Disentangled Interest Network for Out-of-Distribution CTR Prediction

YU ZHENG, Tsinghua University, China
CHEN GAO, Tsinghua University, China
JIANXIN CHANG, Kuaishou, China
YANAN NIU, Kuaishou, China
YANG SONG, Kuaishou, China
DEPENG JIN, Tsinghua University, China
MENG WANG, Hefei University of Technology, China
YONG LI, Tsinghua University, China

Click-through rate (CTR) prediction, which estimates the probability of a user clicking on a given item, is a critical task for online information services. Existing approaches often make strong assumptions that training and test data come from the same distribution. However, the data distribution varies since user interests are constantly evolving, resulting in the out-of-distribution (OOD) issue. In addition, users tend to have multiple interests, some of which evolve faster than others. Towards this end, we propose Disentangled Click-Through Rate prediction (DiseCTR), which introduces a causal perspective of recommendation and disentangles multiple aspects of user interests to alleviate the OOD issue in recommendation. We conduct a causal factorization of CTR prediction involving user interest, exposure model, and click model, based on which we develop a deep learning implementation for these three causal mechanisms. Specifically, we first design an interest encoder with sparse attention which maps raw features to user interests, and then introduce a weakly supervised interest disentangler to learn independent interest embeddings, which are further integrated by an attentive interest aggregator for prediction. Experimental results on three real-world datasets show that DiseCTR achieves the best accuracy and robustness in OOD recommendation against state-of-the-art approaches, significantly improving AUC and GAUC by over 0.02 and reducing logloss by over 13.7%. Further analyses demonstrate that DiseCTR successfully disentangles user interests, which is the key to OOD generalization for CTR prediction. We have released the code and data at https://github.com/DavyMorgan/DiseCTR/.

CCS Concepts: • **Information systems** → **Personalization**.

Additional Key Words and Phrases: Recommendation, Out-of-Distribution, Disentanglement

1 INTRODUCTION

Click-through rate (CTR) prediction plays a crucial role in today's online information services, including recommender systems [5, 7, 10, 34, 41, 43, 54, 65], information retrieval [21, 25, 71] and computational advertising [52], which takes attributes of users and items as input, and predicts the probability of user interacting with a given item. In general, CTR prediction can be regarded as a classification task on a binary target variable Y

Authors' Contact Information: Yu Zheng, Tsinghua University, Beijing, China, zhengyu.davy@foxmail.com; Chen Gao, Tsinghua University, Beijing, China, chgao96@tsinghua.edu.cn; Jianxin Chang, Kuaishou, Beijing, China, jianxin.chang@outlook.com; Yanan Niu, Kuaishou, Beijing, China, alanniu@outlook.com; Yang Song, Kuaishou, Beijing, China, yangsong@kuaishou.com; Depeng Jin, Tsinghua University, Beijing, China, jindp@tsinghua.edu.cn; Meng Wang, Hefei University of Technology, Hefei, China, eric.mengwang@gmail.com; Yong Li, Tsinghua University, Beijing, China, liyong07@tsinghua.edu.cn.

ACM 1558-2868/2025/11-ART
https://doi.org/10.1145/3777368

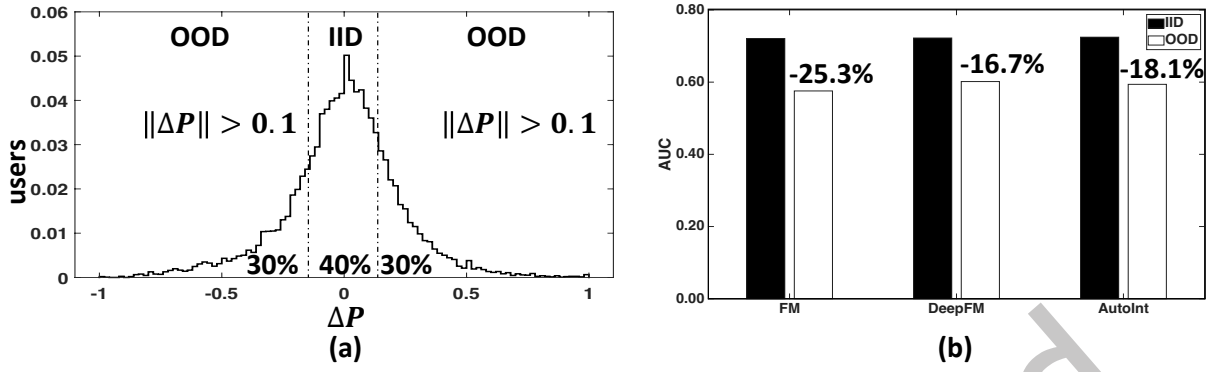


Fig. 1. (a) OOD issue on a video recommendation dataset shown with histogram of ΔP, where $\Delta P = P_{eval}(Y = 1|X) - P_{train}(Y = 1|X)$ is the difference of CTR on popular videos between training and evaluation data. About 60% of users have a distribution variation larger than 0.1, and only 40% of the data can be considered as IID. (b) Performance of three popular CTR prediction models on both IID and OOD data, showing their degraded performance when OOD issue occurs.

(interaction) from a high-dimensional variable X (user ID, item ID, and features), which has been widely studied in both academia and industry [7, 8, 14, 39, 42, 45, 52, 81]. Following classical machine learning theories [35], existing CTR models rely on the assumption that training and evaluation data are *independent and identically distributed (IID)*. Particularly, they assume that the conditional distribution $P(Y|X)$ remains stable. In practice, however, the data distribution tends to drift over time, causing distribution discrepancies between training and test data, which is known as the *out-of-distribution (OOD)* issue [2, 16, 29, 30, 51, 66]. For example, the CTR of a football-related video is often high within the first few days/hours when the match concludes, and then declines sharply over time. To illustrate the distribution variation, we conduct an analysis on a real-world video recommendation dataset. (details of the dataset will be introduced in Section 4). Specifically, for each user in the dataset, we calculate the differences of CTRs $P(Y|X)$ on *popular* videos between training and test data, denoted as ΔP for short. The histogram of ΔP for all users is shown in Figure 1(a). We can observe that the IID assumption only holds for about 40% of the users, while the remaining 60% of the users have large distribution variation with $\|\Delta P\|$ greater than 0.1, which indicates the severe OOD issue. In other words, in real-world scenarios, the OOD issue is quite common and the IID assumption no longer holds. As a result, the performance of existing CTR models trained under the IID assumption can no-longer be guaranteed. To illustrate that, we trained three popular CTR prediction models (FM [45], DeepFM [12] and AutoInt [52]), and evaluated their performance with both IID and OOD data. Figure 1(b) shows that the AUC of all three models drops significantly in OOD scenarios. Therefore, it is critical to develop a robust recommender system that can generalize well under OOD circumstances.

To approach the problem, we first investigate the sources of distribution variation and why existing CTR models fail. From the view of users, they are not interacting with items based on *low-level features*. In real-world recommendation, users tend to have multiple unobserved *interests* ($Z = \{Z_1, \cdots, Z_M\}$), such as aesthetic-aspect interest, social-aspect interest, and economic-aspect interest, which determines whether a user will interact with an item (Y). Each interest can be related to a few raw features (X), *e.g.*, the economic-aspect interest is largely related to the price feature and brand feature. When the users browse the items, the interests rather than features, are the intrinsic drives of user-item interactions. Formally, there exists a set of unobserved interest variables $Z = \{Z_1, \cdots, Z_M\}$, each of which is related to a certain part of features, determining whether a user will interact with an item. Meanwhile, interests usually change in a *partial* manner, *i.e.*, only a few vary and the majority

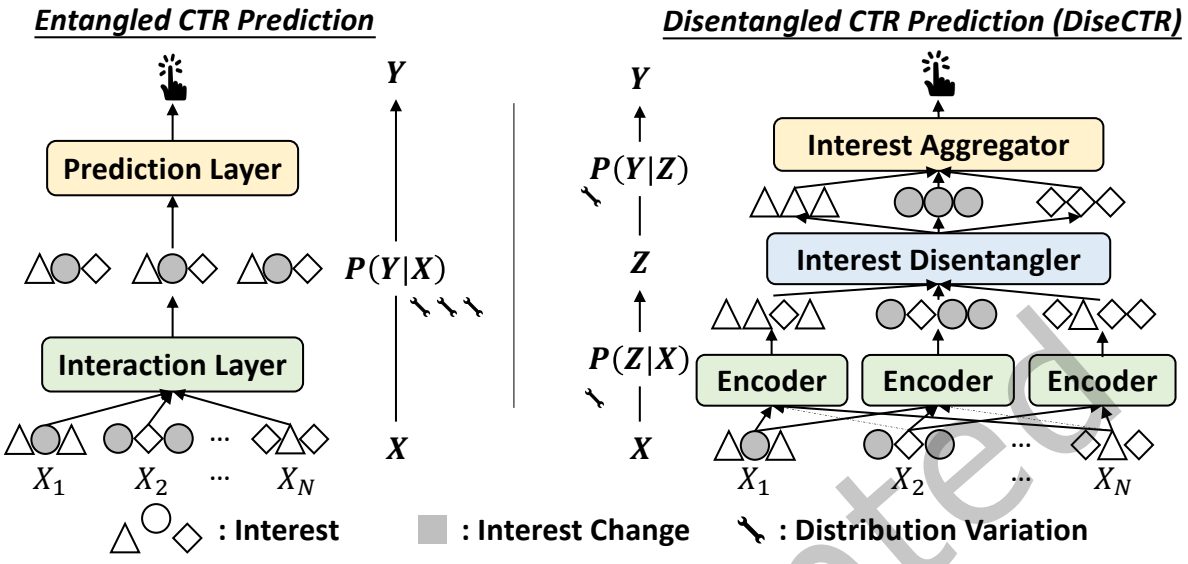


Fig. 2. Existing entangled methods directly learn $P(Y|X)$, while DiseCTR learns $P(Z|X)$ and $P(Y|Z)$ separately with disentangled interests. Under partial changes of interests (gray in the figure), existing methods tend to fail due to the large variation of $P(Y|X)$, while DiseCTR is more robust since variation of $P(Z|X)$ and $P(Y|Z)$ is relatively small.

remains stable [47, 48]. There is a commonly existing example: *giving coupons to e-commerce users tends to change their economic interest, while other interests are largely stable* [13, 20]. In other words, the partial changes on Z indicate small variation of $P(Z|X)$ and $P(Y|Z)$, while the resulted shift of $P(Y|X)$ can be large, since one affected Z_i can be related to many features. Here we follow the above example, and there will be: *change of economic interest can influence features like price, brand, and category.* As a result, existing CTR models fail to generalize in OOD scenarios since they ignore Z and directly capture $P(Y|X)$ which is vulnerable to the partial distribution variation on Z.

To address the OOD issue, a CTR model can take advantage of the partial property by learning disentangled interests. Specifically, based on a causal factorization of recommendation decomposing the joint distribution of features, interests, and interactions into an interest model $P(Z)$, an exposure model $p(X|Z)$, and a click model $P(Y|X,Z)$, we can capture $P(Z|X)$ and $P(Y|Z)$ separately rather than directly learning $P(Y|X)$. In this way, the impact of distribution variation is reduced to a few affected interests, and most parts of the model will not be influenced. Thus it can generalize to OOD scenarios efficiently with only the affected part being updated. However, learning disentangled interests is non-trivial with the following challenges. First, interests are unobserved causal variables that implicitly correspond with input features. There is no label for each Z_i as well as which features are related to Z_i. Second, distribution variation can independently occur on any Z_i, thus it is necessary to decouple different interests to ensure that the model can handle all possible changes.

It is worth noting that two recent works [15, 57] investigated the OOD issue in recommendation, however, they focused on simplified scenarios (collaborative filtering [15] and user income increase [57]), and they can not handle general OOD issues in CTR prediction. In this paper, we propose a disentangled interest network named DiseCTR to achieve OOD CTR prediction under both simple and complicated distribution variation. To address the above-mentioned challenges, we first design an interest encoder that utilizes sparse attention [80] to achieve effective interest embeddings, of which each interest only involves a few related features. We then

propose a weakly supervised interest disentangler which can regularize each interest to be both meaningful and independent. With the encoder and disentangler, we can generate robust interest embeddings from low-level features, which can well capture $P(Z|X)$. We then propose an attentive interest aggregator to accomplish $P(Y|Z)$, combining different interests for the final prediction. Overall speaking, DiseCTR takes full advantage of the *partial-distribution-variation* property, and thus it can generalize better towards OOD scenarios compared with existing approaches of CTR prediction. We illustrate the critical differences between DiseCTR and existing methods in Figure 2. To evaluate the accuracy and robustness of DiseCTR, we experiment on three real-world datasets. Results show that DiseCTR outperforms state-of-the-art approaches with significant improvements by over 0.02 on AUC and GAUC (improvement of 0.001 is acknowledged as significant [8, 52]) and over 13.7% on logloss. Moreover, we demonstrate the superior robustness of DiseCTR in OOD scenarios, especially for large distribution variation. Further in-depth analyses empirically show that DiseCTR successfully disentangles user interests, which is crucial to OOD generalization.

The contribution of this paper can be summarized as follows.

- We highlight the OOD issue and the partial-distribution-variation property in recommendation, and take the pioneer step to study OOD generalization for the general CTR prediction task.
- We design a novel model DiseCTR which introduces a causal perspective to disentangle user interests for OOD CTR prediction. The proposed method captures the partial-variation property with a sparse attentive interest encoder and a weakly supervised interest disentangler.
- We conduct extensive experiments on three real-world datasets. Experimental results demonstrate the superior performance of DiseCTR against state-of-the-art CTR models.

The remainder of the paper is as follows. We first formulate the problem of robust CTR prediction in Section 2 and elaborate on our proposed DiseCTR in Section 3. We then conduct experiments in Section 4 and carefully review related works in Section 5. Last, we conclude the paper and discuss the limitations and future works in Section 6.

2 PROBLEM FORMULATION

The CTR prediction problem is formulated as estimating a binary target Y from a feature vector $\mathbf{X}$ that is composed of N fields $\{X_1, X_2, \cdots, X_N\}$. Let $\mathcal{D}$ denote the joint distribution, $P(X, Y)$. We aim at learning a CTR model on training data $\mathcal{O}_{train}$, that can generalize to the test data $\mathcal{O}_{test}$ with $\mathcal{D}_{test}$ different from $\mathcal{D}_{train}$. Then the problem can be formulated as follows,

Input: Training dataset $\mathcal{O}_{train}$ and OOD test dataset $\mathcal{O}_{test}$, which both contain users, items and their attributes as feature vectors $\mathbf{X}$, and corresponding interaction labels Y.

Output: A model estimating the label with strong generalization ability towards unseen distributions.

It is worthwhile noting that the generalization ability of a CTR model is measured comprehensively following existing literature [3], which includes both accuracy and efficiency of transferring to OOD senarios, *i.e.* $\mathcal{O}_{test}$. as follows,

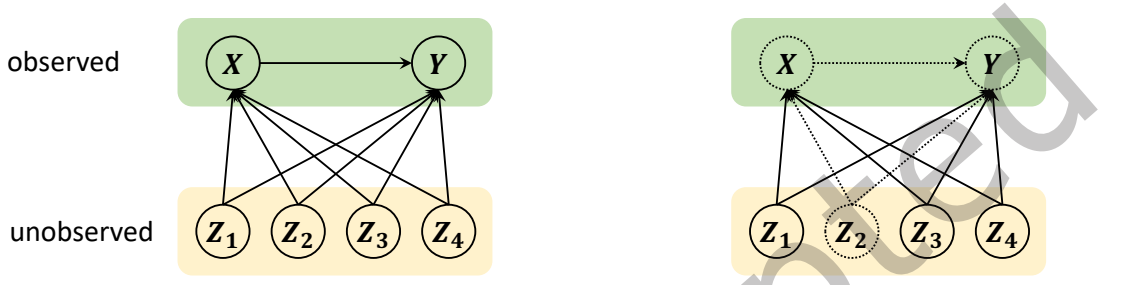
- **Transfer Accuracy.** The accuracy after transferring to $\mathcal{O}_{test}$.
- **Transfer Efficiency.** The sample cost of transferring to $\mathcal{O}_{test}$.

3 METHOD

3.1 A causal view of CTR prediction

It is acknowledged that high-dimensional observations such as images, are the causal effect of a set of low-dimensional independent factors [3, 37, 38, 47, 48]. For example, an image of a cube containing thousands of pixels

can be fully determined by a few causal variables such as the position and rotation of the cube. Therefore, learning representations that can keep consistent with the underlying causal mechanisms is expected to achieve better OOD generalization [11, 47, 48]. It can be explained that distribution variation usually comes from real-world interventions, which tend to influence only a few causal variables. In this paper, we view CTR prediction from the perspective of causal inference, and develop a causal modeling of interests, features, and interactions in recommendation. Specifically, building upon well-established causal theories [3, 37, 38, 47, 48], DiseCTR first decodes the underlying causal variables (interests) from high-dimensional observations (features), and then infer the effect (interaction) of these causal variables.



$P(X, Y, Z) = P(Z)P(X|Z)P(Y|X, Z)$ (a)

partial distribution shift on interest Z_2 (b)

Fig. 3. (a) Causal graph of features X, interactions Y and interests Z. (b) Partial distribution shift .

Intuitively, interactions between users and recommender systems can be modeled as a causal process of three types of causal variables, which are interests Z, features X and interactions Y. Figure 3(a) illustrates the causal graph, where we show four interests for example. The causal factorization [38, 47] of the joint distribution $P(X, Y, Z)$ is composed of three causal mechanisms, which are the intrinsic distribution of user interests $P(Z)$, the exposure model $P(X|Z)$ and the click model $P(Y|X, Z)$. Firstly, $P(Z)$ describes the user interests which are unobserved causal variables. Secondly, what contents are exposed to users is determined by the recommender system which learns to capture user interests. In other words, what users are interested in ($P(Z)$) decides what users see ($P(X)$) from the recommender system, which is the exposure model $P(X|Z)$ in the causal factorization. Thirdly, whether a user clicks an item ($P(Y)$) is determined by both what the user see ($P(X)$) and whether the user is interested in the item ($P(Z)$), which is the final click mechanism $P(Y|X, Z)$ in the causal factorization.

Most existing CTR prediction approaches directly ignore Z and learn $P(Y|X)$, since user interest Z is unobserved and difficult to track, However, such straightforward solutions are vulnerable to distribution shift which is common in real-world applications. As introduced previously, user interests usually change in a *partial* way, indicating that the variation of Z is sparse. Figure 3(b) demonstrates a sparse user interest shift. Specifically, in the example, one of the four interests, Z_2, changes, while the other three interests (Z_1, Z_3, and Z_4) remain largely stable. Since Z_2 is causally related to both X and Y, the conditional distribution $P(Y|X)$ is significantly affected by the drifted interest. Traditional CTR models directly learn $P(Y|X)$, which is vulnerable to the distribution variation occurring partially on interest variables Z. As a consequence, they tend to suffer large performance drops.

Unlike existing CTR predictions, we propose to learn $P(Z|X)$ and $P(Y|Z)$ instead of $P(Y|X)$, grounding DiseCTR on the actual causal factorization of recommendation illustrated by Figure 3(a). Using the above example, since the remaining majority of user interests, Z_1, Z_3 and Z_4, keep largely unchanged, DiseCTR can still take

effect under the partial variation on Z_2. By leveraging the partial distribution variation property, DiseCTR is more robust against interventions and can better generalize to OOD scenarios.

The proposed DiseCTR consists of the following three main components, interest encoder, interest disentangler and interest aggregator, as shown in Figure 2.

- **Interest Encoder.** To capture multiple aspects of user interests, we design an encoder to generate a set of interest embeddings. We utilize sparse attention [80] in the encoder which can force each interest to be related to only a small part of input features.
- **Interest Disentangler.** In order to leverage the partial property of interest shift, the obtained embeddings are first clustered to a set of interest prototypes, and further disentangled with pairwise weak supervision. In this way, interest embeddings are regularized to be independent and meaningful, which reduces the impact of distribution variation in OOD scenarios.
- **Interest Aggregator.** We design an attentive interest aggregator to predict CTR based on disentangled interests, and the attention weights of different interests are adapted to unseen distributions.

From the causal perspective, we first capture $P(Z|X)$ with the interest encoder and enhance robustness using the interest disentangler, then capture $P(Y|Z)$ with the interest aggregator.

3.2 Interest Encoder

Users tend to have multiple high-level interests Z which are implicitly related to low-level features X. Take e-commerce recommendation as an example, Z_i may be content interest related to item category and item image. While another Z_j may be economic interest correlated with item price and item brand. In other words, each interest corresponds with only a part of features, and different interests focus on distinct fields. Inspired by the recent study of sequential modeling that aims at attending to a few dominant timestamps, we utilize a sparse attention based encoder [80] on input features. Specifically, we develop multiple sub-encoders, and each one generates an interest embedding by attending to a small fraction of features. Note that features are transformed into dense vectors by an embedding layer before being fed into sub-encoders.

Embedding Layer. Since raw features are usually high dimensional, they are commonly transformed into dense vectors with embedding matrices [12, 52] as follows,

$$\mathbf{e} = [\mathbf{E_1}(x_1); \mathbf{E_2}(x_2); \cdots; \mathbf{E_N}(x_N)], \tag{1}$$

where N is the number of features, and x_i is the one-hot encoding of each feature. Here $\mathbf{E_i} \in \mathbb{R}^{N_i \times d}$ is an embedding matrix, with N_i as the number of possible values of x_i and d as the embedding size. With the embedding layer, we can express each sample as a tensor of shape (d, N), and each column represents one feature. Notice that the embedding layer is shared across the following M parallel sub-encoders, thus the number of embedding parameters in DiseCTR is the same with existing CTR models.

Parallel Sparse Sub-encoder. We design M parallel sub-encoders to capture multiple interests based on the multi-head self-attention mechanism [53], where M is a hyper-parameter. We first generate query, key, and value for attention as follows,

$$\mathbf{q} = \mathbf{W_q}\mathbf{e};\ \mathbf{k} = \mathbf{W_k}\mathbf{e};\ \mathbf{v} = \mathbf{W_v}\mathbf{e}, \tag{2}$$

where $\mathbf{W_q}, \mathbf{W_k}, \mathbf{W_v} \in \mathbb{R}^{d' \times d}$ are learnable transformation matrices and $d' = Hd$ with H as the number of attention heads. In the original full attention [31], we have to compute the similarity between any pair of query and key to obtain attention scores. However, the learned interest embedding is then related to all the features. As each interest only corresponds with a small fraction of features, we leverage *ProbSparse* self-attention in Informer [80] which only attends to a few dominant features, and the interest embedding is obtained by flattening the output

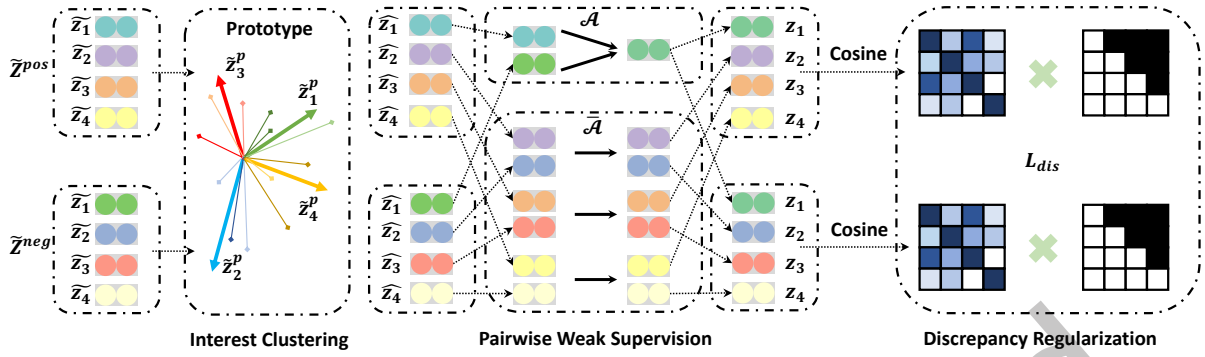


Fig. 4. We first cluster $\tilde{\mathbf{Z}}$ towards a set of interest prototypes, then impose weak supervision by averaging the pair adaptively which makes $\mathcal{A}$ and $\bar{\mathcal{A}}$ capture the overlapped and distinguishing interests of the pair respectively. Cosine similarity loss is further minimized to reduce mutual information between different interests. (Best viewed in color).

of the attention layer, as follows:

$$\tilde{\mathcal{A}} = \texttt{SoftMax}(\frac{\tilde{\mathbf{q}}\mathbf{k}^T}{\sqrt{d}})\mathbf{v},\ \tilde{\mathbf{z}} = \texttt{concat}(\{\mathbf{a}_1, \cdots, \mathbf{a}_N\}), \tag{3}$$

where $\tilde{\mathbf{q}}$ contains c queries with $c \ll N$, and a_i is the i-th row of the attentive output $\tilde{\mathcal{A}}$. In this way, the interest embedding is forced to be related to only a small part of input features. Readers can refer to Eqn (2-4) in [80] for details on the computation of Eqn (3).

To capture multiple interests, we develop M sub-encoders to generate a set of interest embeddings,

$$\tilde{\mathbf{Z}} = [\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2, \cdots, \tilde{\mathbf{z}}_M], \tag{4}$$

where each $\tilde{\mathbf{z}}_i \in \mathbb{R}^{d'}$ is generated with Eqn(3). Although the sparse attention layer makes each $\tilde{\mathbf{z}}_i$ correspond to only a few features, simply paralleling M sub-encoders is insufficient to capture M different interests, because these sub-encoders can be redundant, and interests in $\tilde{\mathbf{Z}}$ can entangle with each other. From the perspective of OOD generalization, it is critical to ensure that different interests are decoupled, which can guarantee that distribution variation on one interest will not affect another one. To this end, we now introduce our proposed disentangler which regularizes different interests in $\tilde{\mathbf{Z}}$ to be both meaningful and independent.

3.3 Interest Disentangler

The output embeddings $\tilde{\mathbf{Z}}$ in Eqn (5) are with high variance since they are directly computed from input features. Intuitively, interests are several different directions in a latent space, while each sample is a different combination of these directions. Therefore, we construct a set of interest prototypes which serve as these directions in the latent space, and perform interest clustering to obtain the portions in each direction. After that we regularize the clustered interests to be both meaningful and independent with pairwise weak supervision and explicit constraints. Figure 4 briefly illustrates the design of the proposed interest disentangler.

Interest Clustering with Prototypes. Since there is no ground-truth for the meaning of each interest, we propose to learn a set of interest prototypes $\mathbf{Z}^p$,

$$\mathbf{Z}^p = [\tilde{\mathbf{z}}_1^p; \tilde{\mathbf{z}}_2^p; \cdots; \tilde{\mathbf{z}}_M^p], \tag{5}$$

where $\tilde{\mathbf{z}}_i^p \in \mathbb{R}^{d'}$ resides in the same latent space and shares the same shape with the encoder outputs $\tilde{\mathbf{Z}}$. These prototypes are learnable parameters, randomly initialized and gradually converged to centroids of user interests

through continuous update using CTR prediction data. Specifically, we cluster interests in $\tilde{\mathbf{Z}}$ towards $\mathbf{Z}^p$ according to the distance between them. Then embeddings in $\tilde{\mathbf{Z}}$ are assigned to different clusters by performing projection to prototypes in $\mathbf{Z}^p$ as follows,

$$p_{j|i} = \frac{\texttt{Norm}(\tilde{\mathbf{z}}_i) \cdot \texttt{Norm}(\tilde{\mathbf{z}}_j^p)}{\sum_{l=1}^{M} \texttt{Norm}(\tilde{\mathbf{z}}_i) \cdot \texttt{Norm}(\tilde{\mathbf{z}}_l^p)}, \tag{6}$$

where $i, j \in 1, 2, \cdots M$, and $p_{j|i}$ is the projection weight of the i-th encoder output to the j-th prototype. The projection weight of $\tilde{\mathbf{z}}_i$ to different prototypes is normalized to sum to 1. Notice that we perform L2 normalization ($\texttt{Norm}(\cdot)$) on encoder outputs and prototypes before calculating the inner product between them. In other words, we cluster interest embeddings according to the *cosine similarity* with prototypes, which is shown to be effective in preventing mode collapse [32, 33], *i.e.* only a few prototypes are active. Particularly, using inner product can easily fall into mode collapse, since there is high probability that all the interest embeddings are clustered to the same prototype with the largest norm. It is worthwhile to notice that the number of prototypes is not necessarily the same with the number of parallel sub-encoders, M, while we keep them consistent for simplicity and leave different setups of encoders and prototypes for future work.

The clustered interest embeddings are projected from encoder outputs as follows,

$$\hat{\mathbf{z}}_k = \sum_{i=1}^{M} p_{k|i} \cdot \tilde{\mathbf{z}}_i, \ \hat{\mathbf{Z}} = [\hat{\mathbf{z}}_1; \hat{\mathbf{z}}_2; \cdots; \hat{\mathbf{z}}_M], \tag{7}$$

where $k = 1, 2, \cdots M$. By projecting to prototypes, the clustered interest embeddings $\hat{\mathbf{Z}}$ are more stable with lower variance than the original encoder outputs $\tilde{\mathbf{Z}}$, as shown in Figure 4. Meanwhile, the projection is fully compatible with back-propagation. Notice that adding M learnable prototypes only introduce Md' extra parameters, which is very small compared with parameters of embedding matrices. With such a small amount of parameters, interest embeddings are gathered in several directions specified by prototypes, and we can further regularize them to be meaningful and independent.

Pairwise Weak Supervision. In order to capture meaningful interests, we impose explicit supervision on the obtained M embeddings. Specifically, inspired by recent advances in Variational Auto-Encoder (VAE) [27, 28], we feed the model with samples in a pair-wise manner (a positive and a negative sample), and each pair can share a few interests in common. Then we regularize a subset of $\hat{\mathbf{Z}}$ to capture the shared interests, and make the rest of $\hat{\mathbf{Z}}$ capture the distinguishing interests of the pair. For example, a user may click on a cheap mobile phone and skip an expensive one with the same brand and similar functions. Then this pair is different in terms of economic-aspect interest, while other interests might be similar. Consequently, we can divide the interests of each pair into two subsets, shared interests $\mathcal{A}$ and distinguishing interests $\bar{\mathcal{A}}$, where $\mathcal{A} \cup \bar{\mathcal{A}} = \{1, 2, \cdots, M\}$ and $\mathcal{A} \cap \bar{\mathcal{A}} = \emptyset$. Particularly, we can have the following similarity constraints,

$$\text{sim}(\hat{\mathbf{z}}_i^{pos}, \hat{\mathbf{z}}_i^{neg}) < \delta_1 < \delta_2 < \text{sim}(\hat{\mathbf{z}}_j^{pos}, \hat{\mathbf{z}}_j^{neg}), i \in \mathcal{A}, j \in \bar{\mathcal{A}}, \tag{8}$$

where *pos* and *neg* indicate the pair, δ_1 and δ_2 are threshold values, and $\text{sim}(\cdot, \cdot)$ measures cosine similarity. In other words, it is the interests in $\bar{\mathcal{A}}$ that separate the pair into positive and negative sample, while the interests in $\mathcal{A}$ are not the reasons for the user's different behaviors towards the two items. In the above example, the economic interest is in $\bar{\mathcal{A}}$, while other interests are in $\mathcal{A}$.

In order to encourage the pair of learned embeddings to share interests in $\mathcal{A}$ and have different interests in $\bar{\mathcal{A}}$, we design the pairwise-similarity based disentangler as follows,

$$\mathbf{Z}^{pos} = F(\hat{\mathbf{Z}}^{pos}, \hat{\mathbf{Z}}^{neg}, \mathcal{A}), \mathbf{Z}^{neg} = F(\hat{\mathbf{Z}}^{neg}, \hat{\mathbf{Z}}^{pos}, \mathcal{A}), \tag{9}$$

where F is the disentangling function of the following form,

$$F(\mathbf{U}, \mathbf{V}, \mathcal{A}) = [f(\mathbf{u}_1, \mathbf{v}_1, \mathcal{A}); \cdots; f(\mathbf{u}_M, \mathbf{v}_M, \mathcal{A})], \tag{10}$$

$$f(\mathbf{u}_i, \mathbf{v}_i, \mathcal{A}) = \begin{cases} (\mathbf{u}_i + \mathbf{v}_i)/2, & \text{if } i \in \mathcal{A}; \\ \mathbf{u}_i, & \text{if } i \in \bar{\mathcal{A}}. \end{cases} \tag{11}$$

Specifically, as illustrated in Figure 4, we replace the interests in $\mathcal{A}$ with the average of the pair, and keep the interests in $\bar{\mathcal{A}}$ unchanged. As a result, $\mathbf{Z}^{pos}$ and $\mathbf{Z}^{neg}$ now share a few rows in common. In this way, we assign specific meanings to the learned interests, making embeddings in $\mathcal{A}$ and $\bar{\mathcal{A}}$ capture the shared and distinguishing interests of the sample pair, respectively.

It is challenging to find the shared interests of a sample pair, thus we propose to determine $\mathcal{A}$ and $\bar{\mathcal{A}}$ adaptively according to the similarity of M interests, which are calculated as follows,

$$\mathcal{A} = \{i \mid \text{sim}(\hat{\mathbf{z}}_i^{pos}, \hat{\mathbf{z}}_i^{neg}) < \delta\}, \ \bar{\mathcal{A}} = \{1, 2, \cdots, M\} \setminus \mathcal{A}, \tag{12}$$

where δ is set dynamically to make $|\mathcal{A}| = 1$ and $|\bar{\mathcal{A}}| = K - 1$, *e.g.* δ can be the second smallest similarity value of the M interests. In other words, we make the sample pair share one interest, which is rational since it is uncommon to obtain two samples that are different everywhere. The computation cost of this adaptive approximation is $O(M)$ which is in *constant time* complexity, and the results turn out to be fine. From our experiments, the difference between the running time of DiseCTR with and without this adaptive design is less than 0.01%, thus this adaptive design is very scalable. We leave other possible strategies for finding shared interests of sample pairs as future work.

Discrepancy Regularization. As introduced previously, distribution variation occurs in a partial way [47, 48], *i.e.* only a few interests are likely to be intervened while the remaining majority tends to remain stable. In order to utilize this property, we add a discrepancy constraint which regularizes the similarity between different interests to be as small as possible, making interests independent with each other. Specifically, we add an loss term on the cosine similarity of each two interests, encouraging M interests to capture different information, formulated as follows,

$$L_{dis} = \sum_{s \in \{pos, neg\}} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \texttt{Norm}(\hat{\mathbf{z}}_i^s) \cdot \texttt{Norm}(\hat{\mathbf{z}}_j^s), \tag{13}$$

which can be jointly optimized with the main CTR prediction loss.

Remark. In short, the interest disentangler encourages embeddings in $\mathbf{Z}$ to be both meaningful and independent. In terms of OOD generalization, independent embeddings guarantee that only a small part of the model parameter will be influenced by distribution variation, and meaningful embeddings ensure that the remaining parts of the model parameter will still take effect. Furthermore, such disentanglement can help DiseCTR accomplish fast transfer to OOD scenarios, since only a small affected part needs to be largely updated. Existing disentangled recommendation approaches [32, 58, 59] only impose independence regularization like the above L_{dis}, while they fail to add explicit supervision on the meaning of the disentangled factors. As a consequence, the learned disentangled representations may capture independent noise signals. In DiseCTR, we address this problem by introducing weak supervision, which can generate interest representations that are both meaningful and independent. We will show in experiments that the pairwise weak supervision design is a very crucial component in DiseCTR.

3.4 Interest Aggregator

As the interest encoder and disentangler accomplish $P(Z|X)$, we now propose an interest aggregator to achieve $P(Y|Z)$ that combines disentangled interests for prediction. The key challenge is that interests contribute differently in the original and OOD scenarios. Specifically, some interests might change in OOD scenarios, thus they become misleading and useless while the importance of other interests increases. For example, the aesthetic-aspect interest is important when users browse videos to kill time, while it is less important than the social-aspect interest when users browse videos uploaded by friends. In other words, the aggregation weights of different interests can vary. Therefore, we design an interest aggregator based on the attention mechanism [31] as follows,

$$\mathbf{A}_{agg} = \texttt{SoftMax}(\mathbf{W}_{agg}\mathbf{Z}), \tag{14}$$

$$\mathbf{z}_{agg} = \texttt{Flatten}(\mathbf{A}_{agg}\mathbf{Z}), \tag{15}$$

$$\hat{Y} = \texttt{sigmoid}(\mathbf{W}_{ctr}\mathbf{z}_{agg}), \tag{16}$$

where $\mathbf{W}_{agg} \in \mathbb{R}^{h \times d'}$ is a learnable attentive matrix with h as the number of attention heads. The final output is estimated with a linear predictor $\mathbf{W}_{ctr} \in \mathbb{R}^{hd' \times 1}$. It is worthwhile noting that the aggregator only introduces $h \times d'$ auxiliary parameters, which makes it easy to transfer to OOD scenarios, especially in practical scenarios where only a few OOD samples are available. When fine-tuning with OOD samples, $\mathbf{W}_{agg}$ can be updated to re-combine different interests with high efficiency due to its low parameter amount. Finally, DiseCTR is jointly trained on two objectives, CTR prediction and discrepancy of interests, formulated as follows,

$$L = L_{bpr}(\hat{Y}^{pos}, \hat{Y}^{neg}) + \alpha L_{dis} + \lambda \|\Theta\|_2, \tag{17}$$

where L_{bpr} is the widely adopted Bayesian Personalized Ranking loss [46]. Θ denotes the parameters of DiseCTR, and α, λ are loss weights for discrepancy and L2 regularization. The parameter scale of DiseCTR is comparable with existing approaches like AutoInt [52]. From the experiments, DiseCTR can be trained to converge within 20 minutes on a single GPU.

In summary, we design an encoder to obtain a set of interest embeddings, and each is forced to be related to only a few features by sparse attention. We further propose a disentangler to make interest embeddings independent and meaningful. At last, we design an attentive aggregator which combines all interests for prediction. We now provide a discussion on the relation between DiseCTR and existing works from the perspective of OOD generalization.

3.5 Discussion

3.5.1 Comparison between DiseCTR and typical CTR prediction approaches. The impact of distribution variation is largely reduced by utilizing the partial-variation property, thus DiseCTR can generalize well towards unseen distributions. From the perspective of model structure, the inner product in Factorization Machine[45] can be regarded as learning an interest for each feature pair, and the self-attention in AutoInt [52] can be considered as learning N interests that each automatically corresponds with N features. However, these approaches impose no supervision on the learned interests which make them entangle with each other. In DiseCTR, we encourage different interests to be both meaningful and independent with weak supervision. Therefore, stronger disentanglement is achieved, which will be further demonstrated in experimental results of Section 4. We summarize the differences between DiseCTR and several typical CTR models in Table 1.

3.5.2 Relation Between DiseCTR and VAE. The idea of learning disentangled representations is mostly investigated under the framework of VAE [17, 19], where images are encoded into low-dimensional vectors with independence regularization by minimizing the KL divergence loss. However, regularizing the independence in such an unsupervised way has been proved to be insufficient to capture meaningful semantics [26]. Variants of

Table 1. Comparison with typical CTR models. We compare the number of features, the number of interests, supervision (shortened as Sup.) on interests and disentanglement (shortened as Disen.) of interests.

Method	#Features	#Interests	Sup.	Disen.
LR	N	0	✗	no
FM	N	$\frac{N(N-1)}{2}$	✗	low
AutoInt	N	N	✗	low
DiseCTR	N	$M \ll N$	✓	high

VAE were proposed which disentangle with weak supervision, either by adding an extra prediction task [28], or by regularizing the similarity between pairs of observations [27]. In DiseCTR, discrepancy regularization and pairwise weak supervision have similar effects with these works. However, DiseCTR extends these methods to a higher dimensional space, *i.e.* DiseCTR learns disentangled vectors while these methods disentangle coordinates in one vector.

4 EXPERIMENTS

In this section, we conduct extensive experiments to answer the following research questions:

- **RQ1**: What is the overall generalization ability of DiseCTR compared with state-of-the-art CTR prediction approaches?
- **RQ2**: How does disentanglement of user interests help achieve OOD generalization for CTR prediction and benefit explainable recommendation?
- **RQ3**: What is the effect of each component in DiseCTR?

4.1 Experimental Settings

4.1.1 Datasets. We utilize the benchmark Amazon[1] dataset and real-world datasets collected from two largest short-video platforms in China, WeChat Channels[2] and Kuaishou[3]. Table 2 shows the statistics of the adopted datasets. The details of the datasets for evaluation are as follows:

- **Amazon**: This is a public benchmark dataset[4] for CTR prediction [36], which contains product reviews written by customers on the Amazon e-commerce website. We split the dataset into training, validation and test sets according to timestamps with the ratio of 8:1:1. The adopted features are: ReviewerID, ItemID, ItemCategory, ItemBrand and Price.
- **Wechat**: This is a public dataset released by WeChat Big Data Challenge 2021[5], which contains the logs on Wechat Channels (short-video platform) within 14 days. We use the data of first 10 days, middle 2 days and last 2 days as training, validation and test set. We use the following features: UserID, VideoID, DeviceID, AuthorID, BGMSongID, BGMSingerID, Duration, UserActivness, VideoPopularity.
- **Kuaishou**: This is an industrial dataset which contains the user interactions with short videos on Kuaishou APP during March 6 to March 15 in 2021. We use the data till March 13 as training set. We utilize the data on March 14 for validation, and evaluate the final performance with the data on March 15. We use the

[1]https://www.amazon.com/
[2]https://www.wechat.com/
[3]https://www.kuaishou.com/
[4]https://nijianmo.github.io/amazon/index.html
[5]https://algo.weixin.qq.com/

Table 2. Statistics of the adopted datasets.

Dataset	Users	Items	Instances
Amazon	84,260	385,897	3,278,991
Wechat	20,000	96,488	7,314,206
Kuaishou	20,303	222,791	10,848,234

Table 3. OOD evaluation on a single affected feature.

X_k	Train		Test	
	Y=1	Y=0	Y=1	Y=0
Cheap/Short	e	$1-e$	e'	$1-e'$
Expensive/Long	$1-e$	e	$1-e'$	e'

following features: UserID, VideoID, AuthorID, VideoCategory, AuthorCategory, Duration, UserActiveness, VideoPopularity.

4.1.2 OOD Evaluation. To evaluate the generalization ability, the distributions of training and test data need to be different. We consider two common OOD issues in real-world scenarios as follows,

- **OOD-easy.** The distribution variation mainly occurs on one single feature X_k, such as price in e-commerce recommendation when users are given coupons. Thus the differences of $P(X, Y)$ between training and test data mostly come from the variation of $P(Y|X_k)$, while other distributions $P(Y|X_i)$ remain largely unchanged. For simplicity, we select price and video duration as the affected feature X_k for different datasets, and transform the training and test data following the standard protocol introduced in related literature investigating OOD generalization [2, 29]. Specifically, the click-through rate for cheap/short and expensive/long items are different in training and test data. As illustrated in Table 3, we set click-through rate e as 0.6 in the training set, while we vary e' to obtain multiple test sets with different intensities of distribution variation. As shown in Table 3, the click-through rate of the single affected feature needs to be different in training and test data. To achieve such goal, we perform data sampling from the original datasets. Specifically, suppose the CTR of the cheap products in the Amazon dataset is 0.6. To obtain an OOD dataset with 0.2 CTR of cheap products, we randomly select 1/6 of the clicked cheap samples, and combine them with all the unclicked cheap samples. Then, CTR of cheap products is (0.6*1/6)/(0.6*1/6+0.4)=0.2. The data we use, including user, item, features, and clicks, are all from real-world applications. The Amazon and Wechat datasets are widely used as open benchmarks and Kuaishou dataset is from a real-world recommender platform. We did not synthesize the click values, and only use different sample ratios to obtain OOD data. In summary, we only perform random sampling to obtain OOD datasets, and all the clicked and unclicked samples are real.
- **OOD-hard.** We use multiple types of behaviors in the datasets, including click and like. Specifically, we train the model with click-behavior data, and evaluate the performance of predicting like-behavior. It is more challenging than OOD-easy since $P(Y|X)$ changes drastically, with $P(Y_{like} = 1|X)$ much smaller than $P(Y_{click} = 1|X)$. Note that the generalization ability for OOD-hard is critical in practice, because models are prone to overfitting when trained to predict like behavior, due to the insufficient training samples. Generalizing from rich click data to scarce like data can largely eliminate the overfitting issue.

In both cases, we follow the evaluation settings in [3] to measure the generalization ability. Specifically, after convergence on the IID training data, we transfer all the models by fine-tuning with a small fraction (10%) of OOD data, and investigate the performance on OOD data.

4.1.3 Baselines and Metrics. We compare DiseCTR with state-of-the-art approaches for CTR prediction. The details of the adopted baselines are as follows,

- **FM** [45]: This method captures feature interaction by taking inner product of feature embeddings. We use the most adopted 2-order FM which learns a cross feature for each pair of input features.
- **DeepFM** [12]: This method is an ensemble model which combines FM and Deep Neural Networks (DNN) to capture both second-order and higher order feature interactions.
- **NFM** [14]: This method proposes a Bi-Interaction layer to replace the inner product operation in FM, which increases the non-linearity modeling of high order feature interactions.
- **AutoInt** [52]: This is the state-of-the-art method which utilizes a multi-head self-attentive neural network to automatically learn feature interactions.
- **AFN** [8]: This is the state-of-the-art approach which designs an adaptive factorization network to learn arbitrary-order feature interactions.
- **DESTINE** [81]: This method proposes a self-attentive neural network to disentangle pairwise feature interaction and unary feature importance.

We do not include AutoFIS [23] due to its high computational cost. For fair comparison, we do not include those models that take users' historical interaction sequence as input like DIN [79] and DIEN [78]. It is worth noting that DiseCTR is fully compatible with sequential models and we leave it for future work. The two OOD recommendation methods [15, 57] are not included since they can not handle the general OOD-easy and OOD-hard problems in CTR prediction. To measure the prediction accuracy, we utilize two widely adopted metrics, AUC and GAUC [4, 79].

4.1.4 Experiment Setups. For both OOD-easy and OOD-hard experiments, we first train the recommendation model with IID data. We use Adam [18] to update the model parameters, and the learning rate is set as 0.0001. The adopted loss function is a combination of BPR loss [46] and the proposed discrepancy loss as shown in Eqn (17), and we set the loss weight α as 0.1. We use a single Nvidia GeForce GPU for model training, and the batch size is 2048. We train the models with early-stopping to avoid over-fitting. Specifically, we stop training when the AUC on validation set does not make progress for 5 epochs. After convergence on the IID training data, we finetune each model with a small fraction (10%) of OOD data. Finally, we evaluate the recommendation performance on OOD test data to compare the generalization ability of different models. To guarantee fair comparison, we fix the embedding size as 32 for all methods, and other hyper-parameters are carefully tuned by grid search. For DiseCTR, we set the number of interests M as 4, and the number of attention heads H and h in the encoder and aggregator as 2 and 8, respectively. Code based on TensorFlow [1] and datasets to reproduce the experimental results in this paper are released at https://github.com/DavyMorgan/DiseCTR/.

4.1.5 Weak supervision for OOD-easy. For OOD-easy, pairwise weak supervision is enhanced to make the last interest capture the affected feature. Specifically in OOD-easy, distribution shift occurs on feature X_k which makes $P(Y|X_k)$ different in training and test data. Since the conditional distributions of other features are largely invariant, we propose to separate the interest related to X_k from other interests. Specifically, we force the last (*i.e.* *M-th*) embedding $\mathbf{z}_M$ in $\mathbf{Z}$ to capture the interest about X_k, and $\mathcal{A}$ is decided according to X_k in the pair:

$$(\mathcal{A}, \bar{\mathcal{A}}) = \begin{cases} (\{M\}, \{1, 2, \cdots, M-1\}), & \text{if } X_k^{pos} = X_k^{neg} \\ (\varnothing, \{1, 2, \cdots, M\}), & \text{if } X_k^{pos} \neq X_k^{neg} \end{cases} \tag{18}$$

In other words, we take average of the M-th interest if X_k in the positive sample is the same with the negative sample.

To further regularize the meanings of $\mathbf{Z}$, we utilize a fully connected classifier $\mathbf{W}_{\mathbf{weak}} \in \mathbb{Z}^{N_k \times d'}$ to reconstruct X_k from $\mathbf{Z}$, inspired by recent advances in disentangled VAE [28]. Specifically, we expect X_K to be predictable from $\mathbf{z}_M$, while not predictable from other interests. Therefore, we apply the extra prediction task on $\mathbf{z}_M$, as well as $\mathbf{z}_i$ with $i < M$ but in an adversarial way :

$$\hat{X}_k^{(i)} = \text{SoftMax}(\mathbf{W}_{\mathbf{weak}} \cdot \mathbf{z}_i), \tag{19}$$

$$L_{weak} = L_{CE}(X_k, \hat{X}_k^{(M)}) - \sum_{i=1}^{M-1} L_{CE}(X_k, \hat{X}_k^{(i)}), \tag{20}$$

where L_{CE} is the CrossEntropy loss function, and the adversarial negative loss on $\mathbf{z}_i$ with $i < M$ is implemented with a gradient reversal layer [9, 76]. By mapping the the M-th interest to the affected feature X_k, we largely reduce the impact of distribution shift, since only one sub-encoder is influenced while other sub-encoders are to great extent stable in OOD cases.

It is worthwhile to notice that baseline methods learn rather entangled interest representations, which means that the influence of the affected feature X_k is almost contained in every learned interest embedding. Thus the impact of X_k can not be separated to one specific interest in the baselines. Intuitive solutions like discarding the affected features X_k is investigated in Section 4.2.2, however, the performance is even worse since useful information of X_k is also removed.

In real-world scenarios, OOD-easy cases are quite common. For example, users can grow loyalty on a specific brand, then the brand-aspect interest varies while other interests may largely remain stable. Another example can be providing coupons to users which almost only changes the economic-aspect interest, as introduced in the paper. Being capable of adapting to specific OOD issues is a great advantage of the proposed DiseCTR, and such advantage comes from the disentanglement design and the effective usage of the partial-variation property. Existing baselines can not handle OOD-easy cases well since different interests are entangled with each other and the impact of the affected feature X_k can not be captured separately.

In fact, learning disentangled interests makes it possible for DiseCTR to separate the influence of distribution variation in OOD-easy. Baseline methods are limited in eliminating the impact of distribution shift. One potential solution is data calibration such as removing the affected X_k from input features. However, the performance is even worse and we show the results in Section 4.2.2.

4.2 Generalization Performance (RQ1)

4.2.1 ***Overall Comparison.*** We set e' as 0.2 that is different from $e = 0.6$ in training data to evaluate OOD-easy performance, and use *like* behavior as the OOD environment against *click* behavior for OOD-hard. Table 4 shows the performance on three datasets. Note that the Amazon dataset only has one type of behavior, and thus we only evaluate OOD-easy performance on the Amazon dataset. From the results we have the following observations:

- **Directly learning correlations of low-level features performs poorly in OOD cases.** FM, DeepFM and NFM, provide much worse performance in most cases compared with other methods. These methods concatenate all the input features, and utilize rather simple feature interaction layers, like inner product and multi-layer perceptrons, for prediction. Since directly learning $P(Y|X)$ is vulnerable to the distribution variation, these methods perform poorly in OOD scenarios.
- **Entangled modeling of interests is insufficient to accomplish OOD generalization.** Although AutoInt and AFN leverage self-attention and adaptive feature interaction, which are large-capacity neural layers, compared with direct matching raw features, they still cannot capture the multiple aspects of interests. In other words, the obtained embeddings are redundant with much mutual information, which

Table 4. Overall performance comparison. We evaluate the accuracy after finetuning with 10% of OOD data. Two evaluation protocols, OOD-easy and OOD-hard are adopted. Underline means the 2nd and 3rd best methods, **bold** means p-value < 0.05.

Protocol	OOD-easy ($e = 0.6, e' = 0.2$)						OOD-hard (click → like)			
Dataset	Amazon		Wechat		Kuaishou		Wechat		Kuaishou	
Method	AUC	GAUC	AUC	GAUC	AUC	GAUC	AUC	GAUC	AUC	GAUC
FM	0.5915	0.4694	0.8356	0.7495	0.6901	0.5034	0.6880	0.5396	0.7256	0.5037
DeepFM	0.6273	0.5156	0.7954	0.7232	0.7064	0.5452	0.6710	0.5400	0.7547	0.5356
NFM	0.6736	0.5815	0.8336	0.7433	0.6955	0.5278	0.6716	0.5237	0.7424	0.5154
AutoInt	0.6279	0.5108	0.8362	0.7508	0.7113	0.5500	0.6686	0.5387	0.7252	0.5051
AFN	0.6614	0.5676	0.8387	0.7557	0.7086	0.5425	0.6850	0.5442	0.7636	0.5251
DESTINE	0.6344	0.5277	0.8335	0.7520	0.7100	0.5482	0.6711	0.5356	0.6792	0.4820
DiseCTR	**0.6762**	**0.5972**	**0.8532**	**0.7678**	**0.7198**	**0.5736**	**0.7071**	**0.5494**	**0.7653**	**0.5449**

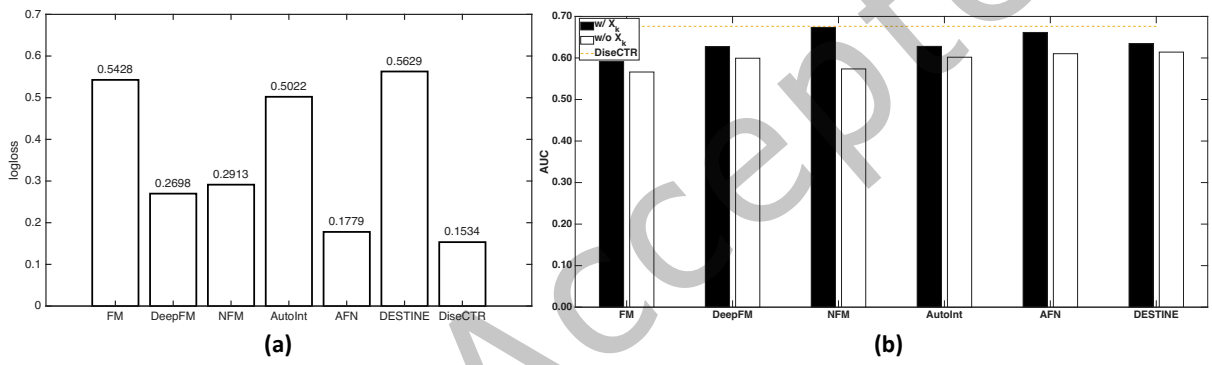


Fig. 5. (a) Logloss of different methods on the Kuaishou dataset under OOD-hard protocol. (b) AUC of baseline methods with the affected feature (price) discarded on Amazon dataset.

entangle different aspects. Therefore, their performance in OOD cases is still limited since distribution variation can influence a large part of these models.

- **DiseCTR can generalize well to OOD scenarios.** Our DiseCTR achieves the best OOD performance in most cases with significant improvements. Specifically, on Amazon dataset, GAUC is improved by 0.016 against NFM; DiseCTR improves AUC by about 0.015 (OOD-easy) and 0.020 (OOD-hard) against AFN on Wechat dataset; on Kuaishou dataset, GAUC is improved by 0.024 (OOD-easy) and 0.040 (OOD-hard) compared with AutoInt. We also calculate the logloss [68] metric of all methods for the challenging OOD-hard scenario on Kuaishou dataset. As illustrated in Figure 5a, our DiseCTR approach achieves the lowest logloss with over 13.7% improvements against AFN. By learning disentangled representations of interests, DiseCTR reduces the impact of distribution shift to only a small fraction of the model, achieving fast transfer towards unseen distributions.

4.2.2 ***Performance of Data Calibration.*** One intuitive solution to OOD-easy is simply discarding the affected feature X_k, then the distribution variation of remaining features will be largely eliminated. However, such straightforward approach also removes useful information which may lead to worse performance. For example,

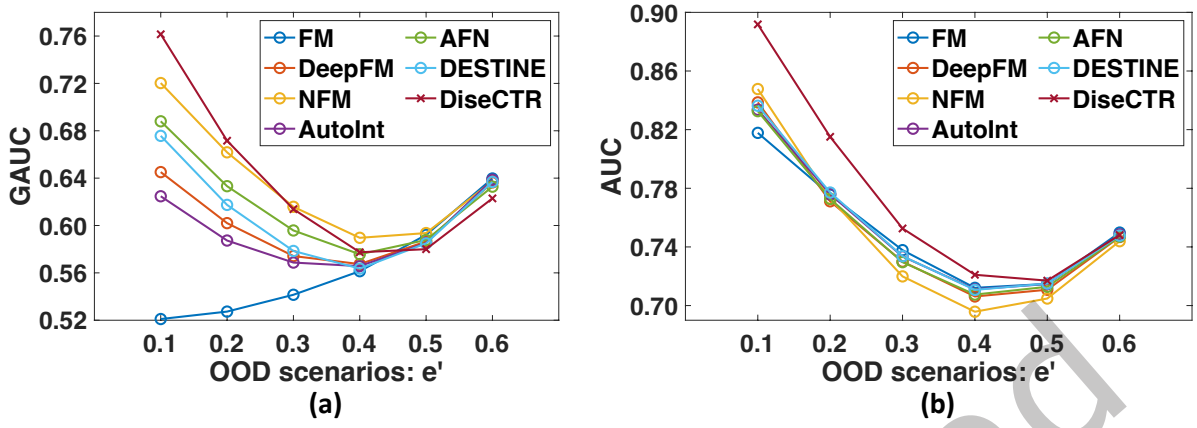


Fig. 6. Transfer accuracy evaluation on (a) Amazon and (b) Kuaishou dataset. Higher means better.

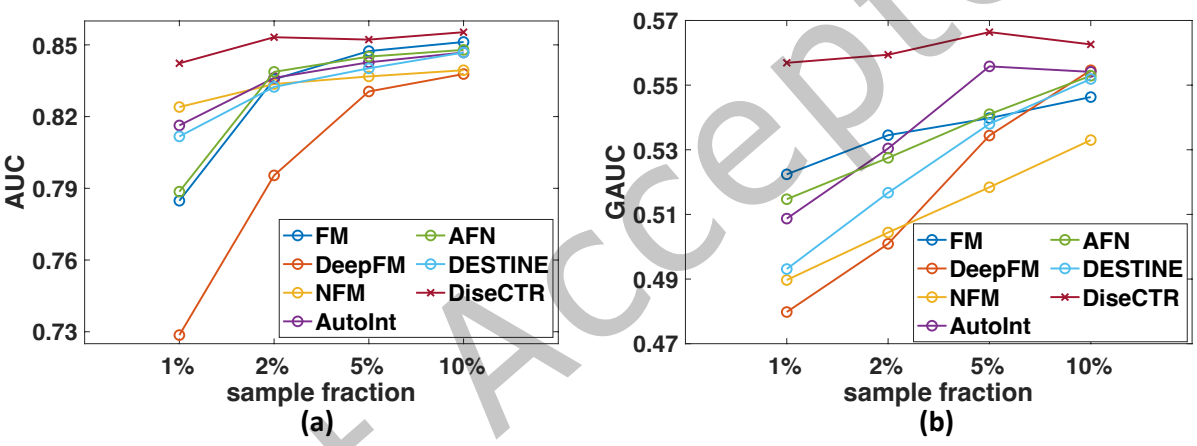


Fig. 7. Transfer efficiency comparison. (a) OOD-easy on Wechat dataset. (b) OOD-hard on the Kuaishou dataset.

providing users with coupons change the distribution $P(Y|X_{price})$, while removing X_{price} from the input features will make the model fail to learn the drifted relation from new data, thus it can not recommend proper items with higher price. We discard the affected feature for all the baselines, and results are shown in Figure 5 (b). We can observe that the performance of all the baselines drops drastically after removing the price feature of Amazon dataset. In other words, it is better to *adapt* to the varied distribution $P(Y|X_k)$ instead of *ignoring* it. Through interest disentanglement, the proposed DiseCTR can achieve fast adaptation to new distributions.

4.2.3 ***Transfer Accuracy on Different OOD Scenarios.*** Since the intensity of OOD issue is usually unknown in practice, we vary the value of e' in the range of 0.1 and 0.6 to simulate multiple environments with different intensities of OOD issue. Intuitively, smaller e' indicates larger distribution variation with training data. Meanwhile, as e' decreases, the affected feature X_k also becomes a stronger predictor in the OOD scenario. Figure 6 illustrates the results on Amazon and Kuaishou dataset. We can find that all the models are of similar performance when $e' = 0.6$ which equals to e in training data, where the IID assumption holds true. However,

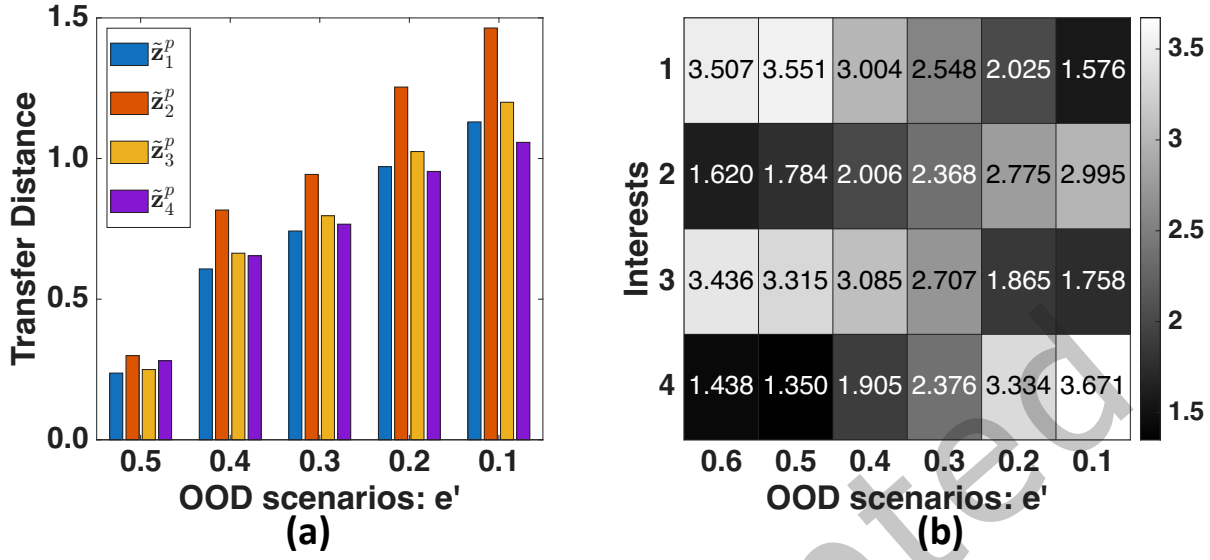


Fig. 8. (a) Transfer distance of prototypes on Kuaishou dataset. (b) Heatmap of attention weights on different interests on Kuaishou dataset.

more importantly, as we decrease e' to simulate OOD scenarios, the proposed DiseCTR starts to outperform other baselines. Specifically, we can observe that DiseCTR outperforms other baselines in almost all levels of distribution variation. Meanwhile, the improvements become larger as e' moves to extreme OOD cases, where AUC on the Kuaishou dataset is improved by over 0.02 when $e' = 0.3$, and over 0.04 when $e' \leq 0.2$. By learning disentangled representations of multiple interests, DiseCTR is more stable in OOD scenarios since distribution variation only influences a few interests and the majority of the model can still take effect. In onther words, only a few parameters related to the affected feature need to be updated significantly, while other parameters only need a small amount of fine-tuning. Therefore, DiseCTR can effectively adapt to multiple OOD scenarios with different intensities of distribution variation.

4.2.4 ***Comparison of Transfer Efficiency.*** Since OOD data is usually expensive and hard to obtain in practice, models are required to transfer with limited data. Therefore, we use different amounts of OOD data for fine-tuning to investigate the transfer efficiency, which has also been adopted in [3]. Figure 7 demonstrates the performance of fine-tuning with 1%, 2%, 5%, and 10% of training data. We can find that DiseCTR consistently outperforms baselines under both protocols, even with only 1% of the OOD training data, and the improvements are consistently significant with different amounts of OOD training data. Meanwhile, the performance with 1% sample fraction is very close to the performance with 10% sample fraction, which means that DiseCTR only requires a very small number of samples to accomplish transfer towards unseen distributions. Overall speaking, the disentangled structures of interest largely reduce the amount of parameters that need to be updated, which makes DiseCTR achieve fast transfer with high data efficiency.

Table 5. Results of different M on Wechat dataset.

M	1	2	4	8
AUC	0.6911	0.6931	**0.7071**	0.6871

4.3 Explainability and Disentanglement Analysis (RQ2)

We analyze the generalization ability of DiseCTR and how it benefits explainable recommendation from the perspective of model parameters. Specifically, we visualize the learned interest embeddings to demonstrate the independence of different interests, which is a critical aspect for explainability. To empirically verify that leveraging the partial-variation property is critical in OOD generalization, we investigate how the parameters in DiseCTR change during the transfer process. Specifically, we calculate the *transfer distance* of interest prototypes, which is the value differences between parameters before and after fine-tuning with OOD data. Note that larger distance indicates more significant updates. Figure 8(a) illustrates the transfer distances to different OOD scenarios with various e'. We can observe that the transfer distances for all the four prototypes increase as we gradually decrease e', which is in line with expectations since smaller e' indicates larger distribution shift. Particularly, the transfer distance of one prototype $\tilde{\mathbf{z}}_2^p$ is much larger than the other three prototypes. In other words, only a small fraction of parameters in DiseCTR need significant updating, while most of the parameters only need small modification. Meanwhile, the results in Figure 8(a) also indicates that the interest shift and evolution are mostly captured by the prototype embedding $\tilde{\mathbf{z}}_2^p$, and other prototype embeddings $\{\tilde{\mathbf{z}}_1^p, \tilde{\mathbf{z}}_3^p, \tilde{\mathbf{z}}_4^p\}$ capture the unchanged user interests, which significantly improve the explainability of recommendation. By learning disentangled interest embeddings, DiseCTR leverages the partial-variation property, which accomplishes explainable and fast transfer to OOD cases.

We also visualize the attention scores in the interest aggregator before and after fine-tuning with OOD data. We compute the rank of four interests according to their attention scores, where larger rank of one interest indicates that it gains more attention against other interests. Specifically, we take one attention head and calculate the attention score for 4,096 samples. We then rank the attention scores of four interests and calculate the averaged ranks of these 4,096 samples. Notice that we force the last interest embedding to capture the interest related to the affected feature (video duration). Meanwhile, e' indicates the positive sample rate of short videos, thus $e' = 0.5$ is the most difficult case to predict CTR from video duration, while $e' = 0.1$ is the easiest case where video duration is a very strong feature. Figure 8(b) demonstrates how attention changes from the IID case ($e' = 0.6$) to different OOD cases on the Kuaishou dataset. We can observe that the attention score of the last interest drops when $e' = 0.5$, and then continuously increases as e' turns down towards extreme OOD cases. In other words, the attention score on the last interest greatly matches the feature importance of video duration, which verifies that DiseCTR successfully disentangles different interests. More importantly, the consistent trends of test data and attention rank observed in Figure 8(b) imply that the 4-th interest captures the user preferences on video duration, and the first 3 interests capture the user preferences on other factors unrelated to video duration, thus facilitating explainable recommendation using the attention ranks of different interests.

4.4 Ablation and Hyper-parameter Study (RQ3)

We verify the design of DiseCTR by comparing with several counterparts of interest encoder and removing specific components of interest disentangler.

4.4.1 ***Effectiveness of Interest Encoder.*** In the encoder, sparse attention forces each interest to only focus on a few input features. We compare with other encoders including MLP, AutoInt [52] and AFN [8]. Figure 9(a) illustrates the results on two datasets. Dense connections like AutoInt provide worse performance since each

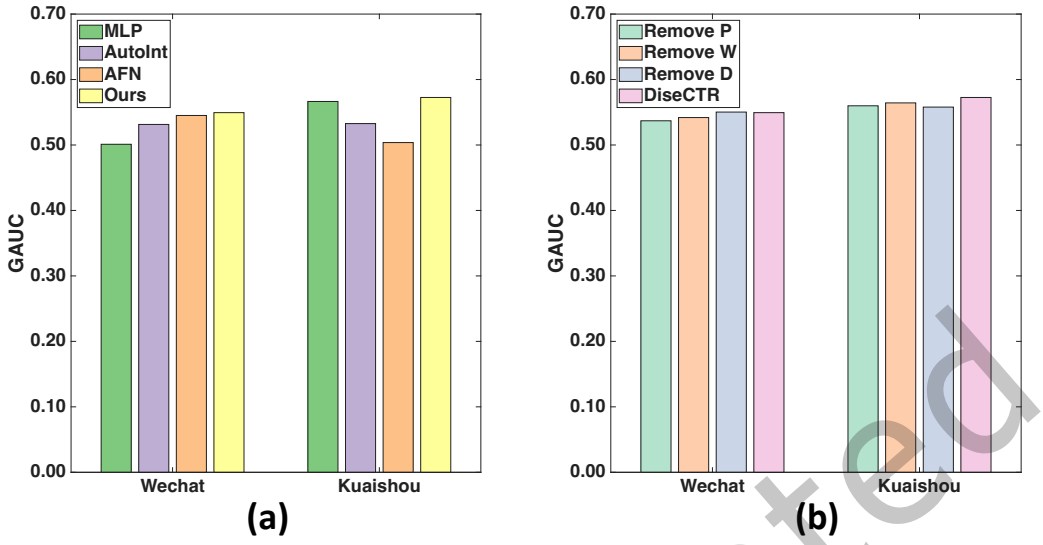


Fig. 9. (a) Performance comparison of different encoders. (b) Ablation on interest disentangler.

interest is related to almost all features, which is vulnerable to distribution variation. Our encoder achieves the best GAUC on both datasets, which confirms the necessity of constructing sparse connections between interests and features.

4.4.2 ***Effectiveness of Interest Disentangler.*** We investigate different components of the disentangler. Figure 9(b) shows the performance of DiseCTR and its variants with specific designs removed including prototypes (P), weak supervision (W) and discrepancy loss (D). We can observe that removing prototypes leads to the worst GAUC on the WeChat dataset and the second worst GAUC on the Kuaishou dataset. Without prototype vectors, interest clustering is conducted directly on the encoder outputs, which have high variance. The prototypes provide learnable cluster centroids, which reduces the variance of interest embeddings greatly after clustering. Moreover, pairwise weak supervision averages interests of sample pairs adaptively, which forces interest embeddings to capture shared and distinguishing interests respectively. We can find that removing it leads to worse performance on both datasets. At last, the discrepancy regularization encourages the independence between different interests, and removing L_{dis} results in drastic performance drop on Kuaishou dataset.

4.4.3 ***Impact of the Number of Interests.*** We compare the performance of DiseCTR with different number of interests. Table 5 shows the results of OOD-hard on Wechat dataset. We can observe that increasing M from 1 to 4 can effectively improve the expressive ability of DiseCTR and achieve better performance. In addition, further increasing M to 8 may introduce redundancy and overfitting, which leads to performance drop.

In summary, we conduct extensive experiments to demonstrate the OOD generalization ability of the proposed method. On the one hand, we show that DiseCTR can generalize well to different OOD scenarios, and the improvements are particularly larger in extreme OOD cases. On the other hand, we illustrate that DiseCTR can achieve OOD generalization with high data efficiency. Furthermore, we show that leveraging the partial-variation property is the key of OOD generalization. Specifically, by disentangling user interests, only a small number of parameters related to the varying interest need a significant update, while most of the parameters only need

a small amount of fine-tuning, which makes DiseCTR generalize towards unseen distributions accurately and efficiently.

5 RELATED WORK

5.1 Interest Learning in CTR Prediction

Learning user interests is critical for CTR prediction, while most studies learn a relatively entangled interest representation. Specifically, existing methods [7, 12, 14, 22, 23, 39, 41, 42, 45, 49, 55, 56, 62, 64, 67–70, 78, 79, 81] aim to perform direct matching from features to interactions. For example, Rendle *et al.* [45] utilized inner product to capture feature interaction between every feature pair. The inner product operation is further replaced by a Bi-interaction layer in [14], and combined with MLP in [12]. Recently, Song *et al.* [52] proposed to learn interests based on self-attention, and Cheng *et al.* [8] further developed an adaptive factorization network to learn arbitrary-order feature interactions. Besides pre-defined feature interactions, Yu et al. [68] proposed an evolutionary search approach to adaptively select proper interactions on features for accurate CTR prediction. However, they largely ignore Z and mainly capture $P(Y|X)$, which can be vulnerable to distribution shift. Unlike existing methods that generalize from one data point to another one in the same distribution, we focus on the generalization from one distribution to unseen distributions and propose to learn disentangled interests to achieve such a goal.

5.2 Disentangled Recommendation

Some studies [6, 44, 58–60, 72, 74, 75, 77, 81] investigate disentangled user interests in recommendation systems. For example, Wang et al. [60] proposed to learn independent representations for different user intents by contrastive learning on a user-item-concept graph. Zhang et al. [72] divided an item graph into multiple subgraphs with distinct semantics, then leveraged graphical disentangled learning to capture user's diverse preferences within each subgraph. Ren et al. [44] further improved the robustness of user interest disentanglement and distilled fine-grained latent factors with adaptive self-supervised augmentation. However, most of these works only tackle the simplified collaborative filtering (CF) task, and they are all based on the IID assumption which hardly holds true in real-world scenarios. Two recent works [15, 57] investigate the OOD issue in recommendation. Nevertheless, He et al. [15] studies OOD in CF and can not handle the more complicated CTR prediction task. Meanwhile, Wang et al. [57] only investigates user feature shift such as income increase, which can not solve the more general OOD-easy and OOD-hard tasks in this paper. Different from these methods, our work investigates general OOD scenarios in the more complicated CTR prediction task, which is quite common in real applications.

5.3 Model Robustness and OOD Generalization

Machine learning theories guarantee optimum accuracy in IID cases, while the performance often drops drastically in OOD scenarios [35, 48]. To obtain a robust model that generalizes to unseen distributions, several studies [2, 29, 30, 50, 51, 63, 73] were proposed to remove spurious correlations in the training data. For example, Arjovsky et al. [2] proposed invariant risk minimization (IRM) which can guarantee generalization ability for linear models. Lu et al. [29] further extends IRM to non-linear cases with a causal inference approach. In terms of recommendation, robustness from the perspective of OOD generalization is largely unexplored, while a few works study the robustness under adversarial attack [24, 40, 61], which is a quite different definition of "robust" compared with our work. As far as we know, we are the first to investigate OOD generalization for CTR prediction, which is a critical problem since distribution variation is common in practice.

6 CONCLUSION AND FUTURE WORK

In this paper, we investigate the problem of OOD generalization for CTR prediction, an emerging issue in many practical online services such as e-commerce and micro-video platforms. A novel model named DiseCTR is introduced to reduce the impact of distribution variation by disentangling the multiple aspects of user interests. The key of DiseCTR's architecture is to impose weak supervision on disentanglement by clustering towards learnable interest prototypes and adaptively averaging the shared interest embeddings of the sample pair. Experiments on real-world datasets demonstrate that DiseCTR can generalize well to unseen distributions. Further analysis on DiseCTR provide valuable insights, that stronger disentanglement can achieve lower transfer distance and more interpretable attention scores.

While DiseCTR has demonstrated promising results in handling OOD issues in CTR prediction, it has some limitations that need to be considered for practical application. First, although DiseCTR only introduced a relatively small number of additional parameters, its computational cost may increase with very large datasets. Specifically, as the number of features and the embedding size increase, the computational overhead of the encoder and disentangler could become significant. The pairwise weak supervision also adds complexity by requiring operations on pairs of samples. Second, the current implementation of DiseCTR is primarily focused on offline training and evaluation. Adapting the model to handle real-time data streams in recommendation systems presents a number of challenges, as models for real-time recommendation need to react and update their recommendations fast. Thus DiseCTR needs to be further optimized to quickly adapt to new data while maintaining its OOD generalization capabilities. Online learning strategies, such as incremental learning, can be adopted for effective real-time adaptation.

As for future work, we plan to further incorporate sequential modeling such as MIMN [39] and UBR4CTR [42] into our proposed DiseCTR. Besides, as a general approach for OOD generalization, it is a promising direction advanced CTR prediction models that incorporate feature interaction search such as CELS [68], and we leave it as important future work. We also plan to evaluate DiseCTR via online A/B tests. In real-world applications, there are numerous OOD scenarios and the multiple types of behaviors in the same domain, *i.e.* the OOD-hard problem in this paper, is only one scenario. Thus it is a promising direction for future work to investigate other OOD scenarios such as cross domain generalization. Overall, we believe DiseCTR makes one step closer to the real-world scenarios where distribution variation is happening all the time.

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 *OSDI 2016)*.

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893* (2019).

[3] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. 2019. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912* (2019).

[4] 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 *SIGIR*. 378–387.

[5] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–28.

[6] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to Debias for Recommendation. *arXiv preprint arXiv:2105.04170* (2021).

[7] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[8] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive factorization network: Learning adaptive-order feature interactions. In *AAAI*. 3609–3616.

[9] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*. 1180–1189.

[10] Chongming Gao, Shiqi Wang, Shijun Li, Jiawei Chen, Xiangnan He, Wenqiang Lei, Biao Li, Yuan Zhang, and Peng Jiang. 2023. CIRS: Bursting filter bubbles by counterfactual interactive recommender system. *ACM Transactions on Information Systems (TOIS)* 42, 1 (2023), 1–27.
[11] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. 2024. Causal Inference in Recommender Systems: A Survey and Future Directions. *ACM Trans. Inf. Syst.* 42, 4, Article 88 (feb 2024), 32 pages. https://doi.org/10.1145/3639048
[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).
[13] Long Guo, Lifeng Hua, Rongfei Jia, Binqiang Zhao, Xiaobo Wang, and Bin Cui. 2019. Buying or browsing?: Predicting real-time purchasing intent using attention-based deep network with multiple behavior. In *KDD*. 1984–1992.
[14] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*. 355–364.
[15] Yue He, Zimu Wang, Peng Cui, Hao Zou, Yafeng Zhang, Qiang Cui, and Yong Jiang. 2022. CausPref: Causal Preference Learning for Out-of-Distribution Recommendation. In *Proceedings of the ACM Web Conference 2022*. 410–421.
[16] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. 2021. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*. 8340–8349.
[17] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-vae: Learning basic visual concepts with a constrained variational framework. (2016).
[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[19] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
[20] Liangwei Li, Liucheng Sun, Chenwei Weng, Chengfu Huo, and Weijun Ren. 2020. Spending Money Wisely: Online Electronic Coupon Allocation based on Real-Time User Intent Detection. In *CIKM*. 2597–2604.
[21] Shijun Li, Wenqiang Lei, Qingyun Wu, Xiangnan He, Peng Jiang, and Tat-Seng Chua. 2021. Seamlessly unifying attributes and items: Conversational recommendation for cold-start users. *ACM Transactions on Information Systems (TOIS)* 39, 4 (2021), 1–29.
[22] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *KDD*. 1754–1763.
[23] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction. In *KDD*.
[24] Yang Liu, Xianzhuo Xia, Liang Chen, Xiangnan He, Carl Yang, and Zibin Zheng. 2020. Certifiable robustness to discrete adversarial perturbations for factorization machines. In *SIGIR*. 419–428.
[25] Yiqun Liu, Junqi Zhang, Jiaxin Mao, Min Zhang, Shaoping Ma, Qi Tian, Yanxiong Lu, and Leyu Lin. 2019. Search result reranking with visual and structure information sources. *ACM Transactions on Information Systems (TOIS)* 37, 3 (2019), 1–38.
[26] 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 *ICML*.
[27] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. 2020. Weakly-supervised disentanglement without compromises. In *ICML*. 6348–6359.
[28] Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. 2019. Disentangling factors of variation using few labels. *arXiv preprint arXiv:1905.01258* (2019).
[29] Chaochao Lu, Yuhuai Wu, Jośe Miguel Hernández-Lobato, and Bernhard Schölkopf. 2021. Nonlinear invariant risk minimization: A causal approach. *arXiv preprint arXiv:2102.12353* (2021).
[30] Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. 2022. Invariant Causal Representation Learning for Out-of-Distribution Generalization. In *ICLR*.
[31] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).
[32] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. *arXiv preprint arXiv:1910.14238* (2019).
[33] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *KDD*.
[34] Chang Meng, Ziqi Zhao, Wei Guo, Yingxue Zhang, Haolun Wu, Chen Gao, Dong Li, Xiu Li, and Ruiming Tang. 2023. Coarse-to-fine knowledge-enhanced multi-interest learning framework for multi-behavior recommendation. *ACM Transactions on Information Systems (TOIS)* 42, 1 (2023), 1–27.
[35] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of machine learning*. MIT press.
[36] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP-IJCNLP*.
[37] Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic books.
[38] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

[39] 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 *KDD*. 2671–2679.
[40] Surabhi Punjabi and Priyanka Bhatt. 2018. Robust factorization machines for user response prediction. In *WWW*. 669–678.
[41] Jiarui Qin, Weinan Zhang, Rong Su, Zhirong Liu, Weiwen Liu, Guangpeng Zhao, Hao Li, Ruiming Tang, Xiuqiang He, and Yong Yu. 2023. Learning to retrieve user behaviors for click-through rate estimation. *ACM Transactions on Information Systems (TOIS)* 41, 4 (2023), 1–31.
[42] Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. 2020. User behavior retrieval for click-through rate prediction. In *SIGIR*. 2347–2356.
[43] Yingrong Qin, Chen Gao, Shuangqing Wei, Yue Wang, Depeng Jin, Jian Yuan, Lin Zhang, Dong Li, Jianye Hao, and Yong Li. 2023. Learning from hierarchical structure of knowledge graph for recommendation. *ACM Transactions on Information Systems (TOIS)* 42, 1 (2023), 1–24.
[44] Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. 2023. Disentangled contrastive collaborative filtering. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1137–1146.
[45] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE ICDM*. IEEE, 995–1000.
[46] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
[47] Bernhard Schölkopf. 2019. Causality for machine learning. *arXiv preprint arXiv:1911.10500* (2019).
[48] 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.
[49] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *KDD*. 255–262.
[50] Zheyan Shen, Peng Cui, Jiashuo Liu, Tong Zhang, Bo Li, and Zhitang Chen. 2020. Stable learning via differentiated variable decorrelation. In *KDD*. 2185–2193.
[51] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. 2021. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624* (2021).
[52] 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 *CIKM*. 1161–1170.
[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.
[54] Chenyang Wang, Weizhi Ma, Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. Sequential recommendation with multiple contrast signals. *ACM Transactions on Information Systems (TOIS)* 41, 1 (2023), 1–27.
[55] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. 2016. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. *Advances in Neural Information Processing Systems* 29 (2016).
[56] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1235–1244.
[57] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal Representation Learning for Out-of-Distribution Recommendation. In *Proceedings of the ACM Web Conference 2022*. 3562–3571.
[58] 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 *WWW*. 878–887.
[59] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *SIGIR*. 1001–1010.
[60] Yuling Wang, Xiao Wang, Xiangzhou Huang, Yanhua Yu, Haoyang Li, Mengdi Zhang, Zirui Guo, and Wei Wu. 2023. Intent-aware recommendation via disentangled graph contrastive learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. 2343–2351.
[61] Chenwang Wu, Defu Lian, Yong Ge, Zhihao Zhu, Enhong Chen, and Senchao Yuan. 2021. Fight Fire with Fire: Towards Robust Recommender Systems via Adversarial Poisoning Training. In *SIGIR*. 1074–1083.
[62] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: learning the weight of feature interactions via attention networks. In *IJCAI*. 3119–3125.
[63] Sang Michael Xie, Ananya Kumar, Robbie Jones, Fereshte Khani, Tengyu Ma, and Percy Liang. 2020. In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. *arXiv preprint arXiv:2012.04550* (2020).
[64] Canran Xu and Ming Wu. 2020. Learning feature interactions with lorentzian factorization machine. In *AAAI*, Vol. 34. 6470–6477.
[65] Mingshi Yan, Zhiyong Cheng, Chen Gao, Jing Sun, Fan Liu, Fuming Sun, and Haojie Li. 2023. Cascading residual graph convolutional network for multi-behavior recommendation. *ACM Transactions on Information Systems (TOIS)* 42, 1 (2023), 1–26.
[66] Haotian Ye, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. 2021. Towards a theoretical framework of out-of-distribution generalization. *NeurIPS* 34 (2021).

[67] Feng Yu, Zhaocheng Liu, Qiang Liu, Haoli Zhang, Shu Wu, and Liang Wang. 2020. Deep interaction machine: A simple but effective model for high-order feature interactions. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2285–2288.

[68] Runlong Yu, Xiang Xu, Yuyang Ye, Qi Liu, and Enhong Chen. 2023. Cognitive Evolutionary Search to Select Feature Interactions for Click-Through Rate Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3151–3161.

[69] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.

[70] Hengyu Zhang, Junwei Pan, Dapeng Liu, Jie Jiang, and Xiu Li. 2024. Deep Pattern Network for Click-Through Rate Prediction. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 1189–1199. https://doi.org/10.1145/3626772.3657777

[71] Junqi Zhang, Yiqun Liu, Jiaxin Mao, Weizhi Ma, Jiazheng Xu, Shaoping Ma, and Qi Tian. 2023. User behavior simulation for search result re-ranking. *ACM Transactions on Information Systems (TOIS)* 41, 1 (2023), 1–35.

[72] Liang Zhang, Guannan Liu, Xiaohui Liu, and Junjie Wu. 2024. Denoising Item Graph with Disentangled Learning for Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[73] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. 2021. Deep Stable Learning for Out-Of-Distribution Generalization. In *CVPR*.

[74] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. *arXiv preprint arXiv:2105.06067* (2021).

[75] Yu Zheng, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2022. Disentangling long and short-term interests for recommendation. In *Proceedings of the ACM Web Conference 2022*. 2256–2267.

[76] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. 2021. DGCN: Diversified Recommendation with Graph Convolutional Networks. In *WWW 2021*.

[77] 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 *WWW 2021*. 2980–2991.

[78] 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 *AAAI*, Vol. 33. 5941–5948.

[79] 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 *KDD*. 1059–1068.

[80] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*.

[81] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Disentangled Self-Attentive Neural Networks for Click-Through Rate Prediction. *arXiv preprint arXiv:2101.03654* (2021).