Disentangling Long and Short-Term Interests for Recommendation

Yu Zheng\textsuperscript{1,}\textsuperscript{†}, Chen Gao\textsuperscript{1,†}, Jianxin Chang\textsuperscript{2}, Yanan Niu\textsuperscript{2}, Yang Song\textsuperscript{2}, Depeng Jin\textsuperscript{1}, Yong Li\textsuperscript{1}

\textsuperscript{1}Beijing National Research Center for Information Science and Technology.
Department of Electronic Engineering, Tsinghua University
\textsuperscript{2}Beijing Kuaishou Technology Co., Ltd.

ABSTRACT

Modeling user’s long-term and short-term interests is crucial for accurate recommendation. However, since there is no manually annotated label for user interests, existing approaches always follow the paradigm of entangling these two aspects, which may lead to inferior recommendation accuracy and interpretability. In this paper, to address it, we propose a Contrastive learning framework to disentangle Long and Short-term interests for Recommendation (CLSR) with self-supervision. Specifically, we first propose two separate encoders to independently capture user interests of different time scales. We then extract long-term and short-term interests proxies from the interaction sequences, which serve as pseudo labels for user interests. Then pairwise contrastive tasks are designed to supervise the similarity between interest representations and their corresponding interest proxies. Finally, since the importance of long-term and short-term interests is dynamically changing, we propose to adaptively aggregate them through an attention-based network for prediction. We conduct experiments on two large-scale real-world datasets for e-commerce and short-video recommendation. Empirical results show that our CLSR consistently outperforms all state-of-the-art models with significant improvements: GAUC is improved by over 0.01, and NDCG is improved by over 4%. Further counterfactual evaluations demonstrate that stronger disentanglement of long and short-term interests is successfully achieved by CLSR. The code and data are available at https://github.com/tsinghua-fib-lab/CLSR.

CCS CONCEPTS

• Information systems → Personalization.

KEYWORDS

Recommendation, Long and Short-Term Interests, Self-supervised Learning, Disentanglement Learning

1 INTRODUCTION

With the deluge of information growing rapidly, recommender systems have been playing crucial roles in numerous online services, such as news [2], e-commerce [51], videos [10, 27], etc. Specifically, recommender systems provide personalized contents by first inferring users’ interests from their historical interactions and then retrieving items that meet these interests. In practice, however, users’ interest are difficult to track since they tend to have both stable long-term interests and dynamic short-term interests. For example, a tech-savvy user may always be willing to browse electronics (long-term interest), while he may also exhibit interest in clothes in a short period (short-term interest). As a result, accurately modeling and distinguishing users’ long and short-term (LS-term) interests is critical.

Let us first review the literature. Collaborative filtering (CF) based recommenders [15, 16, 23, 35, 51] mainly capture the long-term interests and ignore the sequential features, thus they are limited in modeling the dynamic short-term interests. Consequently, sequential models [17, 41, 50, 55] were proposed to exploit convolutional neural networks or recurrent neural networks to learn sequential features of user interests. However, those methods tend to have short-term memory hence recommend items that are more relevant to users’ recent behaviors. As a result, recently, a series of approaches [2, 29, 47, 48] were proposed to combine CF-based recommenders and sequential recommenders to cover both long-term and short-term interests. Specifically, in these approaches, CF-based models such as matrix factorization are adopted for long-term interests, and sequential models are utilized to learn short-term interests. However, whether LS-term interests can be effectively captured by the corresponding models is not guaranteed, since they impose no explicit supervision on the learned LS-term interests. In other words, the learned LS-term interests in those methods can be entangled with each other [28].

Overall speaking, disentangling LS-term interests faces the following challenges.

• First, LS-term interests reflect quite different aspects of user preferences. Specifically, long-term interests can be regarded as user’s overall preferences which can remain stable for a long period of time, while short-term interests indicate a user’s dynamic
preferences that evolve rapidly according to recent interactions. Therefore, learning a unified representation of LS-term interests is insufficient to capture such differences. On the contrary, it is more proper to model the two aspects separately.

- **Second**, it is hard to obtain labeled data for learning LS-term interests. The collected behavior log data only always contains users’ implicit feedback such as clicks. Hence the separate modeling of LS-term interests lacks explicit supervision for distinguishing the two aspects.

- **Last**, for the final prediction of users’ future interactions, both long and short-term interests should be taken into consideration. Nevertheless, the importance of two kinds of interests varies on different user-item interactions. For example, users’ short-term interests are more important when they continuously browse similar items, while users’ behaviors are largely driven by long-term interests when they switch to quite different items. Therefore, it is critical but challenging to adaptively fuse these two aspects for predicting future interactions.

To address the above challenges, we propose a contrastive learning framework that disentangles LS-term interests leveraging the interaction sequences to build self-supervision signals. Specifically, in order to independently capture LS-term interests, we propose to decompose each interaction into three mechanisms: long-term interests representation, short-term interests evolution, and interaction prediction. We design two separate encoders with different dynamics over time to model LS-term interests respectively, which addresses the first challenge. To overcome the key challenge of lacking labeled data for LS-term interests, we propose to use self-supervision [7]. We first generate proxy representations for long/short-term interests by extracting users’ entire historical interactions and recent interactions, respectively. We then supervise the interest representations obtained from the two separate encoders to be more similar with their corresponding proxies than the opposite proxies, in a contrastive manner. Different from existing methods which impose no explicit supervision on the learned LS-term interests [2, 47], our self-supervised approach can learn better-disentangled representations for LS-term interests and remove the dependency on labeled data. With the disentangled interest representations, we design an attention-based fusion network that adaptively aggregates the two aspects for prediction, which solves the last challenge.

We evaluate the recommendation performance of our method on two real-world datasets. Experimental results illustrate that CLSR outperforms state-of-the-art (SOTA) methods with significant improvements. Specifically, AUC and GAUC are improved by over 0.02, and NDCG are improved by over 10.7%, which can be considered as quite promising gain by existing works [39, 47]. To further investigate the effectiveness of the self-supervised disentanglement design, we conduct counterfactual evaluations with intervened historical interaction sequences which block long or short-term interests. The results demonstrate that CLSR achieves steadily stronger disentanglement of LS-term interests against SOTA methods.

In summary, the main contributions of this paper are as follows:

- We highlight the different dynamics of users’ long and short-term interests, and take the pioneer step of disentangling the two factors is critical for accurate recommendation.

- We propose a contrastive learning framework to separately capture LS-term interests. Disentangled representations are learned with self-supervision by comparing with proxy representations constructed from the original interaction sequences. An attention-based fusion network is further designed which adaptively aggregates LS-term interests to predict interactions.

- We conduct extensive experiments on real-world datasets. Experimental results validate that our proposed CLSR achieves significant improvements against SOTA methods. Further counterfactual analyses illustrate that much stronger disentanglement of LS-term interests can be achieved by CLSR.

The remainder of the paper is organized as follows. We first formulate the problem in Section 2 and introduce the proposed method in Section 3. We then conduct experiments in Section 4, and review the related works in Section 5. Finally, we conclude the paper in Section 6.

## 2 PROBLEM FORMULATION

**Notations.** Let $M$ denote the number of users, and $\{x^u\}_{u=1}^M$ denote the interaction sequences for all users. Each sequence $x^u = [x_1^u, x_2^u, \ldots, x_{T_u}^u]$ denotes a list of items which are ordered by the corresponding interaction timestamps. Here $T_u$ denotes the length of user $u$’s interaction history, and each item $x_t^u$ is in $[1, N]$, where $N$ denotes the number of items.

Since a user’s interaction history $x^u$ reflects both long and short-term interests, the recommender system will first learn LS-term interests from $x^u$, and then predict future interactions based on the two aspects. We then can formulate the problem of learning LS-term interests for recommendation as follows:

**Input:** The historical interaction sequences for all users $\{x^u\}_{u=1}^M$.

**Output:** A predictive model that estimates the probability of whether a user will click an item, considering both LS-term interests.

## 3 METHODOLOGY

In this section, we elaborate on the proposed Contrastive learning framework of Long and Short-term interests for Recommendation (CLSR).

### 3.1 User Interests Modeling

Since users’ LS-term interests are quite different in terms of the dynamics over time, it is more appropriate to model the two aspects separately instead of using a unified representation to express them. Specifically, long-term interests are relatively stable, while short-term interests are dynamic and changing frequently. Meanwhile, each interaction is determined by both aspects as well as the target item. Therefore, we propose to frame user interests modeling as the following three separate mechanisms:

$$
\begin{align*}
\hat{\xi} &= \left\{ \begin{array}{l}
U_I = f_1(U), \\
U_s^{(t)} = f_2(U_s^{(t-1)}, V^{(t-1)}, Y^{(t-1)}, U), \\
Y^{(t)} = f_3(U_t, U_s^{(t)}, V^{(t)}, U),
\end{array} \right.
\end{align*}
$$

where $f_1$, $f_2$ and $f_3$ are the underlying functions for user $U$’s long-term interests ($U_I$), short-term interests ($U_s^{(t)}$) and interaction ($Y^{(t)}$) with item $V^{(t)}$. Current and last timestamps are denoted as $t$ and
disentangling LS-term interests means that \( U \) only captures long-term interests and \( U_s \) models pure short-term interests. Such disentanglement is helpful to achieve interpretable and controllable recommendation, since we can track and tune the importance of each aspect by adjusting the fusion weights. Meanwhile, effective adjustment of LS-term interests requires the learned representations to only contain the information of the desired aspect. Take the linear case as a toy example, suppose a recommendation model entangles LS-term interests as follows,

\[
U'_l = 0.6U_l + 0.4U_s, \quad U'_s = 0.4U_l + 0.6U_s, \tag{4}
\]

\( t = 1 \), respectively. It is worthwhile noting that \( U \) denotes user profile, which contains the user ID and the interaction history \( x^U \).

The proposed user interests modeling \( \zeta \) decomposes each interaction into three mechanisms: \( f_1 \) long-term interests representation, \( f_2 \) short-term interests evolution, and \( f_3 \) interaction prediction, which are briefly illustrated in Figure 1. We now explain the details of the three mechanisms.

- **Long-term Interests Representation in Eqn (1).** Long-term interests reflect a holistic view of user preferences, and hence it is stable and less affected by recent interactions. In other words, long-term interests can be inferred from the entire historical interaction sequence, thus we include \( U \) as the input of \( f_1 \), which contains the interaction history \( x^U \).

- **Short-term Interests Evolution in Eqn (2).** Short-term interests are evolving as users continuously interact with recommended items. For example, users may establish new interests after clicking an item. Meanwhile, users may also gradually lose certain interests. That is to say, short-term interests are time-dependent variables, and thus in \( f_2 \), short-term interests \( U_s(t) \) at timestamp \( t \) are evolved recursively from \( U_s(t-1) \), affected by the last interaction \( V(t-1) \).

- **Interaction Prediction in Eqn (3).** When predicting future interactions, whether long or short-term interests play a more important role depends on a wide variety of aspects, including the target item \( V(t) \) and the interaction history \( x^U \) of \( U \) [47]. Therefore, we fuse \( U_l \) and \( U_s(t) \) according to \( V(t) \) and \( U \) in an adaptive manner to accurately predict interactions.

Disentangling LS-term interests means that \( U_l \) only captures long-term interests and \( U_s \) models pure short-term interests. Such disentanglement is helpful to achieve interpretable and controllable recommendation, since we can track and tune the importance of each aspect by adjusting the fusion weights. Meanwhile, effective adjustment of LS-term interests requires the learned representations to only contain the information of the desired aspect. Take the linear case as a toy example, suppose a recommendation model entangles LS-term interests as follows,

\[
U'_l = 0.6U_l + 0.4U_s, \quad U'_s = 0.4U_l + 0.6U_s, \tag{4}
\]
We now introduce the proposed encoders for LS-term interests. To extract long-term interests, which reflect recent interactions, we utilize another attentive network on top of a recurrent neural network. As introduced previously, long-term interests provide a holistic view of user preferences which we use to supervise the two interests encoders. Specifically, we calculate the mean representation of the entire interaction history as the proxy for long-term interests, and use the average representation of recent $k$ interactions as the proxy for short-term interests. Formally, the proxies of LS-term interests for a given user $u$ at timestamp $t$ can be calculated as follows,

$$p_{1}^{u,t} = \text{MEAN}(x_{t}^{u}, \ldots, x_{T}^{u}),$$  

where $E(x)$ means the embedding of item $x$. Note that we only calculate proxies when the sequence length is longer than a threshold $l_t$, since there is no need to distinguish long and short-term if the whole sequence only contains a few items [26]. The threshold $l_t$, the length of the recent-behavior sequence $k$ are hyper-parameters in our method. Furthermore, we use mean pooling here for its simplicity and the performance turns out to be good enough. In fact, our self-supervised paradigm is capable of exploiting more complex design for proxies which we leave for future work.

With proxies serving as labels, we can utilize them to supervise the disentanglement of LS-term interests. Specifically, we perform contrastive learning between the encoder outputs and proxies, which requires the learned representations of LS-term interests to be more similar to their corresponding proxies than the opposite proxies. We illustrate the contrastive tasks in Figure 2 (A). Formally, there are four contrastive tasks as follows:

$$\text{sim}(u_{1}^{u,t}, p_{1}^{u,t}) > \text{sim}(u_{1}^{u,t}, p_{s}^{u,t}),$$  

$$\text{sim}(p_{s}^{u,t}, u_{1}^{u,t}) > \text{sim}(p_{s}^{u,t}, u_{s}^{u,t}),$$  

$$\text{sim}(u_{s}^{u,t}, p_{s}^{u,t}) > \text{sim}(u_{s}^{u,t}, p_{1}^{u,t}),$$  

$$\text{sim}(p_{s}^{u,t}, u_{s}^{u,t}) > \text{sim}(p_{s}^{u,t}, u_{s}^{u,t}),$$

where Eqn (19)-(20) supervise long-term interests, and Eqn (21)-(22) supervise short-term interests, and $\text{sim} (\cdot , \cdot)$ measures embedding similarity. Take long-term interests modeling as an example, Eqn (19) encourages the learned long-term interests representation, $u_{1}^{u,t}$, to be more similar to the long-term proxy, $p_{1}^{u,t}$ than the short-term proxy, $p_{s}^{u,t}$. Meanwhile, Eqn (20) requires that $u_{1}^{u,t}$ is closer to $p_{1}^{u,t}$ compared with the short-term interests representation, $u_{s}^{u,t}$. With four symmetric contrastive tasks on the similarity between encoder outputs and proxies, we add self-supervision on LS-term interests modeling which can achieve stronger disentanglement compared with existing unsupervised approaches.

We implement two pairwise loss functions based on Bayesian Personalized Ranking (BPR) [35] and triplet loss to accomplish contrastive learning in Eqn (19)-(22). Formally, the two loss functions, which use inner product and Euclidean distance to capture embedding similarity, are computed as follows,

$$\mathcal{L}_{\text{bpr}}(a, p, q) = \sigma((a, q) - (a, p)),$$  

$$\mathcal{L}_{\text{tri}}(a, p, q) = \max (d(a, p) - d(a, q) + m, 0),$$

where $\sigma$ is the softplus activation function, $(\cdot, \cdot)$ denotes inner product of two embeddings, $d$ denotes the Euclidean distance, and $m$...
denotes a positive margin value. Both \( L_{bpr} \) and \( L_{tri} \) are designed for making the anchor \( a \) more similar to the positive sample \( p \) than the negative sample \( q \). Thus the contrastive loss for self-supervised disentanglement of LS-term interests can computed as follows,

\[
L_{con}^{u,t} = f(u_t; p_t; p_x) + f(p_t; u_t; u_x) + f(u_x; p_t; p_t) + f(p_x; u_x; u_t) 
\]

(25)

where we omit the superscript of interest representations and proxies, and \( f \) can be either \( L_{bpr} \) or \( L_{tri} \).

**Remark.** Users’ LS-term interests can also overlap with each other to some extent. For example, a user who only purchases clothes on an e-commerce application tends to have consistent LS-term interests. Therefore, unlike existing disentangled recommendation approaches [43, 49] which add an independence constraint forcing the learned disentangled factors to be dissimilar with each other, we do not include such regularizarion term and only supervise the learned representations of LS-term interests to be similar with their corresponding proxies. This is also why we do not use loss functions like InfoNCE [33] which impose too strong punishment on the similarity between opposite encoders and proxies.

In summary, we implement two separate encoders \( \phi \) and \( \psi \) to learn representations for LS-term interests, respectively. In order to achieve disentanglement of LS-term interests, we compute proxies from the historical interaction sequences. We further propose contrastive-learning loss functions that guide the two encoders only to capture the desired aspect in a self-supervised manner.

### 3.2.5 **Adaptive Fusion for Interaction Prediction.**

With the learned disentangled representations by self-supervised learning, how to aggregate the two aspects to predict interactions remains a challenge. Simple aggregators, such as sum and concatenation, assume that contributions of LS-term interests are fixed, which is invalid in many cases. In fact, whether long or short-term one is more important depends on the historical sequence. For example, users are mainly driven by short-term interests when they are continuously browsing items from the same category. Meanwhile, the importance of LS-term interests also depends on the target item. For instance, a sports lover may still click on a recommended bicycle due to long-term interests, even after he/she browses several books. Therefore, we include both the historical sequence and the target item as input of the aggregator, where historical sequence is compressed with a GRU. The proposed attention-based adaptive fusion model is illustrated in Figure 2 (D), which dynamically determines the importance of LS-term interests to aggregate \( u^t \) and \( u^s \). Formally, the final fused interests are obtained as follows,

\[
h^u_t = \text{GRU}((E(x^u_t), \ldots, E(x^u_T))),
\]

(26)

\[
\alpha = \sigma(\tau_f(h^u_t||E(x^u_{t+1}))),
\]

(27)

\[
u^t = \alpha \cdot u^t + (1 - \alpha) \cdot u^s,
\]

(28)

where \( \sigma \) is the sigmoid activation function, and \( \tau_f \) is a MLP for fusion. Here \( \alpha \) denotes the estimated fusion weight based on historical interactions, target item, and user’s LS-term interests.

To predict the interaction, we use the widely adopted two-layer MLP [47] shown in Figure 2 (E). Then the estimated score given a user \( u \) and an item \( v \) at timestamp \( t + 1 \) can be predicted as follows,

\[
s^t+1_{u,v} = \text{MLP}(u^t||E(v)).
\]

(29)

### 4 EXPERIMENTS

In this section, we conduct experiments to show the effectiveness of the proposed contrastive learning framework. Specifically, we aim to answer the following research questions:

- **RQ1:** How does the proposed framework perform compared with state-of-the-art recommendation models?
- **RQ2:** Can CLSR achieve stronger disentanglement of LS-term interests against existing unsupervised baselines?
- **RQ3:** What is the effect of different components in CLSR?

**Datasets.** We conduct experiments on two datasets collected from real-world e-commerce and video platforms, Taobao\(^1\) and Kuaisihou\(^2\).

Basic statistics of the two datasets are summarized in Table 1, where Average Length indicates the average length of user interaction sequences. We leave the details of the datasets in Section A.1.

**Baselines and Metrics.** We compare CLSR with state-of-the-art methods. With respect to long-term interests modeling, we include NCF [16], DIN [51] and LightGCN [14]. For short-term interests modeling, we compare with Caser [41], GRU4REC [17], DIEN [50], SASRec [20] and SURGE [6]. We also include SLi-Rec [47] which is the state-of-the-art model of LS-term interests modeling. We evaluate the models with two widely-adopted accuracy metrics including AUC and GAUC [51], as well as two commonly used ranking metrics MRR and NDCG@K. We leave the details of baselines, implementations, and hyper-parameters in Section A.2-A.3.

#### 4.1 Overall Performance Comparison (RQ1)

We illustrate the overall performance on two adopted datasets in Table 2. From the results, we have the following observations:

\(^1\)https://www.taobao.com

\(^2\)https://www.kuaisihou.com
Table 2: Overall performance on Taobao and Kuaishou datasets. Underline means the best two baselines, bold means $p$-value $< 0.05$, * means $p$-value $< 0.01$, and ** means $p$-value $< 0.001$.

| Dataset   | Category | Method  | Taobao          | Kuaishou         |
|-----------|----------|---------|-----------------|------------------|
|           |          | AUC     | GAUC | MRR | NDCG@2 | AUC     | GAUC | MRR | NDCG@2 |
| Long-term | NCF      | 0.7128  | 0.7221 | 0.1446 | 0.0829   | 0.5559  | 0.5531 | 0.7734 | 0.8327  |
|           | DIN      | 0.7637  | 0.8524 | 0.3091 | 0.2352   | 0.6160  | 0.7483 | 0.8863 | 0.9160  |
|           | LightGCN | 0.7483  | 0.7513 | 0.1669 | 0.1012   | 0.6403  | 0.6407 | 0.8175 | 0.8653  |
|           | Caser    | 0.8312  | 0.8499 | 0.3508 | 0.2890   | 0.7795  | 0.8097 | 0.9100 | 0.9336  |
|           | GRU4REC  | 0.8635  | 0.8680 | 0.3993 | 0.3422   | 0.8156  | 0.8298 | 0.9166 | 0.9384  |
| Short-term| DIEN     | 0.8477  | 0.8745 | 0.4011 | 0.3404   | 0.7037  | 0.7800 | 0.9030 | 0.9284  |
|           | SASRec   | 0.8598  | 0.8635 | 0.3915 | 0.3340   | 0.8199  | 0.8293 | 0.9161 | 0.9380  |
|           | SURGE    | 0.8906  | 0.8888 | 0.4228 | 0.3625   | 0.8525  | 0.8610 | 0.9316 | 0.9495  |
| LS-term   | SLi-Rec  | 0.8664  | 0.8669 | 0.3617 | 0.2971   | 0.7978  | 0.8128 | 0.9075 | 0.9318  |
|           | Ours     | **0.8953** | **0.8936** | **0.4372** | **0.3788** | **0.8563** | **0.8718** | **0.9382** | **0.9544** |

- **Short-term models generally perform better than long-term models.** Long-term models fail to capture temporal patterns of user interactions, hence their performance is rather poor. From the results, we can observe that AUC of NCF, DIN, and LightGCN are all less than 0.8 on Taobao dataset and less than 0.7 on Kuaishou dataset. On the other hand, short-term models outperform long-term models in most cases. For example, SURGE is the best baseline on both datasets, which uses graph convolutional propagation and graph pooling to capture the dynamics of user interests. The better performance of short-term models comes from their ability to capture the sequential pattern of user interactions. In fact, we conduct data analysis on the interaction sequences and discover that, in average, over 31% of interacted items are of the same category as the previous item, which verifies the sequential pattern and explains the better performance of short-term models.

- **Joint modeling of LS-term interests does not always bring performance gains.** SLi-Rec is the SOTA approach that models both LS-term interests. However, the two aspects are entangled with each other, which increases model redundancy and leads to inferior accuracy. Results demonstrate that SLi-Rec is not consistently effective across different metrics and datasets. For example, SLi-Rec is the best baseline on Taobao dataset with respect to AUC, but its ranking performance is poorer than GRU4REC by about 10%, indicating that it is insufficient to disentangle LS-term interests with no explicit supervision.

- **Disentangled modeling of LS-term interests can achieve significant improvements.** CLSR outperforms baselines with significant progress. Specifically, CLSR improves GAUC by about 0.005 ($p$-value $< 0.001$) on Taobao dataset and 0.01 ($p$-value $< 0.05$) on Kuaishou dataset, against SOTA methods. Besides, NDCG is improved by about 5% on Taobao dataset. The consistent and significant progress indicate that disentangling LS-term interests is critical for accurate recommendation.

4.2 Study on Disentanglement of Long and Short-Term Interests (RQ2)

Both SLi-Rec and CLSR explicitly model LS-term interests, however, CLSR achieves the best performance while SLi-Rec shows inferior accuracy. We argue that it is because SLi-Rec entangles LS-term interests which increases the internal dependency of the model and leads to poor performance. On the contrary, CLSR disentangles LS-term interests with the help of self-supervision. In this section, we empirically prove that stronger disentanglement of LS-term interests is indeed achieved by CLSR.

4.2.1 Performance of One-side Interests. In CLSR, we utilize two separate representations for LS-term interests. Therefore, it is crucial that each side only captures the desired single aspect. In order to evaluate the effectiveness of each side, we reserve one-side interests and discard the other side of CLSR and SLi-Rec. Results on two datasets are illustrated in Figure 3, from which we can observe that CLSR outperforms SLi-Rec in all cases. Specifically, on Taobao dataset, CLSR improves AUC against SLi-Rec by about 0.03 with short-term interests and full interests. On Kuaishou dataset, the improvements of AUC are about 0.1, 0.2, and 0.4 for long-term interests, short-term interests, and full interests, respectively. It indicates that CLSR attains more meaningful representations for both LS-term interests. Moreover, for both methods on both datasets, combining LS-term interests achieves better performance than using one-side interests. This further supports our motivation to model both long and short-term interests for accurate recommendation.

4.2.2 Counterfactual Evaluation. Learning disentangled representations of underlying factors is very helpful especially when the importance of different factors changes [37, 38, 49]. For example, behaviors of higher costs, such as purchase (cost of money) in Taobao dataset and like (cost of time) in Kuaishou dataset, tend to be more driven by users’ long-term interests, and behaviors of lower costs such as click in both datasets indicate more about short-term interests, which has been acknowledged by existing works [11]. Therefore, to investigate whether CLSR achieves disentanglement of LS-term interests, we design counterfactual evaluations where
the importance of different interests changes. Specifically, we use models well-trained on click data to predict both clicked items and purchased/liked items, where the importance of LS-term interests is different. Since purchase/like behavior reflects more long-term interests, the importance of long-term interests is supposed to be higher when predicting purchase/like than click. In other words, when the model predicts purchase/like behavior, it is expected that the attention weight for long-term interests when fusing the two aspects, i.e., \( \alpha \), to be also larger than predicting click behavior.

Table 3 illustrates the AUC and the average of \( \alpha \) for clicked items and purchased/liked items. We have the following findings:

- **CLSR outperforms SLi-Rec for all behaviors.** Although predicting purchase/like with models trained on click data is challenging, AUC of CLSR is significantly larger than SLi-Rec by over 0.03. Meanwhile, the average \( \alpha \) of CLSR is much lower in all cases, unlike SLi-Rec whose average \( \alpha \) is even over 0.7 on Kuaishou dataset. In fact, low \( \alpha \) in CLSR is consistent with previous findings in Table 2 that long-term interests are less important than short-term interests, which means that LS-term interests are successfully disentangled in CLSR. On the contrary, high \( \alpha \) in SLi-Rec indicates that the learned long-term interests representations contain much information of the undesired short-term interests, i.e., the two aspects entangles with each other.

- **Since purchase/like reflects more long-term interests than click, \( \alpha \) is supposed to be also larger when predicting purchase/click.** On Taobao dataset, \( \alpha \) of CLSR for purchase behavior is larger than click by about 4%. However, for SLi-Rec, \( \alpha \) for purchase is even less than click by over 6%. On Kuaishou dataset, though \( \alpha \) for like is larger than click in both SLi-Rec and CLSR, the relative increment of \( \alpha \) for CLSR is over two times larger than SLi-Rec (+9.06% v.s. +3.91%). This further validates that CLSR achieves much stronger disentanglement of LS-term interests.

Meanwhile, we also evaluate under special cases where long or short-term interests are blocked by re-arranging interaction sequences with two protocols, namely shuffle and truncate, as illustrated in Figure 4. The details are as follows.

- **Shuffle:** The historical sequence is randomly shuffled, and thus short-term interests are removed under this protocol.

- **Truncate:** Early history is discarded and only recent history is available. Thus long-term interests are weakened.

Table 4 shows the results under shuffle protocol on two datasets. Since shuffling operation blocks short-term interests, predicting click behavior is much more difficult than the original case, while predicting purchase behavior is relatively easier. We can observe that the results in Table 4 compared with Table 3 is consistent with the expectation. Specifically, for both SLi-Rec and CLSR, AUC decreases by over 0.04 and increases by about 0.02 on click-prediction task and purchase/like-prediction task, respectively. Meanwhile, CLSR improves the AUC of click-prediction by over 0.04, and improves the MRR of purchase-prediction by over 30%, against SLi-Rec. Although short-term interests are invalid under this protocol, CLSR can still achieve better performance since LS-term interests are disentangled and long-term interests can still take effect.

We further present the results of CLSR under truncate protocol with varying available length (k) of historical sequences in Figure 5 (a). We can observe that the performance of purchase prediction improves significantly as k grows. Meanwhile, the performance of click prediction increases much slower when k grows larger. This observation verifies our assumption that short-term interests can be effectively captured by mining the recent history, while for long-term interests, it is essential to take the entire history into consideration. In addition, we also show the performance of only using long-term interests representation under truncate protocol in Figure 5 (b). We can find that the accuracy of purchase-prediction increases drastically as k getting larger, while the accuracy of click-prediction is barely changed. The different trends of click and purchase tasks confirm that the learned long-term interests representations only capture the desired interests and distill short-term interests.

In summary, by comparing CLSR and SLi-Rec, which both explicitly model LS-term interests, we empirically show that disentanglement of the two aspects is the reason of better recommendation performance. Moreover, it is insufficient to disentangle LS-term interests in an unsupervised way, and CLSR effectively overcomes the challenge of lacking labeled data with self-supervision.
we also evaluate the performance of replacing the short-term in-
Markov chains-based methods [36] and advanced deep learning
LS-Term Interests Modeling in Recommendation.

disentangled. More experimental results are left in Section A.4.
evaluations demonstrate that LS-term interests are successfully
superior performance of the proposed CLSR model. Counterfactual
disentangle long and short-term interests, and our proposed attention-
adaptive fusion of LS-term interests, and our proposed attention-
alpha all different values of fixed
we can discover that adaptive fusion outperforms
two datasets, where the dashed line represents the performance of
two aspects. Figure 7 shows the recommendation performance on
accuracy on Taobao dataset
accuracy on Kuaishou dataset
(a) Abalation of contrastive learning
(b) Performance of different \( \beta \).

Figure 6: (a) Ablation study of contrastive loss. (b) Hyper-
parameter study of \( \beta \).

Figure 7: Comparison between adaptive and fixed fusion.

4.3 Ablation and Hyper-parameter Study (RQ3)
4.3.1 Contrastive Learning. Contrastive tasks on the similarity
between learned representations and proxies for LS-term interests
help achieve stronger disentanglement than existing unsupervised
methods. We conduct ablation study to compare the performance of
CLSR with and without the contrastive loss \( L_{con} \). In addition,
we also evaluate the performance of replacing the short-term in-
terests encoder \( \psi \) with DIEN. Figure 6 (a) illustrates the results on
Kuashishou dataset. We can find that GAUC of CLSR drops over 0.01
after removing the contrastive tasks which verifies the necessity of
self-supervision. Meanwhile, adding self-supervision can also
significantly improve the performance of DIEN, which means that
CLSR can serve as a general framework to disentangle LS-term
interests for existing recommendation models. We also investigate the
performance under different loss weights of \( L_{con} \). Figure 6 (b)
illustrates the results on Kuaishou dataset. We can observe that 0.1
is an optimal value, and too large \( \beta \) may contradict with the main
interaction prediction task which leads to low accuracy.

4.3.2 Adaptive Fusion of LS-Term Interests. In CLSR, we pro-
pose to aggregate LS-term interests adaptively according to the tar-
get item and the historical sequence. Here we investigate whether
this adaptive fusion is effective. To be specific, we compare with a
static version, which means using a fixed \( \alpha \) when combining the
two aspects. Figure 7 shows the recommendation performance on
two datasets, where the dashed line represents the performance of
adaptive fusion. We can discover that adaptive fusion outperforms
all different values of fixed \( \alpha \). These results verify the necessity
of adaptive fusion of LS-term interests, and our proposed attention-
based network successfully accomplishes this goal.

To conclude, we conduct extensive experiments to show the
superior performance of the proposed CLSR model. Counterfactual
evaluations demonstrate that LS-term interests are successfully
disentangled. More experimental results are left in Section A.4.

5 RELATED WORK
LS-Term Interests Modeling in Recommendation. Traditional
Markov chains-based methods [36] and advanced deep learning
models [17, 20, 24, 25, 30, 40, 41, 50, 55] fail to distinguish be-
tween LS-term interests, since a unified representation is insufficient
to fully capture user interests. Therefore, several methods
[2, 11, 19, 29, 47, 48] were proposed that explicitly differentiate
between LS-term interests. For example, Zhao et al. [48] used matrix
factorization for long-term interests and use RNN for short-term
interests. Yu et al. [47] develop a variant of LSTM for short-term
interests and adopt asymmetric SVD [22] for long-term interests.
However, disentanglement of LS-term interests is not guaranteed
since these approaches impose no supervision on the learned inter-
ests representations. Unlike existing unsupervised approaches, we
propose a self-supervised method that attains stronger disentangle-
ment of long and short-term interests.

Self-supervised Learning in Recommendation. Self-supervised
learning [4, 7, 8, 12, 13] was recently adopted by several recom-
modation algorithms [32, 45, 46, 52]. For example, Zhou et al.
[52] developed a self-supervised sequential recommender based on
mutual information maximization. And Ma et al. [32] proposed
to supervise sequential encoders with latent intention prototypes.
However, those methods ignore the differences between long and
short-term interests, which are crucial for accurate recommenda-
tion. In our paper, we design a self-supervised learning method to
disentangle long and short-term interests for recommendation.

Disentanglement in Recommendation. Disentangled represen-
tation learning in recommendation is largely unexplored until re-
cently [31, 42, 43, 49]. Ma et al. [31] propose to learn users’ multiple
preferences based on Variational Auto-Encoders. Wang et al. [42]
leverage Knowledge Graph to learn different user intentions and
regularize them to be differ from each other. However, most of
these works fail to impose specific semantics to the learned multiple
representations because of lacking labeled data, i.e. unsupervised
disentanglement, which has been shown to be ineffective [28].
In this paper, we propose to disentangle with self-supervision by de-
signing contrastive tasks between the learned representations and
interest proxies extracted from the original interaction sequences.

6 CONCLUSION AND FUTURE WORK
In this paper, we propose to disentangle long and short-term inter-
ests for recommendation with a contrastive learning framework.
CLSR. Extensive experiments and counterfactual evaluations on
two large-scale datasets demonstrate that CLSR consistently out-
performs SOTA baselines with significant improvements. More
importantly, we empirically show that unsupervised LS-term in-
terests modeling can easily entangle the two aspects and lead to
even poorer performance. With the help of self-supervision, CLSR
can effectively disentangle LS-term interests and achieve much bet-
ter performance. As for future work, CLSR can be easily extended
since it is a highly general framework. For example, other designs
of encoders or proxies can be explored. Deploying the proposed
method to industrial systems is another important future work.

ACKNOWLEDGMENTS
This work is supported in part by The National Key Research and
Development Program of China under grant 2020AAA0106000.
This work is also supported in part by the National Natural Sci-
ence Foundation of China under 61972223, U1936217, 61971267 and
U20B2060.
[49] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In Proceedings of the Web Conference 2021. 2980–2991.

[50] Guoru Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In Proceedings of the AAAI conference on artificial intelligence, Vol. 33. 5941–5948.

[51] Guoru Zhou, Xiaqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1059–1068.

[52] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 1893–1902.

[53] Han Zhu, Daojing Chang, Ziru Xu, Pengye Zhang, Xiang Li, Jie He, Han Li, Jian Xu, and Kun Gai. 2019. Joint optimization of tree-based index and deep model for recommender systems. In Advances in Neural Information Processing Systems. 3971–3980.

[54] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning tree-based deep model for recommender systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1079–1088.

[55] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to Do Next: Modeling User Behaviors by Time-LSTM. In IJCAI, Vol. 17. 3602–3608.
A APPENDIX

A.1 Datasets
We use two datasets to conduct experiments, including a public e-commerce dataset and an industrial short-video dataset, which are also adopted by the SOTA sequential recommendation model, SURGE [6]. Both of them are in million scale and collected from real-world applications.

The details of the adopted datasets are introduced as follows,
• Taobao\textsuperscript{3}. This dataset \cite{li2020session} is collected from the largest e-commerce platform in China, and it is widely used as a benchmark dataset for recommendation research \cite{pan2019dssm, li2019dssm, li2020session}. It contains the user behaviors, including click, cart, and purchase from November 25 to December 3, 2017. We use the click data and adopt 10-core settings to filter out inactive entities. To evaluate the recommendation performance, we use all the instances till December 1 as training data. We use the instances on December 2 for validation and evaluate the final performance with the instances on December 3.
• Kuaishou\textsuperscript{4}. This industrial dataset is collected from Kuaishou APP, one of the largest short-video platforms in China. Users can browse short videos uploaded by other users. We extract a subset of the logs from October 22 to October 28, 2020. The dataset contains user interactions with short videos, including click, like, follow (subscribe), and forward. We use the click data and also adopt 10-core settings to guarantee data quality. We keep the instances of the first 6 days as training set, and reserve the last day for validation (before 12 pm) and test (after 12 pm).

Table 5 shows the statistics of the two datasets after splitting.

A.2 Baselines
We compare the proposed approach with the following competitive recommenders:
• NCF \cite{he2017neural}: This method is the state-of-the-art general recommender which combines matrix factorization and multi-layer perceptrons to capture the non-linearity of user interactions.
• DIN \cite{dong2018deep}: This method uses attention mechanism to aggregate the historical interaction sequences. Attention weights are computed according to the target item.
• LightGCN \cite{he2020lightgcn}: This method is the state-of-the-art GCN based recommender and it utilizes neighborhood aggregation to capture the collaborative filtering effect.
• Caser \cite{lin2019neural}: This method regards the sequence of items as images and extract sequential patterns with a convolutional network.
• GRU4REC \cite{hidasi2015session}: This is the first approach that applies RNN to session-based recommendation system, with modified mini-batch training and ranking loss.
• DIEN \cite{fang2019dien}: This method improves DIN by combining attention with GRU to model the sequential pattern of user interests, and takes interests evolution into consideration.
• SASRec \cite{li2018self}: This method is the state-of-the-art sequential recommendation model which utilizes self-attention to capture sequential preferences.

\textsuperscript{3}https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
\textsuperscript{4}https://www.kuaishou.com

| Table 5: Statistics of the datasets. |
|--------------------------------------|
| dataset     | train       | validation | test         |
|------------|-------------|------------|--------------|
| Taobao     | 1,094,775   | 191,946    | 184,434      |
| Kuaishou   | 12,925,390  | 641,580    | 1,385,689    |

| Table 6: Performance of different $k$ on Taobao dataset. |
|-------------|-------------|-------------|-------------|
| $k$ | AUC | GAUC | MRR | NDCG@2 |
| 1  | 0.8975 | 0.8927 | 0.4306 | 0.3717 |
| 2  | 0.8956 | 0.8938 | 0.4364 | 0.3798 |
| 3  | 0.8953 | 0.8936 | 0.4372 | 0.3788 |
| 4  | 0.8936 | 0.8924 | 0.4331 | 0.3747 |

• SURGE \cite{6}: This is the state-of-the-art recommendation approach which utilizes graph convolutional networks (GCN) to model user interest from sequential interactions.
• SLi-Rec \cite{47}: This is the state-of-the-art algorithm which captures long-term interests with asymmetric-SVD and models short-term interests with a modified LSTM.

A.3 Implementation Details
We implement all the models with the Microsoft Recommenders framework \cite{3} based on TensorFlow \cite{1}. We use the Adam optimizer \cite{21}. Embedding size $d$ is set as 40. We use a two-layer MLP with hidden size [100, 64] for interaction estimation. Batch normalization is enabled for the MLP, and the activation function is ReLU. The maximum length for user interaction sequences is 50 for Taobao dataset and 250 for Kuaishou dataset. We use grid-search to find the best hyper-parameters. The optimal settings for our proposed implementation are: $L_2$ regularization weight is 1e-6. Batchsize is 500. Learning rate is 0.001. $\beta$ is 0.1. $I_i$ is 5 for Taobao dataset and 10 for Kuaishou dataset. $k$ is 3 for Taobao dataset and 5 for Kuaishou dataset. $L_{con}$ is $L_{tri}$ for Taobao dataset and $L_{bpr}$ for Kuaishou dataset.

A.4 More Studies on the Proposed Method
In this section, we conduct experiments to investigate how the proposed method performs under different values of several introduced hyper-parameters. We also include further ablation studies on several components.

Short-term Proxy $k$. In the proposed method, we use mean pooling of the recent $k$ interacted items as the proxy representation for short-term interests. Table 6 illustrates the results of different $k$ on Taobao dataset. We can observe that setting $k$ as 1 achieves poorer performance except for AUC, which means only using the last interacted item as proxy for short-term interests is not a good choice since one interaction can be noise with large possibilities.

Interests Evolution Short-term interests are quite different from long-term interests with respect to their dynamics over time, thus we utilize a GRU to generate query vectors in Eqn (7) which simulates the evolution of short-term interests. We study the effect of interests evolution and results are shown in Table 7. We can
observe that removing interests evolution causes a significant decrease of accuracy on both datasets, which confirms the necessity of modeling different semantics of LS-term interests.

**Study of Different Design Choices** We further compare different design choices in CLSR. Specifically, we investigate different options for short-term interests encoder and contrastive loss function in Eqn (14) and (25). For the RNN $\rho$ in the short-term interests encoder $\psi$, we compare LSTM [18], GRU [9], and Time4LSTM proposed by SLi-Rec [47]. For $L_{con}$, we compare BPR loss and triplet loss. Table 8 shows the results of different design choices. We can observe that Time4LSTM outperforms LSTM and GRU on both datasets, indicating that the time interval feature is helpful for LS-term interests modeling, which is ignored by LSTM and GRU. As for contrastive loss, each loss function fails to consistently outperform the competitor, which can be explained by the different scales of the two datasets. In fact, CLSR is a highly general framework in which many sequential encoders and loss functions can be utilized. We leave the further study as future work.

**Fusion Predictor GRU** In the proposed adaptive fusion model based on the attention technique, we incorporate both the target item and the historical sequence to predict whether the next interaction is driven by long or short-term interests. Specifically, we adopt a separate GRU that takes the historical sequence as input, and we use the final state as the input of MLP. We conduct experiments to investigate whether taking the historical sequence into consideration is necessary. Table 9 illustrates the results of the proposed method with and without the fusion predictor GRU. We can observe that removing the fusion predictor GRU makes the recommendation performance drop significantly, which confirms that the importance of long or short-term interests is largely determined by the historical sequence.

**Attentive Encoder** As introduced in Equation (11), the proposed attentive encoder adopts a MLP to compute attention weights. The inputs of the MLP are composed of the key vector, query vector, the element-wise difference, and multiplication of key and query. We compare the MLP based attention with simple inner product based attention:

$$a'_k = \langle \psi_k, \psi_t \rangle. \quad (32)$$

Table 10 illustrates the comparison of MLP and inner product based attention. Results in Table 10 show that MLP based attention outperforms inner product-based attention with a significant margin, which indicates that the relation between user interests and historical sequences is non-linear and can not be well captured by linear operators like the inner product.

**Discrepancy Supervision** In our proposed method, we do not add an extra discrepancy loss on the LS-term interests to make them independent with each other as other works [43, 49], since we believe self-supervision is enough to accomplish disentanglement. During our experiments, we tried to add an independent loss between the two interests as, and AUC drops by 0.01, which verifies our point. It is worthwhile to notice that many existing works [5, 32, 44] also did not use the independent loss.

**More Counterfactual Evaluations** Figure 8 illustrates the AUC of click and like on Kuaisohu dataset with available history length $k$ varying from 50 to 250. Results on Kuaisohu dataset are in line with results on Taobao dataset in Figure 5(a). Specifically, AUC of like is more sensitive to the length of available history and improves drastically as $k$ increases, while AUC of click does not improve much as we increase $k$. Since like reflects more about the user’s long-term interest, it is necessary to have access to the entire user interaction sequence. Meanwhile, click is more about short-term interest, and thus it can be largely captured from the recent history, and looking back to early history will not bring further gains.

**Complexity** We use a single GPU to compare the complexity. The training time of CLSR and typical baselines on Taobao dataset are shown in Table 11. The parameter scale of CLSR is comparable with SLi-Rec (both takes 4.1Gb GPU memory).