of the item set by selecting top-Z items, the ranking stage is still time-consuming.

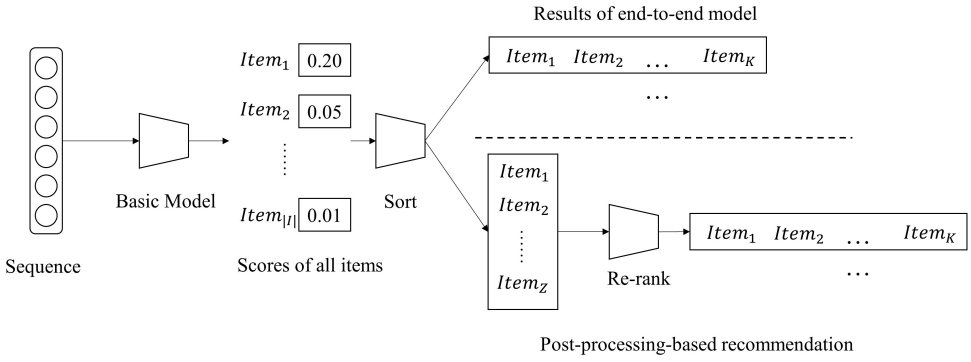


Figure 1. An illustration of end-to-end and post-processing-based recommendation.

To handle these problems, we focus on providing accurate and calibrated results for sequential recommendation in an end-to-end framework. We propose a **D**ecoupled-**A**ggregated **C**alibrated **S**equential **R**ecommendation framework, namely "**DACSR**". First, we define a loss function for calibration to allow models can be optimized by accuracy and calibration simultaneously. We combine the prediction scores of all items and the item

attribute information to estimate the distribution of the recommendation list. Then we use cosine similarity to measure the consistency between distributions of the recommendation list and behavior sequence. We also provide distribution modification approaches to further improve the diversity and mitigate the problem of amplification of main interests. Next, to handle the objectives of accuracy and calibration, we propose a decoupled-aggregated framework to provide accurate and calibrated recommendations. We utilize two individual sequence encoders which only focus on accuracy and calibration, respectively. Then we concatenate the item embeddings and sequence representations, and obtain final representations of the sequence and items by extractor networks with feed-forward networks and residual connections. The scores of all items are computed by the final representations, and the model is optimized according to the weighted sum of accuracy and calibration loss functions.

The contributions of this paper are listed as follows:

- We propose an end-to-end framework to provide accurate and calibrated recommendation lists for sequential recommendation.
- We design a calibration loss function for model optimization, which aligns the preference distribution of recommendations to the historical distribution. In addition, we provide distribution modification methods for diversity and imbalanced interests.
- We propose a decoupled-aggregated framework which aggregates information from individual sequence encoders which are optimized by calibration and accuracy separately.
- Experiments on benchmark datasets show that our model can achieve accurate and calibrated recommendations, with less time consumption than post-processing-based models.

The rest of this paper is organized as follows. We first review the existing literature in Section 2 and provide some preliminaries about sequential and calibrated recommendation in Sections 2 and 3. Then we introduce our model in Section 4. We provide experimental settings in Section 5 and show results and analysis in Section 6. Finally, we discuss our work in Section 7 and conclude our paper and indicate some future work in Section 8.

2. Related Work

In this section, we provide a literature review of our work. We first introduce existing studies in sequential recommendation. Then we review the recent advances in calibrated recommendation.

2.1. Sequential Recommendation

Sequential recommendation relies on users' historical behavior sequences to predict their future behaviors. Existing studies focused on modeling sequences and obtained better sequence representations to achieve higher recommendation accuracy. Hidasi et al. [1] first utilized gated recurrent units for sequential recommendations and provided a parallel training strategy. Li et al. [22] further proposed an attention mechanism to capture the main purposes of the sequence. Tang et al. [23] utilized convolutional neural networks to extract information from short-term sequences. With the development of self-attention, self-attentive-based models were proposed to achieve better sequence representations. For example, SASRec applied a self-attentive mechanism to learn both long-term and short-term preferences of user behavior sequences, which achieved satisfactory performance of recommendation accuracy [24]. Xu et al. [25] divided sequences into subsequences and captured users' long- and short-term preferences by applying two self-attention networks. In addition, transformer-based sequence encoders were proposed, such as BERT4Rec and Transformer4Rec [26,27]. In recent years, graph neural networks were also utilized for sequential recommendation [2,3,28]. For example, Wu et al. [3] applied GGNN to learn item transitions from historical behaviors which treated sequences as graphs. In addition, multi-interest-based models were proposed that used multiple vectors to represent a sequence in order to disentangle users' diverse intentions [5,6,29,30].

2.2. Calibrated Recommendation

Existing sequential recommendation models achieved satisfactory recommendation accuracy. From the concerns of fairness and filter bubble, we focus on the calibration of recommendation lists of sequential recommendation algorithms. In recent years, calibration was proposed, which aimed to generate recommendation lists whose preference distributions were less divergent with the users' profile [17]. Steck [17] also provided a post-processing greedy re-ranking model which considered both accuracy and calibration at each step of generating results. Abdollahpouri et al. [31] studied the connections between popularity bias and calibration. They found that users who were affected more by popularity bias tend to achieve less calibrated recommendation lists. Kaya and Bridge [18] compared intent-aware algorithms and calibration algorithms. They found that the diversity-oriented intent-aware models can achieve calibrated recommendations and calibration-oriented models can obtain diversity to some extent. Seymen et al. [20] proposed weighted total variation to measure the consistency between two distributions and a constrained optimization model to improve the ranking stage for calibration. Silva et al. [21] proposed new metrics to evaluate calibrated recommendations and adaptive selection strategies for the trade-off weight in the post-processing algorithms.

The calibration of recommendations is connected to two types of concepts. One is diversity, which aims to provide diversified recommendations for users. Seeking accuracy may lead to skewed recommendation lists which only focused on the main interest area of users, but users may be interested in diversified lists [17,29]. The calibrated recommendation constrained the recommendations to match the user's historical preference distribution to avoid the problem. However, it is a type of limited diversity because recommendations are limited by users' historical behaviors. Despite the limitation of interests, it is still considered as a solution of homogeneous contents [18,32]. Calibration is also a type of fairness. Fairness in recommender systems aimed to provide unbiased results for users. From the perspective of stakeholder, it can be defined as C-fairness, P-fairness and CP-fairness, which stand for consumers, providers, and the combination, respectively [19]. The calibrated recommender system can be treated as one of C-fairness [21]. The fairness is reflected by the less divergence of preference distributions between the user's profile and the recommendation list.

Despite previous advances in calibrated recommendation, it still suffers from the following problems. First, existing methods for calibration required re-ranked the candidate items generated by a basic recommendation model, which required more time in generating recommendations. In addition, the post-processing models may sacrifice accuracy to improve calibration, because they separated the process of achieving the accuracy and calibration. In our work, we would like to explore whether calibrated recommendations can be provided in an end-to-end way without post-processing. In addition, we investigate whether considering both calibration and accuracy can contribute to the performance of sequential recommendation.

3. Preliminary

3.1. The Sequential Recommendation Paradigm

The sequential recommendation predicts items that the user may interact in the future based on the user's historical behavior sequence. In general, it can be decomposed into two parts, the sequence encoder and the prediction layer. The sequence encoder takes the historical behavior sequence as the input, and represents it to a vector. Formally, the procedure can be written as:

$$h = f(s \mid E^I, \theta, L) \tag{1}$$

where $f(\cdot)$ is the sequence encoder and h is the sequence representation of sequence s. E^I represents the item embedding matrix of all items I, and θ stands for the parameters of the sequence encoder. L is the loss function that is used to optimize the sequence encoder f.

The sequence representation h is further used to predict the score of all items. The prediction layer is usually a linear transformation layer:

$$\hat{y} = Wh^{\mathsf{T}} + b \tag{2}$$

where W and b are $\mid I \mid \times d$ and $\mid I \mid$ dimensional learnable parameters. A common setting is that W is the item embedding matrix which is used in f and bias b is removed:

$$\hat{y} = E^I h^{\mathsf{T}} \tag{3}$$

where E^I is the item embedding matrix. This prediction layer is widely used in existing studies [3,22,24,25,33], and we also follow this setting in our work. In our work, we select SASRec [24] as the basic sequence encoder, because our goal is not to investigate modeling sequences and the SASRec model has achieved satisfactory performances in existing studies.

3.2. Preference Distributions

The calibrated recommendations aim to provide unbiased results, where preferences reflected from historical behaviors and recommendation lists are consistent. We use symbols $p(s)$ and $q(s)$ which stand for the preference distributions from historical behaviors and recommendation lists, respectively. We follow previous work which applied item attributes to define preference distributions [17], which are introduced below in detail.

- $p(s)$ is the preference distribution from the sequence s. For each attribute g, the distribution value is computed as:

$$p(g \mid s) = \frac{\sum_{x \in s} p(g \mid x)}{\mid s \mid} \tag{4}$$

  where $p(g \mid x)$ is the indicator function of item x and attribute g, which satisfies $\sum_{g \in G} p(g \mid x) = 1$. If item x does not contain attribute g, the value of $p(g \mid x)$ is 0. If the item contains two attributes, the value of $p(g \mid x)$ equals to 0.5 for each attribute g. Finally, the preference distribution can be represented as a G-dimensional vector $\{p(g = 1 \mid x), p(g = 2 \mid x), \ldots, p(g = G \mid x)\}$, where $\mid G \mid$ is the total amount of item attributes.
- $q(s)$ is the preference distribution of the recommendation list. For each attribute g, the distribution value is computed as:

$$q(g \mid RL_s) = \frac{\sum_{x \in RL_s} p(g \mid x)}{K} \tag{5}$$

  where RL_s is the recommendation list of the sequence s, and K is the size of RL_s. Similar to $p(s)$, the distribution $q(s)$ can also be represented as a G-dimensional vector $\{q(g = 1 \mid x), q(g = 2 \mid x), \ldots, q(g = G \mid x)\}$.

4. Methodology

In this section, we introduce our DACSR model in detail. We first introduce our proposed calibration loss function for end-to-end sequential recommendation. Then we introduce the Decoupled-Aggregated architecture in detail.

4.1. The Calibration Loss Function

Loss Function for Calibration To generate a calibrated recommendation list, we design the loss function for the model training. The calibration measures the consistency between the recommendation list and the historical sequence. The distribution of historical

sequence $p(s)$ can be computed as Equation (4). For the recommendation list, we estimate its preference distribution $\hat{q}(s)$ as follows:

$$\hat{q}(g \mid s) = \sum_{i}^{|I|} \hat{y}_i \cdot p(g \mid i) \tag{6}$$

$$\hat{y}_i = softmax(\hat{y}_i / \tau) \tag{7}$$

where $\hat{y}_i$ is the score of the item i predicted by the model, and it is further processed by a softmax function. If an item has a higher prediction score, it contributes more to $\hat{q}(g \mid s)$. The softmax function also amplifies the difference in scores. Items with high scores will still be given higher weights, while weights of other items are close to 0. τ ($\tau > 0$) is the temperature parameter of softmax function. If $\tau < 1$, the score distribution becomes sharper and items with higher scores get more emphasis. In contrast, an extremely large value of τ will make the score distribution more uniform.

After estimating $\hat{q}(g \mid s)(g \in G)$, we define the loss function of calibration as:

$$L_{Calib}(\hat{y}) = 1 - cos(\hat{q}(s), p(s)) \tag{8}$$

where $\hat{q}(s)$ is the estimated distribution vector $\{\hat{q}(g = 1 \mid s), \ldots, \hat{q}(g = G \mid s)\}$ and $cos(v_1, v_2)$ is the cosine similarity between two vectors. If the two distributions are more consistent, the value of L_{Calib} will be lower.

Distribution Modification Although calibration is related to diversity, a calibrated recommendation list is not always a diversified list. For example, a user who focuses on a few types of items will receive less diversified recommendations when calibration is considered. Therefore, if we want a diversified recommendation list to a certain degree, we can modify the distribution as follows:

$$p_d(s) = softmax(p(s) / \tau_{div}) \tag{9}$$

In this equation, the historical distribution $p(s)$ is normalized by a softmax function. For item attribute g that the user did not interacted (i.e., $p(g \mid s) = 0$), it will obtain a positive value. Therefore, all attributes are considered. The parameter τ_{div} is also used to control the distribution, similar to τ.

Meanwhile, for users who have homogeneous interests, their main interests may be amplified under the end-to-end framework. This is similar to the imbalanced classification tasks which tend to predict the major labels. To this end, we propose a mask-based modification method:

$$p_m(s) = softmax(mask(p(s)) / \tau_{div}) \tag{10}$$

where the $mask(p(s))$ give all attributes whose $p(g \mid s) = 0$ an extremely little negative value (e.g., -10^{10}). Therefore, scores of these attributes in $p_m(s)$ will be 0. In this equation, τ_{div} can be larger than 1 so that the distribution becomes more uniform, and the scores of the main interest and other interests are more close. The difference from Equation (9) is that the scope of interests are still limited in those the user have interacted with, while the $p_d(s)$ distribution can explore new interests for the user.

Loss Function for Accuracy and Calibration To obtain recommendation lists with accuracy and calibration, an intuitive way is directly optimizing the sequential recommendation model with a weighted sum of loss function:

$$L_w(\hat{y}) = (1 - \lambda) L_{Acc}(\hat{y}) + \lambda L_{Calib}(\hat{y}) \tag{11}$$

where $\lambda \in [0, 1]$ is the trade-off factor between accuracy and calibration. A higher value of λ means more consideration on calibration. L_{Acc} is the accuracy-based loss function. In our work, we choose the cross-entropy loss function:

$$L_{Acc}(\hat{y}) = \sum_{i=0}^{|I|} y_i \log(\hat{y}_i) \tag{12}$$

where y and $\hat{y}$ are the vectors of the ground-truth and predicted scores of all items, respectively. The y is an one-hot vector where $y_i = 1$ means item i is the next item of the sequence, and 0 otherwise.

4.2. The Decoupled-Aggregated Framework

Directly optimizing a sequence encoder by the weighted loss function may lead to a seesaw problem [34,35]. For example, the performance of calibration increases by sacrificing the recommendation accuracy. This is because optimizing by two objectives based on shared parameters limits the ability of the model to obtain better representations of sequences and items. Therefore, we propose our Decoupled-Aggregated framework, which includes two basic sequence encoders, as shown in Figure 2. The two sequence encoders are optimized separately. One is to accurately predict the next behavior, while the other one is to provide a fully calibrated recommendation list. Formally, this can be represented as:

$$h_p = f_p(s \mid E_p^I, \theta_p, L_{Acc}) \tag{13}$$

$$h_c = f_c(s \mid E_c^I, \theta_c, L_{Calib}) \tag{14}$$

where f_p and f_c are two different sequence encoders with different parameters and objective functions. E_p^I and E_c^I are item embedding matrices of the two sequence encoders. θ_p and θ_c are their parameters. Note that the two encoders f_p and f_c do not share the same parameters and item embedding matrices, and are optimized by their unique loss functions L_{Acc} and L_{Calib}, respectively.

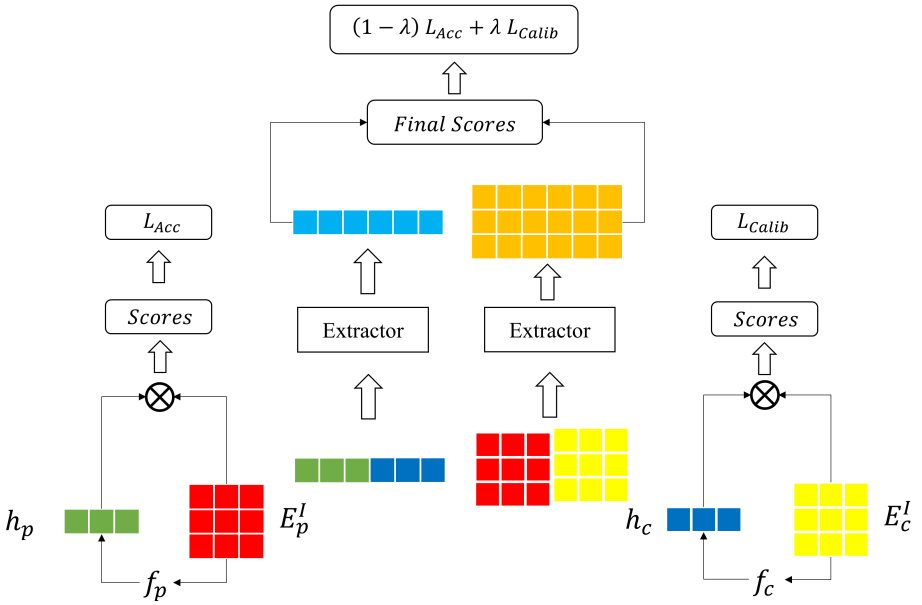


Figure 2. The architecture of our DACSR model.

Next, we use the sequence representations and item embeddings of the two encoders to provide calibrated recommendation list. A direct way is to concatenate the representations of two sequence encoders:

$$h_a = [h_p, h_c] \tag{15}$$

$$E_a^I = [E_p^I, E_c^I] \tag{16}$$

where $[,]$ is the concatenation execution of two vectors or matrices. Based on this, we use the concatenated vectors as input to two extractor nets:

$$h_o = EX_{seq}(h_a) \tag{17}$$

$$E_o^I = EX_{emb}(E_a^I) \tag{18}$$

EX_{seq} and EX_{emb} are two extractor nets. The extractor net is a feed-forward network which can be defined as follows:

$$h^t = W^t ReLU(h^{t-1}) + b^t \quad (t = 1, 2, 3 \dots) \tag{19}$$

where $W^0, W^1, \dots, W^t$ and $b^0, b^1 \dots, b^t$ are $2d \times 2d$ and $2d$ dimensional parameters need to learn (d is the dimension of hidden states). t is the number of layers in the extractor net. $ReLU$ is the activation function. h^0 is the input of the feed-forward network (i.e., $[h_p, h_c]$ and $[E_p^I, E_c^I]$). Inspired by [36], we add the original input as the final representation:

$$h_{out} = h^t + h^0 \tag{20}$$

where h_{out} is the final output the extractor network, which can be either sequence representation h_o or item embedding E_o^I. Finally, the scores of all items can be computed as:

$$\hat{y}_o = E_o^I h_o^\top \tag{21}$$

which is similar to Equation (3). We also use the weighted loss function L_w to optimize the model. In conclusion, the loss function of the DACSR model can be written as:

$$L = L_w(\hat{y}_o) + L_{Acc}(\hat{y}_p) + L_{Calib}(\hat{y}_c) \tag{22}$$

where $\hat{y}_p$ and $\hat{y}_c$ are scores of all items generated by f_p and f_c.

5. Experiments

5.1. Dataset

We adopted two commonly-used benchmark datasets to evaluate the performance of our model. The first one is *Movelens-1m* (https://grouplens.org/datasets/movielens/1m, accessed at 14 September 2021), whichcontains interaction logs of more than 6000 users and 3000 movies. The other is *Tmall* (https://tianchi.aliyun.com/dataset/dataDetail?dataId=53, accessed at 3 January 2022), which includes user behavior logs on an e-commerce platform. We retained the "buy" behaviors for the Tmall dataset. For both datasets, we sorted each user's behaviors according to the timestamp. We followed a 5-core and 20-core strategy for the Ml-1m and Tmall dataset that removes the users and items whose number of occurrences is less than 5 or 20. We also applied the leave-one-out evaluation protocol [24]. The latest clicked item of a user belongs to the testing set, and the previous one of this item belongs to the validation set. The remaining sequences construct the training set. To augment the training data, we extended the user's sequence following [3,22]. We set the maximum sequence length equal to 200 and 100 for the Ml-1m and Tmall dataset, respectively. The statistics are listed in Table 1.

Table 1. Statistics of Datasets.

Statistics	Ml-1m	Tmall
Number of users	6040	31,854
Number of Items	3883	58,343
Number of Training Sequences	981,504	832,603
Number of Testing Sequences	6040	31,854
Number of Attributes	18	70
Average Length of Sequence	164.50	28.13

5.2. Comparison Models

We selected the following methods as baselines:

- **SASRec** [24] is a self-attentive-based sequential recommendation model, and is a strong baseline. We apply the SASRec model as the sequence encoder for f_p and f_c, and compare our model with SASRec.
- **CaliRec** [17] is a post-processing model which re-ranks the results generated by the sequence encoder. It makes a trade-off between accuracy and calibration at each time step.
- **CaliRec-GC** [21] utilizes an adaptive selection for the trade-off factor between calibration and accuracy in the CaliRec model. The higher coverage of item attributes of the user historical sequence results in more consideration for calibration.

5.3. Evaluation Metrics

We evaluate the performances of our model and baselines in terms of accuracy and calibration. Following previous works [3,22], we use Recall and MRR as evaluation metrics to measure the recommendation accuracy.

- **Recall@K** (Rec@K) is a widely used metric in recommendation and information retrieval areas. Recall@K computes the proportion of sequences whose next behaviors are included in the recommendation lists.

$$Recall@K = \frac{1}{N}\sum_{s} 1(x_{n+1} \in RL_s) \tag{23}$$

  where $1(\cdot)$ is an indication function whose value equals 1 when the condition in brackets is satisfied and 0 otherwise. N is the number of testing cases.
- **MRR@K** is another important metric that considers the rank of correct items. The score is computed by the reciprocal rank when the rank is within K; otherwise the score is 0.

$$MRR@K = \frac{1}{N}\sum_{s} \frac{1}{rank(x_{n+1}, RL_s)} \tag{24}$$

To evaluate the effectiveness in terms of calibration, we adopt C_{KL} which is a common metric used for calibrated recommendation [17]. The C_{KL} compares the consistency between two distribution:

$$C_{KL}(RL, s) = \frac{1}{N}\sum_{s}\sum_{g \in G} p(g \mid s)\frac{p(g \mid s)}{\widetilde{q}(g \mid s)} \tag{25}$$

The lower C_{KL} value means we provide more calibrated recommendation lists. To avoid the division-by-zero error, we use $\widetilde{q}(g \mid s) = (1-\alpha)(g \mid s) + \alpha p(g \mid s)$ to replace the original preference distribution $q(g \mid s)$. The value of α also equals to 0.01 according to [17,18,20,21].

In addition, to better compare performances of our model and baseline models, we define the improvement as follows:

$$Improv. = \frac{Metric_{DACSR} - Metric_{Baseline}}{Metric_{Baseline}} \tag{26}$$

where *Metric* can be any metric mentioned above.

5.4. Experimental Setup

We fixed the dimension of sequence representations and item embeddings equal to 64 for the DACSR model. For a fair comparison, we set the dimension of hidden states of the SASRec model to 128, so that the numbers of parameters are close. The number of layers in the extractor net t of the DACSR model equals 2 for all datasets. We used the Adam [37] optimizer with the batch size of 256 and the learning rate of 0.001. We reported the performance under the model parameters with the optimal prediction accuracy on the validation set. We made hyper-parameter $\lambda = 0.5$ and $\tau = 1$ as the default

setting, and analyzed their influence in following sections. To accelerate the training procedure, we initialized the parameters of sequence encoders used in our models by pre-trained parameters. For the top-K recommendations, we set $K = 10$ and 20 which is a common setting.

6. Results and Analysis

In this section, we provided results and analysis of our work. In general, we aimed to answer the following research questions:

- **RQ1** How are the performances and efficiency of our DACSR model in achieving accurate and calibrated recommendation lists?
- **RQ2** How do the performances of our model change as the parameters change?
- **RQ3** How do the modules of our DACSR model contribute to performance improvement?
- **RQ4** How do the distribution modification approaches work on the two datasets?

6.1. RQ1: Overall Performance

In this section, we answer the research question RQ1 about whether our model can provide calibrated and accurate recommendations. The performances of baselines and our model are listed in Table 2, where the best performance is marked in bold.

Table 2. Performances of our models and baselines (best performances are marked in bold).

Datasets	Metrics	SASRec	CaliRec	CaliRec-GC	DACSR
ML-1m	Rec@10	0.2627	0.2636	0.2225	**0.2811**
	MRR@10	0.1203	0.1101	0.0712	**0.1267**
	C_{KL}@10	1.2385	0.9722	**0.4553**	1.0615
	Rec@20	0.3613	0.3616	0.3258	**0.3844**
	MRR@20	0.1271	0.1168	0.0784	**0.1338**
	C_{KL}@20	0.8548	0.7322	**0.3847**	0.7262
Tmall	Rec@10	0.1451	0.1464	0.1454	**0.1517**
	MRR@10	**0.0862**	0.0846	0.0861	0.0857
	C_{KL}@10	2.5004	2.0710	2.4139	**2.0114**
	Rec@20	0.1749	0.1753	0.1751	**0.1855**
	MRR@20	**0.0883**	0.0866	0.0882	0.0881
	C_{KL}@20	2.1103	1.7943	2.0459	**1.6240**

Recommendation Accuracy We first analyze the performance from the perspective of accurate recommendation (i.e., Rec@K and MRR@K). In general, on both datasets, our model achieves the best prediction accuracy in terms of Recall and MRR. By considering calibration, users' preference distributions are incorporated. In addition, our model decoupled the two objectives by two sequence encoders and aggregated their outputs. Therefore, the preference distribution contributed to the prediction of the next item, leading to the improvement of accuracy. For example, on the Ml-1m dataset, the Recall and MRR of our model are higher than the original SASRec model (e.g., 0.1338 vs. 0.1271 in terms of MRR@20). In contrast, the post-processing-based models fragmented the relationship between accuracy and calibration and therefore resulted in a reduction in accuracy. For example, on the Ml-1m dataset, the MRR@20 of our model is 0.1338, while it is 0.1168 and 0.0784 for CaliRec and CaliRec-GC model, with the improvement of 18.67% and 74.66%, respectively. On the Tmall dataset, the CaliRec model also decreases the prediction accuracy.

Calibrated Recommendation Our model can provide more calibrated recommendation lists compared to the original sequential recommendation model. On both datasets,

C_{KL}@10 and C_{KL}@20 of our model are lower than the original SASRec model. For example, the C_{KL}@20 of our model is 0.7262, which is 15.04% better than the 0.8548 of the SASRec model. On the Tmall dataset, our model also achieves a 23.04% improvement in terms of C_{KL}@20. Compared to the post-processing-based CaliRec model, our model still achieves competitive performances in terms of calibration. For example, on the ML-1m dataset, the performances of C_{KL}@20 of our model and the CaliRec model are 0.7262 and 0.7322. On the Tmall dataset, our model achieves an improvement of 9.49% in terms of C_{KL}@20. The comparisons show the ability of our model to achieve better accuracy while obtaining competitive performances of calibration compared to the post-processing-based models. As for the possible reasons, on the one hand, the proposed loss function calibrated the preference distribution of items with the highest scores to the historical preference distribution. On the other hand, the decoupled-aggregated framework ensures accuracy when improving the calibration.

We also observe that the CaliRec-GC model performs differently on the two datasets. On the Ml-1m dataset, the CaliRec-GC model achieves the lowest C_{KL} value among all models, including our proposed model (e.g., 0.3847 vs. 0.7262 of C_{KL}@20). While on the Tmall dataset, the CaliRec-GC model cannot provide calibrated recommendation lists. For example, the C_{KL}@20 of CaliRec-GC is 2.0459, which is close to the original SASRec model. We think this phenomenon results from two aspects. On the one hand, the number of item attributes of the Tmall dataset is much more than that of the Ml-1m dataset. The Ml-1m dataset contains 18 different item attributes, while the Tmall dataset has 70 attributes. On the other hand, the average length of the user behavior sequence of the Tmall dataset is less than the Ml-1m dataset, as shown in Table 1. The shorter sequence and larger item attributes set lead to the lower coverage of item attributes. The CaliRec-GC model adopts an adaptive selection of the trade-off factor λ for calibration based on the coverage of item attributes. The greater coverage leads to the higher value of λ. Therefore, it performs best in terms of calibration on the Ml-1m dataset, and almost does not work on the Tmall dataset.

The differences between the two datasets also lead to the different calibration performances of the two datasets. In general, performances of C_{KL} on the Ml-1m dataset are better than the Tmall dataset. For example, the C_{KL}@20 of our DACSR model is 1.6240 on the Tmall dataset, which is much higher than the 0.7262 on the Ml-1m dataset. This is similar for the original SASRec model, with 2.1103 vs. 0.8548 on the two datasets. A possible reason is that the lower coverage of item attributes mentioned above results in the higher divergence. The large amount of 0 in $p_(s)$ makes it difficult in achieving calibration, especially with the concern of accuracy.

Time Consumption In addition, we also compared the response time of our model against the post-processing-based CaliRec model and the original SASRec model. We focused on the average time required to generate the recommended list for each sequence. For the SASRec model, we reported the time consumption when the dimension of hidden states equals to 64 and 128 (namely $SASRec_{D64}$ and $SASRec_{D128}$). This is because the size 64 is the setting of each sequence encoder of our DACSR model. We conducted experiments on the same device, and removed the GPU acceleration for a fair comparison. The performance is listed in Table 3.

Table 3. Average time consumption for each sequence.

Response Time (10^{-4} s)	Ml-1m	Tmall
$SASRec_{D64}$	2.01	1.32
$SASRec_{D128}$	3.28	1.96
CaliRec	580.15	668.69
DACSR	4.24	5.76

Compared to the original SASRec model, our model needs more computation. For example, on the Ml-1m dataset, the time consumption for each sequence of the $SASRec_{D64}$ model is 2.01 $\times$ 10^{-4} s, which is approximately half of our DACSR model. This is because it incorporates two SASRec encoders and an extraction net, which is more complex than the single SASRec model. In contrast, our model can provide more accurate and calibrated recommendations than the original SASRec model. The $SASRec_{D128}$ requires more time than $SASRec_{D64}$ because it contains a larger scale of parameters. Compared to the CaliRec model, our model costs much less time to generate recommendation lists. For a single sequence, our model only needs 4.24 and 5.76 $\times$ 10^{-4} s on the Ml-1m and Tmall datasets, respectively. In contrast, the CaliRec requires approximately 0.06 s for a sequence, which needs 200 times more time than our model. This is because the CaliRec model needs an extra ranking stage. The original SASRec model provided scores of all items, and selected the top-100 items with the highest scores. The post-processing-based CaliRec model then re-ranks the top-100 items with K steps (K stands for the top-K recommendations). At each step, it computes the gains of the candidate items when they are added to the recommendation list. However, our model follows an end-to-end framework only with a sorting stage to select top-K items after the scores of all items are computed. Therefore, our model obtains better performance and requires less time consumption than the CaliRec model for each sequence.

Generalization of DACSR Model We are also interested in whether our model is also effective when the sequence encoder changes. We incorporated the GRU4Rec model [1] as the sequence encoder, which is also a widely used sequential recommendation model. The experimental settings were same to the previous section with $\lambda = 0.5$. We use DACSR(G) to denote our DACSR model which takes GRU4Rec as the sequence encoder, and CaliRec(G) to denote the post-processing-based CaliRec model with candidates provided by the GRU4Rec model. The performances are listed in Table 4.

Table 4. Performances of our DACSR(G) model with the GRU4Rec sequence encoder (best performances are marked in bold).

Dataset	Method	Rec@20	MRR@20	C_{KL}@20
Ml-1m	GRU4Rec	0.3460	0.1142	0.8356
	CaliRec(G)	0.3454	0.1105	0.7013
	DACSR(G)	**0.3472**	**0.1161**	**0.6840**
Tmall	GRU4Rec	0.1724	0.0863	2.2123
	CaliRec(G)	0.1732	0.0846	1.8597
	DACSR(G)	**0.1750**	**0.0878**	**1.7266**

As shown in the table, our model can still achieve calibrated and accurate recommendation lists when we use GRU4Rec as the sequence encoder. On the Ml-1m dataset, the performances of C_{KL}@20 are 0.6840 and 0.8356 of our DACSR(G) model and the GRU4Rec model, respectively. On the Tmall dataset, our model also obtains an improvement of 21.95% in terms of calibration. Toward recommendation accuracy, the performances of our model are still better than the original GRU4Rec model. The improvement is not as great as the DACSR model with the SASRec sequence encoder. We think that this is because the ability to model sequences of GRU4Rec is worse than that of the SASRec model. The SASRec model with self-attention mechanisms can better find the user's preference and represent the sequence. The CaliRec(G) model also sacrificed the ranking performance to improve the calibration, which is similar to the CaliRec model with the SASRec model. Compared to the CaliRec(G) model, our DACSR(G) model can also achieve better performance in terms of accuracy and calibration, as listed in Table 4. The performance comparisons indicate that our model can be used for other basic sequence encoders, which is not specifically designed for the SASRec model.

6.2. RQ2: Parameter Influence

In this section, we answer the research question RQ2 about the influence of hyperparameters. Specifically, we investigate the two hyperparameters λ and τ (see Section 4.1). The trade-off parameter λ controlled the importance of calibration during the optimization stage of our model. The parameter τ reshaped the predicted scores of all items, which can affect the computation of the calibration loss function.

6.2.1. Trade-Off Factor λ

We first analyze the influence of hyperparameter λ by changing it from 0.1 to 1.0. Since the CaliRec model also required a parameter to control the importance of calibration and accuracy which was similar to our model, we displayed the changes in the performance of our DACSR model and compared it with the CaliRec model simultaneously. The performances on the two datasets are shown in Figure 3, where red lines and blue lines represent our DACSR model and the CaliRec model, respectively.

In general, a greater value of λ leads to a lower recommendation accuracy and higher calibration. When λ has a high value (e.g., $\lambda = 0.7$), the performance of Rec@20 of our model decreases greatly, while the CaliRec model does not decrease as large as our model. A possible reason is the difference in the process of generating recommendation lists. The CaliRec re-ranks the candidate list generated by the SASRec model whose size is 100 and finally selects top-K (K = 10 or 20) items as the final recommendation list. However, our model computes scores of all items, and directly selects top-K items. For some sequences, the users' next items achieve relatively lower ranks. When considering more for calibration, these items may be replaced by items whose attributes are consistent with the historical distributions but not sequential correlated to the sequences. While for post-processing methods, the range of items for re-ranking is narrowed. Therefore, it can preserve the performance of Rec@20. In contrast, our model outperforms CaliRec when considering ranking performances in terms of MRR. As shown in Figure 3c,d, the MRR@20 of our model is always higher than the CaliRec model. This is because although target items may be replaced, these items do not contribute to the ranking performance greatly. In contrast, our model improves the ranking of the target items for most of the sequences, resulting in the improvement in overall ranking performance.

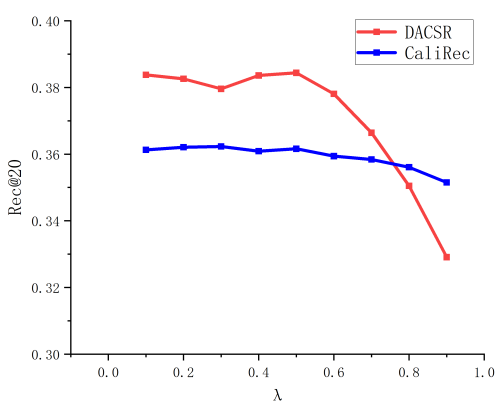
In terms of calibration, our model achieves better performances compared to the CaliRec model under the same λ in general. On the Tmall dataset, the performances of calibration of our DACSR model are better than the CaliRec model during the major changing process of λ. On the Ml-1m dataset, the performances are close. Our model performs better than the CaliRec model when λ is greater than 0.5. When λ is close to 1, our DACSR model cannot perform as well as the CaliRec model. A possible reason is that the CaliRec model utilized a greedy-based ranking strategy so that it can select the most calibrated recommendations from top-100 items when λ is close to 1. In contrast, the deep learning-based optimization strategy always has a gap in fitting the target distribution and therefore does not perform as well as the CaliRec model at very high values of λ. However, a large λ leads to lower recommendation accuracy. Though it achieves better calibration, it is not suggested to use high value of λ because accurately predicting the user's next behavior is still an important concern.

6.2.2. Temperature Factor τ

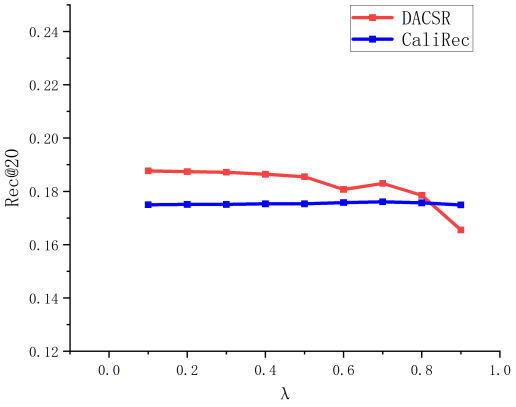
In this section, we analyze the influence of parameter τ, which controls the sharpness of the distribution in the calibration loss function. We tune τ in $\{0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0\}$, and show the performances in Figure 4. Red lines represent the performance of MRR@20 and correspond to the right axis, and blue lines stand for the C_{KL}@20 performance which follows the left axis.

On the two datasets, a relatively lower value of τ can achieve a better performance of calibration and lower accuracy in general. For example, on the Ml-1m dataset, the C_{KL}@20 performances are 0.6466 and 0.6915 when τ equals to 0.25 and 1.0 respectively. This is

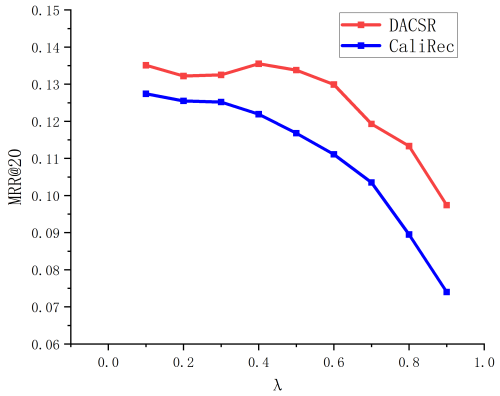
because a lower value of τ amplifies the scores of top items, and other items are ignored because their scores are normalized to 0. The lower value of τ increases the calibration performance, but decreases the recommendation accuracy, as shown by the blue lines in Figure 4. In contrast, a higher value of τ ($\tau > 1$) causes a negative impact on calibration, because scores of all items are normalized to close values (i.e., $1/\mid I \mid$ for all items). Therefore, no useful information for calibration can be propagated to the model.



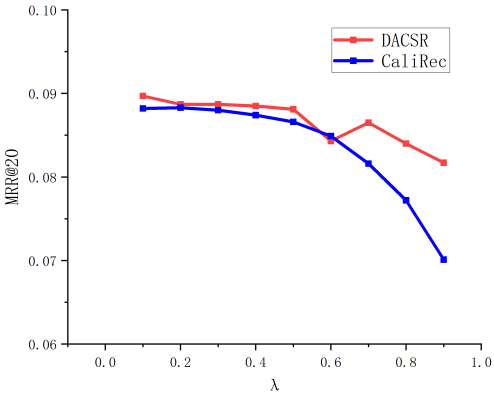
(a) Rec@20 on the Ml-1m dataset



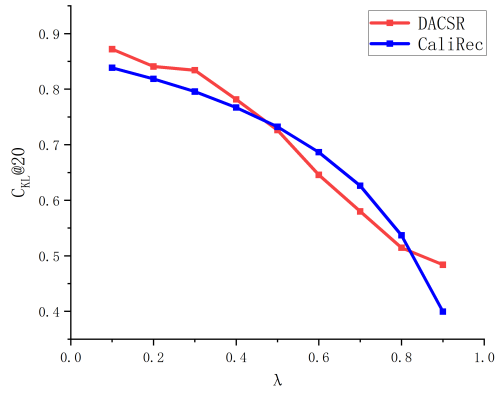
(b) Rec@20 on the Tmall dataset



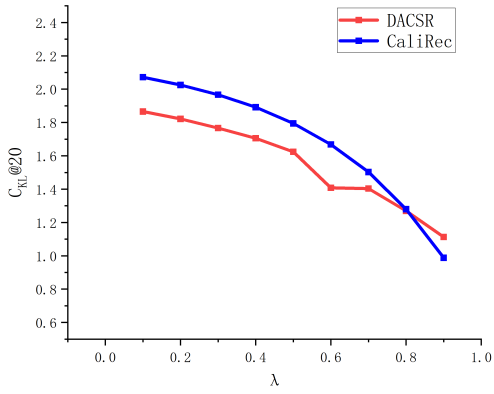
(c) MRR@20 on the Ml-1m dataset



(d) MRR@20 on the Tmall dataset

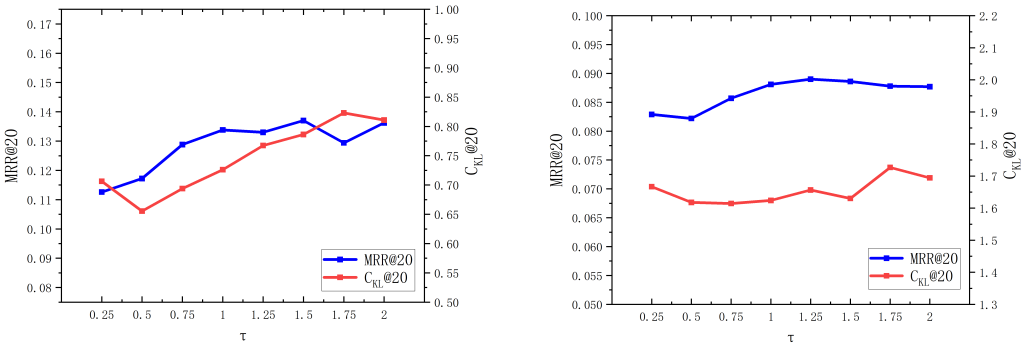


(e) C_{KL}@20 on the Ml-1m dataset



(f) C_{KL}@20 on the Tmall dataset

Figure 3. Performance comparison when λ changes.



(**a**) Performance change on the Ml-1m dataset (**b**) Performance change on the Tmall dataset

Figure 4. Performance comparison when τ changes.

6.3. RQ3: Ablation Studies

To answer the research question RQ3, we conduct ablation experiments in this section by comparing our model with two variants. The first variant is the original SASRec model optimized by the loss function L_w. We aim to investigate the performance of a single sequence encoder optimized by both accuracy and calibration. We also reported the performances when the dimension of hidden states equals to 64 and 128 (namely $SASRec^{L_w}_{D64}$ and $SASRec^{L_w}_{D128}$). In addition, we directly add the extractor nets to the $SASRec^{L_w}_{D128}$ model, namely $SASRec^{L_w}_{D128}EX$. The other one is the direct concatenation of sequence representations and item embedding matrices without extraction nets (namely DACSR-C). We compare our model with these variants by setting $\lambda = 0.5$. The performances are listed in Table 5. In general, our DACSR model obtains the best performance in terms of recommendation accuracy and calibration.

Table 5. Performance comparisons between our DACSR model and its variants.

Dataset	Metrics	$SASRec^{L_w}_{D64}$	$SASRec^{L_w}_{D128}$	$SASRec^{L_w}_{D128}EX$	DACSR-C	DACSR
Ml-1m	Rec@10	0.2551	0.2730	0.2715	0.2710	**0.2811**
	MRR@10	0.1088	0.1206	0.1213	0.1187	**0.1267**
	C_{KL}@10	1.0780	1.0765	1.0835	1.0909	**1.0615**
	Rec@20	0.3599	0.3657	0.3748	0.3719	**0.3844**
	MRR@20	0.1161	0.1269	0.1283	0.1257	**0.1338**
	C_{KL}@20	0.7281	**0.7253**	0.7315	0.7433	0.7262
Tmall	Rec@10	0.1508	0.1482	0.1462	0.1464	**0.1517**
	MRR@10	0.0854	0.0862	**0.0864**	0.0813	0.0857
	C_{KL}@10	2.1490	2.2499	2.3119	**1.9816**	2.0114
	Rec@20	0.1830	0.1787	0.1777	0.1804	**0.1855**
	MRR@20	0.0876	0.0884	**0.0886**	0.0837	0.0881
	C_{KL}@20	1.7810	1.8845	1.9322	**1.5960**	1.6240

The effectiveness of our designed loss function for calibration can be reflected by the performance of $SASRec^{L_w}_{D64}$ and $SASRec^{L_w}_{D128}$. By applying the loss function L_w, the SASRec model is able to provide more calibrated recommendation lists than the original SASRec model only optimized by L_{Acc}. For example, the C_{KL}@20 of SASRec on the Tmall dataset decreases from 2.1103 to 1.8845. Also, it achieves close performance compared to our DACSR model on the Ml-1m dataset in terms of calibration. The calibration performances of the SASRec model optimized by the weighted loss function L_w verified the effectiveness of our proposed loss function.

The performances between our model and variants also demonstrate the effectiveness of the decoupled-aggregated framework. For example, on the Ml-1m dataset, the MRR@20 of our DACSR model is 0.1338, while it is 0.1269 for the $SASRec^{L_w}_{D128}$ model, and the per-

formances of calibration are close (0.7262 vs. 0.7253). On the Tmall dataset, our DACSR model can achieve competitive recommendation accuracy, and provide more calibrated recommendations (e.g., 1.6240 vs. 1.8845 in terms of C_{KL}@20). Compared to the $SASRec^{L_w}_{D128}$ and $SASRec^{L_w}_{D128}EX$ model which shares parameters for two objectives, the decoupled-aggregated framework can achieve better performance. We believe that such a framework can learn the information of two objectives and combine them to obtain better representations of sequences and items. While a single sequence encoder that improves the performance in one aspect may negatively affect performance in the other because their parameters are shared. In addition, the DACSR-C model removed the extraction net and directly concatenated the representations of sequences and items from two sequence encoders. It obtained worse performance than the DACSR model, showing the importance of the extraction net. On the Ml-1m dataset, the C_{KL}@20 of the DACSR model is 0.7262, which is slightly better than the 0.7433 of the DACSR-C model. But the recommendation accuracy of the DACSR model is higher than the DACSR-C model (e.g., 0.1338 vs. 0.1257 in terms of MRR@20). On the Tmall dataset, our DACSR model also obtains better performances in terms of accuracy, and close performance of calibration. The extraction net takes the concatenation of sequence/item representations as inputs, and provides more suitable representations for the two objectives.

6.4. RQ4: Distribution Modification

In this section, we answer the research question RQ4 about the effectiveness of the proposed distribution modification approaches. We proposed the modified distribution $p_d(s)$ and $p_m(s)$ to further improve the diversity and mitigate the imbalanced interest problem. These approaches are related to the diversity. Therefore, we adopted the ILD metric with Jaccard similarity to measure the diversity of the recommendation list:

$$ILD(RL_s) = \frac{2}{\mid RL_s \mid (\mid RL_s \mid -1)} \sum_{(i,j \in RL_s)} (1 - \frac{\mid Attr_i \cap Attr_j \mid}{\mid Attr_i \cup Attr_j \mid}) \tag{27}$$

where $Attr_i$ is the item attribute set that the item i has, and RL_s is the generated recommendation list for sequence s. The larger ILD value represents the higher diversity of the recommendation list. We set the factor $\tau_{div} = 0.5$ and 2 for the distribution $p_d(s)$ and $p_m(s)$, respectively.

We first listed the performances of our DACSR model with the raw historical preference distribution (namely DACSR-$p(s)$) and the modified preference distribution for diversity (namely DACSR-$p_d(s)$) along with the original SASRec model in Table 6.

Table 6. Performances of calibration and diversity (best performances are marked in bold).

Datasets	Models	ILD@10	C_{KL}@10	ILD@20	C_{KL}@20
Ml-1m	SASRec	0.6499	1.2385	0.6677	0.8548
	DACSR-$p(s)$	0.6654	**1.0615**	0.6789	**0.7262**
	DACSR-$p_d(s)$	**0.7012**	1.1347	**0.7123**	0.7649
Tmall	SASRec	0.7086	2.4871	0.7405	2.1092
	DACSR-$p(s)$	0.6714	**2.0114**	0.7045	**1.6240**
	DACSR-$p_d(s)$	**0.7419**	2.2662	**0.7725**	1.8412

On the two datasets, the diversity of our model is improved by sacrificing the calibration. For example, on the Ml-1m dataset, the performances of ILD@10 are 0.7012 and 0.6654 for the normalized distribution $p_d(s)$ and the original distribution $p(s)$, respectively. However, the C_{KL}@10 increases from 1.0615 to 1.1347, which means the ability of calibration of our model is weakened. On the Tmall dataset, the performance comparisons are similar. This is because applying the normalized distribution amplifies the effect of item attributes

that the user did not interacted in the behavior sequence. Though it does not largely affect the true distribution $p(s)$, it deviates from the calibration to a certain degree.

We observe that our DACSR model performed differently on the two datasets. On the Ml-1m dataset, the diversity is higher for our DACSR model than the original SASRec model, while it is totally different on the Tmall dataset. On the Tmall dataset, our DACSR model achieves worse performance in terms of diversity (e.g., 0.6714 vs. 0.7086 of DACSR and SASRec model). It is possibly due to the difference between two datasets. On the Ml-1m dataset, the coverage of item attributes is higher than the Tmall dataset. Users have more historical behaviors than the Tmall dataset. On the Tmall dataset, users always interacted with several types of items, so that the $p(g \mid s)$ score of most attributes equals to 0. The limited interest areas resulted in less diversified recommendation lists under the calibration objective.

We also investigate the imbalanced interest problem. We find that main interests are amplified on the Tmall dataset. As illustrated in Figure 5, the attribute A occupies the 80% of the sequence, while attribute B and C only account for the 20%. For such an imbalanced distribution, our DACSR model amplifies the major interest, as shown in Figure 5. The recommended list only contains items with attribute A. This gives our model a negative impact in terms of diversity. By applying the distribution $p_m(s)$, the diversity increases and the calibration performance remains stable. As shown in Table 7, the performances in terms of diversity are improved by the $p_m(s)$ distribution. This indicates that the modification of distribution with the mask mechanism can mitigate the amplification of major interest.

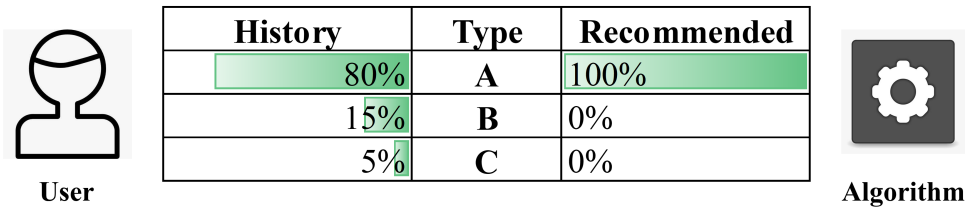


Figure 5. Illustration of amplified main interest.

Table 7. The performances of distribution $p_m(s)$ on the Tmall dataset.

Models	ILD@10	C_{KL}@10	ILD@20	C_{KL}@20
$p(s)$	0.6714	**2.0114**	0.7045	**1.6240**
$p_m(s)$	**0.7201**	2.0539	**0.7516**	1.6288

In conclusion, calibrated recommendations do not always improve the diversity. Considering calibration limits the range of recommendations in users' interacted interests, so that the diversity of recommendations may decrease. For users with homogeneous interests, their main interests are amplified by our end-to-end framework. By applying the modified preference distribution for diversity, our model further increases the diversity that can explore new interests. The proposed modification of distribution based on mask mechanism can mitigate the problem of imbalanced interests. This also indicates us that whether it is necessary to provide these users with diversified recommendations. We believe this is a question worth investigating in the future.

7. Discussion

In this paper, we proposed the DACSR model to provide accurate and calibrated recommendation lists for end-to-end sequential recommendation. We conducted experiments on benchmark datasets to demonstrate the effectiveness of our model. In general, our model achieved higher accuracy in predicting the next item and provided more calibrated recommendations compared to the post-processing-based model. This is because our

model considered the relationship between calibration and accuracy, which was isolated in post-processing-based models. Meanwhile, the end-to-end framework required much less time to provide recommendations than the post-process-based models.

We displayed the trend in the model's performance as the two main hyperparameters change. As the parameter λ varies which stands for the importance of calibration in the objective function, our model achieved better performance in terms of accuracy and calibration. We also analyzed the influence of the parameter τ which can change the predicted score distribution so that the model focused on items in different score segments.

In the ablation study, we first demonstrated the effectiveness of the proposed loss function for calibration. By applying the calibration loss function, the sequential recommendation models became aware of the preference distribution of recommended items, and aligned it to the historical preference distribution. Therefore, the process of training and prediction was conducted in an end-to-end paradigm. Furthermore, with the decoupled-aggregated framework, the positive information for the calibrated and accurate recommendation was extracted from two individual sequence encoders to improve the performance. The performance comparisons between our model and its variants verified the necessity of the decoupled-aggregated framework.

We finally investigated the effectiveness of our proposed distribution modification approaches. Because calibration is connected to diversity to a certain degree, we analyzed the relation between diversity and calibration. We also investigate the effect of imbalanced distribution of homogeneous interests. We found that it differed on the two datasets because of the item attribute coverage and the length of the sequence. For sequences with homogeneous preferences, considering calibration reduced the diversity of recommendations and amplified the main interests of the user. We adopted two distribution modification methods to improve the diversity and mitigate the effect of imbalanced distribution. The performances on two datasets verified the effectiveness of these approaches.

However, there are some limitations in our work:

- Calibration is not always equivalent to diversity, as analyzed in Section 6.4. For users with homogeneous interests, calibration decreases the performance of diversity. Therefore, user studies need to be conducted to analyze the acceptance of these users towards diversified and calibrated recommendations. This can provide data and theory support for recommendations.
- Our model followed the conventional training framework which takes a sequence as input, predicts scores of all items and is optimized by a certain loss function. However, other advanced technologies and theories can be incorporated, such as contrastive learning [38,39], few-shot learning [11,40] and game theory [41,42]
- We focused on providing calibrated recommendations for sequential recommendation models by designing loss functions and the decoupled-aggregated framework. From a different perspective, it would be valuable to investigate the reasons that cause miscalibration in recommendation. For example, previous work has shown there exists correlations between popularity bias and miscalibration [31]. Besides popularity bias, whether there are other factors contributing to miscalibration and how these factors can be incorporated into the sequence recommendation model are directions worth exploring.

8. Conclusions and Future Work

In this paper, we were committed to exploring the provision of accurate and calibrated recommendations based on user behavioural sequences. We proposed a DACSR model to provide accurate and calibrated results under the end-to-end sequential recommendation framework. Specifically, we designed a loss function that estimates the preference distribution of the recommendation list by predicted scores of all items, and measures the consistency with the preference distribution of the user's historical behaviors. In addition, we proposed distribution modification approaches to improve the diversity and mitigate the effect of imbalanced interests. To better handle the goals of accuracy and calibration,

we proposed a decoupled-aggregated framework which includes two individual sequence encoders that were assigned with the accuracy and calibration objectives, respectively. Then we utilized an aggregation module to extract information from two sequence encoders to make both accurate and calibrated recommendations. According to experiments on benchmark datasets, our model can achieve better accuracy and calibration than the original sequence encoder and the post-processing methods. The ablation studies proved the effectiveness of the general architecture and the extractor net of our model. Finally, we investigated the connection between calibration and diversity, and prove the effectiveness of our proposed distribution modification approaches.

For the future work, as mentioned in the discussion section, we first want to conduct user studies about the acceptance of diversity and calibration, so that diversified and calibrated recommendations can be supported. Furthermore, besides the conventional sequential recommendation training framework, incorporating advanced technologies and theories such as contrastive learning and game theory to debiasing recommendation is also one of our interests. In addition, we are also interested in exploring the reasons that cause the miscalibration, and applying reasons to improve the calibration performance. Finally, besides the calibrated recommendation, we hope to deal with different types of bias in recommendation algorithms.

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

Funding: This work is funded by National Natural Science Foundation of China (under project No. 61907016), Science and Technology Commission of Shanghai Municipality, China (under project No. 21511100302), and Natural Science Foundation of Shanghai (under project No. 22ZR1419000). It is also supported by The Research Project of Shanghai Science and Technology Commission (20dz2260300) and The Fundamental Research Funds for the Central Universities.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We conducted experiments on two benchmark datasets: MovieLens 1M (link: https://grouplens.org/datasets/movielens/1m, accessed at 14 September 2021), and Tmall (link: https://tianchi.aliyun.com/dataset/dataDetail?dataId=53, accessed at 3 January 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based Recommendations with Recurrent Neural Networks. In Proceedings of the International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016.
2. 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 ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2021, Virtual Event, 11–15 July 2021; pp. 378–387.
3. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-Based Recommendation with Graph Neural Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, 27 January–1 February 2019; pp. 346–353.
4. Huang, J.; Zhao, W.X.; Dou, H.; Wen, J.R.; Chang, E.Y. Improving sequential recommendation with knowledge-enhanced memory networks. In Proceedings of the International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, 8–12 July 2018; pp. 505–514.
5. Lu, Y.; Zhang, S.; Huang, Y.; Wang, L.; Yu, X.; Zhao, Z.; Wu, F. Future-Aware Diverse Trends Framework for Recommendation. In Proceedings of the World Wide Web Conference, WWW 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 2992–3001.
6. Cen, Y.; Zhang, J.; Zou, X.; Zhou, C.; Yang, H.; Tang, J. Controllable Multi-Interest Framework for Recommendation. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, SIGKDD 2020, Virtual Event, CA, USA, 23–27 August 2020; pp. 2942–2951.
7. Zheng, Y.; Gao, C.; Chen, L.; Jin, D.; Li, Y. DGCN: Diversified Recommendation with Graph Convolutional Networks. In Proceedings of the World Wide Web Conference, WWW 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 401–412.

8. Parapar, J.; Radlinski, F. Towards Unified Metrics for Accuracy and Diversity for Recommender Systems. In Proceedings of the ACM Conference on Recommender Systems, RecSys 2021, Amsterdam, The Netherlands, 27 September–1 October 2021; pp. 75–84.
9. Liu, S.; Zheng, Y. Long-tail Session-based Recommendation. In Proceedings of the ACM Conference on Recommender Systems, RecSys 2020, Virtual Event, Brazil, 22–26 September 2020; pp. 509–514.
10. Kim, Y.; Kim, K.; Park, C.; Yu, H. Sequential and Diverse Recommendation with Long Tail. In Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, 10–16 August 2019; pp. 2740–2746.
11. Sreepada, R.S.; Patra, B.K. Mitigating long tail effect in recommendations using few shot learning technique. *Expert Syst. Appl.* **2020**, *140*, 112887. [CrossRef]
12. Chen, L.; Yang, Y.; Wang, N.; Yang, K.; Yuan, Q. How Serendipity Improves User Satisfaction with Recommendations? A Large-Scale User Evaluation. In Proceedings of the World Wide Web Conference, WWW 2019, San Francisco, CA, USA, 13–17 May 2019; pp. 240–250.
13. Xu, Y.; Yang, Y.; Wang, E.; Han, J.; Zhuang, F.; Yu, Z.; Xiong, H. Neural Serendipity Recommendation: Exploring the Balance between Accuracy and Novelty with Sparse Explicit Feedback. *ACM Trans. Knowl. Discov. Data* **2020**, *14*, 50:1–50:25. [CrossRef]
14. Li, P.; Que, M.; Jiang, Z.; Hu, Y.; Tuzhilin, A. PURS: Personalized Unexpected Recommender System for Improving User Satisfaction. In Proceedings of the ACM Conference on Recommender Systems, RecSys 2020, Virtual Event, Brazil, 22–26 September 2020; pp. 279–288.
15. Dai, E.; Wang, S. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. In Proceedings of the ACM International Conference on Web Search and Data Mining, WSDM 2021, Virtual Event, Israel, 8–12 March 2021; pp. 680–688.
16. Ge, Y.; Liu, S.; Gao, R.; Xian, Y.; Li, Y.; Zhao, X.; Pei, C.; Sun, F.; Ge, J.; Ou, W.; et al. Towards Long-term Fairness in Recommendation. In Proceedings of the ACM International Conference on Web Search and Data Mining, WSDM 2021, Virtual Event, Israel, 8–12 March 2021; pp. 445–453.
17. Steck, H. Calibrated recommendations. In Proceedings of the ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, 2–7 October 2018; pp. 154–162.
18. Kaya, M.; Bridge, D.G. A comparison of calibrated and intent-aware recommendations. In Proceedings of the ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, 16–20 September 2019;pp. 151–159.
19. Burke, R. Multisided Fairness for Recommendation. *arXiv* **2017**, arXiv:abs/1707.00093.
20. Seymen, S.; Abdollahpouri, H.; Malthouse, E.C. A Constrained Optimization Approach for Calibrated Recommendations. In Proceedings of the ACM Conference on Recommender Systems, RecSys 2021, Amsterdam, The Netherlands, 27 September–1 October 2021; pp. 607–612.
21. da Silva, D.C.; Manzato, M.G.; Durão, F.A. Exploiting personalized calibration and metrics for fairness recommendation. *Expert Syst. Appl.* **2021**, *181*, 115112. [CrossRef]
22. Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; Ma, J. Neural attentive session-based recommendation. In Proceedings of the ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, 6–10 November 2017; pp. 1419–1428.
23. Tang, J.; Wang, K. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In Proceedings of the ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina del Rey, CA, USA, 5–9 February 2018; pp. 565–573.
24. Kang, W.; McAuley, J.J. Self-Attentive Sequential Recommendation. In Proceedings of the IEEE International Conference on Data Mining, ICDM 2018, Singapore, 17–20 November 2018; pp. 197–206.
25. Xu, C.; Feng, J.; Zhao, P.; Zhuang, F.; Wang, D.; Liu, Y.; Sheng, V.S. Long- and short-term self-attention network for sequential recommendation. *Neurocomputing* **2021**, *423*, 580–589. [CrossRef]
26. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In Proceedings of the International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, 3–7 November 2019; pp. 1441–1450.
27. de Souza Pereira Moreira, G.; Rabhi, S.; Lee, J.M.; Ak, R.; Oldridge, E. Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation. In Proceedings of the ACM Conference on Recommender Systems, RecSys 2021, Amsterdam, The Netherlands, 27 September–1 October 2021; pp. 143–153.
28. Wang, Z.; Wei, W.; Cong, G.; Li, X.; Mao, X.; Qiu, M. Global Context Enhanced Graph Neural Networks for Session-based Recommendation. In Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, 25–30 July 2020; pp. 169–178.
29. Chen, W.; Ren, P.; Cai, F.; Sun, F.; de Rijke, M. Improving End-to-End Sequential Recommendations with Intent-aware Diversification. In Proceedings of the International Conference on Information and Knowledge Management, CIKM 2020, Virtual Event, Ireland, 19–23 October 2020; pp. 175–184.
30. Tan, Q.; Zhang, J.; Yao, J.; Liu, N.; Zhou, J.; Yang, H.; Hu, X. Sparse-Interest Network for Sequential Recommendation. In Proceedings of the ACM International Conference on Web Search and Data Mining, WSDM 2021, Virtual Event, Israel, 8–12 March 2021; pp. 598–606.

31. Abdollahpouri, H.; Mansoury, M.; Burke, R.; Mobasher, B. The Connection Between Popularity Bias, Calibration, and Fairness in Recommendation. In Proceedings of the ACM Conference on Recommender Systems, RecSys 2020, Virtual Event, Brazil, 22–26 September 2020; pp. 726–731.
32. Zhao, X.; Zhu, Z.; Caverlee, J. Rabbit Holes and Taste Distortion: Distribution-Aware Recommendation with Evolving Interests. In Proceedings of the Web Conference, WWW 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 888–899.
33. Zhang, L.; Wang, P.; Li, J.; Xiao, Z.; Shi, H. Attentive Hybrid Recurrent Neural Networks for sequential recommendation. *Neural Comput. Appl.* **2021**, *33*, 11091–11105. [CrossRef]
34. Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; Chi, E.H. Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, SIGKDD 2018, London, UK, 19–23 August 2018; pp. 1930–1939.
35. Tang, H.; Liu, J.; Zhao, M.; Gong, X. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations. In Proceedings of the ACM Conference on Recommender Systems, RecSys 2020, Virtual Event, Brazil, 22–26 September 2020; pp. 269–278.
36. 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, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
37. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
38. Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; Cui, B. Contrastive Learning for Sequential Recommendation. In Proceedings of the IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 1259–1273.
39. Chen, Y.; Qian, W.; Liu, D.; Su, Y.; Zhou, Y.; Han, J.; Li, R. Contrastive Learning for Session-Based Recommendation. In Proceedings of the International Conference on Artificial Neural Networks, ICANN 2022, Bristol, UK, 6–9 September 2022; pp. 358–369.
40. Wang, Y.; Yao, Q. Few-shot Learning: A Survey. *arXiv* **2019**, arXiv:abs/1904.05046.
41. Arena, P.; Fazzino, S.; Fortuna, L.; Maniscalco, P. Game theory and non-linear dynamics: The Parrondo Paradox case study. *Chaos Solitons Fractals* **2003**, *17*, 545–555. [CrossRef]
42. Benkessirat, S.; Narhimene, B.; Nachida, R. A New Collaborative Filtering Approach Based on Game Theory for Recommendation Systems. *J. Web Eng.* **2021**, *20*, 303–326. [CrossRef]