Neural Computing and Applications
https://doi.org/10.1007/s00521-024-10330-y

ORIGINAL ARTICLE

A knowledge-enhanced interest segment division attention network for click-through rate prediction

Zhanghui Liu[1,2] · Shijie Chen[1,2] · Yuzhong Chen[1,2] · Jieyang Su[1,2] · Jiayuan Zhong[1,2] · Chen Dong[1,2]

Received: 20 October 2023 / Accepted: 29 July 2024

Abstract
Click-through rate (CTR) prediction aims to estimate the probability of a user clicking on a particular item, making it one of the core tasks in various recommendation platforms. In such systems, user behavior data are crucial for capturing user interests, which has garnered significant attention from both academia and industry, leading to the development of various user behavior modeling methods. However, existing models still face unresolved issues, as they fail to capture the complex diversity of user interests at the semantic level, refine user interests effectively, and uncover users' potential interests. To address these challenges, we propose a novel model called knowledge-enhanced Interest segment division attention network (KISDAN), which can effectively and comprehensively model user interests. Specifically, to leverage the semantic information within user behavior sequences, we employ the structure of a knowledge graph to divide user behavior sequence into multiple interest segments. To provide a comprehensive representation of user interests, we further categorize user interests into strong and weak interests. By leveraging both the knowledge graph and the item co-occurrence graph, we explore users' potential interests from two perspectives. This methodology allows KISDAN to better understand the diversity of user interests. Finally, we extensively evaluate KISDAN on three benchmark datasets, and the experimental results consistently demonstrate that the KISDAN model outperforms state-of-the-art models across various evaluation metrics, which validates the effectiveness and superiority of KISDAN.

Keywords Click-through rate prediction · Knowledge graph · Interest segment division · Contrastive learning

1 Introduction

With the rapid development of recommendation systems, click-through rate (CTR) prediction, which is capable of estimating the probability of a user clicking on a particular item, has emerged as a core module. CTR ranks items' click probabilities to generate a list of candidate items that users may be interested in. This approach helps users avoid direct exposure to a massive amount of information and assists them in finding valuable and relevant content. Moreover, CTR has become a crucial metric for business evaluation in various applications.

In recent years, user behavior sequence modeling has become popular for CTR prediction. Several models have been proposed [1–12], which have achieved promising

✉ Yuzhong Chen
yzchen@fzu.edu.cn

Zhanghui Liu
lzh@fzu.edu.cn

Shijie Chen
211027144@fzu.edu.cn

Jieyang Su
221027175@fzu.edu.cn

Jiayuan Zhong
221027220@fzu.edu.cn

Chen Dong
dongchen@fzu.edu.cn

[1] College of Computer and Data Science, Fuzhou University, Fuzhou 350108, Fujian Province, China

[2] Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou 350108, Fujian Province, China

Published online: 17 September 2024

performance in the research community and industry field. These models treat behavior sequences as fixed-length vectors that represent user interests, which are then fed into deep neural networks for CTR prediction. The key feature of these models is their effective utilization of attention mechanisms, which enable the aggregation of user behavior toward the target item. Although these models have greatly advanced the study of CTR prediction, there are still several major challenges that need to be addressed:

C1: How to accurately capture the complex diversity of user interests? Existing models usually employ attention mechanisms to calculate the relevance between the target item and each behavior in user behavior sequence, enabling the adaptive differentiation of contribution scores for each behavior concerning user interests. However, user behavior sequence typically consists of multiple interest segments, with each segment representing the user's preferences in a particular interest domain. Taking movie recommendation systems as an example, a user may be interested in specific actors, directors, or particular movie genres. These aspects, namely actors, directors, and movie genres, can be seen as user interest segments, each containing relevant movies. Therefore, existing models fail to segment user behavior sequence into multiple interest segments, limiting their capability of accurately modeling user preferences in different interest domains.

C2: How to effectively capture the different levels of user interests and the associations among them? Existing models usually represent the user behavior sequence with a single interest vector. However, user interests often have different levels, and there are associations and mutual influences among user interests. For example, a user may be an enthusiast of electronic products but occasionally browse through home decor products. An e-commerce platform should recommend not only the latest smartphones, computers, and related accessories but also some smart home products. Therefore, compressing user behavior sequence into a single user interest vector may not capture different levels of user interests.

C3: How to utilize items that users may be interested in but have not clicked on? Existing models overlook items that users may be interested in but have not clicked on. Typically, user behavior is limited to clicked or purchased items, while the unclicked items are often ignored. However, these unclicked items may represent potential interest domains for users. Ignoring these unclicked items means that the recommendation system cannot fully understand users' interest preferences, limiting the diversity and exploration capability of personalized recommendation systems.

Recently, some models have been proposed to alleviate these issues [4, 7, 8]. These models propose some rules to partition user behavior sequence into multiple parts so that they can extract interests at a finer granularity. DSIN [4] defines interactions occurring within a certain time interval as a session and observes that user behavior is highly homogeneous both within and across sessions. This suggests that users typically have a clear and unique intent within each session, and their interests may change drastically when they start a new session. Consequently, DSIN segments user behavior sequence into multiple parts based on sessions. TGIN [7] samples two additional items in different orders within the item–item graph for each item in the user behavior sequence, forming a triangular shape. These triangles are considered the fundamental units of user interests, leading TGIN to segment user behavior sequences into multiple triangles. RACP [8] assumes that a user's feedback is correlated with the browsed page and is influenced by surrounding items and the overall page context. Furthermore, user interest is considered a gradually converging process, where later interactions are more relevant to the final decision. Therefore, RACP segments user behavior sequence into multiple parts based on page numbers. However, these models solely rely on the interactions between users and items. Consequently, they fail to capture implicit semantics in user behavior sequences, such as a user's click on a movie possibly reflecting an interest in a certain movie director, which is critical for discovering a user's interests. In recommendation system scenarios, knowledge graphs not only contain items (such as movies and books) but also contain entities related to these items (such as directors, actors, and authors) and their semantic associations. Specifically, a knowledge graph is a graph composed of nodes and edges, where nodes represent entities or items, and edges represent semantic relationships between them. For example, in a movie recommendation scenario, a node might represent a movie, while other nodes might represent directors, actors, or genres related to that movie, with edges indicating the "directed by" relationship between the movie and the director, the "acted in" relationship between the movie and actors, and the "belongs to" relationship between the movie and its genres. Since knowledge graphs can provide rich auxiliary information to help discover user interests, they have gained increasing attention in the field of recommendation systems [11–27]. However, most knowledge graph-based models focus on learning representations of users and items based on the user–item interaction graph and the knowledge graph, rather than considering the items not clicked by a user, which may imply a user's potential interests. This motivates us to employ the knowledge graph to mine user interests in user behavior sequences.

To address the above challenges, we propose a knowledge-enhanced interest segment division attention network (KISDAN). Firstly, to accurately capture the complex

diversity of user interests (C1), KISDAN proposes an interest segment division method. KISDAN segments user interests by identifying entities in the knowledge graph that is adjacent to items within the user behavior sequence. Specifically, when an item in a user behavior sequence is connected to an entity in the knowledge graph, it is considered to belong to an interest segment related to that entity. In this way, a user behavior sequence can be divided into multiple interest segments, each reflecting the user's interest preferences in a specific domain. For example, a user behavior sequence includes multiple movies, through the knowledge graph, can be identified as belonging to different directors, actors, or genres. Thus, KISDAN can segment the user behavior sequence into multiple interest segments, such as "liking movies by a certain director", "preferring movies starring a certain actor", or "favoring a certain genre of movies". Additionally, an item may appear in multiple interest segments, representing different preferences of the user toward that item. In contrast to methods focusing solely on explicit patterns of user behavior, KISDAN utilizes the knowledge graph to explore the implicit semantics of user behavior sequences. This reveals the deep structure of user interests and uncovers the diverse domains and themes inherent in user interests.

Secondly, to effectively capture the different levels of user interests and the associations among them (C2), KISDAN categorizes interests into strong interests and weak interests. Strong interests are defined as the explicit interests that users demonstrate during interactions, such as frequently watching movies by a certain director. These interests usually feature in user behavior sequence and exhibit strong associations with specific entities in the knowledge graph. In contrast, weak interests typically reflect domains users occasionally explore, representing interests that have not yet been explicitly expressed through frequent interactions. KISDAN also introduces a strong-to-weak attention mechanism that leverages strong interests to extract weak interests while taking account of the relationships and interactions between user interests. This mechanism enables KISDAN to pay more attention to a user's strong interests while filtering out key information related to strong interests from weak interests.

Lastly, to effectively utilize items that users may be interested in but have not clicked on (C3), KISDAN extracts users' potential interests from two perspectives, i.e., the item co-occurrence graph and the knowledge graph. The knowledge graph facilitates the understanding of the complex structure of user interests by revealing the deep semantic relationships underlying user behaviors, while the item co-occurrence graph reveals potential connections between items by analyzing the similarity relationships among different items. Utilizing these two graphs, we model the relationship between items that users have clicked on and target items. By finding the shortest path between them, we can uncover implicit interests that the user may not have directly expressed. KISDAN also introduces a contrastive learning method to explore the complementary relationship between potential interests, which can help us better understand users' interests.

Compared to existing methods that extract user interests using attention mechanisms or GNN models, KISDAN distinguishes itself by its comprehensive utilization of both the knowledge graph and the item co-occurrence graph. Attention mechanisms mainly adjust the model's focus on different information by identifying key items in user behavior sequence, while GNN models focus on capturing the complex relationships between items as well as between items and entities through the graph structure. KISDAN not only delves into users' historical behavior patterns but also explores users' potential interests, which may not have been fully expressed. By combining the knowledge graph and item co-occurrence graph, KISDAN provides a more comprehensive method for modeling user interests. This approach enables KISDAN to reveal the diversity and complexity of user interests, thereby facilitating recommendation systems in exploring more personalized recommendation strategies. The main contributions of this paper are summarized as follows:

1. We effectively explore the construction of interest segments in user behavior sequence and propose an interest segment division method based on the connections between entities and items in the knowledge graph. This method captures user preferences in different interest domains, revealing the diversity of user interests. Furthermore, we identify strong interests from interest segments that contain more items than a given threshold and extract weak interests from the remaining interest segments. Strong interests represent the core interests of users, while weak interests reflect their broader interests. This refines user interests and captures interactions between interests.
2. We propose a method for constructing potential interest segment sequences based on the semantic associations between items and entities in the knowledge graph, as well as the similarity between items in the item co-occurrence graph. From these sequences, we extract both semantic-based potential interests and similarity-based potential interests. This method introduces items that users may be interested in but have not clicked on, thus enhancing the exploration capability of the recommendation system. Additionally, we design a novel contrastive learning method that takes two different types of potential interests as input and learns their complementary relationships, providing a more

comprehensive understanding of a user's potential interests from different perspectives.

3. We conduct extensive experiments on three public datasets. The experimental results demonstrate that KISDAN outperforms all baseline models. In addition, we conduct ablation experiments to validate the effectiveness of different components of KISDAN. Finally, we conduct parametric experiments to evaluate the impact of some hyperparameters on the overall performance of KISDAN.

2 Related work

2.1 Click-through rate prediction

One critical issue of click-through Rate (CTR) prediction is the high sparsity of input features. Directly using raw features often yields suboptimal results. Moreover, conventional linear models only consider each feature independently without modeling the interactions between features. To explore feature interactions, some models combine sparse features to generate new dense feature representations and capture nonlinear relationships between input features. FM [28] utilizes low-dimensional vectors to represent each feature field and learns second-order feature interactions through inner product operation, leading to significant improvements over linear models. With the development of deep learning, deep neural networks have been able to model complex feature interactions. Models based on deep neural networks, such as DeepFM [29] and xDeepFM [30], have been widely adopted in industrial recommendation systems.

Since user's historical behaviors contain rich information for inferring user preferences, some models attempt to mine user interests from users' historical behaviors, in addition to learning feature interactions. DIN [1] is the first model that introduces attention mechanisms in user behavior modeling. It adaptively calculates embeddings of user interest by evaluating the relevance between candidate items and users' historical behaviors. Subsequently, GIN [2] builds an item co-occurrence graph to mine user intention through multi-layer graph propagation. DIEN [3] introduces GRU networks and auxiliary losses to mine the dependencies between user behaviors. DIEN also proposes an AUGRU structure to learn the evolution of user interests. DSIN [4] segments a user's behavior sequence into multiple sessions based on interaction time and then tries to capture the relationships within and between sessions. MIMN [5] utilizes a neural turing machine to compress a user's long-term behaviors. Then, it uses a joint online-offline design optimization scheme to model long-term user behavior sequences. DMIN [6] employs the multi-head self-attention to capture representations of users' historical behaviors and utilizes another multi-head self-attention to transform each behavior in the sequence into multiple heads. Furthermore, attention units are employed to capture the relevance of each head's output with respect to the target item, enabling the extraction of multiple interests of the user. TGIN [7] introduces triangle structures in the item co-occurrence graph for each clicked item in a user behavior sequence. This model treats these triangles as basic units of user interests and reflects user interests at different levels through multi-order triangles. RACP [8] captures specific user preferences by incorporating contextual information within a page and interest variations between pages. SAM [9] computes the user interest vector at each step and feeds it into a GRU to obtain the memory vector for the next iteration. It proposes a point-wise dual-query attention mechanism and applies it to each user behavior. This attention mechanism treats the target item and the memory vector as dual queries to learn the importance of each behavior. DBPMaN [10] constructs the behavior paths, matching a user's current path with their historical paths to depict the dependency relationships before and after user path decisions. Although these models have made significant improvements, they still do not fully leverage the semantic information latent in user behavior sequences and rely solely on historical behaviors to summarize user interests, thus lacking the ability to explore users' potential interests.

2.2 Knowledge-aware recommendation

As knowledge graphs contain rich auxiliary information, some research works have started incorporating knowledge graphs into recommendation systems. Existing knowledge-enhanced recommendation models can generally be divided into three categories: embedding-based models, path-based models, and graph-based models. Embedding-based models utilize the relationships and entities in knowledge graphs to enhance the semantic representations of items and users. CKE [13] designs three components to extract semantic features from the knowledge graph's structural content, textual content, and visual content of items, respectively. DKN [14] enriches the information in news content by associating each word with relevant entities in the knowledge graph. It designs a knowledge-aware convolutional neural network that integrates word-level and knowledge-level representations of news, resulting in knowledge-aware embeddings for each news article. However, these models cannot fully utilize the higher-order connectivity information in knowledge graphs, thereby limiting their ability to explore the complex relationships among entities.

Path-based models explore various connection patterns among items in the knowledge graph to provide additional guidance for recommendations. PER [15] and MCRec [16] generate effective meta-paths and learn representations of users and items along different types of relation paths. KPRN [17] automatically extracts paths between users and items from the knowledge graph, where each path consists of relevant entities and relationships. Then, a LSTM network is employed to model the sequential dependencies among entities and relationships. Path-based models mainly involve the design of meta-paths to generate meaningful connection patterns. However, the construction of meta-paths can be a challenging and time-consuming task, as it requires domain experts to manually define and validate the relevant paths.

In recent years, graph neural networks (GNNs) have shown great potential in learning high-order node information through information propagation between adjacent nodes. RippleNet [18] utilizes GNNs to propagate users' latent preferences and explore users' hierarchical interests on the knowledge graph. KGCN [19] obtains item embeddings by iteratively aggregating neighborhood information of items in the knowledge graph. It can obtain the high-order dependency information among items through graph convolutions. KGAT [20] merges the user-item graph with the knowledge graph into a unified heterogeneous graph and updates the embeddings of nodes based on the embeddings of their neighboring nodes, recursively propagating embeddings. It utilizes attention mechanisms to learn the weights of each neighbor during the propagation process, where the cascaded attention weights can reveal the importance of higher-order connections. CKAN [21] integrates the explicitly encoded collaborative signals from user–item interactions and the auxiliary knowledge from the knowledge graph. By leveraging these two critical pieces of information, it effectively represents the latent semantics of users and items in the vector space. KGIN [22] introduces intent nodes and utilizes the knowledge graph to explore the user intents behind user–item interactions. It proposes a graph propagation mechanism that is aware of relation paths, distinguishing different knowledge graph relations, and emphasizing the contributions of different knowledge graph relations to node embeddings. This mechanism improves the performance and interpretability of recommendations. CG-KGR [23] encapsulates historical interactions as interactive information summaries. It then utilizes these summaries as guidance to aggregate the information from interactive data and the knowledge graph through graph convolutions. By employing different learning strategies, CG-KGR masks irrelevant information from the knowledge graph, resulting in more accurate personalized recommendations. KGIC [24] constructs both local and non-local graphs. The local graph comprises the first-order components from the user–item interaction graph and knowledge graph. The non-local graph includes the higher-order components from both graphs, facilitating intragraph and intergraph contrastive learning between the local and non-local graphs. MCCLK [25] considers three different graph views, including a global structural view, a local collaborative view, and a semantic view. It performs contrastive learning on these three views at both the local and global level to mine comprehensive graph structure information. HAKG [26] embeds users and items, as well as entities and relations into a hyperbolic space. It designs a hyperbolic aggregation scheme to gather related context on the knowledge graph and introduces an angular constraint to preserve item characteristics in the embedding space. DCLKR [27] disentangles the knowledge graph and user–item interaction graph into multiple aspects. It then performs intraview contrastive learning to learn the differences between representations in these two views and applies interview contrastive learning to transfer knowledge between these two views. Although these graph-based models utilize GNNs to aggregate neighborhood information between the target user and target item, they lack the ability to capture the mutual influence between user behavior and target item during the information aggregation process.

Recent research has begun to combine knowledge graphs with user behavior modeling. ATBRG [11] explores multi-layer neighbors on the knowledge graph for each item involved in the user behavior sequence as well as the target item, constructing an adaptive target-behavior relational graph. It employs a relationship-aware attention mechanism to aggregate structural knowledge of each user behavior and target item on the relational graph, effectively representing the structural relationship between a given target user and target item. MTBRN [12] constructs multiple paths between user behavior and target item on the knowledge graph and item–item similarity graph. It captures multiple relationships through a graph search algorithm and encodes each path using Bi-LSTMs. Finally, MTBRN utilizes an attention mechanism to aggregate different path representations into the final representation that reveals user preferences for target item from different perspectives. However, these models only enrich the representation of user behavior using auxiliary information from the knowledge graph. They still face the challenge of effectively capturing user's potential interests in items that they have not interacted with from the user's historical behavior.

2.3 Contrastive learning

Contrastive learning [31], as a self-supervised learning method, has been proven effective in computer vision [32] and natural language processing [33]. Contrastive learning constructs positive and negative samples without manual annotation and then uses these samples as the supervisory signal for representation learning. There is also a growing trend of introducing contrastive learning into the field of recommendation systems.

For instance, some methods have proposed new data augmentation techniques for contrastive learning. CL4Srec [34] proposes three data augmentation methods (i.e., cropping, masking, and reordering). It randomly applies two of these three strategies to obtain two augmented views of a user behavior sequence. Then, CL4Srec maximizes the consistency between the two augmented views derived from the same user behavior sequence through contrastive learning. SGL [35] introduces three data augmentation methods: node dropout, edge dropout, and random walk. These methods are applied to the user–item interaction graph, resulting in different augmented views. Subsequently, GCN is employed to extract representations of individual nodes within these augmented views. Furthermore, a contrastive learning loss is utilized to maximize the consistency between representations of the same node in different views.

In addition, some studies have designed more complex contrastive learning methods based on user behaviors. CLCRec [36] uses contrastive learning to address the cold-start problem. GDCL [37] proposes a diffusion-based graph contrastive learning method to learn implicit feedback from users. CLSR [38] generates self-supervisory signals from long-term and short-term interests. GCL4SR [39] constructs a weighted item transfer graph (WITG) from all user interaction sequences. By sampling its neighborhoods, two augmented views of a user behavior sequence are obtained. It proposes two auxiliary tasks. One task aims to maximize the consistency between the two augmented views, while the other aims to minimize the discrepancy between the two augmented views and the user behavior sequence. MISS [40] transforms user behavior sequence into a matrix, where the horizontal direction represents sequences of items with the same features, while the vertical direction represents different features of the same item. Different sizes of convolutional kernels are employed to compute representations at both the interest level and feature level through horizontal and vertical convolutions, respectively. From the representations computed with the same convolutional kernel, a pair of representations is randomly selected to serve as two distinct augmented views corresponding to the same interests. CCL4Rec [41] designs a difficulty-aware data augmentation method. This data augmentation method treats the items that users did not click on as substitutes and the items within the user behavior sequence as replaced items and calculates the importance between replaced items and the relevance between substitutes. Then, CCL4Rec replaces important items with highly correlated substitutes to generate negative samples, while replacing unimportant items with unrelated substitutes to generate positive samples. CL4CTR [42] introduces three self-supervised learning signals: contrastive loss, feature alignment, and field uniformity. It constructs positive feature pairs through data augmentation and minimizes the distance between representations of each positive feature pair through contrastive loss. The feature alignment constraint brings feature representations from the same domain close, while the field uniformity constraint pushes apart feature representations from different domains. These self-supervised learning signals enable CL4CTR to generate high-quality feature representations.

Although these contrastive learning-based models have achieved significant improvement, they mainly focus on generating augmented user behavior sequences for learning self-supervised signals. However, user behavior sequences only reflect the user–item interaction information and cannot provide the semantic information that elucidates the reason why user clicks on items. Therefore, these models have limitations in their ability to learn high-quality user interest vectors.

3 Preliminaries

In recommendation scenarios, such as e-commerce platforms and news platforms, we typically have a series of historical interaction records between users and items, such as purchase and click behaviors. Let $\mathcal{U}$ represents a set of users, $\mathcal{I}$ represents a set of items, and we denote the interaction records as $\mathcal{H} = \{(u, v, \mathcal{B}_u, y) \mid u \in \mathcal{U}, v \in \mathcal{V}\}$. Here, $\mathcal{B}_u \in I$ represents historical behavior (i.e., item lists) for user u, and $y \in \{0, 1\}$ represents the implicit feedback of user u toward item v when item v is recommended to the user u. If user u interacts with item v, then $y = 1$; otherwise, $y = 0$. To effectively incorporate auxiliary information about items (i.e., item attributes and external knowledge) into recommendation, we define the recommendation task on the knowledge graph and item co-occurrence graph as follows:

Definition 1 (Knowledge Graph). The knowledge graph describes the semantic relationships between items and real-world entities and can be represented as $\mathcal{G}_{\text{kg}} = \{(h, r, t) \mid h, t \in \mathcal{V}_{\text{kg}}, r \in \mathcal{R}\}$, where $\mathcal{V}_{\text{kg}}$ is the set of

items and entities in the knowledge graph, and $\mathcal{R}$ is the set of relations between entities, and each triple (h, r, t) represents there is a relation r between the head entity h and the tail entity t.

For example, the triple $(J.K.Rowling, Authorof, Harry\ Potter)$ represents that J.K.Rowling is the author of the novel *Harry Potter*.

Definition 2 (Item Co-occurrence Graph). The item co-occurrence graph is an undirected graph denoted as $\mathcal{G}_{\mathrm{cf}} = (\mathcal{V}_{\mathrm{cf}}, E_{\mathrm{cf}})$, where $\mathcal{V}_{\mathrm{cf}}$ is the set of nodes and E_{cf} is the set of edges. Each node $v \in V_{\mathrm{cf}}$ represents an item. The graph is constructed based on the user behavior sequence of all users in the user set U. If there exists at least one user $u \in U$, whose user behavior sequence B_{u} contains both items v_k and v_j, then an edge is established between these two items, i.e., $(v_k, v_j) \in E_{\mathrm{cf}}$.

According to the definition of interaction records, knowledge graph, and item co-occurrence graph, the recommendation task is to predict the probability $\hat{y}_{u,i}$ of user u clicking on item v given the historical interaction record $\mathcal{H}$, as well as the knowledge graph $\mathcal{G}_{\mathrm{kg}}$ and item co-occurrence graph $\mathcal{G}_{\mathrm{cf}}$.

4 Model

4.1 Model overview

In this section, we introduce the KISDAN model, which aims to leverage the knowledge graph and item co-occurrence graph to improve recommendation. The overall framework of KISDAN is shown in Fig. 1. KISDAN consists of four layers: The interest segment division layer, the feature embedding layer, the interest refinement layer, and the potential interest contrastive learning layer.

The interest segment division layer divides the user behavior sequence into multiple user interest segments based on the knowledge graph and item co-occurrence graph. These interest segments semantically reflect the users' diverse interests, enabling the layer to accurately capture the complex diversity of user interests.

The feature embedding layer reduces data dimensionality and learns feature representations by transforming high-dimensional sparse features into low-dimensional dense vectors.

The interest refinement layer refines user interests into four categories: strong interests, weak interests, semantic-based potential interests, and similarity-based potential interests. This layer learns the interaction relationships between strong interests and weak interests, emphasizing the dominant role of strong interests and the complementary role of weak interests, which ultimately improves the accuracy of interest extraction.

The potential interest contrastive learning layer employs contrastive learning to maximize the consistency between two types of potential interests for the same user. By leveraging this layer, KISDAN effectively learns the complementary relationships between these two types of potential interests, thereby enhancing its ability to explore the diversity of user interests. In the following subsections, we provide a detailed description of these four layers.

4.2 Interest segment division layer

To accurately capture the complex diversity of user interests, previous works mainly adopt two strategies. The first strategy uses the target item as the query for the attention mechanism to extract important user behaviors. However, this strategy neglects modeling behaviors belonging to the same interest domain. The second strategy segments the user behavior sequence based on rules such as interaction time [4] or page numbers [8] to simulate different interest domains of users. However, this strategy only considers user behaviors within a certain time window as belonging to the same interest, lacking semantic guidance for extracting interests from the user behavior sequence. Therefore, these strategies can only achieve suboptimal recommendation performance.

Relying solely on user behavior sequences may not unearth implicit information, whereas knowledge graphs and item co-occurrence graphs offer semantic information between items and entities, as well as similarity information between items. Semantic information guides KISDAN in understanding why users click on specific items, while similarity information recommends items similar to those clicked by users. Therefore, KISDAN introduces an interest segment division algorithm, which searches the knowledge graph and the item co-occurrence graph to divide the user behavior sequence into fine-grained interest segments: strong interest segments, weak interest segments, and potential interest segments. These interest segments accurately reflect users' behavior patterns and preferences in various interest domains. Thus, KISDAN can extract user interest characteristics in more detail. The interest segment division algorithm is given in Algorithm 1.

Lines 1–7: Firstly, in order to divide a user's behavior sequence into multiple interest segments, we traverse each clicked item b in a user's behavior sequence $\mathcal{B}_{\mathrm{u}}$. Then, for each item b, we further traverse each entity o adjacent to item b in the knowledge graph $\mathcal{G}_{\mathrm{kg}}$, and then obtain an item set o_{adj} containing all items that are adjacent to entity o in the knowledge graph $\mathcal{G}_{\mathrm{kg}}$. We define the item set o_{adj} as an

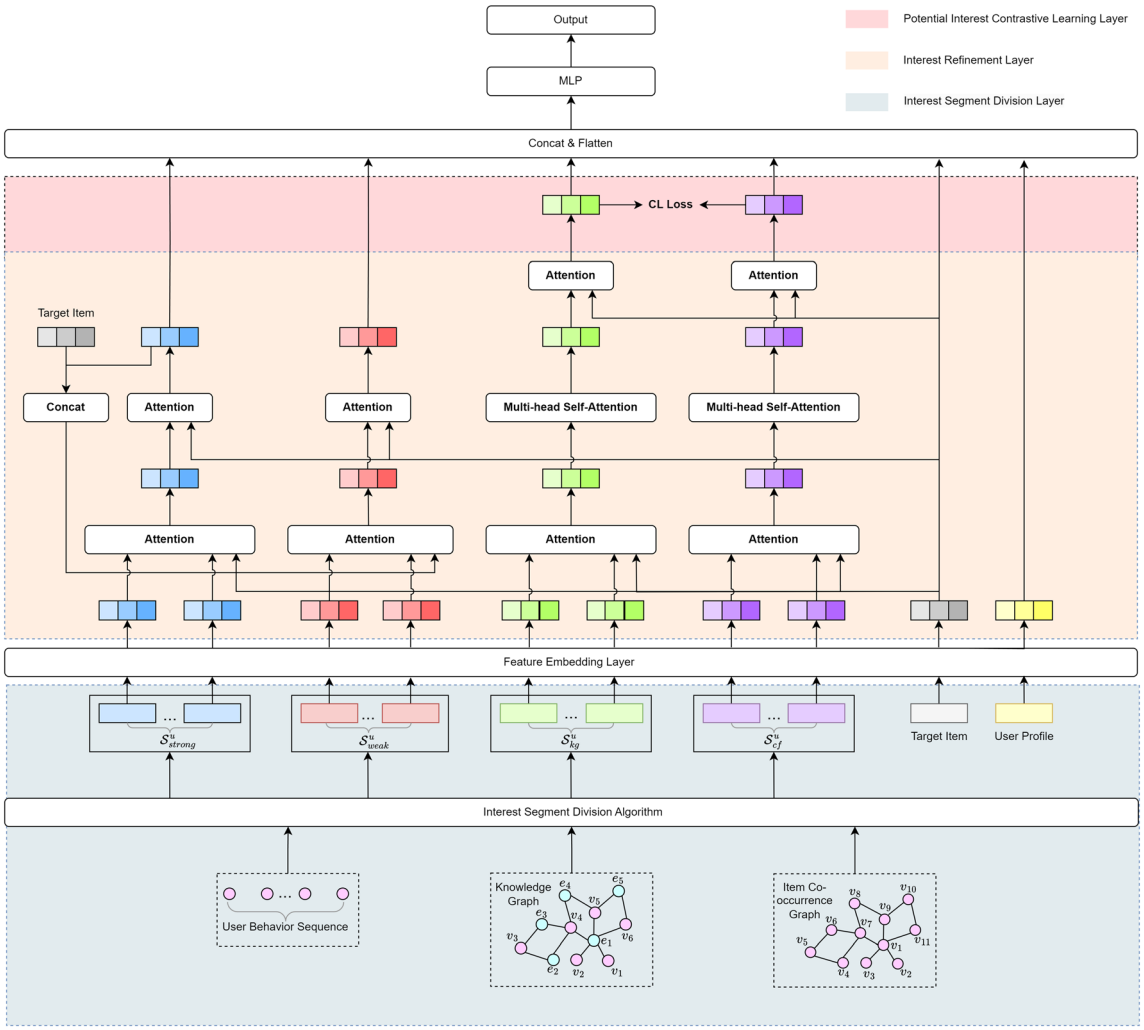

Fig. 1 Framework of the proposed KISDAN model

interest segment and consider the items within the same interest segment as belonging to the same interest. Finally, KISDAN obtains an interest segment sequence $\mathcal{S}^u_{\text{seg}}$ for a user.

Lines 8–15 : We categorize interest segments into strong interest segments and weak interest segments, where strong interest segments represent domains in which the user frequently exhibits preferences, while weak interest segments indicate less frequent user interest. We traverse through each interest segment s in the interest segment sequence $\mathcal{S}^u_{\text{seg}}$. For each interest segment s, if the number of items in it exceeds the given threshold τ, it will be categorized as a strong interest segment. Otherwise, it will be categorized as a weak interest segment. Subsequently, the strong interest segments and the weak interest segments are added to the strong interest segment sequence $\mathcal{S}^u_{\text{strong}}$ and the weak interest segment sequence $\mathcal{S}^u_{\text{weak}}$, respectively.

Lines 16–22: Existing models have a notable limitation in modeling user interest because they ignore items that users may be interested in but have not yet clicked on. These items may reflect user interests that have not been explicitly expressed. Therefore, the recommendation diversity and exploration ability of existing models is limited. To address this problem, KISDAN intends to obtain a user's potential interests from two perspectives: the knowledge graph $\mathcal{G}_{\text{kg}}$ and the item co-occurrence graph $\mathcal{G}_{\text{cf}}$. For each item b in the user behavior sequence $\mathcal{B}_{\text{u}}$, to explore its association with the target item v, we search for the shortest path between it and the target item v in the knowledge graph $\mathcal{G}_{\text{kg}}$. There are direct semantic associations between items adjacent to the same entity in the knowledge graph. Furthermore, the shortest path between the item clicked by the user and the target item can reveal implicit semantic associations. Thus, KISDAN considers all items within the shortest path to form a semantic-based potential interest segment and adds this potential interest

segment to the semantic-based potential interest segment sequence $\mathcal{S}^u_{\text{kg}}$. Similarly, as the item co-occurrence graph contains similarity relationships between items, KISDAN performs the same operation on $\mathcal{G}_{\text{cf}}$ to obtain the similarity-based potential interest segment sequence $\mathcal{S}^u_{\text{cf}}$. Based on these two potential interest segment sequences, KISDAN can utilize a series of unclicked items to discover user's potential interests.

After interest segment division, KISDAN can obtain various types of interest segment sequences, which enables a more comprehensive understanding of users' interest preferences.

Algorithm 1 Interest segment division algorithm

Input: Target item v; User behavior sequence $\mathcal{B}_u$; Knowledge graph $\mathcal{G}_{kg}$; Item co-occurrence graph $\mathcal{G}_{cf}$; threshold τ

Output: Strong interest segment sequence $\mathcal{S}^u_{strong}$; weak interest segment sequence $\mathcal{S}^u_{weak}$; Semantic-based potential interest segment sequence $\mathcal{S}^u_{kg}$; Similarity-based potential interest segment sequence $\mathcal{S}^u_{kg}$

Initialize $\mathcal{S}^u_{seg}$ as an empty list
for each $b \in \mathcal{B}_u$ **do**
  for each $o \in adjacent(\mathcal{G}_{kg}, b)$ **do**
    $o_{adj} \leftarrow adjacent(\mathcal{G}_{kg}, o)$
    $\mathcal{S}^u_{seg}.append(o_{adj})$
  end for
end for
Initialize $\mathcal{S}^u_{strong}$ and $\mathcal{S}^u_{strong}$ as an empty list
for each $s \in \mathcal{S}^u_{seg}$ **do**
  if $s.size > \tau$ **then**
    $\mathcal{S}^u_{strong}.append(s)$
  else
    $\mathcal{S}^u_{weak}.append(s)$
  end if
end for
Initialize $\mathcal{S}^u_{kg}$ and $\mathcal{S}^u_{cf}$ as an empty list
for each $b \in \mathcal{B}_u$ **do**
  $seg^u_{kg} \leftarrow$ search for the shortest path between b and v in $\mathcal{G}_{kg}$
  $seg^u_{cf} \leftarrow$ search for the shortest path between b and v in $\mathcal{G}_{cf}$
  $\mathcal{S}^u_{kg}.append(seg^u_{kg})$
  $\mathcal{S}^u_{cf}.append(seg^u_{cf})$
end for
return $\mathcal{S}^u_{strong}, \mathcal{S}^u_{weak}, \mathcal{S}^u_{kg}, \mathcal{S}^u_{kg}$

4.3 Feature embedding layer

In the real-world recommendation scenarios, both users and items have multiple types of features. KISDAN encodes the following types of features, namely the user profile, the strong interest segment sequence $\mathcal{S}^u_{\text{strong}}$, the weak interest segment sequence $\mathcal{S}^u_{\text{weak}}$, the semantic-based potential interest segment sequence $\mathcal{S}^u_{\text{kg}}$, the similarity-based potential interest segment sequence $\mathcal{S}^u_{\text{cf}}$, and the target item, into low-dimensional dense vectors, which are represented as $e^u \in \mathbb{R}^d, e^s \in \mathbb{R}^{L_s \times N_s \times d}, e^w \in \mathbb{R}^{L_w \times N_w \times d}, e^k \in \mathbb{R}^{L_k \times N_k \times d}$, $e^c \in \mathbb{R}^{L_c \times N_c \times d}$ and $e^t \in \mathbb{R}^d$, respectively. Here

L_s, L_w, L_k, and L_c respectively, represent the maximum number of interest segments in the strong interest segment sequence $\mathcal{S}^u_{\text{strong}}$, weak interest segment sequence $\mathcal{S}^u_{\text{weak}}$, semantic-based potential interest segment sequence $\mathcal{S}^u_{\text{kg}}$, and similarity-based potential interest segment sequence $\mathcal{S}^u_{\text{cf}}$. $N_{\text{s}}, N_{\text{w}}, N_k$, and N_c represent the maximum number of items in the strong interest segment, weak interest segment, semantic-based potential interest segment, and similarity-based potential interest segment, respectively. d is the embedding size.

4.4 Interest refinement layer

The core of a recommendation system is to provide personalized recommendations to users. Each user behavior (e.g., clicks, purchases, comments, etc.) directly or indirectly reflects how much a user prefers a certain item. Recommendation systems rely on user behaviors to infer their interests on items. Therefore, KISDAN utilizes an interest refinement layer to extract user interests from user behaviors. Compared to existing models, KISDAN refines user interests and categorizes them into strong interests, weak interests, and potential interests, thereby enhancing the accuracy of interest extraction.

Each item in the strong interest segment has its unique contribution and importance relative to the target item. KISDAN incorporates a two-layer attention mechanism to accurately estimate the relationship between the item in the strong interest segment and the target item. Specifically, KISDAN treats the target item as the query to estimate the importance of each item within the strong interest segment to get the intra-aggregated representation of each strong interest segment. The intra-aggregated representation x^s_i for the i-th strong interest segment is calculated as follows:

$$\alpha^s_{i,j} = \frac{\exp\left(e^s_{i,j} W_1 e^t\right)}{\sum_{l=1}^{N_s} \exp\left(e^s_{i,l} W_1 e^t\right)} \tag{1}$$

$$x^s_i = \sum_{j=1}^{N_s} \alpha^s_{i,j} e^s_{i,j} \tag{2}$$

where $\alpha^s_{i,j}$ is the attention weight, $e^s_{i,j}$ is the representation of the j-th item in the i-th strong interest segment, and $W_1 \in \mathbb{R}^{d\times d}$ represents the learnable parameters.

KISDAN further uses an attention mechanism to distinguish the influence of different strong interest segments on the target item. In this way, KISDAN aggregates various intra-aggregated representations of strong interest segments, and the representation of the strong interest u^s can be calculated as follows:

$$\beta^s_i = \frac{\exp\left(x^s_i W_2 e^t\right)}{\sum_{j=1}^{L_s} \exp\left(x^s_j W_2 e^t\right)} \tag{3}$$

$$u^s = \sum_{i=1}^{L_s} \beta^s_i x^s_i \tag{4}$$

where β^s_i is the attention weight, and $W_2 \in \mathbb{R}^{d\times d}$ represents the learnable parameters.

Existing models usually use the target item as the query to estimate the importance of various items within the user behavior sequence and capture the user's overall interest. However, this approach ignores the fact that user interests are divided into different levels, and interactions occur between these different levels of interests. Therefore, KISDAN divides user interests into strong interests and weak interests. Strong interests predominate in both categories, while weak interests provide supplementary information and may also contain some noise. Consequently, KISDAN proposes a strong-to-weak attention mechanism. This attention mechanism concatenates the strong interest u^s with the target item e^t to create a joint query. This joint query encapsulates the information from both the strong interest and target item, allowing for the effective utilization of the strong interests to extract information related to the weak interests and reducing the contribution score of noise information within the weak interests. Using this strong-to-weak attention mechanism, KISDAN can estimate the relevance between each weak interest segment and the strong interest, thereby obtaining the intra-aggregated representation for each weak interest segment. The intra-aggregated representation x^w_i for the i-th weak interest segment is calculated as follows:

$$\alpha^w_{i,j} = \frac{\exp\left(e^w_{i,j} W_3 \text{Concat}(u^s, e^t)\right)}{\sum_{l=1}^{N_w} \exp\left(e^w_{i,l} W_3 \text{Concat}(u^s, e^t)\right)} \tag{5}$$

$$x^w_i = \sum_{j=1}^{N_w} \alpha^w_{i,j} e^w_{i,j} \tag{6}$$

where $\alpha^w_{i,j}$ is the attention weight, $e^w_{i,j}$ is the representation of the j-th item in the i-th weak interest segment, $W_3 \in \mathbb{R}^{d\times d}$ represents the learnable parameters, and Concat is the concatenation operation.

After obtaining the intra-aggregated representation of each weak interest segment, an attention mechanism further uses the target item e^t as the query to distinguish the influence of different weak interest segments on the target item. In this way, KISDAN aggregates different intra-aggregated representation of weak interest segments and obtains the representation u^w of the weak interests:

$$\beta_i^w = \frac{\exp\left(x_i^w W_4 e^t\right)}{\sum_{j=1}^{L_w} \exp\left(x_j^w W_4 e^t\right)} \tag{7}$$

$$u^w = \sum_{i=1}^{L_w} \beta_i^w x_i^w \tag{8}$$

where β_i^w is the attention weight, and $W_4 \in \mathbb{R}^{d\times d}$ represents the learnable parameters.

Previous models only consider items in the user behavior sequence. However, items that the user with may still reflect the user's potential interests. For instance, if a user clicks on an item in a certain category, it is plausible that other items in that same category, even if unclicked by the user, may also match their interests. To better capture the user's potential interests, KISDAN introduces the item co-occurrence graph and the knowledge graph. These two graphs implicitly contain similarity information and semantic information between items, respectively. In other words, nodes in these two graphs that are adjacent to those items with which the user has interacted can reflect the user's potential interests. Therefore, KISDAN uses the semantic-based potential interest segment sequence S_{kg}^u and the similarity-based potential interest segment sequence S_{cf}^u extracted from these two graphs as a bridge to learn the user's potential interests.

First, KISDAN uses the target item e^t as the query to compute the relevance scores for each item within a semantic-based potential interest segment. Thus, the intra-aggregated representation x_i^k for the i-th semantic-based potential interest segment is calculated as follows:

$$\alpha_{i,j}^k = \frac{\exp\left(e_{i,j}^k W_5 e^t\right)}{\sum_{l=1}^{N_k} \exp\left(e_{i,l}^k W_5 e^t\right)} \tag{9}$$

$$x_i^k = \sum_{j=1}^{N_k} \alpha_{i,j}^k e_{i,j}^k \tag{10}$$

where $\alpha_{i,j}^k$ is the attention weight, $e_{i,j}^k$ is the representation of the j-th item in the i-th semantic-based potential interest segment, and $W_5 \in \mathbb{R}^{d\times d}$ represents the learnable parameters.

In the semantic-based potential interest segment sequence S_{kg}^u, interest segments include items that users have not clicked on and may contain some noise. Then, KISDAN utilizes multi-head self-attention to capture the interactions between different potential interest segments and reduce the noise. The input of the self-attention module includes three parts: query, key, and value, all of which are identical. Multi-head self-attention is an attention mechanism that performs multiple attention functions in parallel and can learn relationships in different representation subspaces [43]. Specifically, KISDAN denotes the output of h-th attention function as head_h and calculate it as follows:

$$x^k = \text{Concat}\left(x_1^k, x_2^k, \ldots, x_{L_k}^k\right) \tag{11}$$

$$\begin{aligned}\text{head}_h &= \text{Attention}\left(x^k W_h^Q, x^k W_h^K, x^k W_h^V\right) \\ &= \text{Softmax}\left(\frac{x^k W_h^Q \left(x^k W_h^K\right)^T}{\sqrt{d_h}} x^k W_h^V\right)\end{aligned} \tag{12}$$

where $W_h^Q, W_h^K, W_h^V \in \mathbb{R}^{d\times d_h}$ are weight matrices for the query, key, and value of the h-th attention function, respectively, d_h is the dimension of each head, and $h \in [1, n_{\text{head}}]$.

Then, the output vectors from n_{head} attention functions are concatenated to generate the refined intra-aggregated representation H^k for the semantic-based potential interest segment sequence $\mathcal{S}_{\text{kg}}$, which is defined as follows:

$$\begin{aligned}H^k &= \text{MultiHead}\left(x^k\right) \\ &= \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_{n_{\text{head}}}) W_6\end{aligned} \tag{13}$$

where n_{head} is the number of heads or parallel attention functions, and $W_6 \in \mathbb{R}^{d\times d}$ is the learnable parameter.

Finally, KISDAN uses the target item e^t as the query to estimate the importance of each semantic-based potential interest segment and obtains the representation u^k of the semantic-based potential interests as follows:

$$\beta_i^k = \frac{\exp\left(H_i^k W_7 e^t\right)}{\sum_{j=1}^{L_k} \exp\left(H_j^k W_7 e^t\right)} \tag{14}$$

$$u^k = \sum_{i=1}^{L_k} \beta_i^k x_i^k \tag{15}$$

where H_i^k is the refined intra-aggregated representation for the i-th semantic-based potential interest segment, β_i^k is the attention weight, and $W_7 \in \mathbb{R}^{d\times d}$ represents the learnable parameters.

Similarly, KISDAN applies the same mechanism to obtain the representation u^c of the similarity-based potential interests.

In this layer, KISDAN refines user interests by modeling strong interests, weak interests, and potential interests, allowing it to learn user behaviors from different perspectives comprehensively.

4.5 Potential interest contrastive learning layer

The item co-occurrence graph and the knowledge graph contain different implicit information. Therefore, the similarity-based potential interests and the semantic-based potential interests extracted from these two graphs may exhibit differences. Contrastive learning is a representation

learning method that can learn high-quality representations by comparing the similarities and differences between positive and negative samples. To identify the complementary information between these two potential interest segment sequences, KISDAN employs contrastive learning to compare the similarities and differences between them. Traditional contrastive learning methods often generate positive samples and negative samples by random masking, replacing, and inserting. However, samples produced in this way have limited diversity in terms of the information they contain. In contrast, KISDAN can generate high-quality samples from two types of potential interests and has a certain level of interpretability. KISDAN treats these two potential interests of the same user as positive samples, while those from different users are considered as negative samples. The contrastive loss $\mathcal{L}_{\text{cl}}$ is defined as follows:

$$\mathcal{L}_{\text{cl}} = -\sum_{i=1}^{\text{N}_\text{u}} \log \frac{\exp\left(\left(u_i^c\right)^T u_i^k\right)}{\exp\left(\left(u_i^c\right)^T u_i^k\right) + \sum_{j=1}^{N_\text{u}} \exp\left(\left(u_i^c\right)^T \tilde{u}_j^k\right)} \tag{16}$$

where N_u is the number of users, u_i^c represents the similarity-based potential interests for the i-th user, u_i^k represents the semantic-based potential interests for the i-th user, and $\tilde{u}_i^k$ denotes the negative sample randomly sampled from the different user's potential interests segment sequences within one mini-batch.

4.6 Model training

We use a multiple layer perceptron (MLP) to achieve better feature interactions and obtain the predicted click-through rate:

$$f(x_i) = \sigma\left(\text{MLP}\left(\text{Concat}\left(e^u, e^t, u^s, u^w, u^k, u^c\right)\right)\right) \tag{17}$$

where σ is the sigmoid function.

Since the CTR prediction task is a binary classification task, the chosen loss function is cross-entropy loss, typically defined as:

$$\mathcal{L}_{\text{target}} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(f(x_i)) + (1 - y_i)\log(1 - f(x_i)) \tag{18}$$

where N is the size of the training set, $y_i \in \{0, 1\}$ is the click label, and $f(x)$ is the predicted output of our network. As we use contrastive loss to capture complementary information between potential interest segment sequences, the overall loss can be defined as:

$$\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{target}} + \lambda \mathcal{L}_{\text{cl}} \tag{19}$$

where λ is a hyperparameter that balances the two subtasks.

5 Experiments

In this section, we first introduce the benchmark datasets and the experimental settings. Then, we conduct extensive experiments to address the following research questions:

- *RQ1 :* How does KISDAN perform compared to baseline models?
- *RQ2 :* How do the main components of KISDAN affect the performance of KISDAN?
- *RQ3 :* How do different hyperparameters affect the performance of KISDAN?
- *RQ4 :* How efficient is KISDAN compared to baseline models?
- *RQ5 :* How dose KISDAN provide meaningful interpretation of the prediction results?

5.1 Datasets

We evaluate the performance of KISDAN on three commonly used datasets: Amazon-Book,[1] MovieLens-1 M[2] and Last.FM.[3] The Amazon-Book dataset is selected from the widely used product recommendation dataset Amazon-review. The MovieLens-1 M is a movie dataset consisting of movie ratings. Each person expresses their preferences for a movie using scores ranging from 1 to 5. The preferences between users and movies are defined as implicit feedback. The Last.FM dataset is a music listening dataset collected from Last.FM online music systems. We preprocess the datasets as follows: (i) We generate the negative samples required for training and testing by randomly selecting unseen items for each user, maintaining the same size as the positive samples. (ii) We construct a knowledge graph for each dataset following previous works [20, 25]. For Amazon-book, following the methodology of KGAT [20], items are mapped to Freebase entities through title matching if a mapping is available. Additionally, to ensure data quality, we employ a 10-core setting [20], which retains users and items with at least 10 interactions and filters out KG entities with fewer than 10 triples. MovieLens-1 M and Last.FM follow the approach of MCCLK [25] and employ Microsoft's Satori for their construction. By matching the names of movies or musicians with the tail of triples, all valid Satori IDs are collected, and then, the item IDs are matched with the head of all triples. This process selects all well-matched triples, where each triple has a confidence level exceeding 0.9. (iii) For each dataset, assuming a user's entire sequence of behaviors as $(b_1, b_2, \ldots, b_k, \ldots, b_n)$, the task is to predict whether the

[1] http://jmcauley.ucsd.edu/data/amazon/.

[2] https://grouplens.org/datasets/movielens/1m/.

[3] https://grouplens.org/datasets/hetrec-2011/.

$(k+1)$-th item will be clicked by the user based on the first k interacted items. We generate training datasets and test datasets for each user. In the training dataset, k is set to $(n-2)$. In the test set, given the first $(n-1)$ behaviors, we aim to predict the last one. The statistics of the three datasets is shown in Table 1.

5.2 Parameter settings and evaluation metrics

KISDAN is implemented under the TensorFlow framework and trained on NVIDIA Tesla P100 GPU. For the baseline models, we follow the official hyperparameter settings provided in the original paper and the default settings in the corresponding code. As for KISDAN, the dimensions of user and item embeddings are set to 18 and the head number in multi-head self-attention is set to 4. To avoid excessive connections in the item co-occurrence graph, we retain only the edges with co-occurrence counts exceeding a threshold, which are set to 4, 11, and 3 for the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively. During training, the hyperparameter λ in the loss function is set to 2, τ in the interest segment division layer is set to 2, 5, 2 for the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively. The batch size is set to 16. We employ the Adam optimization algorithm as the training optimizer with a learning rate of 0.001. For KGIC, the contrastive loss weight is set to 1×10^{-6}, 1×10^{-7} and 1×10^{-6}, and the L2 regularization weight is set to 1×10^{-4}, 1×10^{-5} and 1×10^{-4} for the Amazon-Book, MovieLens-1 M and Last.FM datasets, respectively. For MCCLK, the local collaborative aggregation depth is set to 2, 2, and 3, and the local semantic aggregation depth is set to 1, 1, and 2 for the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively. For HAKG, the number of negative samples per user is set to 200, 400, and 200, and the margin of contrastive loss is set to 0.7, 0.8, and 0.6 for the Amazon-Book, MovieLens-1 M and Last.FM datasets, respectively. For DCLKR, the aggregation depth is set to 2, 3 and 2, the intraview contrastive loss weight is set to 0.1, 0.01, and 0.01, and the interview contrastive loss weight is set to 0.1, 0.01, and 0.01 for the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively.

In terms of evaluation metrics, we follow previous research such as [1, 23]. We assess the comparative performance of different methods using widely adopted metrics in the field of click-through rate (CTR), namely AUC (Area under the ROC curve) and F1. Higher AUC and F1 indicates better performance. Moreover, we follow previous studies such as [44, 45] and run KISDAN and the best-performing baseline models (HAKG and DCLKR) for five times using random seeds and perform a two-tailed unpaired t-test to calculate p-values for significance analysis.

5.3 Baseline models

To verify the effectiveness of KISDAN, we compare it with some state-of-the-art models on the Amazon-Book, MovieLens-1 M, and Last.FM datasets. We consider two kinds of representative click-through rate prediction models: user behavior-based models and knowledge graph-based models. DIN, DIEN, DMIN, and DBPMaN are user behavior-based models, while CG-KGR, KGIC, MCCLK, HAKG, and DCLKR are knowledge graph-based models. These baseline models are listed below:

1. *DIN* [1]: DIN employs attention mechanism to learn adaptive representations of user behavior related to the target item.
2. *DIEN* [3]: DIEN designs an auxiliary network to capture user's temporal interests and introduces AUGRU to model interest evolution.
3. *DMIN* [6]: DMIN incorporates a behavior refinement layer to capture enhanced user historical item representations and applies a multi-interest extraction layer to extract multiple user interests.
4. *DBPMaN* [10]: DBPMaN designs a deep neural network for behavior path matching and takes into account the influence of sequential behaviors that include user decision trajectories.
5. *CG-KGR* [23]: CG-KGR encapsulates historical interactions into interactive information summaries. Then, utilizing it as a guide, CG-KGR extracts information from the knowledge graph to achieve comprehensive and coherent learning of both the knowledge graph and user-item interactions.
6. *KGIC* [24]: KGIC constructs local and non-local graphs for users/items in the knowledge graph. It performs intragraph contrastive learning within each local/non-local graph and conducts intergraph contrastive learning between the local and non-local graphs. Therefore, KGIC can effectively integrate sparse interactions and redundant facts from the knowledge graph.
7. *MCCLK* [25]: MCCLK performs contrastive learning on three views at both the local and global levels to self-supervise the exploration of comprehensive graph features and structural information. MCCLK also designs a k-nearest neighbor item–item semantic graph construction module to capture important item–item semantic relationships.
8. *HAKG* [26]: HAKG embeds users and items, as well as entities and relations, in the hyperbolic space. It designs a hyperbolic aggregation scheme to gather

Table 1 Statistics of three datasets

Datasets		Amazon-Book	MovieLens-1 M	Last.FM
User–item interaction	#Users	70679	6036	1872
	#Items	24915	2445	3846
	#Interactions	847733	753772	42346
Knowledge graph	#Entities	88572	182011	9366
	#Relations	39	12	60
	#Triplets	2557746	1241996	15518

relationship contexts on the KG and introduces a novel angle constraint to preserve item features in the embedding space.

9. *DCLKR* [27]: DCLKR disentangles the item knowledge graph into multiple aspects for the knowledge view, and the user–item interaction graph for the collaborative view. DCLKR performs intraview contrastive learning to learn differences among disentangled representations in each view. It also performs interview contrastive learning to transfer knowledge between both the knowledge view and collaborative view.

5.4 Main results (RQ1)

The performance results of KISDAN and all baseline models on three benchmark datasets are shown in Table 2. Here, bold* denotes a p-value less than 0.005. KISDAN achieves AUC values of 0.9273, 0.9360, and 0.8735, F1 values of 0.8575, 0.8638, and 0.7914 on the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively. Among all baseline models, DCLKR performs best with the AUC values of 0.9214, 0.9219, and 0.8702, F1 values of 0.8453, 0.8302, and 0.7827 on the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively.

Firstly, it is obvious that KISDAN outperforms all models, surpassing DCLKR by 0.64%, 1.53%, and 0.38% in AUC values, 1.44%, 4.05%, and 1.11% in F1 values on the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively. In contrast, compared to HAKG, which performs the second best among baseline models, DCLKR only surpasses HAKG by 0.25%, 0.67%, and 2.93% in AUC values, and 0.25%, 1.07%, and 2.76% in F1 values on the same datasets. The statistical comparison of the performance improvement between KISDAN, DCLKR, and HAKG shows that KISDAN achieves significant performance improvements over the best-performing existing models. The p-values are well below 0.05, indicating that KISDAN achieves statistically significant improvement over the best-performing baseline models: HAKG and DCLKR.

Secondly, compared to those user behavior-based baseline models, KISDAN effectively utilizes the knowledge graph to divide user behavior sequence into multiple interest segments, fully leveraging the implicit semantic information within the knowledge graph. KISDAN can also distinguish the difference between strong interests and weak interests so that it can effectively evaluate the importance of interactions between different user interests. Furthermore, KISDAN introduces items that users have not clicked on to reveal potential interests. Compared to those knowledge graph-based models, KISDAN can seamlessly integrate the knowledge graph with the item co-occurrence graph to model user behavior sequence. In addition, KISDAN employs attention mechanisms to capture the mutual influence between user behavior and target item. Thus, it can capture personalized user interests more effectively than those knowledge graph-based models.

Thirdly, we can also observe that knowledge graph-based models generally perform better than user behavior-based models. The performance gap between knowledge graph-based models and user behavior-based models may be attributed to the fact that knowledge graph-based models leverage the rich semantic information in the knowledge graph to learn relationships between users and items. In contrast, user behavior-based models solely rely on user's historical behaviors to extract user interests and cannot fully understand user interests.

From a conceptual perspective, KISDAN outperforms all baseline models on three benchmark datasets, validating its effectiveness of KISDAN in capturing the diversity of user interests. By defining the concept of interest segment and dividing user behavior sequence into multiple interest segments, KISDAN accurately models a user's interests in different interest domains. This highlights the importance of utilizing the knowledge graph to mine implicit semantic information in user behavior sequence. Additionally, it also demonstrates the necessity of distinguishing a user's strong and weak interests and exploring users' potential interests from different perspectives.

From a practical perspective, KISDAN excels in capturing user interest in a much finer granularity. It utilizes items that have not been clicked to mine user's potential interest, enhancing the model's exploration capabilities.

This approach enables recommendation systems to more accurately predict content that users may be interested in, thereby enhancing user satisfaction. This not only can improve user experience but also can help businesses achieve their marketing objectives more effectively. Furthermore, experiments in the domains of books, movies, and music offer new possibilities for the application of recommendation systems across different fields.

5.5 Ablation study (RQ2)

In this section, we conduct experiments on several ablation models to analyze the contribution of different modules on the overall performance of KISDAN. We introduce six ablation models, including KISDAN w/o SI, KISDAN w/o WI, KISDAN w/o KGI, KISDAN w/o CFI, KISDAN w/o SWAT, and KISDAN w/o CL. Specifically, KISDAN w/o SI, KISDAN w/o WI, KISDAN w/o KGI, and KISDAN w/o CFI indicate the removal of strong interest, weak interest, semantic-based potential interest, and similarity-based potential interest, respectively, from the input of the final MLP. KISDAN w/o SWAT removes the strong-to-weak attention mechanism and directly uses the target item as the query. KISDAN w/o CL removes the potential interest contrastive learning layer. The experimental results on Amazon-Book, MovieLens-1 M, and Last.FM datasets are shown in Table 3.

Strong interests represent the core interests of users. Identifying users' strong interests helps the model to provide more relevant content accurately, increasing user satisfaction. From Table 3, we can see that KISDAN w/o SI performs the worst among all ablation models. Compared to KISDAN, the AUC values of KISDAN w/o SI drop by 3.82%, 3.46%, and 4.79% on the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively. This indicates that strong interests have a significant impact on learning user interests, as strong interests represent the preferences that users often exhibit.

Similarity-based potential interests are extracted from the item co-occurrence graph, reflecting the similarity relationship between each item interacted with by the user and those not clicked, thereby expanding the diversity of recommendations. KISDAN w/o CFI performs the second worst among the ablation models. The AUC values of KISDAN w/o CFI drop by 3.10%, 3.19%, and 3.70% on the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively. The performance drop proves that incorporating the similarity information from the item co-occurrence graph is crucial for mining user's potential interests.

Contrastive learning methods are used to explore the complementary relationship between potential interests. This approach leads to a comprehensive understanding of user interests. As for KISDAN w/o CL, it performs slightly better than KISDAN w/o SI and KISDAN w/o CFI. Compared to KISDAN, KISDAN w/o CL exhibits a drop of 2.06%, 2.72%, and 2.49% in AUC values on the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively. This performance gap indicates that contrastive learning can capture complementary information between semantically-based potential interests and similarity-based potential interests, leading to a more comprehensive understanding of user interests.

The knowledge graph is employed to mine semantic-based potential interests, KISDAN can understand the deep semantic relationships behind user behaviors. As for KISDAN w/o KGI, it achieves a better performance than KISDAN w/o CFI. This indicates that the similarity-based potential interest learned from the item co-occurrence graph is more important than the semantic-based potential interests learned from the knowledge graph.

Weak interests represent interests not explicitly expressed through frequent interactions. However, weak interests may also have an indirect connection with strong interests and influence the user's clicking behaviors. As for KISDAN w/o WI, the performance gap between KISDAN w/o WI and KISDAN is relatively smaller, indicating that weak interests play a supplementary role in learning user interests.

Strong-to-weak attention mechanism is designed to effectively utilize strong interests to extract information related to weak interests, reducing the impact of noise within weak interests. Finally, KISDAN w/o SWAT performs the best among the ablation models. Compared to KISDAN, it only exhibits a drop of 0.28%, 0.81%, and 0.70% in AUC values on the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively. However, the performance gap between KISDAN w/o SWAT and KISDAN still highlights the importance of exploring the interaction between strong interests and weak interests.

5.6 Parameter analysis (RQ3)

In this section, we conduct some experiments to analyze the impact of some hyperparameters on the performance of KISDAN.

5.6.1 Threshold for interest segment division τ

In the interest segment division layer of KISDAN, the hyperparameter τ is used to determine the number of items within the strong interest segments and the weak interest segments. This hyperparameter helps KISDAN to learn a user's different levels of interests and the correlations between these interests. Thus, the hyperparameter τ is one of the main factors affecting the overall performance of KISDAN. We evaluate the performance of KISDAN with τ

Table 2 Performance results of KISDAN and baseline models on three benchmark datasets

Models	Amazon-Book		MovieLens-1 M		Last.FM	
	AUC	F1	AUC	F1	AUC	F1
DIN	0.8294	0.7396	0.8196	0.7479	0.8201	0.7314
DIEN	0.8431	0.7607	0.8402	0.7645	0.8308	0.7410
DMIN	0.8531	0.7635	0.8611	0.7786	0.8372	0.7657
DBPMaN	0.8649	0.7796	0.8763	7842	0.8396	0.7760
CG-KGR	0.9060	0.8256	0.8819	0.7868	0.8263	0.7334
KGIC	0.8907	0.8104	0.9011	0.8147	0.8529	0.7787
MCCLK	0.9078	0.8259	0.9050	0.8153	0.8459	0.7647
HAKG	0.9191	0.8432	0.9158	0.8214	0.8454	0.7617
DCLKR	0.9214	0.8453	0.9219	0.8302	0.8702	0.7827
KISDAN	**0.9273***	**0.8575***	**0.9360***	**0.8638***	**0.8735***	**0.7914***

ranging from 1 to 8. The experimental results are shown in Fig. 2a and b. On the Amazon-Book dataset and the Last.FM dataset, the AUC value and F1 value reaches its peak when τ is set to 2 and then gradually decreases as τ increases. On the MovieLens-1 M dataset, the AUC value and F1 value reaches its peak when τ is set to 5 and then gradually decreases as τ increases. We attribute it to the reason that a small τ value may result in too many interest segments being classified as strong interest segments. Consequently, these strong interest segments, which consists of a small number of items, may not adequately represent a user's strong interests. Conversely, if τ is set too high, KISDAN may only extract an insufficient number of strong interest segments, potentially overlooking some of the core interests of users. In addition, the optimal values of τ for the Amazon-Book, MovieLens-1 M, and Last.FM datasets are different. KISDAN achieves the best performance when τ is set to 2, 5, and 2 on the Amazon-Book, MovieLens-1 M, and Last.FM datasets, respectively. This could be attributed to the fact that the MovieLens-1 M dataset has a longer average user behavior sequence length, so each interest segment needs to contain more items to be considered as a user's strong interests.

5.6.2 Loss weight λ

KISDAN introduces contrastive learning loss as an auxiliary loss in the training process, and the weight of the contrastive learning loss influences its role in CTR prediction. Therefore, we evaluate the performance of KISDAN by setting the loss weight λ to 0.2, 1, 2, 4, and 8, respectively. The experimental results are shown in Fig. 2c and d. We observe that KISDAN achieves the best performance on both datasets when λ is set to 2. When λ is too small or too large, KISDAN's performance is not optimal. This indicates that the contrastive learning loss plays a supportive role in CTR prediction. A too small contrastive learning loss cannot fully leverage the benefits of contrastive learning to effectively capture complementary information between user's two potential interests. Conversely, a too large contrastive learning loss may reduce the impact of the main task and lead to model bias.

5.7 Complexity analysis (RQ4)

In this section, we comprehensively analyze the model complexity exhibited by KISDAN and some typical baseline models. Specifically, among these baseline models,

Table 3 Performance of KISDAN and its ablation models on three datasets

Models	Amazon-Book		MovieLens-1 M		Last.FM	
	AUC	F1	AUC	F1	AUC	F1
KISDAN	**0.9273**	**0.8575**	**0.9360**	**0.8638**	**0.8735**	**0.7914**
KISDAN w/o SI	0.8932	0.8098	0.9047	0.8388	0.8336	0.7523
KISDAN w/o WI	0.9238	0.8542	0.9205	0.8520	0.8636	0.7855
KISDAN w/o SWAT	0.9247	0.8404	0.9285	0.8551	0.8674	0.7920
KISDAN w/o KGI	0.9164	0.8358	0.9182	0.8598	0.8585	0.7734
KISDAN w/o CFI	0.8994	0.8176	0.9071	0.8346	0.8423	0.7567
KISDAN w/o CL	0.9086	0.8257	0.9112	0.8410	0.8523	0.7657

The bold values represent the best values for each evaluation metric

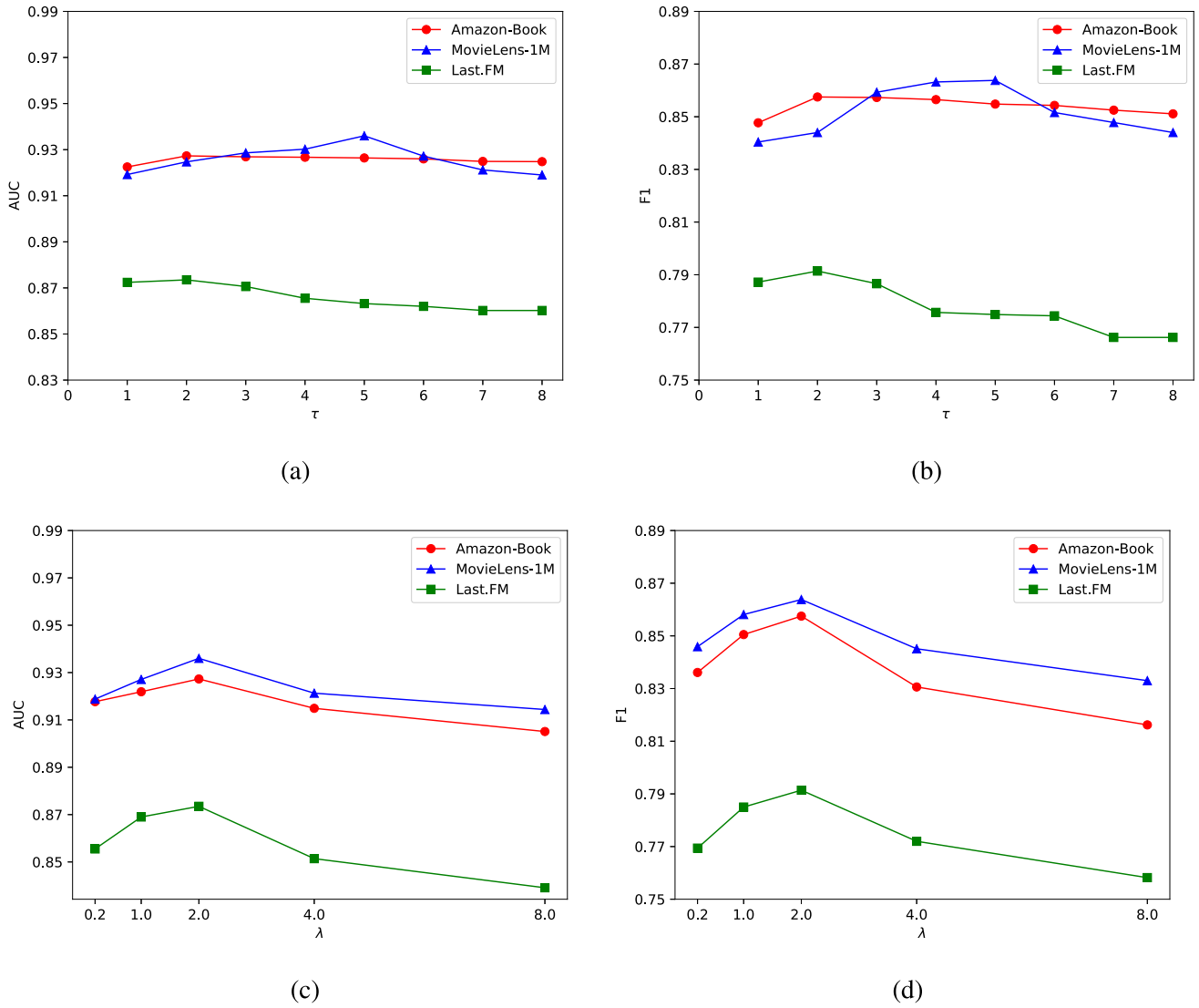

Fig. 2 Effect of some hyperparameters on the performance of KISDAN

Table 4 Complexity of KISDAN and some typical baseline models

Models	Params
DIN	165217
DIEN	194949
DMIN	287488
DBPMaN	208045
CG-KGR	197792
KGIC	611520
MCCLK	751872
HAKG	997632
DCLKR	743552
KISDAN	259840

DIN, DIEN, DMIN, and DBPMaN are user behavior-based models, while CG-KGR, KGIC, MCCLK, HAKG, and DCLKR are knowledge graph-based models. Leveraging the official codebases provided by the authors, these models are executed with default settings on the Last.FM dataset, and the complexity results are presented in Table 4. Additionally, Table 5 represents the training time and inference time of KISDAN and several best-performing baseline models. The training time concerns one epoch, while the testing time concerns one batch. All models are trained with a batch size of 16 on a NVIDIA P100 GPU.

From Table 4, it is evident that KISDAN has more parameters than most user behavior-based models. This discrepancy arises from the fact that user behavior-based models only consider the user behavior sequence. In contrast, the knowledge graph-based models need to perform multi-hop propagation on the user–item interaction graph and the knowledge graph using GNNs, thus requiring more parameters. Furthermore, KISDAN applies attention mechanisms to both the knowledge graph and the item co-

Table 5 Training speed and inference speed of KISDAN and several best-performing baseline models

Models	Training time (s)	Inference time (s)
KGIC	130.91	0.0271
MCCLK	1043.84	0.3575
HAKG	632.32	0.0938
DCLKR	1423.39	0.0899
KISDAN	62.47	0.0178

occurrence graph, rather than performing multi-hop propagation using GNNS. This allows KISDAN to effectively leverage the rich information from the knowledge graph and item co-occurrence graph while maintaining a balance between model complexity and model performance.

From Table 5, it can be observed that KISDAN improves both training and inference efficiency while achieving superior performance. Knowledge graph-based baseline models, such as KGIC, MCCLK, HAKG, and DCLKR, utilize GNNs to perform feature propagation. Therefore, these baseline models are computationally expensive for both training and inference. KISDAN divides user behavior sequence into multiple interest segments and applies attention mechanisms to these segments to directly weigh important information. This approach avoids the need for feature propagation across the entire graph. Therefore, it can reduce computational complexity while maintaining model performance. This demonstrates the advantages of KISDAN in terms of model complexity and effectiveness.

5.8 Case study (RQ5)

In this section, we further analyze the effectiveness of the proposed interest segment division strategy through case studies. Specifically, we select one user behavior sequence from the Amazon-Book dataset for detailed analysis. To analyze the importance of both strong and weak interest segments in click-through rate prediction, we visualize the attention weights assigned by KISDAN to each strong interest segment and weak interest segment. The *X*-axis represents the entities corresponding to each interest segment, and the results are shown in Figs. 3 and 4. Additionally, we select DIN as the comparison model in this case study. DIN is a classic model, which uses the target item as the query and applies attention mechanisms to each item in the user behavior sequence to assess the impact of each item on user interests. In contrast, KISDAN evaluates the impact of each interest segment on user interests. Through the comparison, we can demonstrate the differences between modeling user interests using interest segments and items. Figure 5 illustrates how DIN allocates attention weights to each item in the user behavior sequence, with the *X*-axis representing the items. In these figures, a deeper color indicates a higher attention weight assigned and vice versa. The target item that the user intended to click on is "Luring A Lady" authored by Nora Roberts.

As shown in Fig. 3, it is apparent that the user demonstrates a frequent preference for clicking on other books authored by Nora Roberts. Notably, KISDAN assigns significant attention to "Nora Roberts". As Nora Roberts is a romance novelist, KISDAN also assigns a higher weight to "Romance novel". Similarly, as illustrated in Fig. 4, KISDAN assigns higher weights to other romance novelists such as "Lisa Kleypas" and "Julie Garwood", indicating that the user's other interests also influence his/her clicks on "Luring A Lady". We attribute this result to the fact that KISDAN incorporates semantic information from the knowledge graph, so KISDAN can uncover the user's complex interests and accurately predict the user's clicking behaviors. Figure 5 shows that DIN only assigns higher weights to some books by Nora Roberts ("Bed of Roses" and "Whiskey Beach"), while other books by Nora Roberts ("Pride of Jared Mackade", "Happy Ever After", and "Vision in White") are assigned lower weights. This discrepancy results in DIN's inability to accurately capture the user's interest in the author Nora Roberts. Therefore, DIN fails to accurately predict the user's click on "Luring A Lady". We attribute this result to the fact that DIN relies solely on user interaction data to learn user interests, without incorporating semantic information from the knowledge graph. This prevents DIN from learning deeper user interests.

6 Conclusion and future work

In this paper, we propose a novel and effective model called knowledge-enhanced interest segment division attention network (KISDAN) for click-through rate prediction tasks. For user behavior sequences, we divide them

Fig. 3 One case with visualized attention weights assigned by KISDAN to each strong interest segment

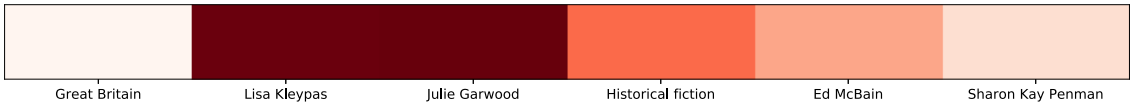

Fig. 4 One case with visualized attention weights assigned by KISDAN to each weak interest segment

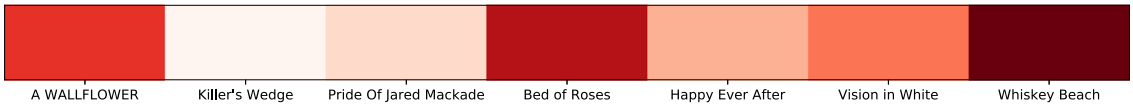

Fig. 5 One case with visualized attention weights assigned by DIN to each item

into multiple interest segments based on the information from the knowledge graph, considering these segments as fundamental units for modeling user interests. These interest segments provide crucial clues for capturing the preferences exhibited by users in multiple interest domains. KISDAN offers a comprehensive and accurate approach to modeling user interests by refining them into strong and weak interests, as well as introducing items that a user has not interacted with as potential interests. This contributes to expanding the exploration capability of KISDAN and improving the performance of click-through rate prediction. Extensive experiments on three commonly used benchmark datasets demonstrate the efficacy of KISDAN compared to several state-of-the-art models.

Although KISDAN performs well in capturing diverse and complex interests of users, refining user interests, and learning user's potential interests, there are still several avenues for future research and improvement. Firstly, the effectiveness of KISDAN depends on the quality and completeness of the knowledge graph. If information in the knowledge graph is missing or erroneous, it may impact the accuracy of interest segmentation. Secondly, the interest segment division method may overlook some complex relationships between user interests. For instance, different interest domains of a user may intersect with or influence each other. Lastly, the decision-making process of users may be influenced by multiple factors, such as temporal elements, which KISDAN has not yet comprehensively considered. Future work will focus on exploring how to reduce dependency on high-quality knowledge graphs and mine user interests in a more granular way. Furthermore, it is also urgent to explore how to capture temporal information within interest segments to reflect real-time interests of users at different time points.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 61672158, 61972097 and U21A20472, in part by the Major Science and Technology project of Fujian Province (China) under Granted No. 2021HZ022007, in part by the Industry-Academy Cooperation Project under Grant 2021H6022, in part by the Natural Science Foundation of Fujian Province under Grant 2020J01494, in part by the Collaborative Innovation Platform Project of Fuzhou City under Grant 2023-P-002.

Data availability The data used in this article are available in the online supplementary material. Supplementary materials are available at http://jmcauley.ucsd.edu/data/amazon/, https://grouplens.org/datasets/movielens/1m/ and https://grouplens.org/datasets/hetrec-2011/.

Code availability The code for this paper has been uploaded to Github: https://github.com/java-jay/KISDAN.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Zhou G, Zhu X, Song C, Fan Y, Zhu H, Ma X, Yan Y, Jin J, Li H, Gai K (2018) Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp 1059–1068
2. Li F, Chen Z, Wang P, Ren Y, Zhang D, Zhu X (2019) Graph intention network for click-through rate prediction in sponsored search. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. pp 961–964
3. Zhou G, Mou N, Fan Y, Pi Q, Bian W, Zhou C, Zhu X, Gai K (2019) Deep interest evolution network for click-through rate prediction. Proceedings of the AAAI conference on artificial intelligence 33:5941–5948
4. Feng Y, Lv F, Shen W, Wang M, Sun F, Zhu Y, Yang K (2019) Deep session interest network for click-through rate prediction. In: Proceedings of the 28th international joint conference on artificial intelligence. pp2301–2307
5. Pi Q, Bian W, Zhou G, Zhu X, Gai K (2019) Practice on long sequential user behavior modeling for click-through rate prediction. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp2671–2679
6. Xiao Z, Yang L, Jiang W, Wei Y, Hu Y, Wang H (2020) Deep multi-interest network for click-through rate prediction. In: Proceedings of the 29th ACM international conference on information & knowledge management. pp2265–2268
7. Jiang W, Jiao Y, Wang Q, Liang C, Guo L, Zhang Y, Sun Z, Xiong Y, Zhu Y (2022) Triangle graph interest network for click-through rate prediction. In: Proceedings of the fifteenth ACM

international conference on web search and data mining. pp401–409
8. Fan Z, Ou D, Gu Y, Fu B, Li X, Bao W, Dai X-Y, Zeng X, Zhuang T, Liu Q (2022) Modeling users' contextualized page-wise feedback for click-through rate prediction in e-commerce search. In: Proceedings of the fifteenth ACM international conference on web search and data mining. pp 262–270
9. Lin Q, Zhou W-J, Wang Y, Da Q, Chen Q-G, Wang B (2022) Sparse attentive memory network for click-through rate prediction with long sequences. In: Proceedings of the 31st ACM international conference on information & knowledge management. pp3312–3321
10. Dong J, Yu Y, Zhang Y, Lv Y, Wang S, Jin B, Wang Y, Wang X, Wang D (2023) A deep behavior path matching network for click-through rate prediction. Companion proceedings of the ACM web conference 2023:538–542
11. Feng Y, Hu B, Lv F, Liu Q, Zhang Z, Ou W (2020) Atbrg: adaptive target-behavior relational graph network for effective recommendation. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. pp 2231–2240
12. Feng Y, Lv F, Hu B, Sun F, Kuang K, Liu Y, Liu Q, Ou W (2020) Mtbrn: multiplex target-behavior relation enhanced network for click-through rate prediction. In: Proceedings of the 29th ACM international conference on information & knowledge management. pp 2421–2428
13. Zhang F, Yuan NJ, Lian D, Xie X, Ma W-Y (2016) Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp 353–362
14. Wang H, Zhang F, Xie X, Guo M (2018) Dkn: deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide Web conference. pp 1835–1844
15. Yu X, Ren X, Sun Y, Gu Q, Sturt B, Khandelwal U, Norick B, Han J (2014) Personalized entity recommendation: A heterogeneous information network appoach. In: Proceedings of the 7th ACM international conference on web search and data mining. pp 283–292
16. Hu B, Shi C, Zhao WX, Yu PS (2018) Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp 1531–1540
17. Wang X, Wang D, Xu C, He X, Cao Y, Chua T-S (2019) Explainable reasoning over knowledge graphs for recommendation. Proceedings of the AAAI Conference on artificial intelligence 33:5329–5336
18. Wang H, Zhang F, Wang J, Zhao M, Li W, Xie X, Guo M (2018) Rippenet: Propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM international conference on information and knowledge management. pp 417–426
19. Wang H, Zhao M, Xie X, Li W, Guo M (2019) Knowledge graph convolutional networks for recommender systems. In: The World Wide Web conference. pp 3307–3313
20. Wang X, He X, Cao Y, Liu M, Chua T-S (2019) Kgat: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp 950–958
21. Wang Z, Lin G, Tan H, Chen Q, Liu X (2020) Ckan: Collaborative knowledge-aware attentive network for recommender systems. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. pp 219–228
22. Wang X, Huang T, Wang D, Yuan Y, Liu Z, He X, Chua T-S (2021) Learning intents behind interactions with knowledge graph for recommendation. Proceedings of the web conference 2021:878–887
23. Chen Y, Yang Y, Wang Y, Bai J, Song X, King I (2022) Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation. In: 2022 IEEE 38th international conference on data engineering (ICDE). IEEE, pp 299–311
24. Zou D, Wei W, Wang Z, Mao X-L, Zhu F, Fang R, Chen D (2022) Improving knowledge-aware recommendation with multi-level interactive contrastive learning. In: Proceedings of the 31st ACM international conference on information & knowledge management. pp 2817–2826
25. Zou D, Wei W, Mao X-L, Wang Z, Qiu M, Zhu F, Cao X (2022) Multi-level cross-view contrastive learning for knowledge-aware recommender system. In: Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. pp 1358–1368
26. Du Y, Zhu X, Chen L, Zheng B, Gao Y (2022) Hakg: hierarchy-aware knowledge gated network for recommendation. In: Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. pp 1390–1400
27. Huang S, Hu C, Kong W, Liu Y (2023) Disentangled contrastive learning for knowledge-aware recommender system. In: International semantic web conference. Springer, pp 140–158
28. Rendle S (2010) Factorization machines. In: 2010 IEEE international conference on data mining. IEEE, pp 995–1000
29. Guo H, Tang R, Ye Y, Li Z, He X (2017) Deepfm: a factorization-machine based neural network for ctr prediction. In: Proceedings of the 26th international joint conference on artificial intelligence. pp 1725–1731
30. Lian J, Zhou X, Zhang F, Chen Z, Xie X, Sun G (2018) xdeepfm: combining explicit and implicit feature interactions for recommender systems. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp 1754–1763
31. Hadsell R, Chopra S, LeCun Y (2006) Dimensionality reduction by learning an invariant mappng. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, pp 1735–1742. IEEE
32. Chen T, Kornblith S, Norouzi M, Hinton G (2020) A simple framework for contrastive learning of visual representations. In: International conference on machine learning. PMLR, pp 1597–1607
33. Yang Z, Cheng Y, Liu Y, Sun M (2019) Reducing word omission errors in neural machine translation: a contrastive learning appoach. In: Proceedings of the 57th annual meeting of the Association for Computational Linguistics. pp 6191–6196
34. Xie X, Sun F, Liu Z, Wu S, Gao J, Zhang J, Ding B, Cui B (2022) Contrastive learning for sequential recommendation. In: 2022 IEEE 38th international conference on data engineering (ICDE). IEEE, pp 1259–1273
35. Wu J, Wang X, Feng F, He X, Chen L, Lian J, Xie X (2021) Self-supervised graph learning for recommendation. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. pp 726–735
36. Wei Y, Wang X, Li Q, Nie L, Li Y, Li X, Chua T-S (2021) Contrastive learning for cold-start recommendation. In: Proceedings of the 29th ACM international conference on multimedia. pp 5382–5390
37. Zhang L, Liu Y, Zhou X, Miao C, Wang G, Tang H (2022) Diffusion-based graph contrastive learning for recommendation with implicit feedback. In: International Conference on database systems for advanced appications. Springer, pp 232–247
38. Zheng Y, Gao C, Chang J, Niu Y, Song Y, Jin D, Li Y (2022) Disentangling long and short-term interests for recommendation. Proceedings of the ACM web conference 2022:2256–2267

39. Zhang Y, Liu Y, Xu Y, Xiong H, Lei C, He W, Cui L, Miao C (2022) Enhancing sequential recommendation with graph contrastive learning. arXiv preprint arXiv:2205.14837
40. Guo W, Zhang C, He Z, Qin J, Guo H, Chen B, Tang R, He X, Zhang R (2022) Miss: multi-interest self-supervised learning framework for click-through rate prediction. In: 2022 IEEE 38th international conference on data engineering (ICDE). IEEE, pp 727–740
41. Zhang S, Li B, Yao D, Feng F, Zhu J, Fan W, Zhao Z, He X, Chua T-s, Wu F (2022) Ccl4rec: contrast over contrastive learning for micro-video recommendation. arXiv preprint arXiv:2208.08024
42. Wang F, Wang Y, Li D, Gu H, Lu T, Zhang P, Gu N (2023) Cl4ctr: A contrastive learning framework for CTR prediction. In: Proceedings of the sixteenth ACM international conference on web search and data mining. pp 805–813
43. Kenton JDM-WC, Toutanova LK (2019) Bert: Pre-training of deep bidirectional transformers for language understanding, 4171–4186
44. Zhu C, Chen B, Zhang W, Lai J, Tang R, He X, Li Z, Yu Y (2021) Aim: automatic interaction machine for click-through rate prediction. IEEE Trans Knowl Data Eng 35(4):3389–3403
45. Zheng Z, Zhang C, Gao X, Chen G (2022) Hien: hierarchical intention embedding network for click-through rate prediction. In: Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. pp 322–331