This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3387276

Unified Representation Learning for Discrete Attribute Enhanced Completely Cold-Start Recommendation

Haoyue Bai, Min Hou, Le Wu, *Member, IEEE,* Yonghui Yang, Kun Zhang, *Member, IEEE,* Richang Hong, *Member, IEEE,* Meng Wang, *Fellow, IEEE*

Abstract—Recommender systems face a daunting challenge when entities (users or items) without any historical interactions, known as the "*Completely Cold-Start Problem*". Due to the absence of collaborative signals, Collaborative Filtering (CF) schema fails to deduce user preferences or item characteristics for such cold entities. A common solution is incorporating auxiliary discrete attributes as the bridge to spread collaborative signals to cold entities. Most previous works involve embedding collaborative signals and discrete attributes into different spaces before aligning them for information propagation. Nevertheless, we argue that the separate embedding approach disregards potential high-order similarities between two signals. Furthermore, existing alignment modules typically narrow the geometric-based distance, lacking in-depth exploration of semantic overlap between collaborative signals and cold entities. In this paper, we propose a novel discrete attribute-enhanced completely cold-start recommendation framework, which aims to improve recommendation performance by modeling heterogeneous signals in a unified space. Specifically, we first construct a heterogeneous user-item-attribute graph and capture high-order similarities between heterogeneous signals in a graph-based message-passing manner. To achieve better information alignment, we propose two self-supervised alignment modules from the semantic mutual information and user-item preference perspective. Extensive experiments on six real-world datasets in two types of discrete attribute scenarios consistently verify the effectiveness of our framework.

Index Terms—Recommender System, Cold-Start Problem, Contrastive Learning.

1 Introduction

PERSONALIZED recommendations have emerged as critical components to alleviate information overloading for users in various online applications, including E-commerce, advertising, and so on [1; 2; 3]. At its core is estimating how likely a user will adopt an item based on historical interactions like purchases and clicks, known as collaborative signals. CF-based methods have shown remarkable success in modern recommender systems [4; 5; 6; 7]. Despite the success in serving regular users and recommending regular items, CF-based methods severely suffer from the cold-start problem, failing on new entities whose interactions are very limited with unsuitable recommendations. In many more extreme cases, models are required to make recommendations to newly registered users or recommend newly launched items, that without any historical records. Due to the absence of collaborative signals, CF-based methods fail to deduce user preferences or item characteristics for such cold entities. We refer to this dilemma as the *"completely cold-start problem"*.

To deal with this issue, many researchers shed light on exploiting *auxiliary discrete attributes*, such as user occupation and gender, item genre, brand , and so on [8; 9; 10; 11; 12; 13]. The discrete attributes have the ability to describe user preferences and item characteristics to a certain extent. More importantly, the attributes are shared by cold entities and warm entities. Thus, they are deemed as a bridge to fill the gap between warm entities with collaborative signals and cold entities. The modeling process of existing works generally consists of two stages, as depicted in Fig. 1. In the first stage, two encoders are trained to separately embed collaborative signals and auxiliary discrete attributes into different spaces. And then perform specific alignment functions (e.g., local geometric similarity, mean square error) to narrow the distance of embedding from different spaces for knowledge transfer, making the attribute representation contains valuable collaborative signals in this way. In the second stage, they generate the representation of cold entities based on the corresponding attribute representations to perform recommendation tasks. For example, Heater [12] employs multiple experts network to embed auxiliary discrete attributes and use pre-trained embedding to provide collaborative signals, then sum square error is used to align them. In the inference phase, the CF-aware attribute representations generated by this multiple experts network are used directly to recommend new entities. CLCRec [14] uses two separate encoders to convert collaborative signals and auxiliary discrete attributes into representations and then uses contrastive loss to align the two kinds of information for new entity recommendations.

Despite the effectiveness, we argue that these methods are not sufficient to yield satisfactory information fusion for collaborative signals and discrete attributes. The key reason is that they perform the modeling process in **two**

- *H. Bai, L. Wu, Y. Yang, Kun Zhang, R. Hong, and M. Wang are with the School of Computer and Information, Hefei University of Technology, Hefei, Anhui, 230009, China. E-mail:baihaoyue621, lewu.ustc, yyh.hfut, zhang1028kun, hongrc.hfut, eric.mengwang@gmail.com.*
- *M. Hou is with the University of Science and Technology of China,Hefei, Anhui, 230026, China. E-mail:minho@mail.ustc.edu.cn*

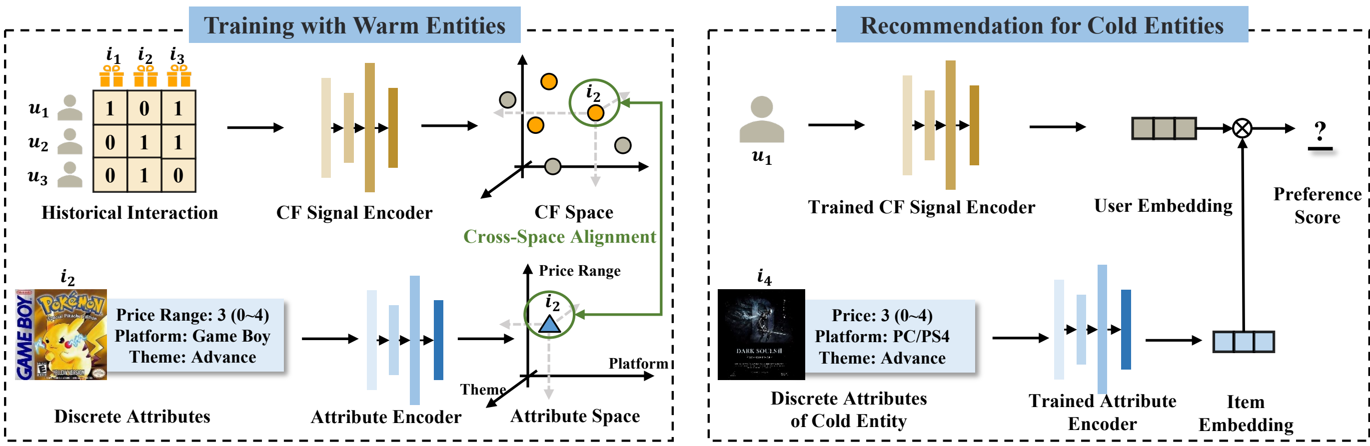


Fig. 1. The scheme of existing models for completely cold-start problem.

separate spaces. To be specific, first, the general practice of independently training collaborative signals and attribute representations in two separate spaces structurally isolates the interactions among warm entities and attributes. The two encoders aim to learn information from two sides separately, without considering the collaborative signals-attributes interactions. The interactions are only used to define the alignment loss function for model training. As a result, when the embeddings are insufficient in perceiving high-order similarities, the methods have to rely on the alignment loss function to make up for the deficiency of suboptimal embeddings. Secondly, due to the two-spaces-based modeling approach, existing alignment modules typically narrow the geometric-based distance of representations from two spaces to make collaborative signals transfer to attribute representations, lacking an in-depth guarantee that cold entities' and warm entities' semantic characteristics are consistent. To overcome these defects, it is necessary to model collaborative signals and discrete attributes in **a unified space**. However, there are still many unique challenges inherent in designing an effective unified framework. On the one hand, the collaborative signals and discrete attributes are heterogeneous. The collaborative signals consist of user-item historical interactions, which are usually represented by an adjacency matrix. And each user and item is initialized as a free embedding. Besides, auxiliary discrete attributes are diverse. And there are also interactions between some attributes, such as knowledge graphs. Therefore, representing two heterogeneous types of information in a unified space and capturing the high-order similarities between them is not non-trivial. On the other hand, how to design an effective alignment method that preserves semantic consistency between warm and cold representation is still an open issue.

Our proposed framework introduces a new approach to enhance completely cold-start recommendations by modeling collaborative signals and discrete attributes in a unified space. Unlike previous works that model heterogeneous information in different spaces, we construct a heterogeneous user-item-attribute graph by representing heterogeneous information as different types of nodes. Then we capture higher-order information and proximity between different types of nodes in a message-passing manner. Meanwhile, to better inject collaborative signals into cold representation, we propose two self-supervised alignments from the perspective of semantic mutual information and user-item preference, including maximizing mutual information between different representations and distance-based constraints between different preference scores. Our more comprehensive and well-designed alignment module results in better collaborative signal-aware cold representations. To prove the effectiveness and universality of the proposed model, we choose six real-world datasets to evaluate the performance of the model. The experiment results clearly verify the superiority and effectiveness of the proposed model and show that our model can be adapted to various types of discrete attributes, including the single and complex relationship between attributes and entities. We summarize our contributions as follows:

- We construct a unified heterogeneous user-item-attribute graph to capture high-order similarities between collaborative signals and discrete attributes in a graph-based message-passing manner.
- We propose two self-supervised alignment modules to achieve better information alignment between collaborative signals and discrete attributes from the semantic mutual information and user-item preference perspective, respectively.
- We conduct extensive experiments on six real-world datasets to demonstrate the superiority and effectiveness of the proposed model in solving the completely cold-start problem.

2 Related Work

2.1 Completely Cold-Start Recommendation

CF-based methods are widely used in recommendation [4; 15; 16; 17; 18]. Rich historical interaction records are key to the success of these methods. However, CF-based methods encounter a significant hurdle with the cold-start problem, where the model struggles to provide effective recommendations for users or items with insufficient historical interaction data. This issue can be classified into two types: completely cold-start and incompletely cold-start, depending on whether there are any previous interaction records [19]. The most common approach to solving the cold-start problem is to incorporate side information, such as content features, social networks, and user profiles, to bridge the gap between the collaborative signal and cold-start items (users) [9; 12; 20].

The completely cold-start problem poses greater challenges, as the sparsity of interactions for cold-start users and items is 100%, making it exceedingly difficult for modeling. Due to the complete absence of collaborative signals, most methods to solve the cold-start problem fail on the completely cold-start problem. Using side information without modeling the collaborative signal may lead to suboptimal performance [10; 21; 22; 23]. To address this issue, existing works usually use two separate encoders to convert heterogeneous CF information and side information into different spaces, and then dedicate to designing various alignment functions to model the correlation and narrow the difference between two embeddings. Specifically, DropoutNet [9] and MTPR [20] strategically discard collaborative filtering information during the training phase, prompting the model to close the gap between different embeddings by simulating a completely cold start scenario. Heater [12], PGD [11] and so on [14; 24] design optimization objectives to explicitly close the distance between collaborative representation and content representation, including reducing the Euclidean distance and mutual information between the representations. Besides, generative methods also be explored in this scenario [13; 25; 26; 27]. Specifically, GAR [27] and LARA [13] use generative adversarial networks to let the discriminator in the model confuse the collaborative representation and the generated content representation to close their gap. However, we argue these methods are insufficient in perceiving high-order similarities and lack an in-depth guarantee that the user and item semantic characteristics are consistent.

In the incomplete cold-start scenario, some methods [28; 29; 30; 31] try to solve the cold-start problem based on the meta-learning paradigm. Here, the global parameters of the model are learned with the existing user data and locally updated to rapidly adapt to the new user preferences with a few interactions. However, these methods require the existence of a small number of interaction records, which is not suitable for the completely cold-start scenario we are concerned about.

2.2 Contrastive Learning and Applications in Recommendations

Contrastive learning, which aims to learn high-quality representation via a self-supervised manner, has achieved remarkable successes in CV, NLP, and other fields [32; 33; 34; 35]. The common motivation behind these works is the InfoMax principle [36; 37]. By identifying the positive pair from some negative ones, contrastive learning maximizes the mutual information between two parts with semantic dependencies and emphasizes learning common features between different views of an instance. When it comes to recommender systems, most existing works apply contrastive learning to improve recommendation performance. Some works [38; 39; 40] organize user behavior data as graphs. The graph structure with slight perturbations may have similar semantics. By contrasting different structures, the shared invariance to structural perturbations is obtained as self-supervised signals. These works extract contrastive self-supervised signals from the data structure perspective, there are also some studies perform contrast between model-level augmentations. For instance, DuoRec [41] applies two different sets of dropout masks to a Transformer-based backbone for two model-level representation augmentations. SRMA [42] proposes to randomly drops some layers of the feed-forward network in the Transformer for model-level augmentation. Compared to the structure-level and model-level contrast, the feature-level contrast is relatively less explored. Inspired by contrastive learning, we maximize the mutual information between collaborative signal and side features to encourage the feature embedding level information fusion.

3 Problem Definition

Considering widespread implicit feedback scenarios, we supposed there are two sets of entities: a user set $U(|U| = M)$ and an item set $V(|V| = N)$. Let $\mathbf{O} \in \mathbb{R}^{M\times N}$ denote the observed implicit feedback matrix, where each entry $o_{ij} = 1$ if there is an interaction between the user i and item j, otherwise $o_{i,j} = 0$. The user-item interaction behavior could be naturally formulated as a user-item bipartite graph. We use $\mathbf{A}^O$ to represent the adjacent matrix that is constructed from the interaction matrix $\mathbf{O}$:

$$\mathbf{A}^O = \begin{bmatrix} \mathbf{O} & \mathbf{0}^{M\times N} \\ \mathbf{0}^{N\times M} & \mathbf{O}^T \end{bmatrix}. \quad (1)$$

In addition to interactions, we also take into account discrete attributes that are sparse and categorical. Take the attribute of the item as an example, We use $\mathbf{X} \in \mathbb{R}^{N\times D}$ to denote the matrix of item attributes, herein D is the dimension of item attributes. Besides, we employ $\mathbf{x}_j \in \mathbb{R}^D$ to denote item j's one-hot attribute $(0 \le j < N)$.

Given the above information, we aim to make recommendations for cold entities which have no prior historical interaction. We term this problem as *the completely cold-start problem*. To assess the performance of the model in real-world scenarios, we categorize our recommendation task into the following three tasks:

- *Task 1:* When a cold (new) item v_{cold} with side information appears, we have to recommend new items to warm (old) users u.
- *Task 2:* When a cold (new) user u_{cold} with side information appears, we recommend warm (old) items v to new users u_{cold}.
- *Task 3:* When cold (new) users and warm (old) items v_{cold} appear at the same time, we have to recommend cold items to cold users u_{cold}.

Our work focuses on the problem of new entities having no historical interaction at all. On the one hand, the initial recommendation experience greatly affects the retention rate of new users, and the feedback after the recommendation of new items affects their value evaluation. On the other hand, most recommendation models are difficult to train incrementally. Limited by the training cost in actual scenarios,

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3387276

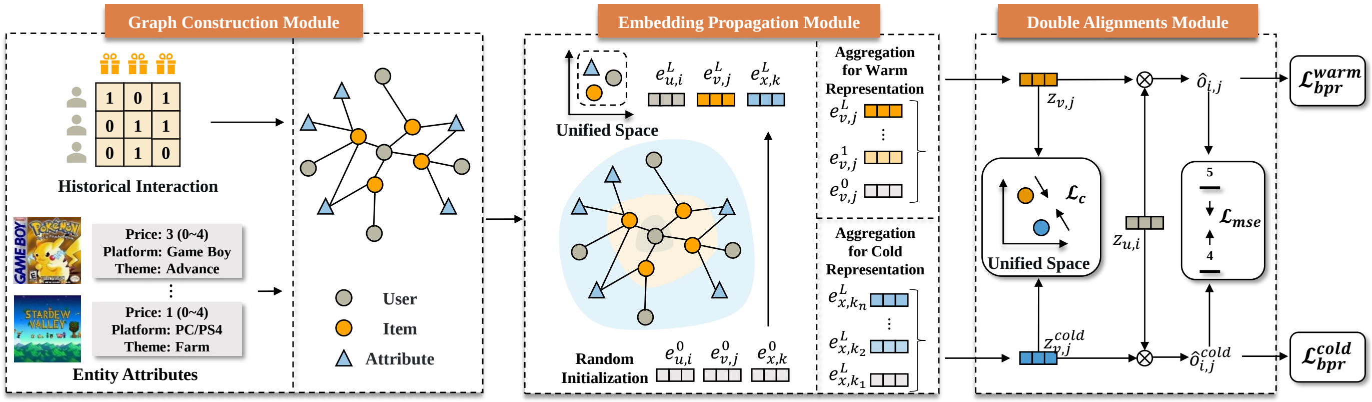


Fig. 2. An illustration of our proposed *URAC* method.

new records are often not used to retrain the model the first time; therefore the completely cold-start scenario will exist for a longer time than expected. We are trying to come up with an effective solution to this practical problem.

4 THE PROPOSED MODEL

We would introduce our proposed ***U**nified **R**epresentation Learning for Discrete **A**ttribute Enhanced Completely **C**old-start Recommendation* (URAC). For clarity, in the subsequent introduction, we use Task 1, i.e., recommend new items to old users, as an example to introduce the technical details of our model and the solutions of the other two tasks can be obtained by a simple analogy.

4.1 Overall Framework

As illustrated in Fig. 2, *URAC* consists of three main components, the graph construction module, the embedding propagation module, and the double alignments module.

- The graph construction module serves to construct heterogeneous graphs based on historical interactions and item attributes, with users, items, and attribute values represented as nodes.
- The embedding propagation module is responsible for learning the embeddings of each node in the graph constructed by the graph construction module and converting these embeddings into cold and warm representations.
- The double alignments module aligns collaborative signals and attributes information at different levels to better inject collaborative signals into cold representation. We propose a representation-level alignment based on mutual information maximization. Then, using both the cold and warm representations, we calculate preference scores and design a distance-based alignment module to align preference scores.

Through the collaboration of the above modules, we model and align the two types of information in a unified space to generate a cold representation with a richer collaborative signal. Finally, we optimize our model using a multi-task learning framework.

4.2 Graph Construction Module

Unlike previous approaches, we construct a heterogeneous graph by combining historical interactions and attributes. Fig. 3 provides an example of how we construct the graph in our model. As shown in the entity attributes part of the figure, the relationship between entities and attributes can be represented as an adjacency matrix. The historical interaction record is also reflected in an adjacency matrix shown in the middle part of the figure. By combining these two adjacency matrices, we can construct a heterogeneous graph where every entity and every potential value of attributes are represented as nodes. The construction of the heterogeneous graph allows for modeling in a unified representation space.

The heterogeneous graph could be formulated as: $\mathcal{G} =< U \cup I \cup X, A >$, where the adjacency matrix is defined as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^O & \hat{\mathbf{X}} \\ \hat{\mathbf{X}}^T & \mathbf{0}^{D\times D} \end{bmatrix}, \tag{2}$$

where $\hat{\mathbf{X}} = [\mathbf{X}, \mathbf{0}^{M\times D}]^T$. Herein, $\mathbf{X} \in \mathbb{R}^{N\times D}$ denotes the one-hot attribute matrix of items.

4.3 Embedding Propagation Module

Based on the heterogeneous graph constructed in the previous section, we employ a graph convolutional neural network to propagate node embeddings, capturing higher-order information and proximity between different types of nodes for improved node representation. Notably, the propagation occurs in a unified representation space, allowing us to model both collaborative signals and auxiliary discrete attributes in a unified representation space. Our model has no additional assumptions about the graph convolution method, and we use the simplest propagation method to verify our proposed *URAC*.

We employ $\mathbf{E}$ to denote the free embedding matrix in the graph encoder. Specifically, we use $\mathbf{E}_u \in \mathbf{O}^{M\times d}$, $\mathbf{E}_v \in \mathbb{R}^{N\times d}$ and $\mathbf{E}_x \in \mathbb{R}^{D\times d}$ denote the free embedding matrix of user, item and item attribute. $\mathbf{e}_{u,i}$, $\mathbf{e}_{v,j}$ and $\mathbf{e}_{x,k}$ are the i^{th}, j^{th} and k^{th} row in user, item, and item attribute matrix, and denote user i's, item j's and item attribute k's embedding. All embeddings are randomly initialized

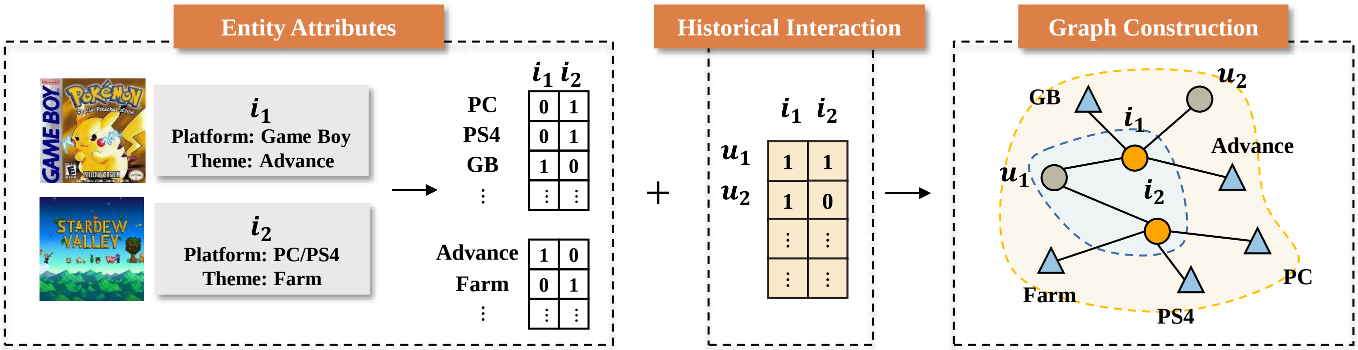


Fig. 3. An example of constructing a graph from historical interaction and attributes.

with Gaussian Distribution. Let $\mathbf{e}^l_{u,i}$, $\mathbf{e}^l_{v,j}$ and $\mathbf{e}^l_{x,k}$ refer to user i's embedding, item j's embedding and item attribute k's embedding at l^{th} layers, respectively. Layer 0 is free embedding after initialization. The propagation of different types of nodes in the graph is as follows:

$$\begin{aligned}
\mathbf{e}^{l+1}_{u,i} &= \mathbf{e}^l_{u,i} + \sum_{j\in A_u} \frac{\mathbf{e}^l_{v,j}}{|A_u|}, \\
\mathbf{e}^{l+1}_{v,j} &= \mathbf{e}^l_{v,j} + \sum_{i\in A_v} \frac{\mathbf{e}^l_{u,i}}{|A_v|} + \sum_{k\in A_v} \frac{\mathbf{e}^l_{x,k}}{|A_v|}, \\
\mathbf{e}^{l+1}_{x,k} &= \mathbf{e}^l_{x,k} + \sum_{j\in A_x} \frac{\mathbf{e}^l_{v,j}}{|A_x|}.
\end{aligned} \tag{3}$$

In the above formula, $A_u = \{j|o_{ij} = 1\}$ denote the item set that user i has interacted; $A_v = \{i|o_{ij} = 1\} \cup \{k|x_{jk} = 1\}$ denote the user set who has interacted with item j and the corresponding attributes of item j; $A_x = \{j|o_{jk} = 1\}$ denote the item which has attribute k. Free embeddings are iteratively propagated L times in the heterogeneous graphs using graph convolutional neural network to obtain the final embeddings $\mathbf{e}^L_{u,i}$, $\mathbf{e}^L_{v,j}$ and $\mathbf{e}^L_{x,k}$.

We propose a general model to handle the completely cold-start problem with the help of attributes. This generic model can also handle scenarios where there are multiple relations between attributes and entities, such as the knowledge graph based recommendation. In such scenarios, a minor tweak is made in that we use graph convolution considering relations, e.g., KGAT[43], R-GCN[44], instead of the propagation methods mentioned above.

After many iterations of the propagation process, we get the embeddings that capture higher-order information and proximity between different types of nodes. We then design a linear aggregation function to generate warm and cold representations by aggregating these embeddings. The warm representation, denoted as $\mathbf{z}$, contains both collaborative signals and attribute information, while the cold representation, denoted as $\mathbf{z}^{cold}$, only contains attribute information. The cold representations are later used to represent cold items to deal with the completely cold start problem. The module aggregates the attribute embeddings learned in the heterogeneous graph to produce these representations. Importantly, the aggregation is linear, which ensures that the warm and cold representations still are both in the same unified representation space. For each item v_j, we use $\mathbf{e}^L_{v,j}$ as the warm representation, and the embeddings corresponding to its attributes are aggregated to form the cold representation $\mathbf{z}^{cold}_{v,j}$. Final representations can be expressed in the following form:

$$\begin{aligned}
\mathbf{z}_{u,i} &= \mathbf{e}^L_{u,i}, \\
\mathbf{z}_{v,j} &= \mathbf{e}^L_{v,j}, \\
\mathbf{z}^{cold}_{v,j} &= \sum_{k=0}^{|\mathbf{x}_j|} \mathbb{I}(\mathbf{x}_{j,k} = 1) \cdot \mathbf{e}^L_{x,k},
\end{aligned} \tag{4}$$

where $\mathbf{x}_{j,k}$ represents the k^{th} value of item j's one-hot attribute and $\mathbb{I}(\cdot)$ represent indicator function, it means when the condition in $(\cdot)$ is satisfied, the outcome of function is 1, otherwise 0.

4.4 Double Alignments Module

Previous studies have focused on applying alignment between entity embeddings and attribute embeddings to inject collaborative signals into attribute embeddings. However, we argue that this is insufficient. On the one hand, only narrowing the geometric-based distance of representations can not guarantee that cold entities' and warm entities' semantic characteristics are consistent. On the other hand, they ignore a different but equally important stage of the recommendation process, i.e., the calculation of the preference score. To improve them, we impose alignments both at the representation-level and preference-level. These double alignments make the performance of our model less dependent on the quality of the attribute information and better infuse collaborative signals into the cold representation.

4.4.1 Representation-level alignment

Cold representations result in poor performance due to a lack of collaborative signals. We design a representation-level alignment that refers to the alignment between the warm and cold representations to accomplish the transfer of the collaborative signal from the warm to the cold representation. We want the semantics of the cold representation to be consistent with the warm representation. To achieve the alignment, we maximize the Mutual Information (MI) between collaborative representation and attribute representation, namely $\mathrm{MI}(\mathbf{Z}_v, \mathbf{Z}^{cold}_v)$. Mutual information measures how much knowing the value of one random variable (or set of variables) informs about another. It captures non-linear statistical dependencies between variables, and thus can act as a measure of true dependence[45]. The maximization of $\mathrm{MI}(\mathbf{Z}_v, \mathbf{Z}_v^{cold})$ effectively aligns the representations of cold and warm items, ensuring that the shared semantic information is captured consistently.

Since the precise value of MI is difficult to compute, a common practice[36] is to utilize neural estimators to maximize the lower bound of MI instead: $\mathrm{MI}(\mathbf{Z}_v, \mathbf{Z}_v^{cold}) \geq log(N) - \mathcal{L}_c$. N is a constant and $\mathcal{L}_c$ is the contrastive learning[36] loss function, which is defined as:

$$\mathcal{L}_c = \sum_{j \in \mathcal{B}} -\log \frac{\exp(\mathbf{z}_{v,j} \cdot \mathbf{z}_{v,j}^{cold})/\tau}{\sum_{p \in \mathcal{B}} \exp(\mathbf{z}_{v,j} \cdot \mathbf{z}_{v,p}^{cold})/\tau}, \tag{5}$$

where $\mathcal{B}$ denote a batch items, τ is the temperature hyper-parameter of softmax. For item j, $\mathbf{z}_{v,j}$ and $\mathbf{z}_{v,j}^{cold}$ denote the corresponding warm and cold representations with L_2 normalization, the same as item p. This objective encourages consistency of warm and cold representations for each item. The contrastive loss encourages the model to bring collaborative representation $\mathbf{z}_{v,j}$ and attribute representation $\mathbf{z}_{v,j}^{cold}$ of the same item j closer and push different items' two representations apart. This process ensures that the shared semantic information is emphasized and preserved.

4.4.2 Preference-level alignment

After obtaining high-quality representations, recommendation models combine user and item representations through some operation (e.g., inner product) to generate a preference score that indicates the degree of preference. This preference score is often the direct basis for the final recommendation, but the semantic agreement between preference scores calculated by cold and warm representations is often ignored in the existing cold-start modelS. We point out that directly aligning the preference scores can better integrate useful collaborative signals into the cold representation. To achieve this, we calculate multiple preference scores using both the cold and warm item representations and the same group of user representations. These preference scores are then used to create two vectors, which we align at the preference-level using a distance-based alignment module, Mean Square Error (MSE). Our goal is to make each pair of specific preference scores in the two vectors as similar as possible.

For warm items, we only use the warm representation to calculate the preference score, which can be calculated with the following function:

$$\hat{o}_{ij} = \mathbf{z}_{u,i}(\mathbf{z}_{v,j})^T. \tag{6}$$

For cold items, we use the cold representation instead of the warm one:

$$\hat{o}_{ij}^{cold} = \mathbf{z}_{u,i}(\mathbf{z}_{v,j}^{cold})^T. \tag{7}$$

We want the preference scores of warm and cold representations of an item to be similar after interacting with the same user, and this similarity is reflected in the calculation with both the interacted user and the non-interacted user. This constraint is achieved by MSE:

$$\mathcal{L}_{mse} = \|\mathbf{Z}_u(\mathbf{Z}_v)^T - \mathbf{Z}_u(\mathbf{Z}_v^{cold})^T\|^2. \tag{8}$$

4.5 Model Optimization

In this section, we give the final optimal objective of our model. We use mixed losses rather than a single loss applicable to warm items, which is commonly used in other works. The warm representation is used to calculate the regular loss, and the cold representation is used to calculate the loss of cold-start. The two are mixed as the optimization objective of the model. This design is inspired by the previous works' idea of "dropping" or random selection training, aiming to simulate the cold-start scenario in the training phase. The reason why we use mixed loss instead of random training is that when random training is carried out, each item will only appear in one scene (for each item, only its warm representation or only its cold representation is involved in training), while in the method of mixed loss, each item will encode information in two scenes every time it participates in training.

The most often used optimization approach for recommender systems based on implicit feedback is BPR-based pair-wise ranking [4] and in this paper we also utilize it to calculate regular loss:

$$\mathcal{L}_{BPR}^{warm} = \sum_{(u,i,j) \in O_{tri}} -\log \sigma(\hat{o}_{ui} - \hat{o}_{uj}). \tag{9}$$

where $\sigma(\cdot)$ is a sigmoid activation function, $O_{tri} = \{(u,i,j)|o_{ui} = 1, o_{uj} = 0\}$ denotes the pairwise training data for user u.

In recommendations involving cold items, we can only use cold representations. Therefore, the cold-start BPR objective is applied to the model to directly capture the interaction in which the cold representation participates:

$$\mathcal{L}_{BPR}^{cold} = \sum_{(u,i,j) \in O_{tri}} -\log \sigma(\hat{o}_{ui}^{cold} - \hat{o}_{uj}^{cold}). \tag{10}$$

Our model should be able to make full use of the collaborative signals in the warm representation, and at the same time make sure that the cold representation of items is meaningful when calculating preference scores. Therefore, the recommended optimization objective form is as follows:

$$\mathcal{L}_{BPR} = \alpha \mathcal{L}_{BPR}^{warm} + (1-\alpha)\mathcal{L}_{BPR}^{cold}, \tag{11}$$

where α is the weight of different BPR loss.

The final optimization objective can be stated as follows:

$$\mathcal{L}_{attr} = \mathcal{L}_{BPR} + \beta \mathcal{L}_c + \gamma \mathcal{L}_{mse} + \|\mathbf{\Theta}\|^2, \tag{12}$$

where $\mathbf{\Theta} = \{\mathbf{E}_u, \mathbf{E}_v, \mathbf{E}_x\}$ represents all free embeddings in the model, β and γ are hyper-parameters to balance the weight of the three losses from different stages.

4.6 Recommendation for New Entities

According to the method presented in section 4.2-4.5, we can get a well-trained model for the completely cold-start problem. When a cold item appears, our model work in an inductive way. We can directly make recommendations without retraining, which is more practical and reasonable. The detail of new entities recommendation is shown in Fig. 4. Suppose a cold item v_{cold} appears, and $\mathbf{x}_{cold}$ is its one-hot attributes. According to the attributes of the cold item, we select corresponding nodes in the heterogeneous graph

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3387276

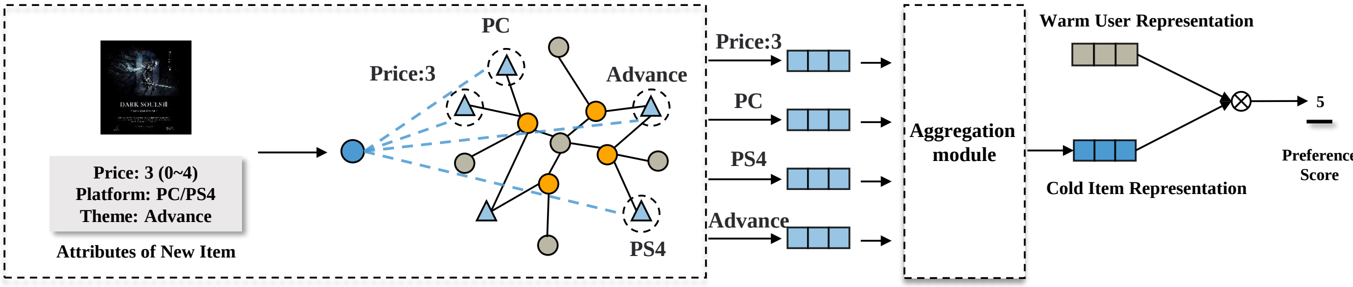


Fig. 4. An example of recommendation for a new item.

and input embeddings of these nodes to generate the cold representation $\mathbf{z}_{cold}$ as follows:

$$\mathbf{z}_{cold} = \sum_{k=0}^{|\mathbf{x}_{cold}|} \mathbb{I}(\mathbf{x}_{cold,k} = 1) \cdot \mathbf{e}_{x,k}^{L}, \tag{13}$$

where $\mathbf{x}_{cold,k}$ represents the k^{th} value of cold item $\mathbf{z}_{cold}$'s one-hot attribute. As we mentioned, attributes are shared by cold entities and warm entities. All attribute node embeddings $\mathbf{e}_{x,k}^{L}$ that we use are trained and informative. Finally, the cold representation $\mathbf{z}_{cold}$ is regarded as the representation of the cold item and is used to calculate the preference score with warm user representation.

5 EXPERIMENTS

In this section, we conduct extensive experiments on six datasets (three datasets with monotonic relations between entities and attributes and three datasets with multiple relations between entities and attributes) to verify the effectiveness of our proposed *URAC* model. Overall, we aim to answer the following questions:

- **RQ1**: How does our model perform compared with state-of-the-art completely cold-start recommendation methods?
- **RQ2**: How do different designs (i.e., unified space, double alignments, mixed loss) affect our model?
- **RQ3**: What are the advantages of using contrastive learning to align representation?

5.1 Experimental Settings

5.1.1 Datasets Description

To evaluate the effectiveness of our proposed model, we conduct experiments on datasets with different characteristics, i.e., have monotonic relations between entities and attributes and have multiple relations. For datasets with monotonic relations, we select three real-world datasets: Yelp [11], XING [46], and Amazon-Video Games [47]. For datasets with multiple relations, we selected three widely used benchmark datasets for our experiments, which are publicly accessible and vary in terms of domain, size, and sparsity: Yelp2018, Last-FM, Book-Crossing[43; 48]. It should be noted that in order to better compare the work of corresponding research lines, the Yelp2018 dataset here and the previous Yelp dataset come from different open sources, and the attribute on the two sides is different.

We manually set up cold-start users and items [11]. Specifically for Task 1, we randomly select 30% items from the original datasets and then remove their connections with all users with whom they have interacted. Similarly, for Task 2, we randomly select 30% users from the original datasets and then remove their connections with all items with which they have interacted. For Task 3, based on the previous processing, we collated interaction records belonging to both the new user and the new item as the test set. In addition, during model training, we divided 10% of warm entities or cold users items) as the valuation set according to the needs. The statistics of all datasets after pre-processing are summarized in Appendix A.

5.1.2 Evaluation Metrics and Baselines.

We select two metrics that are widely used in personalized recommender systems to evaluate our model: Hit Radio (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K). HR@K measures the number of successfully predicted items in the top-K ranking list that the user likes in the test data. NDCG@K further considers the hit positions of the items.

To verify the effectiveness of our framework, we categorize existing approaches according to their problem-solving approaches and pick novel approaches representative of each category as the baseline: (1) content-based methods, which only model preferences or characteristics of entities based on their side information to make recommendations, including **KNN** [8], **xDeepFM** [21] and **CDL** [23]. (2) Robustness-based methods, which strategically discard collaborative information during the training phase to enhance the robustness of the model by simulating cold start scenarios, including **DropoutNet** [9] and **MTPR** [20]. (3) Regularization-based methods, which explicitly align between collaborative representation and content representation by adding regular terms to the optimization target, including **Heater** [12], **CLCRec** [14] and **CCFCRec** [24]. (4) Graph-based methods, such as using graphs to model the relationship between collaborative signals and side information, including **PGD** [11]. (5) Generation-based methods, which generate pseudo-collaborative representations for new items based on side information, including **GAR** [27].

It is important to note that datasets with relations between attributes and entities are knowledge graph datasets. As far as we know, there is still no work considering the completely cold-start problem based on such data, and most existing models cannot be applied to such data. This is

TABLE 1
Comparison of different completely cold-start recommendation models on Yelp. '-' represents an unavailable result.

Model	Yelp(Task 1)				Yelp(Task 2)				Yelp(Task 3)			
	HR		NDCG		HR		NDCG		HR		NDCG	
	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20
KNN	0.0159	0.0278	0.0099	0.0137	0.0181	0.0310	0.0153	0.0207	-	-	-	-
xDeepFM	0.0202	0.0349	0.0128	0.0174	0.0198	0.0323	0.0161	0.0215	0.0131	0.0244	0.0075	0.0112
CDL	0.0196	0.0341	0.0128	0.0175	0.0193	0.0326	0.0160	0.0216	0.0127	0.0200	0.0081	0.0105
DropoutNet	0.0173	0.0282	0.0105	0.0141	0.0201	0.0328	0.0168	0.0221	0.0114	0.0214	0.0069	0.0100
MTPR	0.0203	0.0332	0.0125	0.0171	0.0200	0.0311	0.0168	0.0224	0.0118	0.0222	0.0072	0.0110
Heater	0.0244	0.0418	0.0150	0.0206	0.0206	0.0337	0.0173	0.0227	0.0123	0.0244	0.0073	0.0113
CLCRec	0.0246	0.0412	0.0156	0.0211	0.0202	0.0333	0.0169	0.0225	0.0122	0.0250	0.0070	0.0113
GAR	0.0233	0.0402	0.0146	0.0208	0.0212	0.0343	0.0171	0.0227	-	-	-	-
CCFCRec	0.0255	0.0432	0.0160	0.0220	0.0200	0.0310	0.0165	0.0220	-	-	-	-
PGD	0.0272	0.0471	0.0166	0.0231	0.0208	0.0340	0.0177	0.0232	0.0144	0.0259	0.0087	0.0124
URAC	**0.0285**	**0.0482**	**0.0179**	**0.0240**	**0.0219**	**0.0348**	**0.0184**	**0.0235**	**0.0156**	**0.0266**	**0.0088**	**0.0126**

because most work converts attributes into a vector input model, but the amount of attributes in knowledge graph datasets is too huge, and each attribute may possibly be connected to any entity. Thus the vector is too sparse and the dimension of it is too high to input into the model. So we mainly compare variations of our model on these datasets. We select the following methods as baseline: **Random**: We randomly select items as candidates to users in the test stage. **KGAT**[43]: It applies an attentive neighborhood aggregation mechanism on a holistic graph, which combines the knowledge graph with the user-item graph, to generate user and item representations. ***URAC*-UR**: We model different information in a unified representation space and only impose representation-level alignment. ***URAC*-UP**: We only impose preference-level alignment.

5.1.3 Parameter Settings

We implement our *URAC* and all baselines with Pytorch framework. We fix the dimension as 64 whether CF embedding or attribute embedding. A Gaussian distribution with a mean of 0 and a variance of 0.01 is employed to initialize the embedding matrices. The batch size is set to 2048. The number of iterations of embedding propagation is searched in $\{1, 2, 3, 4\}$. During training, we employ Adam [49] as the optimizer and set the learning rate at 0.001, the early stop strategy is employed to avoid overfitting. For contrastive learning, we carefully turn the temperature and find *URAC* performs the best performance when temperature $\tau = 0.07$. We analyze the results of three hyper-parameters on five tasks in $\{0.0001, \cdots, 1000\}$, respectively.

5.2 Overall Performances (RQ1)

The overall results of the baseline comparison of our model on datasets with monotonic relations are reported in Table 1 and Tabel 2. The best performance is in bold, and the strongest baselines are underlined. We find that:

- *URAC* consistently outperforms all baselines across five tasks regarding all measures. Compared to the strongest baseline, our models improved by up to 7.7%, 5.3%, 8.2%, 6.8%, and 33%, respectively, which demonstrates the effectiveness of our proposed model. Besides, we find that *URAC* achieves higher improvements on the small-length ranking task, e.g, 7.72% relative improvement on NDCG@10 and 4.08% on NDCG@20 compared to PGD on the Yelp(Task 1), which is more suitable to real-world recommendation scenarios. This improvement is the result of the rational design of each module, and the main reasons are the modeling of different information in the unified space and the combination of two-level alignments. Subsequent ablation experiments have verified our views.
- CLCRec does not perform well in our experiments, we guess that the reason is that it was originally designed for multimedia recommendation, and performance depends on the quality of side information representation. In contrast, the representation of attributes generated by MLP is not rich in information. As such, CLCRec performs better on Yelp(Task 1), which has denser attributes than other tasks.
- PGD is the strongest baseline in all tasks. The reason is that it distills the preference matrix information, that is, unconsciously imposes preference-level constraints, so as to obtain better performance. Nevertheless, *URAC* also consistently outperforms PGD, which shows the effectiveness of learning in a unified space instead of two graph encoders.

Table 3 shows the overall result of our model on datasets with multiple relations between entities and attributes. The best performance is in bold, and the strongest baselines are underlined. We have the following observations:

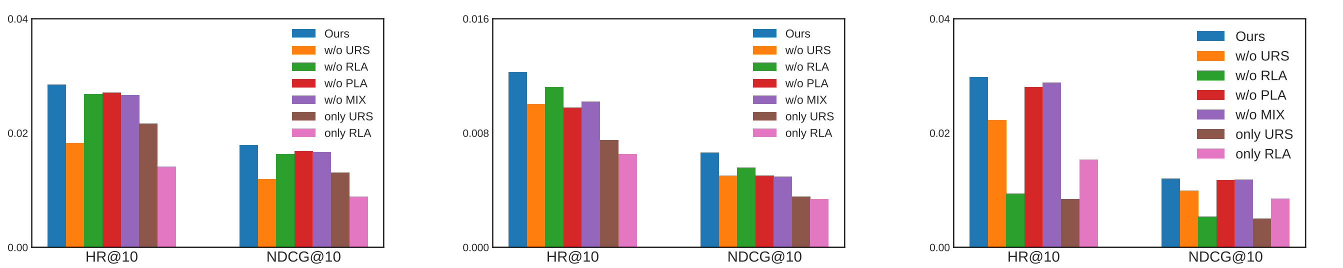
- Our proposed *URAC* still consistently outperforms all baselines under different settings. Specifically, *URAC* improves the strongest baseline $w.r.t$ NDCG@10 by 29.66%, 4.09% and 21.86% on Yelp2018, Last-FM and Amazon-Book dataset, respectively. Extensive empirical studies have shown that the proposed *URAC* remains effective even after taking into account multiple relations between entities and attributes.
- Alignment between warm and cold representations is necessary for the completely cold-start recommendation. To be specific, our model uses KGAT as a method of embedding propagation on datasets with multiple relations, so KGAT can be viewed as only guaranteeing a uniform space, but not aligning

TABLE 2
Comparison of different completely cold-start recommendation models on Xing and Amazon-Video Games.

Model	XING(Task 2)						Amazon-Video Games(Task 1)					
	HR			NDCG			HR			NDCG		
	@10	@20	@50	@10	@20	@50	@10	@20	@50	@10	@20	@50
KNN	0.0030	0.0060	0.0125	0.0016	0.0024	0.0039	0.0013	0.0019	0.0084	0.0008	0.0010	0.0024
xDeepFM	0.0079	0.0148	0.0275	0.0042	0.0062	0.0090	0.0183	0.0249	0.0390	0.0084	0.0095	0.0130
CDL	0.0077	0.0153	0.0282	0.0042	0.0063	0.0024	0.0185	0.0250	0.0419	0.0083	0.0095	0.0144
DropoutNet	0.0067	0.0128	0.0235	0.0033	0.0050	0.0074	0.0114	0.0161	0.0288	0.0054	0.0067	0.0099
MTPR	0.0067	0.0133	0.0240	0.0035	0.0055	0.0088	0.0210	0.0271	0.0420	0.0064	0.0087	0.0119
Heater	0.0069	0.0152	0.0272	0.0034	0.0057	0.0085	0.0203	0.0266	0.0410	0.0093	0.0103	0.0133
CLCRec	0.0068	0.0130	0.0252	0.0033	0.0051	0.0080	0.0121	0.0177	0.0302	0.0060	0.0083	0.0105
GAR	0.0086	0.0202	0.0306	0.0047	0.0068	0.0099	0.0213	0.0286	0.0430	0.0098	0.0113	0.0148
CCFCRec	0.0101	0.0210	0.0342	0.0053	0.0072	0.0111	0.0221	0.0297	0.0460	0.0100	0.0116	0.0167
PGD	0.0115	0.0220	0.0406	0.0065	0.0092	0.0133	0.0224	0.0295	0.0451	0.0101	0.0116	0.0160
URAC	**0.0123**	**0.0226**	**0.0421**	**0.0066**	**0.0096**	**0.0141**	**0.0298**	**0.0351**	**0.0522**	**0.0120**	**0.0133**	**0.0181**

TABLE 3
Comparison of different completely cold-start recommendation models on Yelp2018, Last-FM, and Amazon-Book.

Model	Yelp2018				Last-FM				Book			
	HR		NDCG		HR		NDCG		HR		NDCG	
	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20
Random	0.0002	0.0003	0.0001	0.0002	0.0003	0.0004	0.0002	0.0003	0.0006	0.0002	0.0035	0.0006
PGD	0.0046	0.0070	0.0030	0.0039	0.0202	0.0305	0.0133	0.0170	0.0035	0.0063	0.0016	0.0024
KGAT	0.0006	0.0009	0.0004	0.0005	0.0008	0.0009	0.0006	0.0006	0.0032	0.0045	0.0017	0.0021
URAC-UR	0.0032	0.0056	0.0020	0.0029	0.0350	0.0441	0.0306	0.0333	0.0035	0.0051	0.0018	0.0023
URAC-UP	0.0030	0.0062	0.0020	0.0032	0.0315	0.0462	0.0257	0.0308	0.0039	0.0063	0.0018	0.0024
URAC	**0.0050**	**0.0081**	**0.0031**	**0.0041**	**0.0372**	**0.0488**	**0.0318**	**0.0354**	**0.0043**	**0.0083**	**0.0022**	**0.0031**



(a) Ablation Study on Yelp. (b) Ablation Study on Xing. (c) Ablation Study on Amazon-Video Games.

Fig. 5. The impact of each component on the model on the datasets of category attribute.

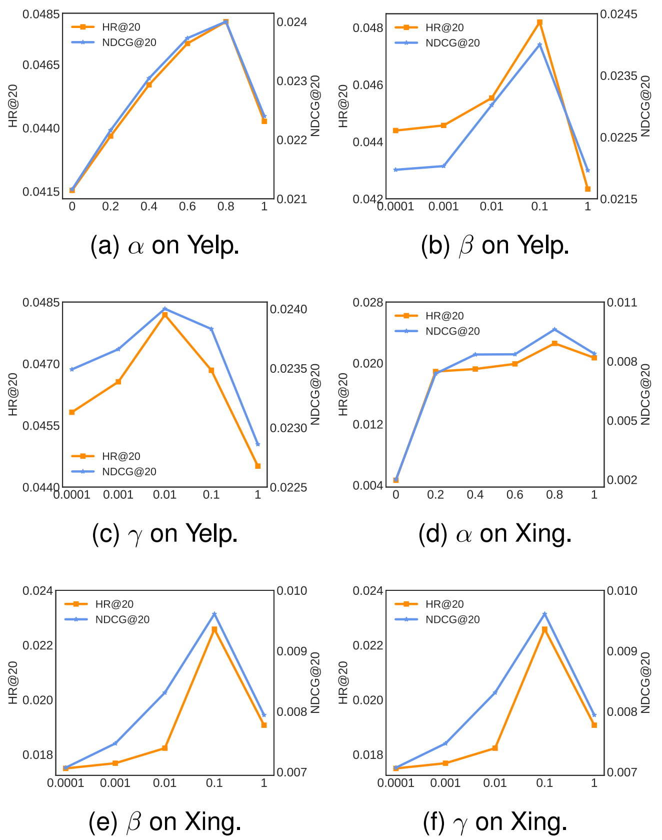
different representations. *URAC*-UR improves KGAT $w.r.t$ NDCG@10 by 451.35% and *URAC*-UP improves KGAT $w.r.t$ NDCG@10 by 448.65% on the Yelp2018 dataset, which shows alignment at any stage can significantly improve model performance.

5.3 Analysis of *URAC* (RQ2)

5.3.1 Ablation Study

In this section, we conduct an ablation study to exploit the effectiveness of each component in *URAC* . We implement some variants of *URAC* and compare performances with *URAC* . We use URS to denote unified representation space, RLA to denote representation-level alignment, PLA to denote preference-level alignment, and MIX to denote the mixed BPR objective. We remove different components on the complete model in turn. In order to verify the influence of the unified representation space, we use an additional LightGCN [50] model to generate warm representations, *URAC* only provides cold representations. For the other ablation experiments, we simply set the corresponding coefficient to 0. In addition, we do separate tests for our highlight URS and RLA modules. only URS and only RLA mean that we use only a single module to test model performance in a cold start scenario. We conduct ablation experiments on datasets with monotonic relation and show the results in Fig. 5. From the figure, we can observe that among these components, URS consistently has a great impact on the model in different datasets, which once again verifies the significance of modeling in the unified representation space. Specifically, damage to the URS condition reduces the performance of our model $w.r.t$ NDCG@10 by 36.01%, 18.17%, and 25.17% on yelp(Task 1), XING(Task 2), and Amazon-Video Games(Task 1), respectively. Besides, removing any other components also can have a significant impact on the final result. This shows that every module in our model is reasonable, and the good performance of *URAC* is the result of the combined action of all modules.

5.3.2 Parameter Sensitivities Analysis.



(a) α on Yelp. (b) β on Yelp. (c) γ on Yelp. (d) α on Xing. (e) β on Xing. (f) γ on Xing.

Fig. 6. The effect of different hyper-parameters on the model. Each row shows the result of a data set, and each column represents a hyper-parameter. If the hyper-parameter α is 0, it indicates that only the cold-start BPR objective is used. If the hyper-parameter α is 1, it indicates that only the common BPR objective is used.

There are three hyper-parameters in *URAC* : α, β, and γ, which are used to balance each part of the optimization objective. Fig. 6 shows how they affect the model on different datasets. For all three hyper-parameters, the model performance increases first and then decreases with the increase of the hyper-parameters. The results show that the three hyper-parameters have important impressions on the model and need to be adjusted appropriately. It is worth noting that even if only a small proportion of cold BPR objective is introduced, the model will improve greatly, but only using cold BPR objective will reduce the performance. For example, as reflected in Fig. 6a and 6d, the model achieves the best performance when $\alpha = 0.8$, that is, the mixed objective is composed of 80% regular BPR objective and 20% cold BPR objective; but when $\alpha = 0$, that is, only cold BPR objective is used as the final optimization objective, the model has the worst effect. This may be because we need a regular BPR objective to ensure adequate collaborative signals in the warm representation, which is a prerequisite for double alignments to be effective.

TABLE 4
Comparison of the results of different alignments at the representation-level.

Alignment	yelp(Task 1)	Amazon_Video Games(Task 1)	XING(Task 2)
	NDCG@10	NDCG@10	NDCG@10
w/o	0.01628	0.00538	0.00560
MSE	0.01570	0.01002	0.00556
SSE	0.01624	0.01012	0.00561
CS	0.01632	0.00999	0.00545
InfoNCE	0.01787	0.01201	0.00664

5.4 Investigation of Representation-level Alignment (RQ3)

In this section, we mainly answer the question, "What are the advantages of using contrastive learning to alignment different information at the representational-level?". Representation-level alignments are more intuitive and effective, so it has been considered in most of the previous work. Different from previous work, we use contrastive learning to maximize the mutual information between different representations, so as to narrow the semantic differences between them. We will discuss how useful contrastive learning and give an explanation of the unique role of contrastive learning.

Table 4 shows the results using different alignments at the representation-level, herein, w/o means no alignments, MSE means using mean square error, SSE means using summation square error, CS means using cosine similarity, and InfoNCE represents our setup. The results showed that InfoNCE played a great role and could not be replaced by MSE. Simply bringing different representations closer together may even degrade model performance.

6 Conclusion and Limitation

In this paper, we pointed out the limitations of combining discrete attributes and collaborative signals in current work to solve the completely cold-start problem. Aiming at the discrete attribute which is not sufficiently explored and difficult to leverage, we proposed a comprehensive model to generate warm and cold representations and to carry out multiple alignments between them to transfer collaborative signals from warm to cold representations. We chose two different discrete side information data to verify our model. A large number of experimental results show that our model is more effective. In future work, we plan to further explore the completely cold-start problem in an attempt to address some of the limitations of the approach presented in this paper. Our work focuses on discrete attributes and designs a heterogeneous graph construction scheme for the data with more discrete attributes. While efficiently utilizing discrete attribute information, our method requires additional processing when using other data. We believe that appropriate data pre-processing methods can enhance the application scope of our model, such as continuous attribute discretization. In addition, how to extend to other types of side information more efficiently remains to be further explored.

REFERENCES

[1] J. Luo, M. He, W. Pan, and Z. Ming, "Bgnn: Behavior-aware graph neural network for heterogeneous session-based recommendation," *Front. Comput. Sci.*, 2023.

[2] C. Wu, D. Lian, Y. Ge, Z. Zhu, and E. Chen, "Influence-driven data poisoning for robust recommender systems," *TPAMI*, 2023.

[3] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, "A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation," *TKDE*, 2022.

[4] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *UAI*, 2009.

[5] M. Nilashi, K. Bagherifard, O. Ibrahim, H. Alizadeh, L. A. Nojeem, and N. Roozegar, "Collaborative filtering recommender systems," *Research Journal of Applied Sciences, Engineering and Technology*, 2013.

[6] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," *WWW*, 2017.

[7] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," *SIGIR*, 2019.

[8] S. Sedhain, S. Sanner, D. Braziunas, L. Xie, and J. Christensen, "Social collaborative filtering for cold-start recommendations," in *RecSys*, 2014.

[9] M. Volkovs, G. Yu, and T. Poutanen, "Dropoutnet: Addressing cold start in recommender systems," in *NeurIPS*, 2017.

[10] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, "Learning attribute-to-feature mappings for cold-start recommendations," *ICDM*, 2010.

[11] S. Wang, K. Zhang, L. Wu, H. Ma, R. Hong, and M. Wang, "Privileged graph distillation for cold start recommendation," *SIGIR*, 2021.

[12] Z. Zhu, S. Sefati, P. Saadatpanah, and J. Caverlee, "Recommendation for new users and new items via randomized training and mixture-of-experts transformation," *SIGIR*, 2020.

[13] C. Sun, H. Liu, M. Liu, Z. Ren, T. Gan, and L. Nie, "Lara: Attribute-to-feature adversarial learning for new-item recommendation," *WSDM*, 2020.

[14] Wei, X. Wang, Q. Li, L. Nie, Y. Li, X. Li, and T.-S. Chua, "Contrastive learning for cold-start recommendation," *MM*, 2021.

[15] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, 2009.

[16] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *ICML*, 2008.

[17] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," *AAAI*, 2020.

[18] L. Wu, Y. Yang, L. Chen, D. Lian, R. Hong, and M. Wang, "Learning to transfer graph embeddings for inductive graph based recommendation," *SIGIR*, 2020.

[19] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, 2017.

[20] X. Du, X. Wang, X. He, Z. Li, J. Tang, and T.-S. Chua, "How to learn item representation for cold-start multimedia recommendation?" *MM*, 2020.

[21] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. zhong Sun, "xdeepfm: Combining explicit and implicit feature interactions for recommender systems," *KDD*, 2018.

[22] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," 2013.

[23] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," *KDD*, 2015.

[24] Z. Zhou, L. Zhang, and N. Yang, "Contrastive collaborative filtering for cold-start item recommendation," *WWW*, 2023.

[25] X. Xu, C. Yang, Q. Yu, Z. Fang, J. Wang, C. Fan, Y. He, C. Peng, Z. Lin, and J. Shao, "Alleviating cold-start problem in ctr prediction with a variational embedding learning framework," *WWW*, 2022.

[26] H. Bai, M. Hou, L. Wu, Y. Yang, K. Zhang, R. Hong, and M. Wang, "Gorec: A generative cold-start recommendation framework," *MM*, 2023.

[27] H. Chen, Z. Wang, F. Huang, X. Huang, Y. Xu, Y. Lin, P. He, and Z. Li, "Generative adversarial framework for cold-start item recommendation," *SIGIR*.

[28] Y. Lu, Y. Fang, and C. Shi, "Meta-learning on heterogeneous information networks for cold-start recommendation," *KDD*, 2020.

[29] M. Dong, F. Yuan, L. Yao, X. Xu, and L. Zhu, "Mamo: Memory-augmented meta-optimization for cold-start recommendation," *KDD*, 2020.

[30] Z. Wu and X. Zhou, "M2eu: Meta learning for cold-start recommendation via enhancing user preference estimation," *SIGIR*, 2023.

[31] X. Lin, J. Wu, C. Zhou, S. Pan, Y. Cao, and B. Wang, "Task-adaptive neural process for user cold-start recommendation," *WWW*, 2021.

[32] Z. Chi, L. Dong, F. Wei, N. Yang, S. Singhal, W. Wang, X. Song, X.-L. Mao, H. Huang, and M. Zhou, "Infoxlm: An information-theoretic framework for cross-lingual language model pre-training," in *NAACL*, 2021.

[33] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *NeurIPS*, 2020.

[34] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," *ICML*, 2020.

[35] C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka, "Debiased contrastive learning," *NeurIPS*, 2020.

[36] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *ArXiv*, 2018.

[37] R. Linsker, "Self-organization in a perceptual network," *Computer*, 1988.

[38] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," *SIGIR*, 2021.

[39] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, "Improv-

ing graph collaborative filtering with neighborhood-enriched contrastive learning," *WWW*, 2022.

[40] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, B. Ding, and B. Cui, "Contrastive learning for sequential recommendation," *ICDE*, 2022.

[41] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *WSDM*, 2022.

[42] Z. Liu, Y. Chen, J. Li, M. Luo, S. Y. Philip, and C. Xiong, "Self-supervised learning for sequential recommendation with model augmentation," 2021.

[43] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," *KDD*, 2019.

[44] M. Schlichtkrull, T. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Extended Semantic Web Conference*, 2017.

[45] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 531–540. [Online]. Available: https://proceedings.mlr.press/v80/belghazi18a.html

[46] F. Abel, Y. Deldjoo, M. Elahi, and D. Kohlsdorf, "Recsys challenge 2017: Offline and online evaluation," *RecSys*, 2017.

[47] R. He and J. McAuley, "Vbpr: Visual bayesian personalized ranking from implicit feedback," *AAAI*, 2016.

[48] D. Zou, W. Wei, Z. J. Wang, X. Mao, F. Zhu, R. Fang, and D. Chen, "Improving knowledge-aware recommendation with multi-level interactive contrastive learning," *CIKM*, 2022.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[50] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," *SIGIR*, 2020.

Haoyue Bai is currently working toward the master's degree at the Hefei University of Technology, Hefei, China. His research interests include data mining and recommender systems.

Min Hou Min Hou is currently pursuing her Ph.D. at the School of Data Science, University of Science and Technology of China, Hefei, China. Her current research interests include data mining, recommender systems and AI for finance. She has published papers in refereed conference proceedings, such as WWW, AAAI, IJCAI, CIKM, ICDM, WSDM.

Le Wu received the Ph.D. degree from the University of Science and Technology of China (USTC). She is currently a professor with the Hefei University of Technology (HFUT), China. Her general area of research interests include data mining, recommender systems, and social network analysis. She has published more than 30 papers in referred journals and conferences. She is the recipient of the Best of SDM 2015 Award, and the Distinguished Dissertation Award from China Association for Artificial Intelligence (CAAI) 2017. She is a member of the IEEE.

Yonghui Yang is currently working towards a Ph.D. degree at Hefei University of Technology, China. He received his master's degree from the same University in 2021. He has published several papers in leading conferences, including SIGIR and IJCAI. His research interests include graph learning and recommender systems.

Kun Zhang received the Ph.D. degree in computer science and technology from University of Science and Technology of China, Hefei, China, in 2019. He is currently a faculty member with the Hefei University of Technology (HFUT), China. His research interests include Natural Language Understanding, Recommender System. He has published several papers in refereed journals and conferences, such as the IEEE Transactions on Systems, Man, and Cybernetics: Systems, the ACM Transactions on Knowledge Discovery from Data, AAAI, KDD, ACL, SIGIR, WWW, ICDM. He received the KDD 2018 Best Student Paper Award.

Richang Hong (M'12) received the Ph.D. degree from the University of Science and Technology of China (USTC), in 2008. He is currently a professor with the Hefei University of Technology, He has co-authored more than 60 publications in the areas of his research interests, which include multimedia question answering, video content analysis, and pattern recognition. He is a member of the Association for Computing Machinery. He was a recipient of the best paper award in ACM Multimedia 2010. He is a member of the IEEE.

Meng Wang received the BE and Ph.D. degrees in the special class for the gifted young from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China, in 2003 and 2008, respectively. He is a professor with the Hefei University of Technology, China. His current research interests include multimedia content analysis, computer vision, and pattern recognition. He has authored more than 200 book chapters, and journal and conference papers in these areas. He is the recipient of the ACM SIGMM Rising Star Award 2014. He is an associate editor of the IEEE Transactions on Knowledge and Data Engineering (TKDE), the IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), the IEEE Transactions on Multimedia (TMM), and the IEEE Transactions on Neural Networks and Learning Systems (TNNLS). He is a fellow of the IEEE.