

MISS: Multi-Interest Self-Supervised Learning Framework for Click-Through Rate Prediction

Wei Guo¹, Can Zhang², Zhicheng He¹, Jiarui Qin³, Huifeng Guo¹, Bo Chen¹,
Ruiming Tang¹, Xiuqiang He¹, Rui Zhang⁴

¹Huawei Noah's Ark Lab, ²National University of Singapore

³Shanghai Jiao Tong University, ⁴www.ruizhang.info

{guowei67, hezhicheng9, huifeng.guo, chenbo116, tangruiming, hexiuqiang1}@huawei.com
can.zhang@u.nus.edu, qinjr@apex.sjtu.edu.cn, rayteam@yeah.net

Abstract—CTR prediction is essential for modern recommender systems. Ranging from early factorization machines to deep learning based models in recent years, existing CTR methods focus on capturing useful feature interactions or mining important behavior patterns. Despite the effectiveness, we argue that these methods suffer from the risk of *label sparsity* (i.e., the user-item interactions are highly sparse with respect to the feature space), *label noise* (i.e., the collected user-item interactions are usually noisy), and the underuse of domain knowledge (i.e., the pairwise correlations between samples). To address these challenging problems, we propose a novel Multi-Interest Self-Supervised learning (MISS) framework which enhances the feature embeddings with interest-level self-supervision signals. With the help of two novel CNN-based multi-interest extractors, self-supervision signals are discovered with full considerations of different *interest representations* (point-wise and union-wise), *interest dependencies* (short-range and long-range), and *interest correlations* (inter-item and intra-item). Based on that, contrastive learning losses are further applied to the augmented views of interest representations, which effectively improves the feature representation learning. Furthermore, our proposed MISS framework can be used as an “plug-in” component with existing CTR prediction models and further boost their performances. Extensive experiments on three large-scale datasets show that MISS significantly outperforms the state-of-the-art models, by up to 13.55% in AUC, and also enjoys good compatibility with representative deep CTR models.

Index Terms—CTR Prediction; Multi-interest; Self-Supervised Learning;

I. INTRODUCTION

Click-Through Rate (CTR) prediction is an essential task in the domain of online advertising and recommender systems, both of which are multi-billion dollar businesses nowadays. As shown in Table I, the data involved in CTR prediction are mostly in a multi-field tabular format. Each row represents a sample¹ described by multiple *fields*² such as gender, click history, item ID, and item category. CTR prediction is to estimate the probability that a user will click an item based on the multi-field features. Due to the powerful feature representation learning ability, the mainstream of CTR prediction research is dominated by deep learning models [1]–[3]. Deep CTR prediction models have made great progresses and have been

^{*}Wei Guo, Can Zhang and Zhicheng He are co-first authors with equal contributions. Ruiming Tang and Rui Zhang are the co-corresponding authors.

¹In conventional practices, a sample is made up by a given user, a candidate item, and other associated information including profiles and behavior histories.

²The instantiation of a field is a *feature*. When there is no ambiguity, we use “field” and “feature” exchangeably.

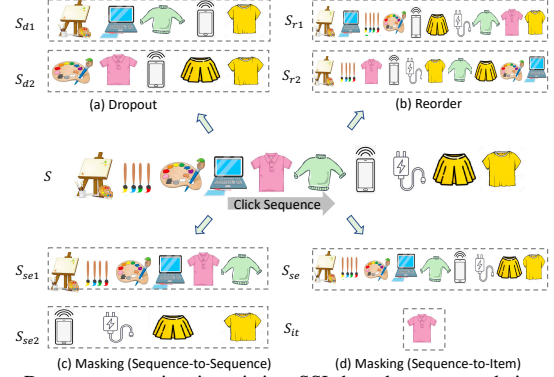


Fig. 1. Data augmentation in existing SSL-based recommendation models. (a) Dropout, (b) Reorder, (c) Sequence-to-Sequence Masking, (d) Sequence-to-Item Masking.

deployed in many commercial recommender systems, such as Wide&Deep [1] in Google Play, DeepFM [2] in Huawei AppGallery, and Deep Interest Network (DIN) [3] in Taobao.

Despite the great successes, CTR prediction models are all faced with the *label sparsity* and *label noise* problems, which deteriorate quickly with the rapid growth of data volume and feature size in online systems. What is more, existing approaches also *underuse the domain knowledge* implicitly contained in the data. Without solving the above three problems, it is difficult for existing CTR prediction models to learn effective feature representations with the sparse and noisy user-item interactions which serve as the supervision signals. In this work, we seek to utilize self-supervised learning (SSL) to solve the above mentioned three problems. On the one hand, the self-supervision signals are extracted based on the understanding of recommendation domain knowledge, which can effectively supplement the original sparse supervision signals. On the other hand, the self-supervision loss also regularizes the learned representations and filters out noises.

The basic framework for SSL mainly contains two key components: data augmentation for enhancing training data and contrastive losses for enhancing supervision signals. Though making great progress in both CV [4]–[6] and NLP [7]–[9] fields, SSL has not been fully explored in recommendation tasks. As illustrated in Figure 1, current SSL-based CTR models generally adopt three kinds of augmentation operators, i.e., *dropout*, *reorder*, and *masking*, all of which are directly introduced from CV or NLP areas without appropriate adaptation to recommendation tasks. After augmentation, each behavior

TABLE I

AN EXAMPLE OF MULTI-FIELD DATA FOR CTR PREDICTION, WHERE EACH ROW INDICATES A SAMPLE, AND EACH COLUMN REPRESENTS A FIELD.

User	Gender	City	Click History	Item	Item Category	Day	Click
Lisa	Female	New York	Honor50, iPhone12, MI11	Mate40 Pro	Cellphone	Mon.	1
David	Male	Los Angeles	Diaper, Milk powder, Shave cream	Draft beer	Beverage	Sat.	1
Yakov	Male	Moscow	Caviar, Vodka, Dark chocolate	Whisky	Wine	Sun.	0
Meimei	Female	Beijing	Umbrella, Chopsticks, Detergent	Running shoes	Outdoor	Fri.	0
Eliza	Female	London	Sun glasses, Boots, Wind-breaker	Heels	Shoes	Fri.	1
Yoshida	Male	Tokyo	Salmon, Sea urchin, Wasabi	Beer	Beverage	Wed.	1

sequence is transformed into two different new sequences (i.e., a pair of views) which are required to be similar in the contrastive learning stage. However, in the recommendation domain, it is natural that user behaviors are of *multi-interest*, as stated in [3], [10], [11], so one augmented pair of views may contain very different interests even when they are obtained from the same user behavior sequence. As a consequence, maximizing the pairwise similarities in contrastive learning may introduce noises and deteriorate representation learning in recommendation tasks.

In this paper, we study the self-supervised learning for CTR prediction with the consideration of multi-interest. To incorporate the multi-interest in user behavior sequences, we design three important interest modeling practices to better utilize the domain knowledge.

- **Point-wise and Union-wise Interest Representations.**

Within a user behavior sequence, an interest can not only be represented as a single behavior in a point-wise manner, but also can be represented as several behaviors in a union-wise manner. For example, as shown in Figure 2, the paint board, the brush, and the palette together represent user’s interest in painting tools, while notebook alone is enough to indicate an interest in electronic products. Therefore, it is important to simultaneously consider both point-wise and union-wise interest representations to provide more sufficient self-supervision signals.

- **Short-range and Long-range Interest Dependencies.** It is possible that user behaviors of one interest are interleaved with behaviors of another interest, in which case modeling long-range dependencies is necessary. For instance, in Figure 2, behaviors on electronic products (i.e., computer, phone, and charger) are interleaved by behaviors on clothes (i.e., red T-shirt and green sweater). It is also common that behaviors of an interest are consecutive without any interruption, e.g., the first three behaviors of painting tools in Figure 2, thus short-range dependencies also need to be learned. Therefore, mining long-range and short-range dependencies are complementary when modeling user behavior sequences with multiple interests.

- **Inter-item and Intra-item Interest Correlations.** Besides the inter-item interest correlations discussed above, the interest correlations between different item attributes (defined as intra-item correlation) also contain useful self-supervision signals. For examples, some people like Nike sneakers while some other people may prefer cheap slippers. Therefore, it is necessary to extract self-supervision signals from both inter-item and intra-item correlations together.

To this end, we propose a novel Multi-Interest Self-Supervised learning (MISS) framework for deep CTR models. To mine self-supervision signals from user behaviors of multi-interests, MISS proposes interest-level contrastive losses to take the place of sample-level losses. Specifically, individual

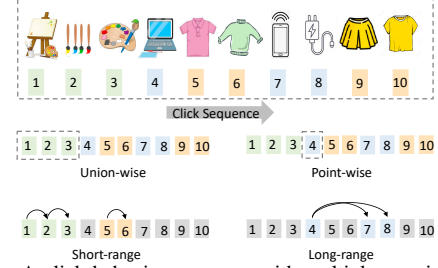


Fig. 2. A click behavior sequence with multiple user interests.

self-supervision signals are extracted for multiple interests from the user behaviors. By means of Convolutional Neural Network (CNN), both point-wise and union-wise interest representations are learned from the local correlations of behaviors on the time line, and the short-range and long-range interest dependencies are extracted by considering different distances of interest representations. Finally, the intra-item correlations are also modeled by sampling different feature combinations with convolution kernels. Furthermore, MISS serves as a model-agnostic embedding learning framework for user behavior sequence features, which is able to work compatibly with the existing deep CTR models, including methods based on both feature interactions and user interest mining.

To summarize, our work makes three major contributions as follows:

- 1) We propose a novel Multi-Interest Self-Supervised Learning framework named MISS, which enhances feature embeddings in an end-to-end manner. As far as we know, our work is the first to apply interest-level contrastive losses for recommendation tasks.
- 2) More specially, we propose CNN-based self-supervision signal extractors with full considerations of different interest representations (point-wise and union-wise), interest dependencies (short-range and long-range) and interest correlations (inter-item and intra-item). Based on that, contrastive learning is implemented to make better use of the domain knowledge and to make the best of interest-level self-supervision knowledge.
- 3) Extensive experiments demonstrate that our MISS framework not only achieves state-of-the-art performances on three large-scale datasets, but also enjoys excellent compatibility with various representative baselines.

II. RELATED WORK

A. Deep CTR Models

In early days, shallow models were proposed, such as LR [12], FM [13], Bayesian models [14], GBDT [15], and Poly2 [16]. However, shallow CTR models rely heavily on hand-crafted features and share a common bottleneck in exploiting possible feature interactions for improvements. The

last decade has witnessed the great successes of deep neural networks (DNN) in many fields like CV [17], [18] and NLP [19], [20], deep learning techniques have also been introduced to CTR prediction research. By virtue of the superior learning ability of DNNs, many recent CTR models are empowered to effectively capture high-order feature interaction patterns [1], [2], [21]–[25]. According to different model architectures, these models can be divided into two categories: feature interaction based models and user interest modeling based models. In the following, we give a brief introduction about these two kinds of models, interested readers can refer to the recent survey paper [26] for more details.

Feature interaction based models focus on learning sophisticated interactions between different features, and the representative models include Wide&Deep [1], DeepFM [2], DCN [22] and DCN-M [27]. Wide&Deep [1] learning builds a wide linear component and a DNN component to model explicit and implicit feature interactions respectively. However, manual efforts for feature engineering are still required in its wide component. To avoid such manual efforts, DeepFM [2] is thus proposed to replace the wide part with FM and share the input features between deep and wide components. DCN [22] explicitly and automatically applies feature crossing for improving accuracy and efficiency of the DNN model. DCN-M [27] further improves DCN by replacing the cross vector into a cross matrix to enhance its learning ability. Some AutoML-based models are also proposed for feature interaction modeling. AutoCTR [28] automatically discovers appropriate feature interaction architectures with a bi-level search space. AutoFIS [29] automatically selects important feature interactions for factorization models with a set of architecture parameters. However, these ML-based models require massive resources to search parameters which are usually not available for regular research groups or even mid-sized companies, so we considered it not appropriate to be included for comparison, and hence is not efficient enough for practical use, especially in industrial settings.

On the other hand, user interest modeling based models dedicate to capture important patterns from sequential behavior fields (e.g., Click History in Table I), which generally involves multiple features that can reflect user’s real interests (e.g., {Honor50, iPhone12, MI11} for Lisa). The mainstream models in this category include DIN [3], DIEN [30], and DSIN [31] which aim to use auto-regressive models to learn users’ diverse interests precisely. DIN [3] proposes a local activation unit to adaptively learn candidate-wise user interest representations from the diverse behavior sequences, based on which the CTR score is estimated. Based on DIN, DIEN [30] further proposes to capture the interest evolving process with an auxiliary loss, thus better deals with interest drifting. Considering the homogeneity of user behaviors within each session, DSIN [31] adopts self-attention and Bi-LSTM to capture intra- and inter-session interest representations respectively. To retrieve more relevant user behavior interests from long history sequences, search-based models like SIM [32] and UBR4CTR [33] have also been proposed. To better utilize the user-item relevance, DMR [34] adopts the attention networks to learn user and item representations from the user-item and item-item interaction networks.

Despite the great progresses achieved by feature interaction and user interest modeling models, there are three common problems hindering the performances of both lines of approaches, i.e., label sparsity, label noise and the underuse of domain knowledge. To tackle these three problems, we propose a self-supervised learning framework tailored for deep CTR models. Through data augmentation and interest-level contrastive learning, self-supervision signals and pairwise correlations between samples can be utilized to enhance user interest representations learned from sparse and noisy data. What is more, our proposed framework is model-agnostic which can be seamlessly applied to both feature interaction and user behavior modeling approaches, as described and verified in the following sections.

B. Self-Supervised Learning

Self-supervised learning [4], [35]–[37] has recently become an emerging trend in CV and NLP areas. SSL models enhance the learned representations with self-supervision signals extracted from unlabeled data, thus alleviating deep models’ heavy dependence on manual labels. According to the model architectures and learning objectives [38], SSL models can be categorized into two genres, i.e., generative models and contrastive models.

Generative SSL models exploit context features by modeling the generation processes, and typical examples include the BERT models [7], [8], [39]. BERT models usually use two pretext tasks for pre-training, i.e., the masked language model and the next sentence prediction task. Through the generation of missing context words or sentences, the pretext tasks effectively transform the co-occurrence information into self-supervision signals and thus improve semantic representations. On the other hand, contrastive SSL models utilize discrimination information in a “learn to compare” manner, which maximizes the correlation between similar instances [40], [41]. Following this paradigm, Deep InfoMax (DIM) [42] explicitly learns the Mutual Information Maximization (MIM) objective between features from the local patches and the whole input image. Similarly, Contrastive Predictive Coding (CPC) [41] learns MIM between audio segments and their context audios. Deep Graph InfoMax (DGI) [43] explicitly maximizes the correlation between a node and its 2-hop neighbors in the context graph with MIM.

Despite the successes achieved in CV and NLP areas, exploiting SSL in recommendation is still an under-explored task where few works have been proposed so far. To characterize the intrinsic data correlations, S3Rec [44] combines SSL with sequential recommendation by utilizing four MIM objectives, i.e., Item-Attribute MIM, Sequence-Item MIM, Sequence-Attribute MIM, and Sequence-Sequence MIM. In large-scale item recommendations, an auxiliary SSL task is employed to explore feature correlations by applying different feature masking patterns [45]. CL4SRec [46] proposes three data augmentation techniques (i.e., cropping, masking and reordering) from which two methods are randomly sampled and applied to each user sequence. Instead of performing self-supervision in the data space, [47] proposes a sequence-to-sequence training strategy to extract extra supervision signals from pairwise sub-sequences in the disentangled latent space.

TABLE II
NOTATIONS USED IN THIS PAPER.

Notations	Descriptions
$\mathcal{U}, \mathcal{U} $	The set of users and its cardinality
$\mathcal{V}, \mathcal{V} $	The set of items and its cardinality
u_i, v_j	The i -th user and the j -th item
$\mathbf{b}, \mathbf{c}, \mathbf{p}$	The item ID, category, and price sequences
$\mathbf{x}, \mathbf{x}, \mathbf{y}$	A sample, its feature vector, and ground-truth label
I, J	The numbers of categorical and sequential features
L, K	The sequence length and the embedding dimension
f_i, \mathbf{e}_i	The i -th categorical feature and its embedding
$s_j, \mathbf{e}_{j,k}$	The j -th sequential feature and its k -th embedding
\mathbf{E}, \mathbf{X}	The original feature embeddings and its aggregation
$\mathbf{a}^{(*)}, \mathbf{W}^{(*)}, \mathbf{o}^{(*)}$	The activation, weight, and bias in an MLP Layer
$\mathbf{r}_*, \mathbf{r}_*, \mathcal{T}, \mathcal{R}$	Interest representations and sequences of them
$ \mathcal{T} $	Number of interest representations
Ω	Number of feature representations for each interest
$\langle \mathbf{h}_*^{*,1}, \mathbf{h}_*^{*,2} \rangle, \mathcal{H}^*$	A pair of interest views and a sequence of pairs
$\langle \mathbf{z}_*^{*,1}, \mathbf{z}_*^{*,2} \rangle, \mathcal{Z}^*$	A pair of encoded views and a sequence of pairs
\mathbf{C}	The 3D tensor of sequential feature embeddings
$g_*, \hat{g}_*, \mathbf{G}_*, \hat{\mathbf{G}}_*$	The convolution kernels and the convolution results
Θ	The set of all model parameters

SGL [48] extends SSL to GCN-based recommendation models by augmenting ID embeddings and graph structures.

However, the above SSL recommendation models directly borrow the ideas from CV and NLP areas without carefully considering the characteristics of recommendation tasks, especially the interest diversity of individual users. In consequence, two instances generated from the same user behavior sequence are unconditionally required to be similar in contrastive learning, even if they are derived from different interests. Such one-size-fits-all practices inevitably introduce noises into representation learning, thus harm the recommendation performance. To this end, we propose a new SSL CTR framework to incorporate self-supervision signals at the interest level. By considering historical behavior dependencies under multiple interests, user representations are enhanced by better exploiting the intra-interest behavioral self-supervision while avoiding inter-interest contrastive learning.

III. PRELIMINARY

For the ease of understanding, in this section, we begin by the formal definition of the CTR prediction task and necessary notations, followed by the limitation analysis of current deep CTR models and the outline of this work.

A. Problem Formulation

In a recommender system, data samples are usually stored in a multi-field format as shown in Table I. For CTR prediction purpose, necessary features are retrieved and combined in a fixed format to describe each sample. For example, sample of user Yoshida can be represented as

[Yoshida]	[Male, ...]	[Salmon, ...]	...	[Beer]	[Beverage]	[Fri.]
User	Profile	Click Seq.	...	Item	Item Cate.	Context

Based on the collected data, CTR prediction is to estimate the probability that a user (i.e., Yoshida) will click a candidate item (i.e., Beer) under the given context (i.e., Friday).

In symbolic language, suppose there are $|\mathcal{U}|$ users $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$ and $|\mathcal{V}|$ items $\mathcal{V} = \{v_1, v_2, \dots, v_{|\mathcal{V}|}\}$, each user or item is accompanied with some attribute information such as user gender and item category, and the behavior sequence of each user is also collected and chronologically

ordered as $\mathbf{b} = \{v_1, v_2, \dots, v_L\}$, where v_l is the l -th interacted item and L is the sequence length. Thus a long raw feature vector is constructed for each sample x through the combination of categorical and sequential features:

$$\mathbf{x} = [f_1, \dots, f_i, \dots, f_I, s_1, \dots, s_j, \dots, s_J], \quad (1)$$

where f_i is a categorical feature, s_j is a sequential feature, I and J denote the numbers of categorical and sequential features respectively. Note that, besides the item ID sequence, the attribute sequences of interacted items are also useful for CTR prediction such as the category sequence $\mathbf{c} = \{c_1, c_2, \dots, c_L\}$ and price sequence $\mathbf{p} = \{p_1, p_2, \dots, p_L\}$, thus we have $\mathbf{s}_j \in \{\mathbf{b}, \mathbf{c}, \mathbf{p}\}$. However, the sequential features are not limited to these three kinds, but can also incorporate other features according to specific tasks. Take \mathbf{x} as input, a CTR model is learned to minimize the following loss function:

$$\min_{\Theta} \sum_{(x,y)} \Delta(y, \text{CTRModel}(\mathbf{x}; \Theta)), \quad (2)$$

where $y \in \{0, 1\}$ is the ground-truth click label, $\text{CTRModel}(\cdot, \cdot)$ is the CTR model with parameter set Θ , and $\Delta(\cdot, \cdot)$ is the loss function. Table II summarizes the notations used in this paper.

B. Limitation Analysis

To tackle the CTR prediction task described in the last subsection, various machine learning approaches have been proposed as described in Section II. Despite the achievements, there are two common problems that seriously influence the performances of existing CTR models, i.e., *label sparsity* and *label noise*. Here we explain these two critical problems in details.

- **Label Sparsity.** The observed user-item interactions, which serve as supervision signals for CTR models, are highly *sparse* with respect to the number of items [49], [50]. Moreover, there are numerous cold-start users and infrequent items that have very sparse history interactions, which makes the training of CTR models non-trivial.
- **Label Noise.** Besides sparsity, the collected user-item interactions are also *noisy* by two reasons. On the one hand, there exist spurious interactions derived from miss clicks or simple curiosity rather than users' true interests. On the other hand, there are items that match one user's potential interests but are not interacted due to underexposure. However, the random negative sampling process may judge them as not interesting to the user.

C. Outline of MISS

The label sparsity and label noise problems deteriorate quickly with the growth of data volume and feature size due to the Matthew Effect. In other words, popular items occupy more and more exposure chances and accumulate richer features and more supervision signals. On the contrary, unpopular items are less likely to be seen by users and suffer from the increasing lack of features and supervision. Without sufficient and correct supervision, it is difficult for CTR models to make accurate predictions. To this end, we propose a novel MISS framework to improve representation learning by means of SSL, as shown in Figure 3. Our MISS

framework is model-agnostic and can be outlined with three major contributions:

- **Interest Augmentation.** Two interest representation extractors are proposed to explore both point-wise and union-wise interest representations while considering inter-item and intra-item correlations. After that, the extracted interest representations are further augmented in consideration of the short-range and long-range dependencies, which makes them more robust to label noise.
- **Contrastive Learning.** Contrastive learning is imposed on the multi-interest representations, which effectively transforms the interest-level correlations into extra supervision signals to alleviate the lack of supervision caused by label sparsity.
- **Model Compatibility.** Under a multi-task learning framework, MISS flexibly combines the SSL component with any CTR prediction model in a plug-and-play manner, which achieves both significant performance boosts and excellent compatibility with little handcrafted model configurations.

IV. FRAMEWORK

In this section, we present the technical details of the proposed MISS framework. As illustrated in the right part of Figure 3, a typical deep CTR model DIN [3] is given as the default base model according to the experimental results in Table V, based on which we explain in detail how the MISS framework is applied in a plug-and-play mechanism. For other advanced deep CTR models, a compatibility analysis is also provided later in the experiments.

A. The Base Model

The base model consists of embedding initialization, representation learning, and CTR prediction components.

1) *Embedding Initialization:* In CTR prediction, input data samples are usually represented as high-dimensional sparse feature vectors as in Equation (1). To facilitate follow-up calculations, feature vectors are first transformed into dense real-valued embedding vectors. For the I one-hot categorical features, f_i is embedded into \mathbf{e}_i by looking up the field-wise embedding table. While a sequential feature \mathbf{s}_j is represented as a list of embedding vectors. By gathering all categorical and sequential feature embeddings, a sample is represented as a set of embedding vectors

$$\mathbf{E} = \{\mathbf{e}_1, \dots, \mathbf{e}_I, \mathbf{e}_{1,1}, \dots, \mathbf{e}_{1,L}, \dots, \mathbf{e}_{J,1}, \dots, \mathbf{e}_{J,L}\}, \quad (3)$$

where L is the sequence length.

2) *Representation Learning:* The behavior sequence length L differs from user to user, thus the number of embeddings in \mathbf{E} also varies. A naive solution is to transform all sequential features into the same length with truncation and padding. However, truncation brings information loss and padding increases redundancy. To handle this problem, researcher generally resort to different pooling techniques to aggregate the embedding sequences, which include max pooling, mean pooling, sum pooling, and the advanced attention-based pooling. In this paper, we adopt the local activation unit based pooling, which was proposed in DIN [3], as it learns to assign an adaptive weight to each feature embedding according to the target item.

For all J sequential features with length L , the embedding vectors are aggregated into the sample representation as:

$$\mathbf{X} = [\mathbf{e}_1, \dots, \mathbf{e}_I, \text{LAUP}(\{\mathbf{s}_1, \dots, \mathbf{s}_J\}, \mathbf{e}^c)], \quad (4)$$

where \mathbf{e}^c is the embedding of the candidate item, and $\text{LAUP}(\cdot, \cdot)$ is the pooling net based on the local activation unit. For space limitation, the technical details of $\text{LAUP}(\cdot, \cdot)$ is omitted here, interested readers can refer to [3]. Thus a fixed-length representation \mathbf{X} is obtained for each sample x .

3) *CTR Prediction:* Based on the integrated feature representation \mathbf{X} , a Multi-Layer Perceptron (MLP) further learns the advanced feature interactions. Suppose a D -layer MLP is adopted, each layer works as

$$\mathbf{a}^{(d)} = \sigma(\mathbf{W}^{(d)} \mathbf{a}^{(d-1)} + \mathbf{o}^{(d)}), \quad (5)$$

where $\mathbf{a}^{(d-1)}$ is the output of the previous layer, σ is the activation function, $\mathbf{W}^{(d)}$ and $\mathbf{o}^{(d)}$ are the weight matrix and bias vector respectively. We set $\mathbf{a}^{(0)} = \mathbf{X}$ for the first layer. Finally, a prediction layer is devised to predict the CTR score

$$\hat{y} = \text{Sigmoid}(\mathbf{W}^{(D+1)} \mathbf{a}^{(D)} + \mathbf{o}^{(D+1)}), \quad (6)$$

where $\text{Sigmoid}(\cdot)$ is the sigmoid activation function, and \hat{y} is the predicted CTR score. Finally, the batch-wise Logloss objective function is adopted to evaluate the predicted CTR score \hat{y} :

$$\mathcal{L}_l = -\frac{1}{|\mathcal{B}|} \sum_{(x,y) \in \mathcal{B}} y \log \hat{y} + (1-y) \log(1-\hat{y}), \quad (7)$$

where \mathcal{B} is a batch of training samples, (x, y) is the pair of sample and label in the batch, and \mathcal{L}_l is an instantiation of Equation (2).

B. The MISS Framework

The base model estimates the CTR score through embedding initialization, representation learning, and CTR prediction, as illustrated in the right part of Figure 3. Based on that, the proposed MISS framework further enhances feature embeddings with SSL by multi-interest augmentation, interest view encoding, and contrastive learning, as shown in the left part of Figure 3. In this subsection, we explain the MISS components step-by-step.

1) *Sample-Level Data Augmentation:* Data augmentation is the first step of our proposed MISS framework, based on which the contrastive learning is implemented. Given a batch of training samples $\mathcal{B} = \{x_1, x_2, \dots, x_{|\mathcal{B}|}\}$, existing SSL-based models all adopt **sample-level** data augmentation methods. For each sample x , two different views are first obtained through augmentation as:

$$\langle \mathbf{h}^1, \mathbf{h}^2 \rangle = \text{Aug}^s(\mathbf{x}), \quad (8)$$

where $\text{Aug}^s(\cdot)$ is the sample-level augmentation function, and $\langle \mathbf{h}^1, \mathbf{h}^2 \rangle$ is the pair of generated views.

After data augmentation, encoder functions are further used to extract high-level semantic representations from \mathbf{h}^1 and \mathbf{h}^2 , based on which a contrastive loss is used to make use of the self-supervision signals. However, due to the multi-interest characteristic of user behavior sequences, **sample-level** data augmentation may inevitably introduce noise. The reason is that the augmented \mathbf{h}^1 and \mathbf{h}^2 may be derived from different interests even if they are obtained from the same \mathbf{x} . To solve this problem, we put forward an **interest-level** SSL

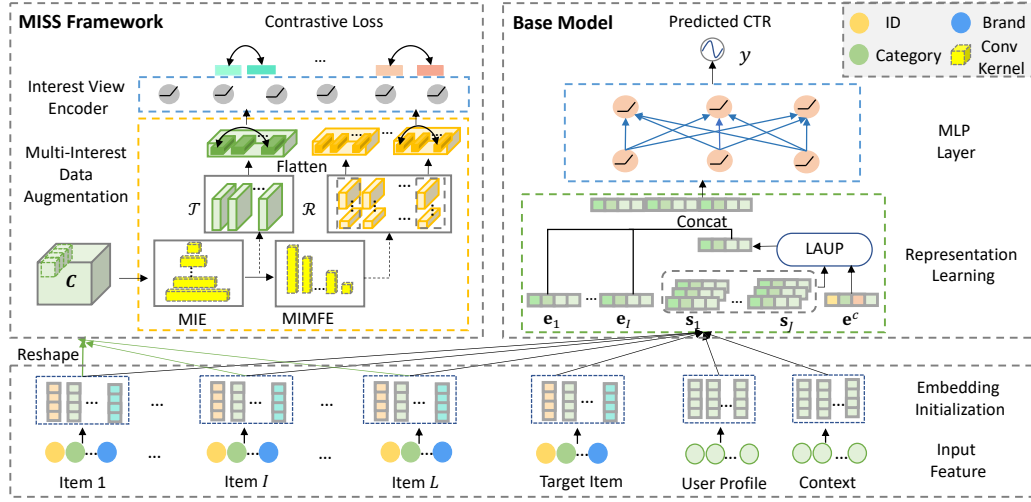


Fig. 3. Overview of our proposed MISS framework. The right part illustrates a typical deep CTR model, and the left part is the proposed multi-interest self-supervised learning component applied to it.

framework, i.e., MISS, which augments the training data at the interest level in an end-to-end fashion.

2) *Multi-Interest Data Augmentation*: In consideration of the multi-interest characteristic of user behaviors, our MISS framework implements SSL within each sample at both the interest level and the feature level. Therefore, not only the semantics provided by each training sample can be enriched, but also the modeling and utilization of long behavior sequences get promoted. To achieve these targets, we design a novel multi-interest extractor for data augmentation purpose.

To augment user behavior data at the interest level, the multiple interest representations of each user should first be extracted. An intuitive augmentation method is to directly divide the user behavior sequences according to item categories. However, item categories are usually defined in coarse granularities and are not always available in data. Therefore, we propose a CNN-based multi-interest extractor which transforms the sample feature \mathbf{x} into a group of implicit interest representations

$$\mathcal{T} = \text{MIE}(\mathbf{x}) = \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{|\mathcal{T}|}\}, \quad (9)$$

where $\text{MIE}(\cdot)$ is the multi-interest extractor network, \mathcal{T} is the output user interest representation sequence, and \mathbf{t}_k is the k -th interest representation extracted from \mathbf{x} . Moreover, for a fine-grained understanding and utilization of interest semantics, another CNN-based feature augmentation component is further designed to augment each interest representation at the feature level

$$\mathcal{R} = \text{MIMFE}(\mathbf{x}) = \{\{\mathbf{r}_{1,1}, \dots, \mathbf{r}_{1,\Omega}\}, \dots, \{\mathbf{r}_{|\mathcal{T}|,1}, \dots, \mathbf{r}_{|\mathcal{T}|,\Omega}\}\} \quad (10)$$

where $\text{MIMFE}(\cdot)$ is the multi-interest multi-feature extractor network that extracts fine-grained representations for each user interest, and Ω is the number of feature representations for each interest. After that, an augmentation function is applied to \mathcal{T} to obtain interest-level augmented views as:

$$\mathcal{H}^i = \text{Aug}^i(\mathcal{T}) = \{\langle \mathbf{h}_1^{i,1}, \mathbf{h}_1^{i,2} \rangle, \dots, \langle \mathbf{h}_P^{i,1}, \mathbf{h}_P^{i,2} \rangle\}, \quad (11)$$

where $\text{Aug}^i(\cdot)$ is the interest-level augmentation function, $\langle \mathbf{h}_p^{i,1}, \mathbf{h}_p^{i,2} \rangle$ is a pair of generated views for sample x , and P is the number of generated view pairs. Similarly, an augmen-

tation function is also applied to \mathcal{R} for a further fine-grained augmentation as:

$$\begin{aligned} \mathcal{H}^{if} &= \text{Aug}^{if}(\mathcal{R}) \\ &= \{\langle \mathbf{h}_1^{if,1}, \mathbf{h}_1^{if,2} \rangle, \dots, \langle \mathbf{h}_Q^{if,1}, \mathbf{h}_Q^{if,2} \rangle\}, \end{aligned} \quad (12)$$

where $\text{Aug}^{if}(\cdot)$ is the feature-level augmentation function, $\langle \mathbf{h}_q^{if,1}, \mathbf{h}_q^{if,2} \rangle$ is a pair of views, and Q is the number of generated pairs. In this section, we focus on the principled explanation of our MISS framework, while the technical details of $\text{MIE}(\cdot)$, $\text{MIMFE}(\cdot)$, $\text{Aug}^i(\cdot)$, and $\text{Aug}^{if}(\cdot)$ are presented later in Section V.

3) *Interest View Encoder*: With the extractor networks and augmentation functions, two sequences of augmented view pairs are obtained, i.e., \mathcal{H}^i and \mathcal{H}^{if} , where the user interest representations are augmented at different granularities. Based on the sequence \mathcal{H}^i , an encoder is adopted to explore high-order abstractions:

$$\mathcal{Z}^i = \text{Enc}^i(\mathcal{H}^i) = \{\langle \mathbf{z}_1^{i,1}, \mathbf{z}_1^{i,2} \rangle, \dots, \langle \mathbf{z}_P^{i,1}, \mathbf{z}_P^{i,2} \rangle\}, \quad (13)$$

where $\text{Enc}^i(\cdot)$ is the encoder network that transforms each interest view representation $\mathbf{h}_p^{i,1}$ (or $\mathbf{h}_p^{i,2}$) into high-order representation $\mathbf{z}_p^{i,1}$ (or $\mathbf{z}_p^{i,2}$). Similarly, an encoder $\text{Enc}^{if}(\cdot)$ is also applied to \mathcal{H}^{if} :

$$\begin{aligned} \mathcal{Z}^{if} &= \text{Enc}^{if}(\mathcal{H}^{if}) \\ &= \{\langle \mathbf{z}_1^{if,1}, \mathbf{z}_1^{if,2} \rangle, \dots, \langle \mathbf{z}_Q^{if,1}, \mathbf{z}_Q^{if,2} \rangle\}. \end{aligned} \quad (14)$$

As we mainly focus on the extraction and utilization of self-supervised signals, two simple MLPs are used to implement $\text{Enc}^i(\cdot)$ and $\text{Enc}^{if}(\cdot)$. However, other advanced networks are also applicable, such as Transformer in [46], [51], and we leave the exploration of other encoder structures to future works.

4) *Contrastive Loss*: Having obtained the high-level semantics for each augmented view, the contrastive losses can finally be applied to exploit the self-supervision signals. Following SimCLR [4], we use the InfoNCE contrastive loss [41] which attempts to maximize the similarity of positive pairs of views and minimize the agreement of negative pairs of views. As a result, similar interests can thus have similar representations (defined as *alignment*) and sufficient information are kept to distinguish different interests (defined as *uniformity*). Both

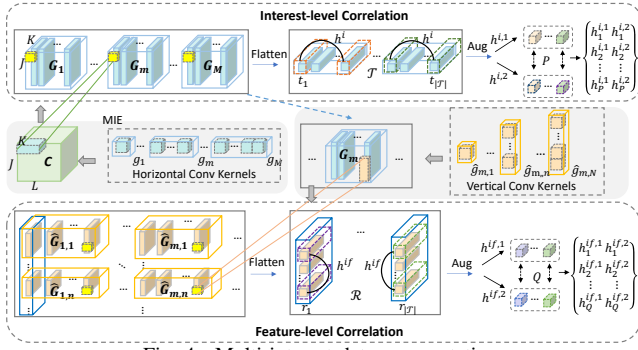


Fig. 4. Multi-interest data augmentation.

the *alignment* and *uniformity* properties are necessary and important for a good SSL system, as proved in [52]. Formally, taking $\langle \mathbf{z}_p^{i,1}, \mathbf{z}_p^{i,2} \rangle$ from the same interest as positive pairs while $\langle \mathbf{z}_p^{i,1}, \mathbf{z}_p^{j,2} \rangle$ and $\langle \mathbf{z}_p^{j,1}, \mathbf{z}_p^{i,2} \rangle$ from different samples as negative pairs, the InfoNCE loss for learning the **interest-level correlation** is formulated as:

$$\mathcal{L}_{ssl} = -\frac{1}{|\mathcal{B}|P} \sum_{x \in \mathcal{B}} \sum_{1 \leq p \leq P} \log \frac{\exp(s(\mathbf{z}_p^{i,1}, \mathbf{z}_p^{i,2})/\tau)}{\sum_{x' \in \mathcal{B}} \exp(s(\mathbf{z}_p^{i,1}, \mathbf{z}_p^{j,2})/\tau)}, \quad (15)$$

where $s(\cdot, \cdot)$ is cosine similarity function, τ is the softmax temperature parameter, and $\exp(\cdot)$ is the exponential function. Similarly, the InfoNCE loss for learning the **feature-level correlation** can be formulated as:

$$\mathcal{L}'_{ssl} = -\frac{1}{|\mathcal{B}|Q} \sum_{x \in \mathcal{B}} \sum_{1 \leq q \leq Q} \log \frac{\exp(s(\mathbf{z}_q^{i,f,1}, \mathbf{z}_q^{i,f,2})/\tau)}{\sum_{x' \in \mathcal{B}} \exp(s(\mathbf{z}_q^{i,f,1}, \mathbf{z}_q^{j,f,2})/\tau)}. \quad (16)$$

C. Multi-task Learning

To better integrate the MISS framework with the CTR prediction component, a multi-task learning strategy is adopted to jointly optimize the auxiliary SSL losses and the main prediction loss in an end-to-end manner. Thus the final loss function is formulated as:

$$\mathcal{L} = \mathcal{L}_l + \alpha_1 \cdot \mathcal{L}_{ssl} + \alpha_2 \cdot \mathcal{L}'_{ssl}, \quad (17)$$

where α_1 and α_2 are the hyper-parameters to control the strength of SSL losses. In experiments, the two-stage pre-training learning strategy is also tried, where the model is first trained with the auxiliary SSL losses, then fine-tuned by the main prediction loss.

V. MULTI-INTEREST DATA AUGMENTATION

As mentioned in Section IV, a multi-interest extractor network MIE(\cdot) and an interest-level feature extractor network MIMFE(\cdot) are used for interest representation learning at different granularities, based on which the augmentation functions $\text{Aug}^i(\cdot)$ and $\text{Aug}^{if}(\cdot)$ are further applied. In this section, we describe the technical details of these extractors and augmentation functions.

A. Multi-Interest Extractor

The multi-interest extractor network MIE(\cdot) aims to discover the potential interests from user behavior sequences. However, it is hard to achieve this goal as the number of interests varies from user to user. Moreover, the sequential pattern

of the same interest is also dynamic in terms of both different users and different time. Therefore, we propose an intuitive multi-interest extractor based on a closeness assumption, i.e., user behaviors derived from the same interest are more likely to be closely located within a sequence.

Based on the closeness assumption, we adopt CNN to extract hidden interest representations due to its effectiveness in capturing the local correlations [17], [18]. Other sequence representation learning models like RNNs or self-attention are also applicable, however, they fail to extract effective interest representation pairs for comparison. We will verify this point later in the experiments. After padding, all J sequential features share the same length L , and the embeddings in \mathbf{E} can be re-organized into a 3D tensor as:

$$\mathbf{C} = \begin{bmatrix} \mathbf{e}_{1,1}, \mathbf{e}_{1,2}, \dots, \mathbf{e}_{1,L} \\ \mathbf{e}_{2,1}, \mathbf{e}_{2,2}, \dots, \mathbf{e}_{2,L} \\ \vdots \\ \mathbf{e}_{J,1}, \mathbf{e}_{J,2}, \dots, \mathbf{e}_{J,L} \end{bmatrix} \quad (18)$$

where $\mathbf{C} \in \mathbb{R}^{J \times L \times K}$, and K is dimension of each embedding vectors $\mathbf{e}_{j,l}$. Hidden interests are extracted through horizontal convolutions along the time axis of \mathbf{C} . Take the click sequence in Figure 2 as an example, one convolution kernel may aggregate the feature embeddings of the paint board ($\mathbf{C}^{:,1,:}$), the brush ($\mathbf{C}^{:,2,:}$), and the palette ($\mathbf{C}^{:,3,:}$) into the interest in painting tools, while another kernel may take the embeddings of the notebook ($\mathbf{C}^{:,4,:}$) alone as the interest in electronic products.

Specifically, a horizontal convolution layer with M branches of kernels are adopted, as shown in left middle part of Figure 4. Denote $g_m \in \mathbb{R}^{1 \times m \times 1}$ as a kernel with width $m \in [1, M]$, where both the kernel height and channel number are set to 1. For simplification, only one kernel is used in each branch. Thus M kernels with different widths are used, which simultaneously capture the point-wise ($m = 1$) and union-wise ($m > 1$) interest representations. Each g_m slides on \mathbf{C} from left to right, and the convolution operation at the j -th row, the l -th to the $(l + m - 1)$ -th column, and the k -th channel in \mathbf{C} is formulated as:

$$G_m^{j,l,k} = \text{ReLU}(\mathbf{C}^{j,l:l+m-1,k} \circ g_m), \quad (19)$$

where $\text{ReLU}(\cdot)$ is the ReLU activation function, \circ denotes the convolution operation, $\mathbf{C}^{j,l:l+m-1,k}$ represents the sliced sub-tensor from \mathbf{C} , and l is restricted as $1 \leq l \leq (L - m + 1)$. After convolution, the final output tensor of g_m is denoted as $\mathbf{G}_m \in \mathbb{R}^{J \times (L-m+1) \times K}$, which is a combination of $(L-m+1)$ interest representations. Take all M filters as a whole, the resulting $|\mathcal{T}| = \sum_{1 \leq m \leq M} (L - m + 1)$ interest representations together make up the interest sequence

$$\begin{aligned} \mathcal{T} &= \text{MIE}(\mathbf{x}) \\ &= \{\dots, \text{Flat}(\mathbf{G}_m^{:,l,:}), \dots\} \\ &= \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{|\mathcal{T}|}\}, \end{aligned} \quad (20)$$

where $\text{Flat}(\cdot)$ is the flatten function that transforms each interest representation $\mathbf{G}_m^{:,l,:} \in \mathbb{R}^{J \times K}$ into a vector $\mathbf{t} \in \mathbb{R}^{JK}$, as shown in the upper part of Figure 4.

B. Interest-Level Augmentation

The multi-interest extractor learns different user interest representations based on the closeness assumption. However,

the closeness assumption not only holds at the behavior level, but also applies to the extracted interest representations. In other words, the more adjacent two interest representations are located on the time line, the more likely they represent the same hidden interest. Therefore, we randomly select a pair of representations as two different views of the same interest from those with the same filter g_m from \mathcal{T}

$$\begin{aligned}\mathcal{H}^i &= \text{Aug}^i(\mathcal{T}) \\ &= \{\dots, \text{RS}^i(\mathbf{G}_m), \dots\} \\ &= \{\dots, \langle \mathbf{t}_l, \mathbf{t}_{l+h} \rangle, \dots\} \\ &= \{\langle \mathbf{h}_1^{i,1}, \mathbf{h}_1^{i,2} \rangle, \dots, \langle \mathbf{h}_P^{i,1}, \mathbf{h}_P^{i,2} \rangle\},\end{aligned}\quad (21)$$

where $\text{RS}^i(\cdot)$ randomly selects two representations derived from the same convolution filter with a given distance $h \in [1, H]$. By repeating the select function $\text{RS}^i(\cdot)$ for P times, the sequence of augmented interest view pairs \mathcal{H}^i is obtained. Here we use different h values to cover both short-range and long-range interest dependencies, and a maximum distance H is pre-defined to prevent overlong dependencies. Note that, we assume a uniform distribution of interest dependency distance h . However, other complex distributions (e.g., Gaussian distribution) are also applicable, and we leave them to future works.

C. Fine-Grained Interest Extractor

The multi-interest extractor $\text{MIE}(\cdot)$ only explores the inter-item correlations of different interests along the time line, while the intra-item relationship between all J features within each interest is ignored. For example, given the resulting interest representations $\mathbf{G}_m \in \mathbb{R}^{J \times (L-m+1) \times K}$, the interest in daily supplies (say $\mathbf{G}_m^{:,l,:}$) is sensitive to price, while the interest in shoes (say $\mathbf{G}_m^{:,l,:}$) is affected by both price and category. To deal with this issue, vertical convolution operators are further applied to each resulting at the feature level for fine-grained augmentation.

As shown in the right middle part of Figure 4, N branches of vertical kernels $\hat{g}_{m,n} \in \mathbb{R}^{n \times 1 \times 1}$ are adopted to learn intra-item correlations, where $n \in [1, N]$ is the kernel height. As n varies from 1 to N , both single and collective feature representations are captured. Each kernel $\hat{g}_{m,n}$ interacts with the sliced tensor $\mathbf{G}_m^{j:j+n-1,l,k}$ from left to right, which yields the result $\hat{\mathbf{G}}_{m,n}^{j,l,k}$:

$$\hat{\mathbf{G}}_{m,n}^{j,l,k} = \text{ReLU}(\mathbf{G}_m^{j:j+n-1,l,k} \circ \hat{g}_{m,n}). \quad (22)$$

Denote the final output via $\hat{g}_{m,n}$ on \mathbf{G}_m as $\hat{\mathbf{G}}_{m,n} \in \mathbb{R}^{(J-n+1) \times (L-m+1) \times K}$, it can also be viewed as a combination of $(J-n+1)$ interest representations. However, the interest representations are now enhanced at the feature level. Through all N kernels, $|\mathcal{T}| \Omega$ enhanced interest representations are obtained, where $\Omega = \sum_{1 \leq n \leq N} (J-n+1)$. Formally,

$$\begin{aligned}\mathcal{R} &= \text{MIMFE}(\mathbf{x}) \\ &= \{\dots, \{\dots, \text{Flat}(\hat{\mathbf{G}}_{m,n}^{j,:}), \dots\}, \dots\} \\ &= \{\{\mathbf{r}_{1,1}, \dots, \mathbf{r}_{1,\Omega}\}, \dots, \{\mathbf{r}_{|\mathcal{T}|,1}, \dots, \mathbf{r}_{|\mathcal{T}|,\Omega}\}\}.\end{aligned}\quad (23)$$

D. Fine-Grained Interest-Level Augmentation

Because of the independence between each feature, a totally random select function $\text{RS}^{if}(\cdot)$ is applied to sample feature-

level interest representation views from each $\hat{\mathbf{G}}_{m,n}$, as

$$\begin{aligned}\mathcal{H}^{if} &= \text{Aug}^{if}(\mathcal{R}) \\ &= \{\dots, \text{RS}^{if}(\hat{\mathbf{G}}_{m,n}), \dots\} \\ &= \{\dots, \langle \mathbf{r}_j, \mathbf{r}_{j'} \rangle, \dots\} \\ &= \{\langle \mathbf{h}_1^{if,1}, \mathbf{h}_1^{if,2} \rangle, \dots, \langle \mathbf{h}_Q^{if,1}, \mathbf{h}_Q^{if,2} \rangle\},\end{aligned}\quad (24)$$

which is repeated for Q times.

E. Complexity Analysis

For the M branches of horizontal convolution kernels, i.e., $g_m \in \mathbb{R}^{1 \times m \times 1}$, there are m learnable parameters. As m ranges from 1 to M , the total number of learnable parameters is $\sum_{1 \leq m \leq M} m = \frac{M(M+1)}{2}$. Similarly, the total number of the N branches of vertical convolution kernels is $\sum_{1 \leq n \leq N} n = \frac{N(N+1)}{2}$. After data augmentation, two encoders are used for high-order abstraction. For simplicity, two MLP encoders are used. Thus the numbers of introduced parameters are $J \times K \times H_1^i + \sum_{2 \leq d \leq D'} H_{d-1}^i \times H_d^i$ for encoder $\text{Enc}^i(\cdot)$, and $K \times H_1^{if} + \sum_{2 \leq d \leq D'} H_{d-1}^{if} \times H_d^{if}$ for encoder $\text{Enc}^{if}(\cdot)$, where D' is the layer depth and H_d^* is the layer size of the d -th layer. All together, the total number of parameters brought by the multi-interest data augmentation in MISS is $\frac{M(M+1)+N(N+1)}{2} + J \times K \times H_1^i + K \times H_1^{if} + \sum_{2 \leq d \leq D'} H_{d-1}^i \times H_d^i + H_{d-1}^{if} \times H_d^{if}$. On the whole, M and N are very small values and the MLP parameters are also negligible compared to the embedding matrices, which makes the space complexity of MISS acceptable.

VI. EXPERIMENTS

A. Experiment Setup

1) *Datasets*: We evaluate the effectiveness of our proposed model on three large-scale datasets, i.e., *Amazon-Cds*, *Amazon-Books*, and *Alipay*. All three of them are real-world datasets described as follows:

- **Amazon Dataset**³: The Amazon dataset collects user review data from one of the largest e-commerce website in the world, i.e., amazon.com. The crawled reviews have a time span from May 1996 to July 2014. The dataset can be divided into many subsets according to the various product categories, such as Amazon-Electronics, Amazon-Cds, and Amazon-Books. In this paper, we pick the Amazon-Cds and Amazon-Books subsets for experiments.
- **Alipay**⁴: The Alipay dataset is provided in the IJCAI-16 contest which is collected from the Tmall.com website, the Taobao.com website, and the Alipay App. It contains user behavior logs between July 1st to November 30th in 2015. Each log contains multiple feature fields, including user ID, item ID, seller, category, online action type, and timestamp. We take the click behaviors as users' interaction records to construct the user behavior sequences.

Table III presents the detailed statistics of the three datasets. As we can see, the datasets are different from each other in many aspects including feature number, field number, and interaction sparsity.

³<http://jmcauley.ucsd.edu/data/amazon/>

⁴<https://tianchi.aliyun.com/dataset/dataDetail?dataId=53>

TABLE III
DATASET STATISTICS.

Dataset	#Users	#Items	#Instances	#Features	#Fields
Amazon-Cds	75,258	64,443	150,516	140,167	5
Amazon-Books	158,650	128,939	317,300	288,577	5
Alipay	326,577	451,631	653,154	788,166	7

2) *Data Processing*: To ensure the data quality, we filter out infrequent users and items with fewer than 5 interactions in the Amazon-Cds dataset. While the threshold value is 10 in the Amazon-Books and Alipay datasets. For all datasets, we aggregate each user’s interaction records and sort them by the action timestamps in chronological order. For evaluation purpose, we adopt the data split strategy in [33], [53]. Specifically, suppose a user has L historical behaviors sorted by time, the behavior sequence $[1, L - 3]$ is used for training and predicts whether she/he will interact with the $(L - 2)$ -th item. Similarly, behavior sequence $[1, L - 2]$ is used to predict the $(L - 1)$ -th item in the validation set, while behavior sequence $[1, L - 1]$ is used to predict the L -th item in the testing set. Further, given a user, a non-interacted item is randomly selected as the negative sample.

3) *Baseline Models*: For a thorough verification of model effectiveness, the proposed MISS framework is compared with three groups of representative CTR prediction models: a) Feature interaction based models (LR [54], FM [13], DeepFM [2], IPNN [49], DCN [22], DCN-M [27], xDeepFM [55]); b) User interest modeling based models (DIN [3], DIEN [30]), SIM(soft) [32], DMR [34]; c) GNN and Transformer based models (AutoInt+ [56], FiGNN [57]).

4) *Evaluation Metrics*: To quantitatively evaluate the model performances, two widely-used metrics are adopted, i.e., *AUC* and *Logloss* [2]. *AUC* measures the goodness of ranking positive samples higher than randomly chosen negative samples. A higher AUC value means better performance. *Logloss* measures the distance between the predicted scores and the true labels for all samples. A lower Logloss value indicates better model performance.

5) *Parameter Settings*: For a fair comparison, we set the embedding dimension of all models as 10, and the batch size is fixed as 128. The learning rate is selected from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, the L_2 norm regularization weight is picked from $\{0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$, and the dropout ratio is tuned from 0 to 0.9 with step size 0.1. The deep layers for all models are set as $\{40, 40, 40, 1\}$. The Adam optimizer [58] is chosen for model optimization. In addition to the above hyper-parameters for all models, we set the layers for interest encoder and feature encoder as $\{20, 20\}$ and $\{10, 10\}$. For simplicity, we set $\alpha_1 = \alpha_2$, and search them and τ within the ranges of $\{0.05, 0.1, 0.5, 1, 5\}$. For the branches of horizontal and vertical convolution kernels, M is tuned from $\{1, 2, 3, 4\}$, and N is tuned from $\{1, 2\}$. The distance H is tuned from $\{1, 2, 3, 4\}$. We use the validation set for parameter tuning, while the final reported performances are obtained on the testing set. Each experiment is repeated for 5 times to remove random noises, and the averaged results are reported.

B. Performance Comparison

In this section, we compare the performances of MISS with the state-of-the-art CTR prediction models. Table IV shows

TABLE IV
THE OVERALL PERFORMANCES ON ALL THREE DATASETS. THE \star MARK INDICATES THE STATISTICAL SIGNIFICANCE (P-VALUE<0.05) OF THE COMPARISON BETWEEN MISS AND THE STRONGEST BASELINE (UNDERLINED VALUES).

Dataset	Amazon-Cds		Amazon-Books		Alipay	
Model	AUC	Logloss	AUC	Logloss	AUC	Logloss
LR	0.6918	0.6308	0.7350	0.5968	0.7848	0.5578
FM	0.7585	0.5851	0.7653	0.5745	0.8470	0.4859
DeepFM	0.8039	0.5369	0.8056	0.5310	0.8718	0.4464
IPNN	0.8053	0.5364	0.8051	0.5308	0.8823	0.4299
DCN	0.7994	0.5412	0.7982	0.5390	0.8700	0.4494
DCN-M	0.8050	0.5363	0.8070	0.5293	0.8757	0.4403
xDeepFM	0.8034	0.5370	0.8028	0.5336	0.8777	0.4382
DIN	0.8055	0.5357	0.8074	0.5289	0.9098	0.3734
DIEN	0.7928	0.5479	0.8016	0.5352	0.9004	0.3950
SIM(soft)	0.7977	0.5437	0.7951	0.5430	0.9101	0.3729
DMR	0.8115	0.5289	0.8082	0.5282	0.9148	0.3642
AutoInt+	0.8008	0.5398	0.8045	0.5317	0.8705	0.4479
FiGNN	0.8012	0.5391	0.8006	0.5366	0.8716	0.4467
MISS	0.8867\star	0.4357\star	0.9180\star	0.3730\star	0.9327\star	0.3295\star

the experimental results of all compared models on all three datasets. From Table IV, we have the following observations:

- MISS consistently performs better than all baselines on all three datasets. More precisely, MISS significantly ($p - value < 0.05$) outperforms the strongest baselines by **9.27%**, **13.55%** and **1.96%** in terms of *AUC* (17.62%, 29.38% and 9.53% in terms of *Logloss*) on the Amazon-Cds, Amazon-Books, and Alipay datasets respectively. The great improvements over baseline models verify the effectiveness of MISS for CTR prediction. By supplementing the CTR prediction task with self-supervised learning, MISS is capable of exploiting the latent correlation information with more supervision signals, while baseline models only utilize the observed user-item interactions as supervision signals.
- The improvements in the Amazon-Cds and Amazon-Books datasets are much more significant than in the Alipay dataset. A possible reason is that the time span of user behaviors in these two datasets (over ten years) is much longer than that in the third dataset (six months). As more diverse interests take place in the relatively longer time span, our proposed MISS obtains more significant improvements by considering the multi-interest characteristic of user behaviors.
- LR and FM perform the worst among all baselines, which indicates that shallow models are insufficient for CTR prediction. By modeling high-order feature interactions with DNNs, DeepFM, IPNN, DCN, DCN-M and xDeepFM perform better than shallow models. DIN, DIEN, SIM(soft), and DMR achieve comparable performances with deep feature interaction models, which demonstrates the usefulness of user interest mining. Due to the weakness of RNN in modeling long sequences, DIEN performs a little worse than DIN. DMR achieves the best performances among all compared baselines. A possible reason is that it learns better representations by utilizing and integrating both user-item and item-item interactions in an attentive manner. AutoInt+ and FiGNN use self-attention or GNN for feature interaction modeling. Similar performances can be found compared with DeepFM and IPNN. It indicates that only using user-item interactions as supervision signals (as by existing deep

CTR models) cannot make a big difference on the model performances.

C. Ablation Study

To better understand the design rational of our proposed MISS, we conduct a series of ablation experiments and analysis in this section.

TABLE V
COMPATIBILITY ANALYSIS RESULTS.

Dataset	Amazon-Cds		Amazon-Books		Alipay	
Model	AUC	Logloss	AUC	Logloss	AUC	Logloss
DIN	0.8055	0.5357	0.8074	0.5289	0.9098	0.3734
DIN-MISS	0.8867	0.4357	0.9180	0.3730	0.9327	0.3295
IPNN	0.8053	0.5364	0.8051	0.5308	0.8823	0.4299
IPNN-MISS	0.8858	0.4368	0.9146	0.3778	0.9004	0.4006
FiGNN	0.8012	0.5391	0.8006	0.5366	0.8716	0.4467
FiGNN-MISS	0.8828	0.4410	0.9170	0.3746	0.8947	0.4160

1) *Compatibility Analysis*: Compatibility is among the key factors that restrict one model’s applications. To verify the compatibility of our proposed MISS framework, apart from the DIN backbone model described in the framework section, we also use it to improve the representation learning in another two representative CTR models, i.e., IPNN, and FiGNN. For a fair comparison, other parts of these models remain unchanged, and the enhanced models are named as DIN-MISS (the same model as MISS), IPNN-MISS, and FiGNN-MISS respectively. We compare the original and enhanced models on the three datasets, and the experimental results are presented in Table V. As can be easily observed, all three enhanced models (DIN-MISS, IPNN-MISS, and FiGNN-MISS) significantly outperform their original models on all three datasets. It validates the compatibility of our embedding enhancement approach by demonstrating its effectiveness when combined with various popular CTR models. The results show that MISS can be used as a general framework to improve the existing CTR models by supplementing self-supervised signals for embedding enhancement.

TABLE VI
SUPERIORITY ANALYSIS RESULTS.

Dataset	Amazon-Cds		Amazon-Books		Alipay	
Model	AUC	Logloss	AUC	Logloss	AUC	Logloss
IPNN	0.8053	0.5364	0.8051	0.5308	0.8823	0.4299
IPNN-Rule	0.8101	0.5303	0.8497	0.4788	0.8818	0.4321
IPNN-IRSSL	0.8050	0.5371	0.8065	0.5295	0.8821	0.4314
IPNN-S3Rec	0.8065	0.5343	0.8073	0.5286	0.8826	0.4299
IPNN-CL4SRec	0.8372	0.5057	0.8759	0.4553	0.8865	0.4250
IPNN-MISS	0.8858	0.4368	0.9146	0.3778	0.9004	0.4006
DIN	0.8055	0.5357	0.8074	0.5289	0.9098	0.3734
DIN-Rule	0.8068	0.5349	0.8397	0.4925	0.9113	0.3719
DIN-IRSSL	0.8058	0.5352	0.8064	0.5295	0.9098	0.3744
DIN-S3Rec	0.8073	0.5348	0.8076	0.5286	0.9100	0.3728
DIN-CL4SRec	0.8364	0.5082	0.8756	0.4563	0.9141	0.3686
DIN-MISS	0.8867	0.4357	0.9180	0.3730	0.9327	0.3295

2) *Superiorty Analysis*: To demonstrate the superiority of our proposed MISS framework, we compare it with state-of-the-art self-supervised learning models. Specifically, we apply MISS, IRSSL [45], S3Rec [44], and CL4SRec [46] to the IPNN, DIN, and FiGNN models for embedding enhancement purpose. Besides above SSL models, we also equip these base CTR models with a rule based model that segments the behavior sequence into several sub-sequences based on item categories and then conduct dropout on each sequence for SSL. The resulting models are named in an “A”-“B” manner

where “A” and “B” represent the base model and the SSL method respectively. Notice that, we adopt the item feature mask strategy in IRSSL as it achieves better performances than feature dropout, and the sequence-segment correlation is adopted in S3Rec thanks to its best performances within the four data augmentation techniques. Comparative experimental results of the original and enhanced models are shown in Table VI. Due to the space limitation and similar trend of evaluation metrics, results of the FiGNN model are not presented. From Table VI, we have the following findings:

- Our MISS model consistently performs the best regardless of the base models or datasets, which further verifies the superiority of our comparative learning strategies.
- Rule based SSL model achieves much better performances than IRSSL on the Amazon-Books dataset, which verifies the effectiveness of interest-level contrastive learning for recommendation tasks. However, comparable performances are achieved on the other two datasets. The reason is that the item categories in different datasets are differently defined. In some cases, item categories indicate user interests well, but in other cases they do not.
- Generally speaking, IPNN-IRSSL performs no better than IPNN and it is the same for DIN-IRSSL and DIN. The reason is that IRSSL only focuses on item features, thus loses efficacy when few item features are available.
- IPNN-S3Rec and DIN-S3Rec perform slightly better than the original models, which supports the effectiveness of SSL at the behavior level. However, there is an obvious semantic difference between a random segment and the whole behavior sequence, hence the correlation learning is biased and limits its performances.
- Models enhanced by the CL4SRec method achieve the second best performances. In CL4SRec, the majority of the behavior sequences remain unchanged after the item crop, mask, and reorder operations, which makes it more robust to random noises. In our MISS, however, a more flexible data augmentation method is put forward to make better use of user interests.

TABLE VII
PERFORMANCES OF DIFFERENT MISS VARIANTS.

Dataset	Amazon-Cds		Amazon-Books		Alipay	
Model	AUC	Logloss	AUC	Logloss	AUC	Logloss
IPNN-MISS	0.8858	0.4368	0.9146	0.3778	0.9004	0.4006
IPNN-MISS/F	0.8741	0.4506	0.9122	0.3880	0.8973	0.4044
IPNN-MISS/F/U	0.8620	0.4649	0.8946	0.4116	0.8937	0.4120
IPNN-MISS/F/L	0.8549	0.4769	0.8931	0.4209	0.8885	0.4214
IPNN-MISS/F/U/L	0.8480	0.4829	0.8739	0.4625	0.8871	0.4220
IPNN-MISS/M/F/U/L	0.8393	0.4953	0.8449	0.4933	0.8838	0.4265
IPNN	0.8053	0.5364	0.8051	0.5308	0.8823	0.4299
DIN-MISS	0.8867	0.4357	0.9180	0.3730	0.9327	0.3295
DIN-MISS/F	0.8813	0.4419	0.9136	0.3823	0.9300	0.3357
DIN-MISS/F/U	0.8636	0.4642	0.8978	0.4100	0.9260	0.3446
DIN-MISS/F/L	0.8568	0.4807	0.8937	0.4188	0.9236	0.3499
DIN-MISS/F/U/L	0.8514	0.4869	0.8717	0.4642	0.9222	0.3509
DIN-MISS/M/F/U/L	0.8429	0.4978	0.8425	0.4948	0.9188	0.3579
DIN	0.8055	0.5357	0.8074	0.5289	0.9098	0.3734

3) *Effectiveness Analysis*: As firstly addressed in the Introduction and also reflected in the model structure, our MISS framework is built upon some important practices including the multi-interest consideration (M), the union-wise interest representation (U), the long-range interest dependencies (L),

and the intra-item feature correlation (F). To evaluate the effectiveness of these different practices, we explore MISS with different settings. By removing some of the practices, five more MISS variants are obtained and named as MISS/F, MISS/F/U, MISS/F/L, MISS/F/U/L, and MISS/M/F/U/L respectively. All MISS variants are applied to the IPNN, DIN, and FiGNN models to verify their performances, where the resulting models are also named in an “A”-“B” manner. The comparison results are presented in Table VII, where the results of the FiGNN model are also omitted to save space. As can be observed, all MISS variants bring about performance boosts to the original IPNN and DIN models, and the complete MISS framework achieves the best results. Therefore, we claim that all four practices (M, U, L, and F) are effective and complementary to each other, and it is necessary to adopt all of them for better performances. What is more, the removal of M results into the worst performance decay, revealing the importance of multi-interest modeling.

TABLE VIII

PERFORMANCES OF DIFFERENT MULTI-INTEREST EXTRACTOR.

Dataset	Amazon-Cds		Amazon-Books		Alipay	
Extractor	AUC	Logloss	AUC	Logloss	AUC	Logloss
DIN	0.8055	0.5357	0.8074	0.5289	0.9098	0.3734
MISS-SA	0.8042	0.5385	0.8128	0.5225	0.9092	0.3758
MISS-LSTM	0.8106	0.5299	0.8172	0.5178	0.9096	0.3753
MISS-CNN	0.8867	0.4357	0.9180	0.3730	0.9327	0.3295

4) *Multi-Interest Extractor Analysis*: To verify the rational of our MIE(\cdot) design formulated in Equation (18-20), we compare the performances of our proposed CNN module with self-attention [59] and LSTM [60] for multi-interest extraction, and the resulting models are named as MISS-CNN (the same model as MISS), MISS-SA, MISS-LSTM respectively. Table VIII summarizes the experimental results. We can see that the CNN extractor achieves the best performances on all datasets. For an in-depth analysis of these extractors, we further visualize the cosine similarity scores between the generated pairs of views from these interest representations on Figure 5. Each training step on the x-axis in Figure 5 corresponds to a batch of training samples fed at that step, and the average similarity score among training batches are reported. As we can see, the similarity scores of MISS-SA and MISS-LSTM are close to 1, thus the generated pairs hardly provide any useful information for contrastive learning. The reason may be as follows. LSTM learns the characteristics of the whole historical behavior sequence, and the histories of two adjacent items of the sequence only differ by one item, so the representations learned for the two adjacent items via LSTM are highly similar. Self-attention based method aggregates all behaviors to generate interest representations, and hence learns similar representations for adjacent items. In comparison, CNN based model considers a sliding window of the past history, and the size of the sliding window is small (at most 3 or 4 in our experiments), so differing by one most recent item will make a notable difference in the representation. This is evidenced by the similarity scores of our proposed CNN model, which are in the range of 0.7 and 0.8. The representations of interest at adjacent timestamps are similar but also distinguishable for contrastive learning. This validates the superiority of using CNN compared to LSTM or self-attention in our problem.

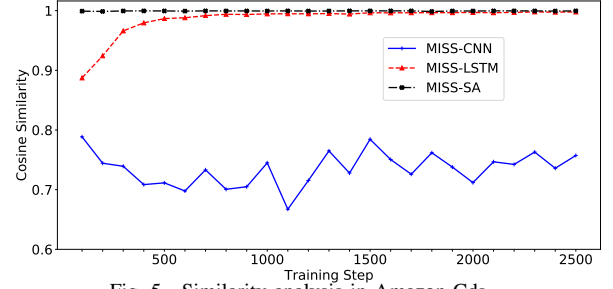


Fig. 5. Similarity analysis in Amazon-Cds.

D. Model Training Analysis

During training, our proposed MISS framework has several key hyper-parameters that may affect the performances, and so do the multi-task training strategies. In this section, we first investigate the importance and sensitivity of these hyper-parameters by changing one hyper-parameter while fixing the others. After that, different training strategies of the two losses are also compared.

1) *Impact of the loss weight*: The final loss function of MISS in Equation (17) is a combination of the CTR prediction loss and the SSL losses. Figure 6 shows the CTR prediction performances under different loss weights where larger weights indicate stronger contributions of the SSL losses. We can observe that the performances grow stably with the increase of the loss weight at the beginning. However, when the weight grows bigger than 1, performance degradation happens. Thus the SSL losses should not dominate the training process. In other words, the SSL part takes the auxiliary role for CTR prediction, and the model can be biased when it is overemphasized.

2) *Impact of the softmax temperature*: The softmax temperature parameters in Equation (15) and Equation (16) tune the distribution of the SSL losses. A large temperature value will draw close the predictions of positive and negative samples in SSL losses, thus weakens the supervision signals in training. We analyze how different temperature parameter values affect the model performances, and the results are illustrated in Figure 7. With the growth of the temperature value, performances on all three datasets increase first and then decrease. The turning point is 0.1 for both metrics on all datasets. With such a small temperature value (significantly less than 1), the supervision signals get strengthened during training as the positive and negative SSL samples are better discriminated. In other words, discriminating positive and negative samples benefits the model performances, which accords with our motivations.

TABLE IX

PERFORMANCES OF DIFFERENT MISS TRAINING STRATEGIES.

Dataset	Amazon-Cds		Amazon-Books		Alipay	
Model	AUC	Logloss	AUC	Logloss	AUC	Logloss
DIN	0.8055	0.5357	0.8074	0.5289	0.9098	0.3734
MISS-Joint	0.8867	0.4357	0.9180	0.3730	0.9327	0.3295
MISS-Pre	0.8848	0.4381	0.9170	0.3746	0.9313	0.3328

3) *Training strategies*: There are two learning targets in our MISS framework, i.e., CTR prediction and self-supervised learning. During training, different multi-task learning strategies can be adopted to optimize the two targets. Right here, we compare and analyze the most widely used joint learning and pre-training strategies. Table IX gives the analysis results, where DIN is used as the backbone model. The MISS model

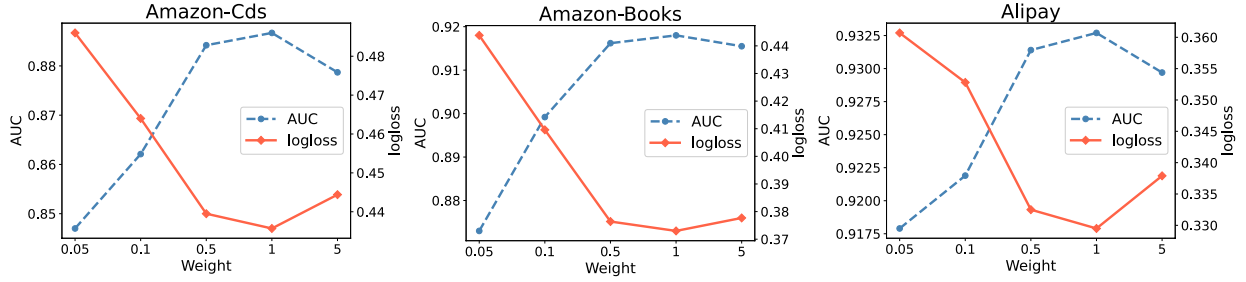


Fig. 6. Performances of MISS w.r.t. different weights assigned to the SSL losses.

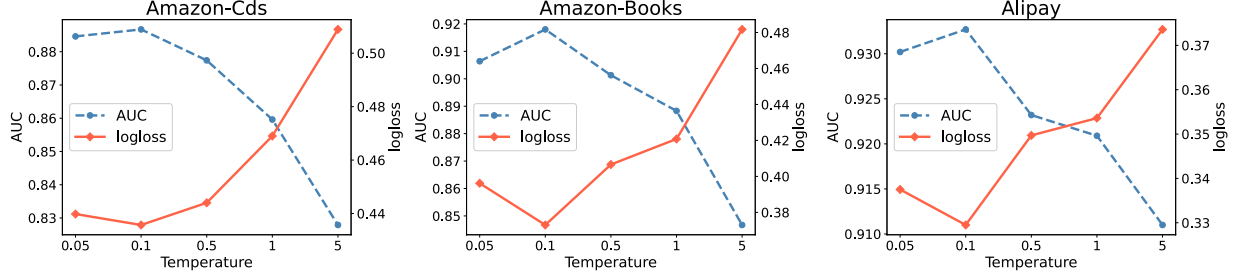


Fig. 7. Performances of MISS w.r.t. different temperature parameter values.

TABLE X

AUC SCORES OF CTR PREDICTION WITH DIFFERENT SAMPLING RATE.

Dataset	Amazon-Cds			Amazon-Books		
SR	DIN	DIN-MISS	RI	Din	DIN-MISS	RI
80%	0.7913	0.8779	10.94%	0.7932	0.9107	14.81%
90%	0.7988	0.8814	10.34%	0.7998	0.9152	14.43%
100%	0.8055	0.8867	10.08%	0.8074	0.9180	13.70%

TABLE XI

AUC SCORES OF CTR PREDICTION WITH DIFFERENT LABEL NOISE RATE.

Dataset	Amazon-Cds			Amazon-Books		
	NR	DIN	DIN-MISS	RI	DIN	DIN-MISS
0%	0.8055	0.8867	10.08%	0.8074	0.9180	13.70%
10%	0.7768	0.8652	11.38%	0.7775	0.8877	14.16%
20%	0.7413	0.8331	12.38%	0.7384	0.8678	17.52%

trained with joint learning is denoted as MISS-Join, while MISS-Pre learns the CTR prediction target based on the pre-trained embeddings by MISS. Both MISS-Join and MISS-Pre achieve better performances than DIN, and MISS-Join perform even better than MISS-Pre. In the joint end-to-end training, complementary supervision signals are shared across the two targets, resulting into mutual enhancements that are beyond the reach of pre-training.

E. Case Study

In this part, we verify that our model can effectively alleviate the label sparsity and label noise problems.

1) *Label Sparsity Analysis*: As explained in the Introduction, CTR models easily suffer from the label sparsity issue. To verify our model's effectiveness in alleviating label sparsity, we down-sample the original training set with sampling rate (SR) 90% and 80%, while the validation and testing sets stay unchanged. Notice that the 100% sampling rate means using the original training set. Table X shows the performances with different SR, where the results on the Alipay dataset are omitted for space limitation. We omit the results on the Alipay dataset for space limitation, which have similar trends. It can be found that the performance drops when the labels become sparse (SR decreases), while the relative improvement (RI) gets larger. Thus our MISS model can effectively alleviate the label sparsity problem.

2) *Label Noise Analysis*: Besides label sparsity, the label noise problem can also be well solved by our proposed MISS model. To check the robustness of MISS to label noise, noises are imposed on the training set by randomly swapping the

labels at an indicated proportion (10% and 20%) of samples, while the validation and testing sets stay unchanged. Notice that 0% noise rate (NR) means using the original training set. Due to space limitation, only the results on Amazon-Cds and Amazon-Books datasets are demonstrated in Table XI. It is obvious that the relative improvement of DIN-MISS over DIN grows more significant when NR increases. In other words, MISS shows good robustness to label noise.

VII. CONCLUSION

In this paper, we proposed a Multi-Interest Self-Supervised learning (MISS) framework for the CTR prediction task to deal with the label sparsity and label noise issues. In view of the multi-interest characteristics of user behaviors, a CNN-based multi-interest extractor component was proposed to learn the hidden interests while considering both point-wise and union-wise interest representations. Further, another CNN-based multi-interest and multi-feature extractor was also proposed to utilize both inter-item and intra-item interest correlations at the fine-grained feature level. With the help of two random selection functions, augmented views of interest representations can be extracted in consideration of both short-range and long-range interest dependencies. Based on the augmented views of interest representations, two contrastive learning losses effectively transforms interest correlation knowledge into self-supervision signals. In this way, not only the label sparsity issue gets alleviated by the self-supervision signals, but also the model robustness gets enhanced to shield label noise. Extensive experimental results on three large-scale datasets verify the effectiveness of the proposed MISS framework.

REFERENCES

- [1] H. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proceedings of DLR@RecSys*, A. Karatzoglou, B. Hidasi, D. Tikk, O. S. Shalom, H. Roitman, B. Shapira, and L. Rokach, Eds. Boston, MA, USA: ACM, 2016, pp. 7–10.
- [2] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for CTR prediction," in *Proceedings of IJCAI 2017*, C. Sierra, Ed. Melbourne, Australia: ijcai.org, 2017, pp. 1725–1731.
- [3] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of SIGKDD*, Y. Guo and F. Farooq, Eds. London, UK: ACM, 2018, pp. 1059–1068.
- [4] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of ICML*, ser. Proceedings of Machine Learning Research, vol. 119. Virtual Event: PMLR, 2020, pp. 1597–1607.
- [5] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proceedings of ICLR*. Vancouver, BC, Canada: OpenReview.net, 2018.
- [6] O. J. Hénaff, S. Koppula, J. Alayrac, A. van den Oord, O. Vinyals, and J. Carreira, "Efficient visual pretraining with contrastive detection," *CoRR*, vol. abs/2103.10957, 2021.
- [7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, MN, USA: Association for Computational Linguistics, 2019, pp. 4171–4186.
- [8] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proceedings of ICLR*. Addis Ababa, Ethiopia: OpenReview.net, 2020.
- [9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.
- [10] C. Li, Z. Liu, M. Wu, Y. Xu, H. Zhao, P. Huang, G. Kang, Q. Chen, W. Li, and D. L. Lee, "Multi-interest network with dynamic routing for recommendation at tmall," in *Proceedings of CIKM*, W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds. Beijing, China: ACM, 2019, pp. 2615–2623.
- [11] Y. Cen, J. Zhang, X. Zou, C. Zhou, H. Yang, and J. Tang, "Controllable multi-interest framework for recommendation," in *Proceedings of SIGKDD*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. Virtual Event, CA, USA: ACM, 2020, pp. 2942–2951.
- [12] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikrur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, and J. Kubica, "Ad click prediction: a view from the trenches," in *Proceedings of SIGKDD*, I. S. Dhillon, Y. Koren, R. Ghani, T. E. Senator, P. Bradley, R. Parekh, J. He, R. L. Grossman, and R. Uthrusamy, Eds. Chicago, IL, USA: ACM, 2013, pp. 1222–1230.
- [13] S. Rendle, "Factorization machines," in *Proceedings of ICDM*, G. I. Webb, B. Liu, C. Zhang, D. Gunopulos, and X. Wu, Eds. Sydney, Australia: IEEE Computer Society, 2010, pp. 995–1000.
- [14] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich, "Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine," in *Proceedings of ICML*, J. Fürnkranz and T. Joachims, Eds. Haifa, Israel: Omnipress, 2010, pp. 13–20.
- [15] C. Cheng, F. Xia, T. Zhang, I. King, and M. R. Lyu, "Gradient boosting factorization machines," in *Proceedings of RecSys*, A. Kobsa, M. X. Zhou, M. Ester, and Y. Koren, Eds. Foster City, Silicon Valley, CA, USA: ACM, 2014, pp. 265–272.
- [16] Y. Chang, C. Hsieh, K. Chang, M. Ringgaard, and C. Lin, "Training and testing low-degree polynomial data mappings via linear SVM," *J. Mach. Learn. Res.*, vol. 11, pp. 1471–1490, 2010.
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in NeurIPS*, P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Lake Tahoe, Nevada, United States: nips.cc, 2012, pp. 1106–1114.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR*. Las Vegas, NV, USA: IEEE Computer Society, 2016, pp. 770–778.
- [19] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of EMNLP*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: ACL, 2014, pp. 1724–1734.
- [20] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of INTERSPEECH*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds. Makuhari, Chiba, Japan: ISCA, 2010, pp. 1045–1048.
- [21] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data - A case study on user response prediction," in *Advances in ECIR*, ser. Lecture Notes in Computer Science, N. Ferro, F. Crestani, M. Moens, J. Mothe, F. Silvestri, G. M. D. Nunzio, C. Hauff, and G. Silvello, Eds., vol. 9626. Padua, Italy: Springer, 2016, pp. 45–57.
- [22] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD*. Halifax, NS, Canada: ACM, 2017, pp. 12:1–12:7.
- [23] W. Guo, R. Tang, H. Guo, J. Han, W. Yang, and Y. Zhang, "Order-aware embedding neural network for ctr prediction," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 1121–1124.
- [24] B. Chen, Y. Wang, Z. Liu, R. Tang, W. Guo, H. Zheng, W. Yao, M. Zhang, and X. He, "Enhancing explicit and implicit feature interactions via information sharing for parallel deep ctr models," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3757–3766.
- [25] W. Guo, R. Su, R. Tan, H. Guo, Y. Zhang, Z. Liu, R. Tang, and X. He, "Dual graph enhanced embedding neural network for ctr prediction," in *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2021, p. 496–504.
- [26] W. Zhang, J. Qin, W. Guo, R. Tang, and X. He, "Deep learning for click-through rate estimation," *CoRR*, vol. abs/2104.10584, 2021.
- [27] R. Wang, R. Shivanna, D. Z. Cheng, S. Jain, D. Lin, L. Hong, and E. Chi, "DCN V2: improved deep & cross network and practical lessons for web-scale learning to rank systems," in *Proceedings of WWW*, J. Leskovec, M. Grobelle, M. Najork, J. Tang, and L. Zia, Eds. ACM / IW3C2, 2021, pp. 1785–1797.
- [28] Q. Song, D. Cheng, H. Zhou, J. Yang, Y. Tian, and X. Hu, "Towards automated neural interaction discovery for click-through rate prediction," in *Proceedings of KDD*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. ACM, 2020, pp. 945–955.
- [29] B. Liu, C. Zhu, G. Li, W. Zhang, J. Lai, R. Tang, X. He, Z. Li, and Y. Yu, "Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2636–2645.
- [30] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, "Deep interest evolution network for click-through rate prediction," in *Proceedings of AAAI*. Honolulu, Hawaii, USA: AAAI Press, 2019, pp. 5941–5948.
- [31] Y. Feng, F. Lv, W. Shen, M. Wang, F. Sun, Y. Zhu, and K. Yang, "Deep session interest network for click-through rate prediction," in *Proceedings of IJCAI*, S. Kraus, Ed. Macao, China: ijcai.org, 2019, pp. 2301–2307.
- [32] Q. Pi, G. Zhou, Y. Zhang, Z. Wang, L. Ren, Y. Fan, X. Zhu, and K. Gai, "Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction," in *Proceedings of CIKM*, M. d'Aquin, S. Dietze, C. Hauff, E. Curry, and P. Cudré-Mauroux, Eds. Virtual Event, Ireland: ACM, 2020, pp. 2685–2692.
- [33] J. Qin, W. Zhang, X. Wu, J. Jin, Y. Fang, and Y. Yu, "User behavior retrieval for click-through rate prediction," in *Proceedings of SIGIR*, J. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen, and Y. Liu, Eds. Virtual Event, China: ACM, 2020, pp. 2347–2356.
- [34] Z. Lyu, Y. Dong, C. Huo, and W. Ren, "Deep match to rank model for personalized click-through rate prediction," in *Proceedings of AAAI*. AAAI Press, 2020, pp. 156–163.
- [35] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of CVPR*. Seattle, WA, USA: IEEE, 2020, pp. 9726–9735.
- [36] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of ECCV*, ser. Lecture Notes in Computer Science, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11218. Munich, Germany: Springer, 2018, pp. 139–156.
- [37] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Advances in NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. virtual: nips.cc, 2020.

- [38] X. Liu, F. Zhang, Z. Hou, Z. Wang, L. Mian, J. Zhang, and J. Tang, “Self-supervised learning: Generative or contrastive,” *CoRR*, vol. abs/2006.08218, 2020.
- [39] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, “Spanbert: Improving pre-training by representing and predicting spans,” *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, 2020.
- [40] M. Gutmann and A. Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in *Proceedings of AISTATS*, ser. JMLR Proceedings, Y. W. Teh and D. M. Titterton, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: JMLR.org, 2010, pp. 297–304.
- [41] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *CoRR*, vol. abs/1807.03748, 2018.
- [42] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, “Learning deep representations by mutual information estimation and maximization,” in *Proceedings of ICLR*. New Orleans, LA, USA: OpenReview.net, 2019.
- [43] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, “Deep graph infomax,” in *Proceedings of ICLR*. New Orleans, LA, USA: OpenReview.net, 2019.
- [44] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J. Wen, “S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization,” in *Proceedings of CIKM*, M. d’Aquino, S. Dietze, C. Hauff, E. Curry, and P. Cudré-Mauroux, Eds. Virtual Event, Ireland: ACM, 2020, pp. 1893–1902.
- [45] T. Yao, X. Yi, D. Z. Cheng, F. Yu, T. Chen, A. Menon, L. Hong, E. H. Chi, S. Tjoa, J. Kang, and E. Ettinger, “Self-supervised learning for large-scale item recommendations,” *CoRR*, vol. abs/2007.12865, 2021.
- [46] X. Xie, F. Sun, Z. Liu, J. Gao, B. Ding, and B. Cui, “Contrastive pre-training for sequential recommendation,” *CoRR*, vol. abs/2010.14395, 2020.
- [47] J. Ma, C. Zhou, H. Yang, P. Cui, X. Wang, and W. Zhu, “Disentangled self-supervision in sequential recommenders,” in *Proceedings of SIGKDD*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. Virtual Event, CA, USA: ACM, 2020, pp. 483–491.
- [48] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, “Self-supervised graph learning for recommendation,” *CoRR*, vol. abs/2010.10783, 2020.
- [49] Y. Qu, B. Fang, W. Zhang, R. Tang, M. Niu, H. Guo, Y. Yu, and X. He, “Product-based neural networks for user response prediction over multi-field categorical data,” *ACM Trans. Inf. Syst.*, vol. 37, no. 1, pp. 5:1–5:35, 2019.
- [50] S. Shalev-Shwartz, O. Shamir, and S. Shammah, “Failures of gradient-based deep learning,” in *Proceedings of ICML*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. Sydney, NSW, Australia: PMLR, 2017, pp. 3067–3075.
- [51] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, “Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer,” in *Proceedings of CIKM*, W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds. Beijing, China: ACM, 2019, pp. 1441–1450.
- [52] T. Wang and P. Isola, “Understanding contrastive representation learning through alignment and uniformity on the hypersphere,” in *Proceedings of ICML*, ser. Proceedings of Machine Learning Research, vol. 119. Virtual Event: PMLR, 2020, pp. 9929–9939.
- [53] K. Ren, J. Qin, Y. Fang, W. Zhang, L. Zheng, W. Bian, G. Zhou, J. Xu, Y. Yu, X. Zhu, and K. Gai, “Lifelong sequential modeling with personalized memorization for user response prediction,” in *Proceedings of SIGIR*, B. Piwowarski, M. Chevalier, É. Gaussier, Y. Maarek, J. Nie, and F. Scholer, Eds. Paris, France: ACM, 2019, pp. 565–574.
- [54] K. Lee, B. Orten, A. Dasdan, and W. Li, “Estimating conversion rate in display advertising from past performance data,” in *Proceedings of SIGKDD*, Q. Yang, D. Agarwal, and J. Pei, Eds. Beijing, China: ACM, 2012, pp. 768–776.
- [55] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, “xdeepfm: Combining explicit and implicit feature interactions for recommender systems,” in *Proceedings of SIGKDD*, Y. Guo and F. Farooq, Eds. London, UK: ACM, 2018, pp. 1754–1763.
- [56] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang, “Autoint: Automatic feature interaction learning via self-attentive neural networks,” in *Proceedings of CIKM*, W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds. Beijing, China: ACM, 2019, pp. 1161–1170.
- [57] Z. Li, Z. Cui, S. Wu, X. Zhang, and L. Wang, “Fi-gnn: Modeling feature interactions via graph neural networks for CTR prediction,” in *Proceedings of CIKM*, W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds. Beijing, China: ACM, 2019, pp. 539–548.
- [58] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of ICLR*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA: OpenReview.net, 2015.
- [59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in NeurIPS*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds. Long Beach, CA, USA: nips.cc, 2017, pp. 5998–6008.
- [60] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.