Disentangling Long and Short-Term Interests for Recommendation

Preprint · February 2022

7 authors, including:

Yu Zheng
Tsinghua University
19 PUBLICATIONS   **364** CITATIONS

SEE PROFILE

Depeng Jin
Tsinghua University
429 PUBLICATIONS   **10,180** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Urban Computing Research View project

Social-Aware D2D Networks View project

Disentangling Long and Short-Term Interests for Recommendation

Yu Zheng[1*], Chen Gao[1†], Jianxin Chang[2], Yanan Niu[2], Yang Song[2], Depeng Jin[1], Yong Li[1]

[1]Beijing National Research Center for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University
[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 **C**ontrastive learning framework to disentangle **L**ong and **S**hort-term interests for **R**ecommendation (**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 $x^u = [x_1^u, x_2^u, ..., x_{T_u}^u]$ 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 **C**ontrastive learning framework of **L**ong and **S**hort-term interests for **R**ecommendation (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:

$$\zeta = \begin{cases} U_l = f_1(U), & (1) \\ U_s^{(t)} = f_2(U_s^{(t-1)}, V^{(t-1)}, Y^{(t-1)}, U), & (2) \\ Y^{(t)} = f_3(U_l, U_s^{(t)}, V^{(t)}, U), & (3) \end{cases}$$

where f_1, f_2 and f_3 are the underlying functions for user U's long-term interests (U_l), short term interests ($U_s^{(t)}$) and interaction ($Y^{(t)}$) with item $V^{(t)}$. Current and last timestamps are denoted as t and
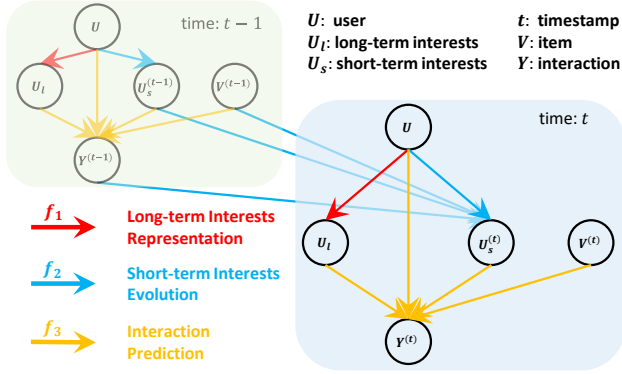
Figure 1: User interests modeling ζ (best viewed in color) which consists of three mechanisms, namely long-term interests representation (red edges), short-term interests evolution (blue edges) and interaction prediction (yellow edges).

$t - 1$, 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 ζ 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 [50]. 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 Y^{t-1} with item $V^{(t-1)}$.
- **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,

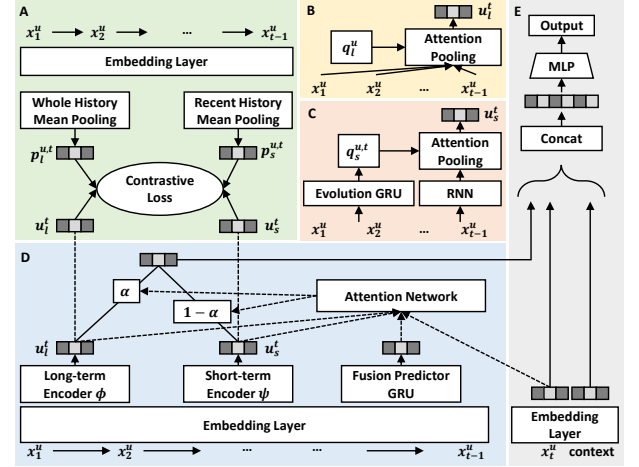$$U_l' = 0.6U_l + 0.4U_s, \quad U_s' = 0.4U_l + 0.6U_s, \tag{4}$$

Figure 2: Our proposed CLSR framework based on self-supervised learning. A) contrastive tasks on the similarity between representations and proxies of LS-term interests to enhance disentanglement; B) long-term interests encoder ϕ; C) short-term interests encoder ψ; D) adaptive fusion of LS-term interests with attention on the target item and historical interactions; E) interaction prediction network.

where U_l' and U_s' are the learned entangled interests. Given the fusion weights (importance) of LS-term interests as 0.8 and 0.2 respectively, the actual fused interests are computed as follows,

$$U_{fuse}' = 0.8U_l' + 0.2U_s' = 0.56U_l + 0.44U_s, \tag{5}$$

which is quite different from the desired interests.

However, disentangling LS-term interests is challenging since there is no labeled data for U_l and U_s. We now elaborate on our contrastive learning framework which can achieve strong disentanglement with self-supervision.

3.2 Our Self-supervised Implementation

In this section, we first provide two separate encoders to implement f_1 and f_2 which learn representations of LS-term interests. Then we introduce our designed contrastive tasks to achieve disentanglement with self-supervision. Last, we introduce the adaptive fusion model based on attention technique to accomplish f_3. The overview of CLSR is illustrated in Figure 2.

3.2.1 Generating Query Vectors for LS-term Interests. Motivated by recent works [2, 29, 47, 48] that learn LS-term interests separately with two different models, we design two separate attentive encoders, ϕ and ψ, to capture the two aspects, respectively. First, we generate query vectors for LS-term interests as follows,

$$q_l^u = \text{Embed}(u), \tag{6}$$

$$q_s^{u,t} = \text{GRU}(\{x_1^u, \cdots, x_t^u\}), \tag{7}$$

where we use a look-up embedding table and a Gated Recurrent Unit (GRU) [9] to capture different dynamics over time. In order to impose extra self-supervision on embedding similarity, all the embeddings need to be in the same semantic space. Thus, we use the historical sequence of items as *keys* of the attentive encoders, thus the obtained LS-term interests representations are in the same

item embedding space as follows,

$$u_l^t = \phi(q_l^u, \{x_1^u, \cdots, x_t^u\}), \tag{8}$$

$$u_s^t = \psi(q_s^{u,t}, \{x_1^u, \cdots, x_t^u\}), \tag{9}$$

where u_l^t and u_s^t are the learned representations of LS-term interests. We now introduce the proposed encoders for LS-term interests.

3.2.2 Long-term Interests Encoder. Figure 2 (B) illustrates the proposed long-term interests encoder ϕ. We use attention pooling to extract long-term interests representations, and the attention score of each item x_j^u can be computed as follows,

$$v_j = W_l E(x_j^u), \tag{10}$$

$$\alpha_j = \tau_l(v_j \| q_l^u \| (v_j - q_l^u) \| (v_j \cdot q_l^u)), \tag{11}$$

$$a_j = \frac{exp(\alpha_j)}{\sum_{i=1}^t exp(\alpha_i)}, \tag{12}$$

where W_l is a transformation matrix, τ_l is a multi-layer perceptrons (MLP) network, and $\|$ denotes the concatenation of embeddings. The final learned long-term interests representation is a weighted aggregation of the entire interaction history, with weights computed from the above attentive network, formulated as follows,

$$u_l^t = \sum_{j=1}^t a_j \cdot E(x_j^u). \tag{13}$$

3.2.3 Short-term Interests Encoder. Sequential patterns of user interaction play a crucial role in short-term interests modeling, thus we utilize another attentive network on top of a recurrent neural network (RNN). Specifically, we feed the historical item embeddings to a RNN model and use the output of RNN as the keys, which can be formulated as follows,

$$\{o_1^u, ..., o_t^u\} = \rho(\{E(x_1^u), ..., E(x_t^u)\}), \tag{14}$$

$$v_j = W_s o_j^u, \tag{15}$$

where W_s is a transformation matrix and ρ represents a RNN model. In Section 4, we conduct experiments to evaluate different implementations of the RNN model, including LSTM [18], GRU [9] and Time4LSTM [47]. Similar as Eqn (18) and (19), we use $q_s^{u,t}$ as the query vector, and obtain attention scores b_k. Then the learned representation for short-term interests can be computed as follows,

$$u_s^t = \sum_{j=1}^t b_j \cdot o_j^u. \tag{16}$$

Although separate encoders are adopted, disentanglement of LS-term interests is not guaranteed since u_l^t and u_s^t are extracted in an unsupervised manner [28]. Particularly, there is no labeled data to supervise the learned interests representations. Therefore, we propose to design contrastive tasks which can achieve disentanglement with self-supervision and overcome the challenge of lacking labeled data.

3.2.4 Self-supervised Disentanglement of LS-Term Interests. As introduced previously, long-term interests provide a holistic view of user preferences which summarize the entire historical interactions, while short-term interests evolve dynamically over time which reflect recent interactions. Therefore, we can obtain *proxies* for LS-term interests from the interaction sequences themselves 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,

$$p_l^{u,t} = \text{MEAN}(\{x_1^u, \cdots, x_t^u\}) = \frac{1}{t} \sum_{j=1}^t E(x_j^u), \tag{17}$$

$$p_s^{u,t} = \text{MEAN}(\{x_{t-k+1}^u, \cdots, x_t^u\}) = \frac{1}{k} \sum_{j=1}^k E(x_{t-j+1}^u), \tag{18}$$

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,

$$sim(u_l^t, p_l^{u,t}) > sim(u_l^t, p_s^{u,t}), \tag{19}$$

$$sim(p_l^{u,t}, u_l^t) > sim(p_l^{u,t}, u_s^t), \tag{20}$$

$$sim(u_s^t, p_s^{u,t}) > sim(u_s^t, p_l^{u,t}), \tag{21}$$

$$sim(p_s^{u,t}, u_s^t) > sim(p_s^{u,t}, u_l^t), \tag{22}$$

where Eqn (19)-(20) supervise long-term interests, and Eqn (21)-(22) supervise short-term interests, and $sim(\cdot, \cdot)$ measures embedding similarity. Take long-term interests modeling as an example, Eqn (19) encourages the learned long-term interests representation, u_l^t, to be more similar to the long-term proxy, $p_l^{u,t}$, than to the short-term proxy, $p_s^{u,t}$. Meanwhile, Eqn (20) requires that u_l^t is closer to $p_l^{u,t}$ compared with the short-term interests representation, u_s^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,

$$\mathcal{L}_{\text{bpr}}(a, p, q) = \sigma(\langle a, q \rangle - \langle a, p \rangle), \tag{23}$$

$$\mathcal{L}_{\text{tri}}(a, p, q) = \max\{d(a, p) - d(a, q) + m, 0\}, \tag{24}$$

where σ is the *softplus* activation function, $\langle \cdot, \cdot \rangle$ denotes inner product of two embeddings, d denotes the Euclidean distance, and m

denotes a positive margin value. Both $\mathcal{L}_{bpr}$ and $\mathcal{L}_{tri}$ 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,

$$\mathcal{L}_{\text{con}}^{u,t} = f(\boldsymbol{u}_l, \boldsymbol{p}_l, \boldsymbol{p}_s) + f(\boldsymbol{p}_l, \boldsymbol{u}_l, \boldsymbol{u}_s) + f(\boldsymbol{u}_s, \boldsymbol{p}_s, \boldsymbol{p}_l) + f(\boldsymbol{p}_s, \boldsymbol{u}_s, \boldsymbol{u}_l) \tag{25}$$

where we omit the superscript of interest representations and proxies, and f can be either $\mathcal{L}_{bpr}$ or $\mathcal{L}_{tri}$.

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 regularization 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 ϕ and ψ 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 $\boldsymbol{u}_l^t$ and $\boldsymbol{u}_s^t$. Formally, the final fused interests are obtained as follows,

$$\boldsymbol{h}_t^u = \text{GRU}(\{E(x_1^u), ..., E(x_t^u)\}), \tag{26}$$

$$\alpha = \sigma(\tau_f(\boldsymbol{h}_t^u \| E(x_{t+1}^u) \| \boldsymbol{u}_l^t \| \boldsymbol{u}_s^t), \tag{27}$$

$$\boldsymbol{u}^t = \alpha \cdot \boldsymbol{u}_l^t + (1 - \alpha) \cdot \boldsymbol{u}_s^t, \tag{28}$$

where σ is the *sigmoid* activation function, and τ_f is a MLP for fusion. Here α 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,

$$\hat{y}_{u,v}^{t+1} = \text{MLP}(\boldsymbol{u}^t \| E(v)). \tag{29}$$

Table 1: Statistics of the two datasets used in experiments.

Dataset	Users	Items	Instances	Average Length
Taobao	36,915	64,138	1,471,155	39.85
Kuaishou	60,813	292,286	14,952,659	245.88

Following the existing works' settings[47], we use the negative log-likelihood loss function as follows,

$$\mathcal{L}_{\text{rec}}^{u,t} = -\frac{1}{N} \sum_{v \in O} y_{u,v}^{t+1} \log(\hat{y}_{u,v}^{t+1}) + (1 - y_{u,v}^{t+1}) \log(1 - \hat{y}_{u,v}^{t+1}), \tag{30}$$

where O is the set composed of training pairs of one positive item x_{t+1}^u and $N - 1$ sampled negative items. We train the model in an end-to-end manner with multi-task learning on two objectives. Specifically, the joint loss function with a hyper-parameter β to balance objectives, can be formulated as follows,

$$\mathcal{L} = \sum_{u=1}^{M} \sum_{t=1}^{T_u} (\mathcal{L}_{\text{rec}}^{u,t} + \beta \mathcal{L}_{\text{con}}^{u,t}) + \lambda \|\Theta\|_2, \tag{31}$$

where $\lambda \|\Theta\|_2$ denotes the $L2$ regularization for addressing overfitting. The computation complexity of our implementation is $O((M + N)d + Q)$ where Q denotes the complexity of MLP and GRU, which is on par with the state-of-the-art SLi-Rec method [47].

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 achieves 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 Kuaishou[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.kuaishou.com

Table 2: Overall performance on Taobao and Kuaishou datasets. <u>Underline</u> means the best two baselines, bold means p-value < 0.05, * means p-value < 0.01, and ** means p-value < 0.001.

Dataset		Taobao				Kuaishou			
Category	Method	AUC	GAUC	MRR	NDCG@2	AUC	GAUC	MRR	NDCG@2
	NCF	0.7128	0.7221	0.1446	0.0829	0.5559	0.5531	0.7734	0.8327
Long-term	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	<u>0.3422</u>	0.8156	<u>0.8298</u>	<u>0.9166</u>	<u>0.9384</u>
Short-term	DIEN	0.8477	<u>0.8745</u>	<u>0.4011</u>	0.3404	0.7037	0.7800	0.9030	0.9284
	SASRec	0.8598	0.8635	0.3915	0.3340	<u>0.8199</u>	0.8293	0.9161	0.9380
	SURGE	<u>0.8906</u>	<u>0.8888</u>	<u>0.4228</u>	<u>0.3625</u>	<u>0.8525</u>	<u>0.8610</u>	<u>0.9316</u>	<u>0.9495</u>
LS-term	SLi-Rec	<u>0.8664</u>	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 performs 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, agaisnt 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
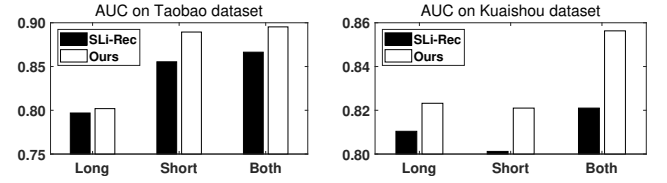
Figure 3: Comparison of using single and both interests between CLSR and Sli-Rec.

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

Table 3: Comparison between CLSR and SLi-Rec on predicting click and purchase/like.

Dataset	Method	Click		Purchase/Like	
		AUC	AVG(α)	AUC	AVG(α)
Taobao	SLi-Rec	0.8572	0.4651	0.8288	0.4350 (-6.47%)
	CLSR	0.8885	0.3439	0.8616	0.3568 (**+3.75%**)
Kuaishou	SLi-Rec	0.8153	0.7259	0.7924	0.7543 (+3.91%)
	CLSR	0.8618	0.2528	0.7946	0.2757 (**+9.06%**)

Table 4: Counterfactual evaluation under shuffle protocol.

| Dataset	Method	Click		Purchase/Like	
		AUC	MRR	AUC	MRR
Taobao	SLi-Rec	0.8092	0.2292	0.8480	0.3151
	CLSR	**0.8413**	**0.2744**	**0.8790**	**0.4194**
Kuaishou	SLi-Rec	0.7992	0.9088	0.8165	0.9113
	CLSR	**0.8431**	**0.9380**	**0.8197**	**0.9167**

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.* α, to be also larger than predicting click behavior.

Table 3 illustrates the AUC and the average of α 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 α of CLSR is much lower in all cases, unlike SLi-Rec whose average α is even over 0.7 on Kuaishou dataset. In fact, low α 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 α 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, α is supposed to be also larger when predicting purchase/click. On Taobao dataset, α of CLSR for purchase behavior is larger than click by about 4%. However, for SLi-Rec, α for purchase is even less than click by over 6%. On Kuaishou dataset, though α for like is larger than click in both SLi-Rec and CLSR, the relative increment of α 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.
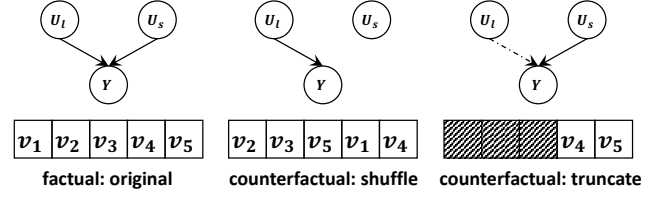
Figure 4: Counterfactual evaluation. Shuffle: short-term interests are removed by shuffling. Truncate: long-term interests are weakened by discarding early history.
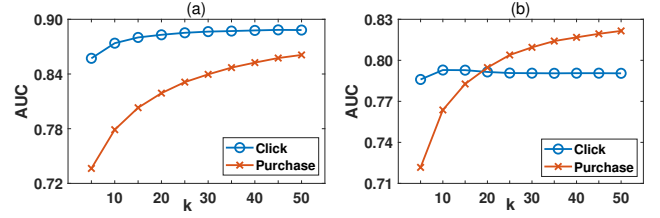
Figure 5: Counterfactual evaluation under truncate protocol. (a) CLSR. (b) CLSR with only long-term interests.

- **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 achieves 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.
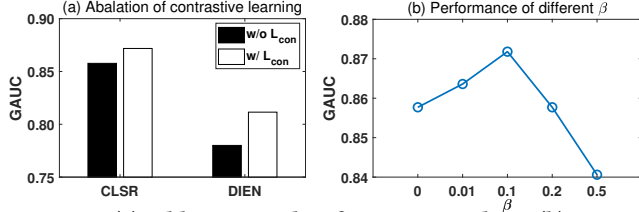
Figure 6: (a) Ablation study of contrastive loss. (b) Hyper-parameter study of β.
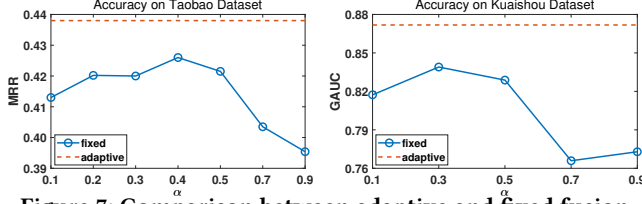
Figure 7: Comparison between adaptive and fixed fusion.

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 $\mathcal{L}_{con}$. In addition, we also evaluate the performance of replacing the short-term interests encoder ψ with DIEN. Figure 6 (a) illustrates the results on Kuaishou 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 $\mathcal{L}_{con}$. Figure 6 (b) illustrates the results on Kuaishou dataset. We can observe that 0.1 is an optimal value, and too large β 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 propose to aggregate LS-term interests adaptively according to the target 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 α 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 α. 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 between 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] use 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 interests representations. Unlike existing unsupervised approaches, we propose a self-supervised method that attains stronger disentanglement of long and short-term interests.

Self-supervised Learning in Recommendation. Self-supervised learning [4, 7, 8, 12, 13] was recently adopted by several recommendation 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 recommendation. In our paper, we design a self-supervised learning method to disentangle long and short-term interests for recommendation.

Disentanglement in Recommendation. Disentangled representation learning in recommendation is largely unexplored until recently [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 designing 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 interests for recommendation with a contrastive learning framework, CLSR. Extensive experiments and counterfactual evaluations on two large-scale datasets demonstrate that CLSR consistently outperforms SOTA baselines with significant improvements. More importantly, we empirically show that unsupervised LS-term interests 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 better 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.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*. 265–283.

[2] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 336–345.

[3] Andreas Argyriou, Miguel González-Fierro, and Le Zhang. 2020. Microsoft Recommenders: Best Practices for Production-Ready Recommendation Systems. In *Companion Proceedings of the Web Conference 2020*. 50–51.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882* (2020).

[5] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.

[6] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential Recommendation with Graph Neural Networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 378–387.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709* (2020).

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* (2020).

[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[10] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[11] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 311–320.

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733* (2020).

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9729–9738.

[14] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 639–648.

[15] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. *arXiv preprint arXiv:1808.03912* (2018).

[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[17] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[19] Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2020. Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management* 57, 2 (2020), 102142.

[20] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[22] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 426–434.

[23] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[24] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[25] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*. 322–330.

[26] Lei Li, Li Zheng, and Tao Li. 2011. Logo: a long-short user interest integration in personalized news recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*. 317–320.

[27] Yongqi Li, Meng Liu, Jianhua Yin, Chaoran Cui, Xin-Shun Xu, and Liqiang Nie. 2019. Routing micro-videos via a temporal graph-guided recommendation system. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1464–1472.

[28] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*. PMLR, 4114–4124.

[29] Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. 2019. SDM: Sequential deep matching model for online large-scale recommender system. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2635–2643.

[30] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. 2020. Memory augmented graph neural networks for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5045–5052.

[31] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 5711–5722.

[32] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled Self-Supervision in Sequential Recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 483–491.

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[34] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2671–2679.

[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[36] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[37] Bernhard Schölkopf. 2019. Causality for machine learning. *arXiv preprint arXiv:1911.10500* (2019).

[38] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proc. IEEE* 109, 5 (2021), 612–634.

[39] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.

[40] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1441–1450.

[41] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 565–573.

[42] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning Intents behind Interactions with Knowledge Graph for Recommendation. In *Proceedings of the Web Conference 2021*. 878–887.

[43] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1001–1010.

[44] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. 2020. Disenhan: Disentangled heterogeneous graph attention network for recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1605–1614.

[45] Xu Xie, Fei Sun, Zhaoyang Liu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive Pre-training for Sequential Recommendation. *arXiv e-prints* (2020), arXiv–2010.

[46] Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M Jose. 2020. Self-Supervised Reinforcement Learning forRecommender Systems. *arXiv preprint arXiv:2006.05779* (2020).

[47] Zeping Yu, Jianxun Lian, Ahmad Mahmoody, Gongshen Liu, and Xing Xie. 2019. Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation.. In *IJCAI*. 4213–4219.

[48] Wei Zhao, Benyou Wang, Jianbo Ye, Yongqiang Gao, Min Yang, and Xiaojun Chen. 2018. PLASTIC: Prioritize Long and Short-term Information in Top-n Recommendation using Adversarial Training.. In *Ijcai*. 3676–3682.

[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] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang 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] Guorui Zhou, Xiaoqiang 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, Daqing 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**[3]. This dataset [54] is collected from the largest e-commerce platform in China, and it is widely used as a benchmark dataset for recommendation research [34, 53, 54]. 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**[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** [16]: 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** [51]: This method uses attention mechanism to aggregate the historical interaction sequences. Attention weights are computed according to the target item.
- **LightGCN** [14]: This method is the state-of-the-art GCN based recommender and it utilizes neighborhood aggregation to capture the collaborative filtering effect.
- **Caser** [41]: This method regards the sequence of items as images and extract sequential patterns with a convolutional network.
- **GRU4REC** [17]: This is the first approach that applies RNN to session-based recommendation system, with modified mini-batch training and ranking loss.
- **DIEN** [50]: This method improves DIN by combining attention with GRU to model the sequential pattern of user interests, and takes interests evolution into consideration.
- **SASRec** [20]: This method is the state-of-the-art sequential recommendation model which utilizes self-attention to capture sequential preferences.

---

[3]https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
[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** [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** [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 [3] based on TensorFlow [1]. We use the Adam optimizer [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. β is 0.1. l_t is 5 for Taobao dataset and 10 for Kuaishou dataset. k is 3 for Taobao dataset and 5 for Kuaishou dataset. $\mathcal{L}_{con}$ is $\mathcal{L}_{tri}$ for Taobao dataset and $\mathcal{L}_{bpr}$ 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

Table 7: Study of interests evolution.

Dataset	Evolution	AUC	GAUC	MRR	NDCG@2
Taobao	yes	0.8953	0.8936	0.4372	0.3788
	no	0.8847	0.8884	0.4320	0.3735
Kuaishou	yes	0.8563	0.8718	0.9382	0.9544
	no	0.8202	0.8333	0.9226	0.9429

Table 8: Comparison of different design choices.

Dataset	LSTM	GRU	Time4LSTM	BPR	Triplet
Taobao	0.8872	0.8860	**0.8953**	0.8909	**0.8953**
Kuaishou	0.8240	0.8259	**0.8563**	**0.8563**	0.8102

Table 9: Study of fusion predictor GRU on Taobao dataset.

Method	AUC	GAUC	MRR	NDCG@2
w/ GRU	0.8953	0.8936	0.4372	0.3788
w/o GRU	0.8817	0.8853	0.4275	0.3692

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 ρ in the short-term interests encoder ψ, we compare LSTM [18], GRU [9], and Time4LSTM proposed by SLi-Rec [47]. For $\mathcal{L}_{con}$, 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

Table 10: Study of attentive encoder on Taobao dataset.

Attention	AUC	GAUC	MRR	NDCG@2
MLP	0.8953	0.8936	0.4372	0.3788
Inner Product	0.8684	0.8706	0.4051	0.3480
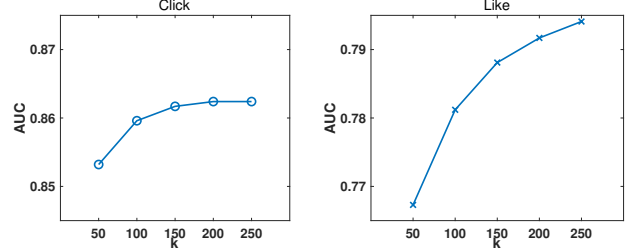
Figure 8: Counterfactual (truncate) evaluation of the proposed method on Kuaisohu dataset.

Table 11: Training time cost on Taobao dataset.

Method	GRU4REC	SLi-Rec	CLSR
Time	27.8min	26.7min	28.2min

based attention:

$$\alpha'_k = \langle v_k, u_l \rangle. \tag{32}$$

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 Kuaishou dataset with available history length k varying from 50 to 250. Results on Kuaishou 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).