

# Modeling Two-Way Selection Preference for Person-Job Fit

Chen Yang<sup>†</sup>  
2021100973@ruc.edu.cn  
Gaoling School of Artificial  
Intelligence, Renmin University of  
China  
Beijing, China

Yupeng Hou<sup>†</sup>  
houyupeng@ruc.edu.cn  
Gaoling School of Artificial  
Intelligence, Renmin University of  
China  
Beijing, China

Yang Song  
songyang@kanzhun.com  
BOSS Zhipin  
Beijing, China

Tao Zhang  
kylen.zhang@kanzhun.com  
BOSS Zhipin  
Beijing, China

Ji-Rong Wen<sup>†</sup>  
jrwen@ruc.edu.cn  
Gaoling School of Artificial  
Intelligence, Renmin University of  
China  
Beijing, China

Wayne Xin Zhao<sup>†‡</sup> ✉  
batmanfly@gmail.com  
Gaoling School of Artificial  
Intelligence, Renmin University of  
China  
Beijing, China

## ABSTRACT

Person-job fit is the core technique of online recruitment platforms, which can improve the efficiency of recruitment by accurately matching the job positions with the job seekers. Existing works mainly focus on modeling the unidirectional process or overall matching. However, recruitment is a two-way selection process, which means that both candidate and employer involved in the interaction should meet the expectation of each other, instead of unilateral satisfaction. In this paper, we propose a dual-perspective graph representation learning approach to model directed interactions between candidates and jobs. To model the two-way selection preference from the dual-perspective of job seekers and employers, we incorporate two different nodes for each candidate (or job) and characterize both successful matching and failed matching via a unified dual-perspective interaction graph. To learn dual-perspective node representations effectively, we design an effective optimization algorithm, which involves a quadruple-based loss and a dual-perspective contrastive learning loss. Extensive experiments on three large real-world recruitment datasets have shown the effectiveness of our approach. Our code is available at <https://github.com/RUCAIBox/DPGNN>.

## CCS CONCEPTS

• Information systems → Recommender systems.

## KEYWORDS

person-job fit, graph neural network, contrastive learning

<sup>†</sup> Beijing Key Laboratory of Big Data Management and Analysis Methods.

<sup>‡</sup> Beijing Academy of Artificial Intelligence, Beijing, 100084, China.

✉ Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

RecSys '22, September 18–23, 2022, Seattle, WA, USA

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-0/22/xx...\$15.00

<https://doi.org/10.1145/3523227.3546752>

## ACM Reference Format:

Chen Yang, Yupeng Hou, Yang Song, Tao Zhang, Ji-Rong Wen, Wayne Xin Zhao. 2022. Modeling Two-Way Selection Preference for Person-Job Fit. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22)*, September 18–23, 2022, Seattle, WA, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3523227.3546752>

## 1 INTRODUCTION

With the rapid development of Internet technology, online recruitment has become a prevalent service for job hunting, which matches qualified candidates with suitable jobs. Due to the massive growth of candidates and recruiters in online recruitment, it is critical to design effective algorithms to automatically establish high-quality connections between job seekers and employers through the recommendation mechanism. Such a task is called *person-job fit* [55].

In the literature, a variety of studies have been conducted for person-job fit, such as reciprocal recommendation [23], job recommendation [18, 29, 48] and job-oriented skill measurement [43]. Since the information of job and candidate is usually organized and described in textual documents, many text-based person-job fit works have been proposed in recent years, which learn matching function by modeling the textual semantics [31, 55]. These methods might also be thought of as content-based reciprocal recommendation approaches.

Despite the performance improvement, existing studies either model the one-way selection process (e.g., recommending qualified candidates conditioned on the job requirement [38, 45]) or overall matching relation (e.g., the text matching approach [55]). Since the recruitment process involves the two sides of candidates and employers, it naturally reflects the two-way selection preference [24] from both perspectives of candidates and job positions. In order to achieve a person-job matching, both sides involved in the interaction should meet the expectation of each other, instead of unilateral satisfaction. Such a kind of bilateral satisfaction is the key to the success of person-job fit. On online recruitment platforms, it is common to see unilateral satisfaction cases. For example, a job seeker applies for a position but is refused by the employer, or an employer sends an interview request but is refused by the job seeker. These unilateral interactions correspond to failed matches, which don't

meet the bilateral expectation. Actually, two-way selection is a common phenomenon in nature and society [53], and it is essential to consider such a bilateral match process in person-job fit.

To improve the person-job fit, we explicitly model the two-way selection preference from the dual perspectives of job seekers and employers. We develop our solution based on a graph representation learning approach. The core idea is to incorporate two different nodes for each candidate (or job): one captures its own preference to select a job (or a candidate) and the other reflects the corresponding preference from others. In this way, we can characterize both successful matching and failed matching via a unified dual-perspective interaction graph. Based on such an interaction graph, we propose a novel dual-perspective graph representation learning approach for modeling two-way selection intentions. There are two major technical contributions in our approach. Firstly, we learn both active and passive representations via a hybrid preference propagation, capturing the node characteristics to actively select and to be passively selected for person-job interaction. Secondly, we design an effective optimization algorithm for learning dual-perspective node representations, which involves a quadruple-based loss and a dual-perspective contrastive learning loss.

To the best of our knowledge, it is the first study that explicitly models the two-way selection preference for person-job fit. In order to demonstrate the effectiveness of our approach, we conduct extensive experiments on real-world recruitment data from three job domains. Experiment results show that our approach is significantly better than both collaborative filtering baselines and content-based baselines.

## 2 RELATED WORK

### 2.1 Person-Job Fit

As an important task in recruitment data mining [15, 35], Person-Job Fit (PJF) has been widely studied. The early work can be traced back to [23], in which expectation maximization algorithm was used to make recommendations by utilizing profile information from both candidates and jobs. Some other works tackled this challenge from the perspective of collaborative filtering [7, 21]. Zhang *et al.* [48] systematically compared a number of user-based and item-based collaborative filtering algorithms in order to recommend suitable jobs for candidates.

In recent years, most research cast this problem as a text matching task in order to make full use of the rich textual semantic information in resumes and job requirements. Some works have been proposed to utilize various kinds of neural networks to represent resumes and job posts, such as CNN [36, 55], RNN [31] and memory network [44]. Besides, some researchers have explored various techniques to improve the expression ability of the text encoders, such as adversarial learning [22], transfer learning [3] and co-teaching mechanisms [2].

Some works also considered the multi-granularity interactions in the person-job fit scenario. Le [17] cast the multi-level interactions as supervision signals and proposed a ranking-based loss function. Fu [9] proposed a bilateral multi-behavior sequence model to describe the dynamic comprehensive preferences of users. Besides, some studies have explored other related behaviors of users to improve person-job fit, such as search history [13] and interactive

feedbacks [8]. However, the two-way selection preference, which naturally exists in online recruitment scenarios, is not explicitly modeled in these methods. In this work, we propose a novel dual-perspective graph representation learning approach that jointly leverages interactions and text.

### 2.2 Recommender Systems

Recommender systems have been widely deployed to alleviate information overload on the web. A prevalent technique in modern recommender systems is collaborative filtering (CF) [11, 12]. The core idea is to predict users' preferences by exploiting their historical interactions. Various collaborative filtering methods have been proposed, such as matrix factorization-based methods [33], neural network-based methods [12] and auto-encoder-based methods [19].

Recently, graph-based collaborative filtering (GCF) is proposed to deepen the use of high-hop neighbors of users and items [11, 40]. These methods usually organize interaction data into a bipartite graph. However, in the person-job fit scenario, it's difficult to leverage directional interactions and model the two-way selection preference via existing GCF methods. Thus, in this work, we redesign the network architecture and incorporate two different nodes for each candidate (or job) and propose a graph collaborative approach on the constructed dual-perspective interaction graph.

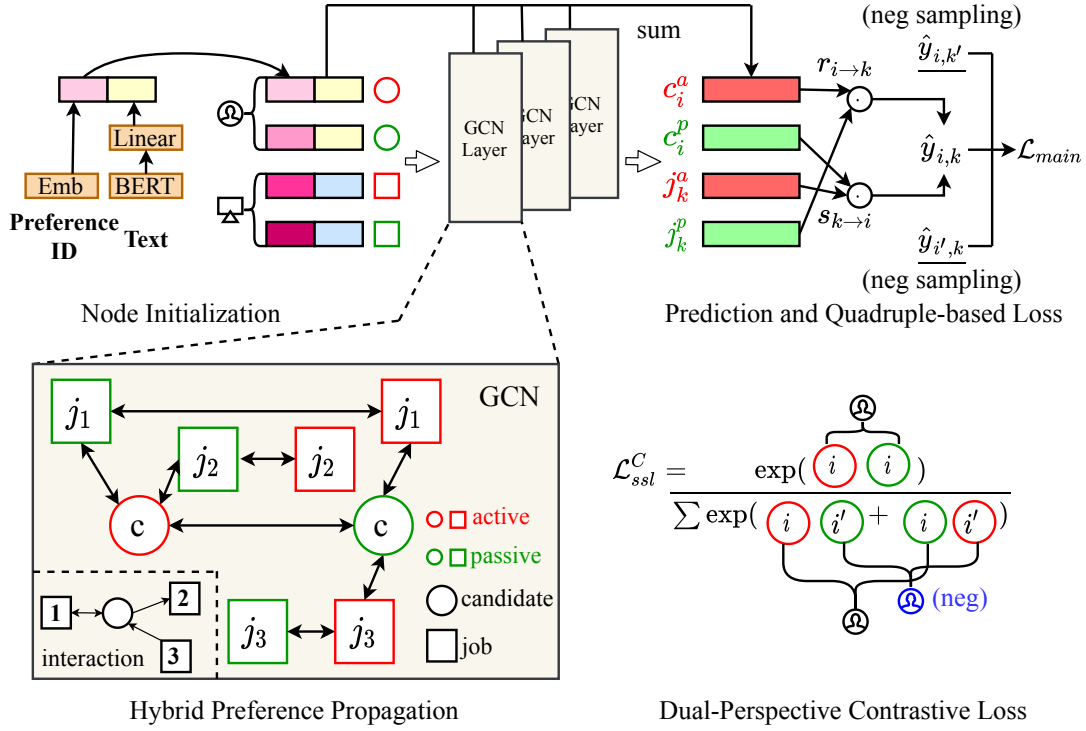
Since recruitment is a two-way selection process, it is also related to a research topic called Reciprocal Recommender System (RRS) [25, 37, 46], a kind of recommender system that recommends users to other users rather than items. RRS is frequently employed in domains with significant societal influences, such as recruitment [25, 38, 46], online dating [30, 39, 42] and social networking platforms [10]. They can be divided into two main categories based on the data source they used: content-based methods based on user profiles [26, 30] and collaborative filtering methods based on user behaviors [16, 27, 42]. As mentioned above, person-job fit methods based on text matching [44, 55] can also be regarded as content-based reciprocal recommender systems. However, existing RRS methods do not model directional behaviors to explicitly capture two-way selection preferences, which is important in person-job fit scenario [24]. In this work, we first construct a dual-perspective interaction graph based on directional interactions, and then propose a hybrid method to jointly model the interactions and text via graph neural networks.

## 3 PROPOSED METHOD

In this section, we first formulate the person-job fit task and then introduce the proposed **Dual-Perspective Graph Neural Network**, named **DPGNN**. The overall architecture is depicted in Figure 1.

### 3.1 Notations and Problem Statement

Assume that we have a set of candidates  $C = \{c_1, c_2, \dots, c_n\}$  and a set of jobs  $\mathcal{J} = \{j_1, j_2, \dots, j_m\}$ , where  $n$  and  $m$  are the total numbers of candidates and jobs. Each candidate or job is associated with a text document describing the resume or job requirements. Besides, they are also associated with a set of (directed) interaction records (*e.g.*, job application or resume review) in the job hunting platform. These interaction sets are formally denoted by  $\mathcal{A}_{c_i} = \{c_i \rightarrow j' | c_i \in C, j' \in \mathcal{J}\}$  and  $\mathcal{A}_{j_k} = \{j_k \rightarrow c' | j_k \in \mathcal{J}, c' \in C\}$ , described as



**Figure 1: Overall framework of the proposed model DPGNN. Note that in the hybrid preference propagation module (Sec. 3.2.1), we construct a new dual perspective interaction graph based on original interactions. Each candidate (job) corresponds to two nodes in the constructed graph, representing her active and passive preferences. Edges are created between nodes that belong to the same user as well as between nodes whose corresponding users are connected by directional interactions.**

directed interactions or links requested by candidate  $c_i$  or employer  $j_k$  (a.k.a., job). Here, we use  $i$  and  $k$  as the indices for candidate and job, respectively.

Through these interaction behaviors, employers and candidates might reach agreements on some job positions, which form a matching set  $M = \{(c_i, j_k) | c_i \in C, j_k \in \mathcal{J}\}$ , denoting all matching pairs in the platform. It should be noted that the matching pairs are removed from the interaction sets of both sides. Actually, a matching record  $(c_i, j_k)$  can also be considered as two directed interaction records, namely  $c_i \rightarrow j_k$  and  $j_k \rightarrow c_i$ . Based on the interactions and matching results, the task of person-job fit is to learn a matching function  $y = f(c_i, j_k)$  that predicts the matching results or score  $y$  between the job-person pair [36], which can be equally cast as a ranking task *w.r.t.* to a job or candidate [17].

In job recruitment, offering and accepting a position is indeed a two-way process [24], where both the candidate and employer can express their intention to each other, called *selection preference*. In prior person-job fit studies [55], they mainly consider the overall matching results (either success or failure), while the two-way selection preference is seldom considered in the matching function. For this purpose, we explicitly model candidate- and employer-specific selection preference, denoted by  $r_{i \rightarrow k}$  and  $s_{k \rightarrow i}$ , indicating the intention degree of candidate  $c_i$  on job  $j_k$  and vice versa. In this way, we aim to incorporate *dual-perspective selection* (candidate’s

perspective and employer’s perspective) for improving the performance of person-job fit. We present the used notations throughout the paper in Table 1.

### 3.2 Dual-Perspective Graph Representation Learning for Directed Selection Intention

As described before, the person-job fit requires considering the two-way selection process, and thus we propose a dual-perspective graph convolution network for modeling the directed interactions between jobs and candidates. We first construct the dual-perspective interaction graph and then design a specific graph convolution layer to learn high-level intention representations for both candidates and jobs.

**3.2.1 Constructing the Dual-Perspective Interaction Graph.** To model the directed behavior for person-job fit, we propose to construct the dual-perspective interaction graph.

Specifically, given a candidate  $c$ , we denote its active and passive representations by  $c^a$  (describing the active selection preference) and  $c^p$  (describing the passive selection preference), respectively. Similarly, let  $j^a$  and  $j^p$  denote the active and passive representations of job  $j$ . Each representation corresponds to one node in the graph, that is, each candidate (job) is represented with two nodes. These two kinds of nodes aim to capture the node characteristics to

**Table 1: Notations and explanations.**

Notation	Explanation
$c_i, c', c_1, c_2$	candidates
$j_k, j', j_1, j_2$	jobs
$\mathcal{C}$	the set of candidates
$\mathcal{J}$	the set of jobs
$\mathcal{A}_{c_i}$	the interaction set that the candidate $c_i$ actively interact
$\mathcal{A}_{j_k}$	the interaction set that the job $j_k$ actively interact
$c_i \rightarrow j_k, j_k \rightarrow c_i$	candidate $c_i$ interact with job $j_k$ actively and passively
$r_{i \rightarrow k}$	candidate-specific selection preference score from candidate $c_i$ to job $j_k$
$s_{k \rightarrow i}$	employer-specific selection preference score from job $j_k$ to candidate $c_i$
$d_E$	the dimension of the preference embedding
$d_O$	the output dimension of BERT for document encoding
$d_T$	the dimension of the document representation
$c_i^a, c_i^p \in \mathbb{R}^{d_E}$	active and passive representations of a candidate $c_i$
$j_k^a, j_k^p \in \mathbb{R}^{d_E}$	active and passive representations of a job $j_k$
$t_n \in \mathbb{R}^{d_O}$	the document representation of a resume or a job description
$z_n^{(0)} \in \mathbb{R}^d$	the initial representation for node $n$
$\mathbf{W} \in \mathbb{R}^{d_T \times d_O}$	a learnable transformation matrix for node representation initialization
$z_n \in \mathbb{R}^d$	the final representation encoded via graph neural network for node $n$
$\omega$	a hyper-parameter to balance the two types of propagation in graph convolution
$\mathcal{M}_n$	the matching set related to node $n$
$\mathcal{N}_n$	the neighbors of node $n$
$\tau$	the temperature hyper-parameter in dual-perspective contrastive learning optimization function
$\lambda$	a hyper-parameter to control the strength of the contrastive loss at the training stage

actively select others (active perspective) or be passively selected by others (passive perspective). Intuitively, the final match can only be achieved when the intention scores from both active and passive perspectives are strong.

The edges are constructed by the different interactions, as well as the association between two nodes of the same candidate (job). Since such a characterization is symmetric, we can model these directed interactions by modeling the following three cases. Examples of these three different interactions and the corresponding edges are illustrated in Fig 1.

- *A candidate applied for a job but was not accepted.* It means that the job meets the expectations of the candidate, but the candidate doesn't meet the expectations of the employer. In this case, such an interaction reflects candidate's active preference. Thus, we add an edge between nodes whose representations are  $c^a$  and  $j^p$ .

- *An employer reached out to a candidate but was refused.* Similar to the first case, an edge between nodes with representation  $j^a$  and  $c^p$  is established.

- *The two sides reached an agreement for offline interviews.* It indicates that the two sides meet the expectation of each other. Under this circumstance, we connect the corresponding active and passive nodes of the involved candidate and job ( $c^a$  and  $j^p$ ,  $j^a$  and  $c^p$ ), forming two edges for matching.

**3.2.2 Initializing the Node Representations.** To initialize the node representations, for each node  $n$  in the dual-perspective interaction graph, we first apply a look-up table operation for obtaining a preference embedding  $e_n \in \mathbb{R}^{d_E}$ , where  $d_E$  is the dimension of

the preference embeddings and this embeddings are generated randomly at first.

Besides, each node is associated with a descriptive text (*i.e.*, job description or candidate resume). This text information provides important evidence to understand and model the node characteristics. Therefore, we further utilize the BERT model [6] to encode the corresponding text for deriving node representations. Specially, we keep the original order of the text and insert a special token [CLS] before the text. Then the concatenated sequence is fed to the BERT model and the document representation  $t_n \in \mathbb{R}^{d_O}$  can be obtained, where  $d_O$  is the output dimension of BERT.

Finally, the two representations mentioned above are fused as the initial representation for node  $n$ :

$$z_n^{(0)} = [e_n; \mathbf{W} \cdot t_n] \in \mathbb{R}^d, \quad (1)$$

where  $\mathbf{W} \in \mathbb{R}^{d_T \times d_O}$  is a learnable transformation matrix,  $[\cdot]$  denotes concatenation operation and  $d = d_E + d_T$  is dimension of the initialized node representations.

**3.2.3 Propagating the Hybrid Preference.** Since we have characterized the interactions in the form of the interaction graph, we adopt the graph convolution networks (GCN) to learn the node representations. Different from prior GCN studies [11], we have two kinds of different edges for each involved candidate and job. Therefore, we propose a hybrid preference propagation algorithm for learning the node representations. It should be noted that the differences are inherent in edge rather than node type, so the preference propagation of all nodes can be defined uniformly.

At the  $l$ -th layer of graph convolution, for each node  $n$  (either job or candidate, either active or passive), we consider the preference propagation from two different sets of interactions: the matching set related to node  $n$  denoted by  $\mathcal{M}_n$  and the interaction set related to node  $n$  denoted by  $\mathcal{A}_n$ . Formally, we adopt a lightweight propagation mechanism [11] for updating the node representations:

$$z_n^{(l)} = \sum_{u \in \mathcal{M}_n} \frac{1}{\sqrt{|\mathcal{N}_n| |\mathcal{N}_u|}} z_u^{(l-1)} + \omega \sum_{v \in \mathcal{A}_n} \frac{1}{\sqrt{|\mathcal{N}_n| |\mathcal{N}_v|}} z_v^{(l-1)}, \quad (2)$$

where  $z_u$  and  $z_v$  denote the neighbors that have matching interaction and unidirectional interaction, respectively, and  $\mathcal{N}_n$ ,  $\mathcal{N}_u$  and  $\mathcal{N}_v$  are neighbors of node  $n$ , node  $u$  and node  $v$ . Here, we incorporate a specific hyper-parameter  $\omega$  to balance the two types of propagation, since these two kinds of interactions convey different levels of preference when learning node representations.

We average the representations from  $(L + 1)$  layers as the final representation for each node  $n$  as follows:

$$z_n = \sum_{l=0}^L \frac{1}{L+1} z_n^{(l)}, \quad (3)$$

where  $L$  denotes the number of graph convolution layers.

### 3.3 Prediction with Two-Way Intentions

Above, we have constructed the interaction graph and introduced the propagation mechanism for updating node representations. Next, we discuss how to learn the model parameters according to the person-job matching records.

After learning node representations, we can compute the two-way selection preference  $r_{i \rightarrow k}$  (the intention that a candidate  $c_i$  selects a job  $j_k$ ) and  $s_{k \rightarrow i}$  (the intention that a job  $j_k$  selects a candidate  $c_i$ ) for modeling the intention from dual perspectives. Formally, given a candidate  $c_i$  and a job  $j_k$ , we use the inner products to compute these two kinds of intention scores:

$$\begin{aligned} r_{i \rightarrow k} &= \mathbf{c}_i^a \cdot \mathbf{j}_k^p, \\ s_{k \rightarrow i} &= \mathbf{j}_k^a \cdot \mathbf{c}_i^p. \end{aligned} \quad (4)$$

Finally, we integrate the two intention degrees to predict the matching score:

$$\hat{y} = \frac{1}{2} r_{i \rightarrow k} + \frac{1}{2} s_{k \rightarrow i}. \quad (5)$$

Different from prior person-job fit studies (either modeling the overall or unidirectional fit), our approach explicitly models the two-way selection preference, which is expected to better capture the actual recruitment process.

### 3.4 Self-Supervised Enhanced Dual-Perspective Ranking Optimization

For person-job fit, it is more difficult to directly predict an absolute matching score, which is likely to result in over-fitting and model bias [17]. Therefore, we adopt a ranking-based approach to optimize the entire model.

**3.4.1 Quadruple-based Loss Function.** Unlike previous studies, which either optimize a cross-entropy loss [31] or a pairwise comparison loss [17], we introduce a novel quadruple-based loss function. In the recruitment scenario, a successful matching often means

that the job position should rank highly from the perspective of the candidate, and vice versa. Both perspectives are equally important, and both must be satisfied at the same time.

Given a matched record  $\langle c_i, j_k \rangle$ , we construct a quadruple  $\langle c_i, j_k, c_{i'}, j_{k'} \rangle$  by sampling a non-matched candidate and a non-matched job, *i.e.*, constructing the negative pairs  $\langle c_i, j_{k'} \rangle$  and  $\langle c_{i'}, j_k \rangle$  for optimization. Intuitively, the matching scores should hold the following conditions:

$$\begin{aligned} f(c_i, j_k) &> f(c_i, j_{k'}), \\ f(c_i, j_k) &> f(c_{i'}, j_k). \end{aligned} \quad (6)$$

For simplicity, we abbreviate  $\langle c_i, j_k, c_{i'}, j_{k'} \rangle$  by  $\langle i, k, i', k' \rangle$ . Then, we extend the widely used BPR loss [33] by modeling the quadruple-based loss as follows:

$$\mathcal{L}_{main} = -\frac{1}{|\mathcal{D}|} \sum_{(i,k,i',k') \in \mathcal{D}} \log \left( \sigma(\hat{y}_{i,k} - \frac{1}{2} \hat{y}_{i,k'} - \frac{1}{2} \hat{y}_{i',k}) \right), \quad (7)$$

where  $\mathcal{D} = \{(i, k, i', k') | (i, k) \in \mathcal{M}, (i, k') \in \mathcal{M}^-, (i', k) \in \mathcal{M}^-\}$  denotes the training data,  $\mathcal{M}$  and  $\mathcal{M}^-$  denote the matched and unmatched sets, respectively. Here, we average the scores of  $\hat{y}_{i,k}$  and  $\hat{y}_{i',k}$  to contrast with the matched score  $\hat{y}_{i,k}$  via the sigmoid function  $\sigma(\cdot)$ . As we can see, such a loss tries to optimize the partial orders in Eqn. (6).

**3.4.2 Dual-Perspective Contrastive Learning.** Besides the above optimization, we further consider another natural constraint for optimizing the model parameters. Since we set up two kinds of representations for each candidate (or job), the two representations should be similar to each other. Inspired by the above insight, we design a dual-perspective contrastive learning optimization function. Specifically, we treated the *active representation* and the *passive representation* of the same candidate (or job) as the positive pairs (*i.e.*,  $\{(c_i^a, c_i^p) | c_i \in C\}$ ), and the representations between candidates (or jobs) as the negative pairs (*i.e.*,  $\{(c_i^a, c_{i'}^p) | c_i, c_{i'} \in C\}$ ). The auxiliary supervision of positive pairs promotes the consistency of representations from different perspectives for the same candidate (or job), while the supervision of negative pairs tries to enlarge the divergence among different subjects. Formally, we adopt the InfoNCE [28] loss to maximize the agreement of positive pairs and minimize that of negative pairs:

$$\mathcal{L}_{ssl}^C = - \sum_{c_i \in C} \log \frac{\exp((c_i^a \cdot c_i^p) / \tau)}{\sum_{c_{i'} \in C} (\exp((c_i^a \cdot c_{i'}^p) / \tau) + \exp((c_{i'}^a \cdot c_i^p) / \tau))}, \quad (8)$$

where  $\tau$  is the temperature hyper-parameter. Similarly, we can obtain the contrastive loss on the job side  $\mathcal{L}_{ssl}^J$ . Combining these two losses, the final objective function of the self-supervised task is given as:

$$\mathcal{L}_{ssl} = \mathcal{L}_{ssl}^C + \mathcal{L}_{ssl}^J. \quad (9)$$

**3.4.3 Training and Complexity Analysis.** At the training stage, we leverage a multi-task training strategy to jointly optimize the proposed quadruple-based ranking loss (Eqn. (7)) and dual-perspective contrastive loss (Eqn. (9)):

$$\mathcal{L} = \mathcal{L}_{main} + \lambda \mathcal{L}_{ssl}, \quad (10)$$

**Algorithm 1:** The training algorithm of DPGNN.

---

**input** : candidate set  $C$  and job set  $\mathcal{J}$ , as well as their associated text document and directed interactions, and matching set  $\mathcal{M}$  defined in Sec. 3.1.

- 1 Construct the dual-perspective interaction graph with directed behaviors in training set. (Sec. 3.2.1)
- 2 Initialize preference embedding  $\mathbf{e}_n$  and document representation  $\mathbf{t}_n$  for each node  $n$ . (Sec. 3.2.2)
- 3 **foreach** mini-batch  $\mathcal{B}$  **do**
- 4     Update node representations  $\mathbf{z}_n^{(0)}$  for each node  $n$  with Eqn. (1)
- 5     Obtain the final node representations  $\mathbf{z}_n$  via propagating the hybrid preference with Eqn. (2)
- 6      $\mathcal{L}_{main} \leftarrow 0$
- 7     **foreach** candidate-job pair  $\langle c_i, j_k \rangle \in \mathcal{B}$  **do**
- 8         Sample a non-matched candidate  $c_{i'}$  (for  $j_k$ ) and a non-matched job  $j_{k'}$  (for  $c_i$ )
- 9          $\hat{y}_{i,k} \leftarrow \frac{1}{2} \mathbf{c}_i^a \cdot \mathbf{j}_k^p + \frac{1}{2} \mathbf{j}_k^a \cdot \mathbf{c}_i^p$  (similar for  $\hat{y}_{i,k'}$  and  $\hat{y}_{i',k}$ )
- 10          $\mathcal{L}_{main} \leftarrow \mathcal{L}_{main} - \log(\sigma(\hat{y}_{i,k} - \frac{1}{2} \hat{y}_{i,k'} - \frac{1}{2} \hat{y}_{i',k}))$
- 11     Calculate  $\mathcal{L}_{ssl}^C$  with candidates in this mini-batch with Eqn. (8) (similar for  $\mathcal{L}_{ssl}^J$ )
- 12      $\mathcal{L} = \mathcal{L}_{main} + \lambda (\mathcal{L}_{ssl}^C + \mathcal{L}_{ssl}^J)$
- 13     Backpropagate  $\mathcal{L}$  and update model parameters

---

where  $\lambda$  is a hyper-parameter to control the strength of the individual-level contrastive learning task.

For training the proposed model, the major time consuming part lies in the propagation via GCN (Eqn. (2)) and calculation of  $\mathcal{L}_{ssl}$  (Eqn. (9)). For propagation of GNN, it requires a time of  $O(|\mathcal{E}| \cdot L \cdot d)$ , where  $\mathcal{E} = \mathcal{D} \cup \mathcal{A}^u \cup \mathcal{A}^c$  denotes the set of all edges in the constructed dual-perspective interaction graph,  $L$  is the number of GNN layers and  $d$  is the dimension of node representations in GNN. For the dual-perspective contrastive loss, we usually sample  $S$  negative candidates (or jobs) for each user in practice. Then the time complexity can be roughly estimated as  $O(|\mathcal{V}| \cdot S \cdot d)$ , where  $\mathcal{V} = C \cup \mathcal{J}$  denotes the set of all nodes in the interaction graph. Note that once users submit their resumes or job description, we can obtain the corresponding BERT representations offline, and thus we ignore the corresponding cost here. Overall, the time complexity for training an epoch is  $O(|\mathcal{E}| \cdot L \cdot d + |\mathcal{V}| \cdot S \cdot d)$ . The full algorithm is described in Algorithm 1.

**3.4.4 Discussion.** We conduct a discussion here with comparisons to existing methods for a better understanding of the proposed optimization techniques. For the proposed quadruple-based ranking loss, the negative examples are sampled from both perspectives to meet the ordering constraints in Eqn. (4). Optimizing the quadruple-based object makes the dual-perspective ranking stable. Besides, inspired by recent advances in self-supervised recommendation [47], we propose the dual-perspective contrastive loss to enhance the whole optimization. Unlike previous self-supervised signals based on attribute correlation [54] and data augmentation [41], we consider the natural correlations between the preference representations from two perspectives of the same user (e.g.,  $\mathbf{c}_i^a$  and  $\mathbf{c}_i^p$  of candidate  $c_i$  in Eqn. (8)) as self-supervised supervision, which is both efficient and effective.

**Table 2: Statistics of the experiment dataset.**

Statistics	# Candidates	# Jobs	# C Interaction	# J Interaction	# Match
Tech	56,634	48,090	3,749,807	907,087	925,193
Sales	15,854	12,772	213,860	2,077,560	145,066
Design	12,290	9,143	1,200,590	76,287	166,270

## 4 EXPERIMENT

In this section, we would like to evaluate the performance of the proposed DPGNN. Extensive experiments conducted on three real-world datasets aim to answer the following research questions:

- **RQ1:** Does our model outperform the content-based, collaborative filtering-based, and hybrid state-of-the-art methods on the person-job fit task with the real-world datasets?
- **RQ2:** How do the key components (e.g., interaction graph, quadruple-based loss, and dual-perspective contrastive loss) of our model benefit the prediction?
- **RQ3:** How does our model perform with users who have varied levels of interaction sparsity?
- **RQ4:** How to select hyper-parameters for better performance of the proposed method?

In what follows, we first set up the experiments, and then present and analyze the results. Finally, we present how the proposed method performs in a specified case.

### 4.1 Experiment Setup

**Dataset.** We evaluate our model on three large real-world datasets provided by a popular online recruiting platform. The datasets were constructed from 106 days' real online logs and contained two kinds of behavior: *Match* and *Interaction*, corresponding to the matching set and interaction set mentioned in Section 3.1. Besides, each candidate (and job) is associated with a descriptive text (i.e., resume or job description). The overall statistics are shown in Table 2.

**Baseline Models.** We compare our method with the following baseline models:

- **BPRMF** [33] learns the latent user/item embeddings via minimizing Bayesian Personalized Ranking (BPR) loss.
- **NCF** [12] leverages an MLP to replace the inner product to model the interaction between user and item.
- **LightGCN** [11] simplifies GCN's design to make it concise for collaborative filtering.
- **LGCNBERT**. It has the same structure as LightGCN, except that the initial embedding is replaced by  $[\mathbf{e}_i; \text{MLP}(\mathbf{o}_T)]$ , which is the same as Eqn. (1).
- **LFRR** [27] is a reciprocal collaborative filtering method based on latent factors.
- **PJFNN** [55] is a method based on convolutional neural network (CNN). Resumes and job descriptions are encoded independently by hierarchical CNN, and the matching degree is calculated by cosine similarity.
- **BPJFNN** [31] leverages bidirectional LSTM to derive the representations of resumes and job descriptions.

**Table 3: Performance comparison of all methods. The best performance and the second best performance methods are denoted in bold and underlined fonts respectively. “\*” indicates that the improvements are significant at the level of 0.01 with paired *t*-test. “–” denotes that the corresponding experiment is not finished after 24 hours’ training.**

Dataset	Direction	Candidates				Jobs			
	Metric	Recall@5	Precision@5	NDCG@5	MRR@5	Recall@5	Precision@5	NDCG@5	MRR@5
Tech	BPRMF	0.2544	0.0906	0.2630	0.2692	0.2987	0.1464	0.3514	0.3403
	NCF	0.2496	0.0902	0.2646	0.2707	0.3040	0.1542	0.3727	0.3609
	LightGCN	0.2557	0.0925	0.2703	0.2762	0.3076	0.1535	0.3674	0.3545
	LFRR	0.2366	0.0817	0.2308	0.2376	0.2776	0.1228	0.3009	0.2856
	PJFNN	0.2104	0.0770	0.2232	0.2362	0.2733	0.1404	0.3578	0.3439
	BPJFNN	0.2136	0.0727	0.2033	0.2167	0.2620	0.1264	0.3146	0.3008
	APJFNN	-	-	-	-	-	-	-	-
	LGCN <sub>BERT</sub>	0.2685	<u>0.1003</u>	<u>0.2951</u>	<u>0.2982</u>	0.3187	<u>0.1713</u>	<u>0.4085</u>	0.3914
	IPJF	0.2591	0.0941	0.2654	0.2699	0.2979	0.1606	0.3818	0.3644
	PJFFF	0.2556	0.0985	0.2785	0.2791	<u>0.3195</u>	0.1691	0.4049	0.3870
	DPGNN	<b>0.2941*</b>	<b>0.1076*</b>	<b>0.3119*</b>	<b>0.3100*</b>	<b>0.3430*</b>	<b>0.1858*</b>	<b>0.4410*</b>	<b>0.4167*</b>
	Sales	BPRMF	0.2041	0.0643	0.1966	0.2156	0.2462	0.0843	0.2320
NCF		0.2031	0.066	0.1950	0.2138	0.2332	0.0796	0.2233	0.2395
LightGCN		0.2084	0.0657	0.1966	0.2152	0.2575	0.0903	0.2473	0.2538
LFRR		0.2244	0.0703	0.2064	0.2213	0.2397	0.0839	0.2312	0.2418
PJFNN		0.1385	0.0422	0.1180	0.1500	0.1550	0.0484	0.1396	0.1689
BPJFNN		0.1641	0.0521	0.1522	0.1777	0.2015	0.0676	0.1879	0.2065
APJFNN		0.1756	0.0588	0.1701	0.1908	0.2187	0.0722	0.2013	0.2198
LGCN <sub>BERT</sub>		<u>0.2272</u>	<u>0.0731</u>	<u>0.2184</u>	<u>0.2310</u>	<u>0.2518</u>	<u>0.0891</u>	<u>0.2404</u>	<u>0.2509</u>
IPJF		0.1393	0.0436	0.1244	0.1571	0.1977	0.0672	0.1955	0.2140
PJFFF		0.1973	0.0644	0.1869	0.2059	0.2478	0.0858	0.2363	0.2445
DPGNN		<b>0.2330*</b>	<b>0.0765*</b>	<b>0.2232*</b>	<b>0.2373*</b>	<b>0.2578*</b>	<b>0.0916*</b>	<b>0.2591*</b>	<b>0.2664*</b>
Design		BPRMF	0.2349	0.0726	0.2252	0.2417	0.2312	0.0919	0.2478
	NCF	0.2386	0.0761	0.2366	0.2505	0.2410	0.1014	0.2728	0.2733
	LightGCN	0.2451	0.0756	0.2336	0.2457	0.2438	0.1019	0.2707	0.2711
	LFRR	0.2342	0.0699	0.2105	0.2272	0.2478	0.1042	0.2789	0.2780
	PJFNN	0.1592	0.0518	0.1597	0.1886	0.1201	0.0511	0.1445	0.1661
	BPJFNN	0.1725	0.0539	0.1568	0.1826	0.1797	0.0694	0.1874	0.2003
	APJFNN	0.2120	0.0624	0.1738	0.2054	0.2022	0.0843	0.2314	0.2413
	LGCN <sub>BERT</sub>	<u>0.2517</u>	<u>0.0800</u>	<b>0.2498*</b>	<b>0.2637*</b>	0.2567	0.1158	0.2998	0.2934
	IPJF	0.2487	0.0798	0.2445	0.2553	0.2294	0.1159	0.3013	0.2987
	PJFFF	0.2400	0.0766	0.2367	0.2484	<u>0.2582</u>	<u>0.1193</u>	<u>0.3156</u>	<u>0.3121</u>
	DPGNN	<b>0.2685*</b>	<b>0.0808*</b>	<u>0.2478</u>	<u>0.2584</u>	<b>0.2783*</b>	<b>0.1342*</b>	<b>0.3524*</b>	<b>0.3379*</b>

- **APJFNN** [31] learns a word-level semantic representation for both resumes and job descriptions based on RNN models with attention hierarchically.
- **IPJF** [17] leverages multiple labels to indicate the propensity of candidates and jobs to reach a match.
- **PJFFF** [14] fuses the representations for the explicit and implicit intentions, which are processed by the historical applications using LSTM.

All baseline models fall into three categories, depending on the data they focus on: (1) *Collaborative filtering based methods*: BPRMF, NCF, LightGCN and LFRR; (2) *Content-based methods*: PJFNN, BPJFNN, APJFNN; (3) *Hybrid methods*: LGCN<sub>BERT</sub>, IPJF and PJFFF. Note that we do not include weaker collaborative filtering-based reciprocal methods as baselines such as RCF [42] and RWS [16]. Besides, as content-based methods are indeed reciprocal methods, we do not include additional content-based reciprocal baselines.

**Table 4: Effectiveness analysis of key components of DPGNN. R@5, P@5, NDCG@5, MRR are adopted for evaluation.**

Model	For candidates				For jobs			
	R@5	P@5	NDCG@5	MRR	R@5	P@5	NDCG@5	MRR
DPGNN	<b>0.2941</b>	<b>0.1076</b>	<b>0.3119</b>	<b>0.3100</b>	<b>0.3430</b>	<b>0.1858</b>	<b>0.4410</b>	<b>0.4167</b>
w/o DPG	0.2770	0.1008	0.2899	0.2912	0.3360	0.1803	0.4257	0.4045
w/o QL	0.2756	0.1010	0.2945	0.2964	0.3294	0.1741	0.4188	0.3994
w/o SSL	0.2897	0.1072	0.3125	0.3108	0.3346	0.1802	0.4286	0.4075

For each compared method, we learn the model parameters on the training set, perform hyper-parameter selection according to the results on the validation set, and finally report the result on the test set. Besides, for fair comparisons of those baselines that don't utilize unidirectional interactions, we treat the unidirectional interaction records as positive instances while training. We observe that it generally improves the performance of these methods, and we report these results in subsequent experiments.

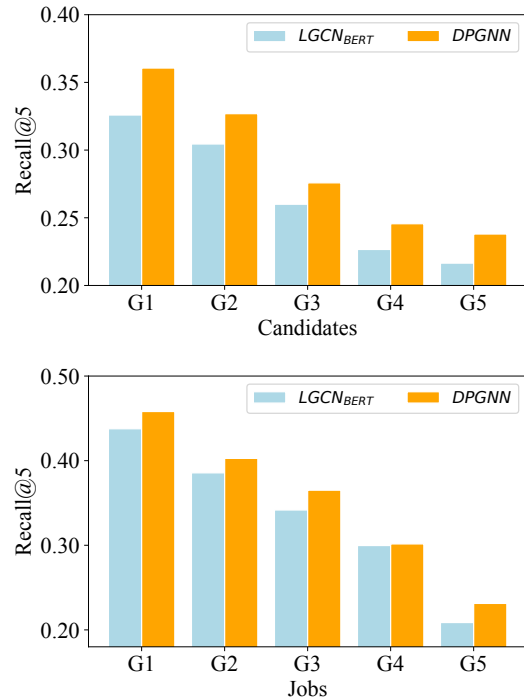
**Evaluation and Implementation Details.** To evaluate the performance of top- $k$  recommendation [50], we employ four widely used metrics, Recall ( $R@k$ ), Precision ( $P@k$ ), Normalized Discounted Cumulative Gain ( $NDCG@k$ ) and Mean Reciprocal Rank (MRR). We set  $k$  to 5 empirically. In particular, the evaluation method is different from general recommendation tasks. We leverage these metrics to do the evaluation for two person-job fit ranking tasks simultaneously, *i.e.*, ranking jobs for candidates and ranking candidates for employers, which is more in line with the real online recruitment scenarios. Concretely, for each positive instance  $(c_i, j_k)$ , we randomly sample 20 jobs for candidate  $c_i$  and 20 candidates for job  $j_k$  as negative instances. We evaluate the ranking list of the overall matching degree and report the average metric scores over all the candidates and jobs respectively.

The baseline models are implemented with a popular open-source recommendation library RECOLE [49, 51]. The interaction records are sorted and split into training, validation and test sets by timestamp. Specifically, the records of the last 11 days and the previous 11 days are used as test set and validation set respectively, and the rest 84 days are used for training. We optimize all the methods with Adam optimizer and carefully search the hyper-parameters of all the baselines. Parameters of the text encoders are initialized by bert-base-uncased<sup>1</sup>. The dimensions of text embeddings and preference embeddings are set to 32 and 128, respectively. The learning rate is tuned in {0.01, 0.001, 0.0001, 0.00001}. Early stopping is used with patience of 10 epochs.

## 4.2 Experiment Result

### Overall Performance (RQ1)

Table 3 presents the comparison between our model and all the baselines. For the four collaborative filtering baselines, LightGCN achieves the best performance, but the improvement is not significant compared with BPRMF, NCF and LFRR. As for the three content-based baselines, BPJFNN, PJFNN and APJFNN, which highly rely on the text content, they do not perform well in most



**Figure 2: Performance analysis for different sparsity level candidates and jobs, G1 denotes the group of candidates or jobs with the lowest average number of interactions.**

cases. The possible reason is that they require resumes and job postings to be structured and unabridged, while in our scenario, users in the platform have different text organization habits. IPJF doesn't perform well in Sales, due to the high imbalance in the number of different types of interactions. The performances of PJFFF are better in most cases, as PJFFF integrates historically interacted resumes or job descriptions. Besides, documents in Tech usually contain more specific skill requirements than others, making the text-based models relatively efficient in Tech. Finally, as we can see, LGCN<sub>BERT</sub> leverages both interactions and text. It mostly performs the best across baselines, indicating that utilizing both text descriptions and interactions is important.

As a comparison, our method achieves the best performance on most metrics across three datasets. Specifically, our model can on average improve the best baseline by 7.12%, 4.81% and 7.73% relatively on Tech, Sales and Design, respectively. Different from

<sup>1</sup><https://github.com/huggingface/transformers>



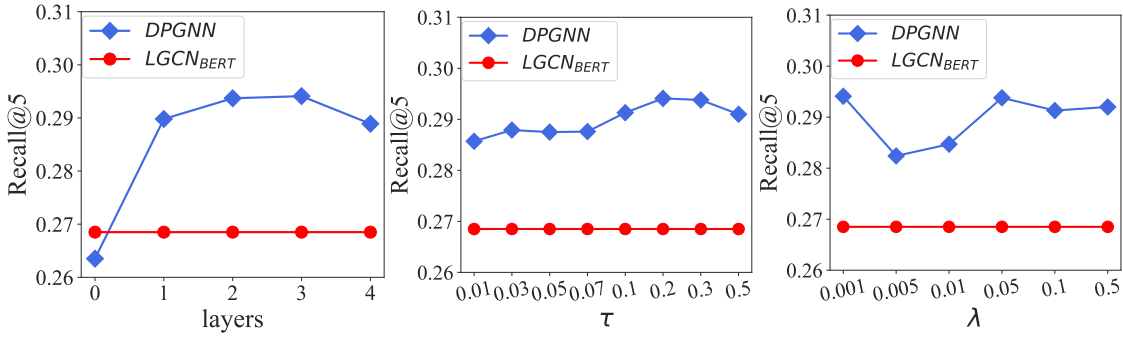


Figure 3: Performance comparison w.r.t. different layers,  $\tau$  and  $\lambda$ .

baselines, our approach models the two-way selection preferences of both candidates and jobs. Thus, it is more appropriate for person-job fit scenario.

#### Ablation Study (RQ2)

The major technical contribution of our approach lies in the dual-perspective interaction graph construction, as well as the involved two optimization objects. We now analyze how each part contributes to the final performance.

We consider the following three variants of DPGNN: **(A)** DPGNN w/o DPG replaces the proposed dual-perspective interaction graph to the conventional interaction graph, where each user has only one representation.; **(B)** DPGNN w/o QL changes quadruple-based loss to BPR loss; **(C)** DPGNN w/o SSL removes the dual-perspective contrastive loss.

In Table 4, we can see that the performance order can be summarized as DPGNN w/o QL < DPGNN w/o DPG < DPGNN w/o SSL < DPGNN. These results indicate that all three components are useful to improve the performance of DPGNN. Especially, the dual-perspective interaction graph and the quadruple-based loss function bring more improvements to our approach.

#### Data Sparsity Levels (RQ3)

As collaborative filtering methods are known to be sensitive to data sparsity, here we give an in-depth analysis to show how the proposed method DPGNN performs on different sparsity levels. Concretely, we divided all the candidates and jobs into five groups based on the number of their interactions, with the overall number of interactions in each group remaining constant. Then, we compare the recommendation performance of DPGNN with a strong baseline LGCN<sub>BERT</sub> on these overall ten groups and report the results in Figure 2. We can find that the performance of DPGNN is consistently better than LGCN<sub>BERT</sub>, even in those groups where users have sparse interactions. On the one hand, the preference representations are partially initialized by textual representations encoded. In this way, the prediction is not only based on behavior data. On the other hand, as contrastive learning has been shown effective on sparse data [20], the proposed dual-perspective contrastive learning manner (Eqn. (9)) can also benefit recommendation for those users with sparse interactions. The twofold techniques help DPGNN provide high-quality recommendations to users with different sparsity levels.

#### Parameter Tuning (RQ4)

In this part, we examine the robustness of our model and perform detailed analysis of key hyper-parameters. For simplicity, we only incorporate the best baseline LGCN<sub>BERT</sub> from Table 3 as a comparison.

*Varying the layers of GNN.* We introduce  $l$  to denote the layer number of GCN in Eqn (2). Here, we vary the number of layers from 0 to 4. As shown in Figure 3, our model achieves the best performance when  $l = 3$ , indicating DPGNN can deepen the usage of interaction histories. Overall, the performance is relatively stable when  $l \geq 1$ . When  $l \geq 4$ , the model may suffer from the over-smoothing issue and get sub-optimal results.

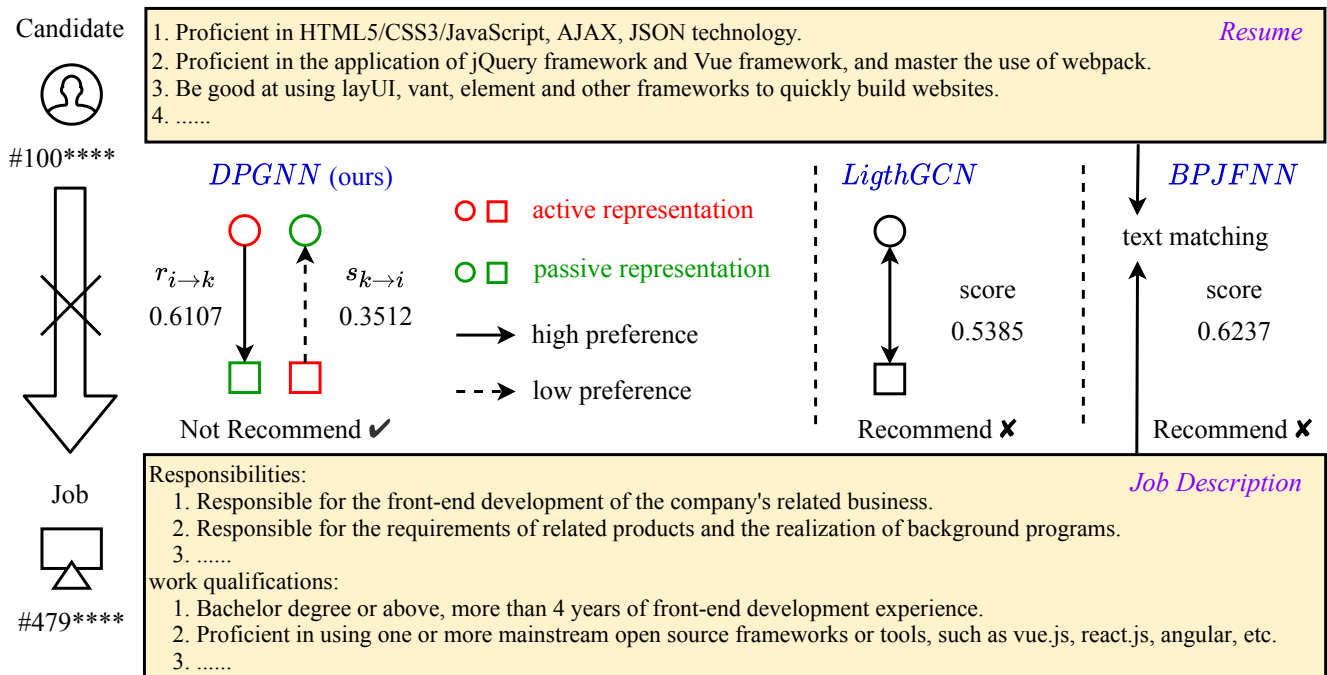
*Varying the temperature of the contrastive loss.* The parameter  $\tau$  of the contrastive loss is tuned in the ranges of {0.5, 0.1, 0.05, 0.01, 0.005, 0.001}. We can see that with different common values of  $\tau$ , the performances of DPGNN are constantly better than LGCN<sub>BERT</sub>. Besides, DPGNN achieves the best performance when the temperature is around 0.2.

*Varying the weight of the contrastive loss.* The weight of the contrastive loss are tuned in the ranges of {0.5, 0.3, 0.2, 0.1, 0.07, 0.05, 0.03, 0.01}. As shown in Figure 3, for our model, a weight above 0.05 works better. The fact that larger  $\lambda$  works better indicates the importance of modeling the correlations between preference representations of the same user.

#### 4.3 Case Study

As shown in Table 3, we can see that our approach can achieve better performance than both collaborative filtering methods and content-based methods. It is interesting to study that in what situation our method can work better. For this purpose, we present an illustrative case in Figure 4.

We randomly sample a job-candidate pair with their text descriptions from the test set of our experimental dataset Tech. It can be observed that our approach calculates two opposite one-way selection preference scores  $r_{i \rightarrow k}$  and  $s_{k \rightarrow i}$ , which means that the candidate may like the job but the recruiter doesn't satisfy him (may have better choices). With these two preference scores, we can see that it is not appropriate to recommend the candidate for the job in this situation. But for some content-based (BPJFNN) or collaborative filtering-based (LightGCN) models, the results may be the opposite with a relatively positive score obtained.



**Figure 4: A case where candidate and job have high text similarity but different preferences. Though the candidate prefers the given job position, the recruiter will not give a positive response. This case shows that DPGNN can capture the two-way selection preference, which is helpful to improve the recommendation performance of the person-job fit.**

This case verifies that person-job fit is a two-way selection process. The preferences from both sides are key factors that determine the final matching. The proposed DPGNN can capture these two-way selection preferences with historical interactions from two sides, making more effective recommendations.

## 5 CONCLUSION AND FUTURE WORK

This paper presented a dual-perspective graph convolution network to model the directed interactions between jobs and candidates for person-job fit. In our approach, we propose to construct a dual-perspective interaction graph to model the directed behavior. A hybrid preference propagation algorithm is leveraged to learn the node representations. To optimize the entire model, we design an effective algorithm including a quadruple-based loss and a dual-perspective contrastive learning loss and jointly optimize them. Extensive experiments indicate that the proposed approach can achieve better performance from both perspectives of candidates and job positions compared with competitive baselines.

Besides textual information and two-way selection behaviors, there are additional features that are essential for inferring user preferences in real online recruitment platforms, such as timestamps, expected salary ranges, working locations, and chronological sequential behavior histories. As a result, we consider designing a more unified framework for future work that incorporates various features into two-way selection preference modeling. Moreover, when we regard candidates and jobs in the person-job fit scenario

as two groups of users, it is related to the group balancing studies in multi-stakeholder recommendation [52], which focuses on designing joint optimization objectives towards diversity [5, 32], novelty [34] or fairness [1, 4]. For future work, we will explore how to balance these two stakeholders as well as benefit both groups.

## ACKNOWLEDGMENTS

This work was partially supported by National Natural Science Foundation of China under Grant No. 61872369, Beijing Natural Science Foundation under Grant No. 4222027, and Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098. This work was partially supported by Beijing Academy of Artificial Intelligence (BAAI). Xin Zhao is the corresponding author.

## REFERENCES

- [1] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in recommendation ranking through pairwise comparisons. In *KDD*.
- [2] Shuqing Bian, Xu Chen, Wayne Xin Zhao, Kun Zhou, Yupeng Hou, Yang Song, Tao Zhang, and Ji-Rong Wen. 2020. Learning to Match Jobs with Resumes from Sparse Interaction Data using Multi-View Co-Teaching Network. In *CIKM*.
- [3] Shuqing Bian, Wayne Xin Zhao, Yang Song, Tao Zhang, and Ji-Rong Wen. 2019. Domain adaptation for person-job fit with transferable deep global match network. In *EMNLP*.
- [4] Asia J Biega, Krishna P Gummedi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *SIGIR*.
- [5] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *SIGIR*.

- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.
- [7] Mamadou Diaby, Emmanuel Viennet, and Tristan Launay. 2013. Toward the next generation of recruitment tools: an online social network-based job recommender system. In *ASONAM*.
- [8] Bin Fu, Hongzhi Liu, Hui Zhao, Yao Zhu, Yang Song, Tao Zhang, and Zhonghai Wu. 2022. Market-Aware Dynamic Person-Job Fit with Hierarchical Reinforcement Learning. In *DASFAA*.
- [9] Bin Fu, Hongzhi Liu, Yao Zhu, Yang Song, Tao Zhang, and Zhonghai Wu. 2021. Beyond matching: Modeling two-sided multi-behavioral sequences for dynamic person-job fit. In *DASFAA*.
- [10] Jianming He and Wesley W Chu. 2010. A social network-based recommender system (SNRS). In *Data mining for social network data*.
- [11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*.
- [12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*.
- [13] Yupeng Hou, Xingyu Pan, Wayne Xin Zhao, Shuqing Bian, Yang Song, Tao Zhang, and Ji-Rong Wen. 2022. Leveraging Search History for Improving Person-Job Fit. In *DASFAA*.
- [14] Junshu Jiang, Songyun Ye, Wei Wang, Jingran Xu, and Xiaosheng Luo. 2020. Learning Effective Representations for Person-Job Fit by Feature Fusion. In *CIKM*.
- [15] Krishnamurthy Kishore, Benjamin Le, and Ganesh Venkataraman. 2017. Personalized job recommendation system at linkedin: Practical challenges and lessons learned. In *RecSys*.
- [16] Akiva Kleinerman, Ariel Rosenfeld, Francesco Ricci, and Sarit Kraus. 2018. Optimally balancing receiver and recommended users' importance in reciprocal recommender systems. In *RecSys*.
- [17] Ran Le, Wengpeng Hu, Yang Song, Tao Zhang, Dongyan Zhao, and Rui Yan. 2019. Towards effective and interpretable person-job fitting. In *CIKM*.
- [18] Danielle H Lee and Peter Brusilovsky. 2007. Fighting information overflow with personalized comprehensive information access: A proactive job recommender. In *ICAS*.
- [19] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW*.
- [20] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *TheWebConf*.
- [21] Yao Lu, Sandy El Helou, and Denis Gillet. 2013. A recommender system for job seeking and recruiting website. In *WWW*.
- [22] Yong Luo, Huaizheng Zhang, Yonggang Wen, and Xinwen Zhang. 2019. ResumeGAN: An Optimized Deep Representation Learning Framework for Talent-Job Fit via Adversarial Learning. In *CIKM*.
- [23] Jochen Malinowski, Tobias Keim, Oliver Wendt, and Tim Weitzel. 2006. Matching people and jobs: A bilateral recommendation approach. In *HICSS*.
- [24] John M McNamara and EJ Collins. 1990. The job search problem as an employer-candidate game. *Journal of Applied Probability* (1990).
- [25] Tsunenori Mine, Tomoyuki Kakuta, and Akira Ono. 2013. Reciprocal recommendation for job matching with bidirectional feedback. In *IIAI*.
- [26] James Neve and Ryan McConville. 2020. ImRec: Learning reciprocal preferences using images. In *RecSys*.
- [27] James Neve and Ivan Palomares. 2019. Latent factor models and aggregation operators for collaborative filtering in reciprocal recommender systems. In *RecSys*.
- [28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv:1807.03748* (2018).
- [29] Ioannis Paparrizos, Barla Cambazoglu, and Aristides Gionis. 2011. Machine learned job recommendation. In *RecSys*.
- [30] Luiz Pizzato, Tomek Rej, Thomas Chung, Irena Koprinska, and Judy Kay. 2010. RECON: a reciprocal recommender for online dating. In *RecSys*.
- [31] Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Liang Jiang, Enhong Chen, and Hui Xiong. 2018. Enhancing person-job fit for talent recruitment: An ability-aware neural network approach. In *SIGIR*.
- [32] Filip Radlinski, Paul N Bennett, Ben Carterette, and Thorsten Joachims. 2009. Redundancy, diversity and interdependent document relevance. In *SIGIR*.
- [33] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*.
- [34] Marco Tulio Correia Ribeiro. 2013. Multi-objective pareto-efficient algorithms for recommender systems. (2013).
- [35] Walid Shalaby, BahaaEddin AlAila, Mohammed Korayem, Layla Pournajaf, Khalifeh AlJadda, Shannon Quinn, and Wlodek Zadrozny. 2017. Help me find a job: A graph-based approach for job recommendation at scale. In *BigData*.
- [36] Dazhong Shen, Hengshu Zhu, Chen Zhu, Tong Xu, Chao Ma, and Hui Xiong. 2018. A joint learning approach to intelligent job interview assessment. In *IJCAI*.
- [37] Zheng Siting, Hong Wexing, Zhang Ning, and Yang Fan. 2012. Job recommender systems: a survey. In *ICCS*.
- [38] Yi Su, Magd Bayoumi, and Thorsten Joachims. 2022. Optimizing Rankings for Recommendation in Matching Markets. In *TheWebConf*.
- [39] Kun Tu, Bruno Ribeiro, David Jensen, Don Towsley, Benyuan Liu, Hua Jiang, and Xiaodong Wang. 2014. Online dating recommendations: matching markets and learning preferences. In *WWW*.
- [40] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*.
- [41] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*.
- [42] Peng Xia, Benyuan Liu, Yizhou Sun, and Cindy Chen. 2015. Reciprocal recommendation system for online dating. In *ASONAM*.
- [43] Tong Xu, Hengshu Zhu, Chen Zhu, Pan Li, and Hui Xiong. 2018. Measuring the popularity of job skills in recruitment market: A multi-criteria approach. In *AAAI*.
- [44] Rui Yan, Ran Le, Yang Song, Tao Zhang, Xiangliang Zhang, and Dongyan Zhao. 2019. Interview choice reveals your preference on the market: To improve job-resume matching through profiling memories. In *KDD*.
- [45] Yuyang Ye, Hengshu Zhu, Tong Xu, Fuzhen Zhuang, Runlong Yu, and Hui Xiong. 2019. Identifying high potential talent: A neural network based dynamic social profiling approach. In *ICDM*.
- [46] Hongtao Yu, Chaoran Liu, and Fuzhi Zhang. 2011. Reciprocal recommendation algorithm for the field of recruitment. *JICS* (2011).
- [47] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2022. Self-Supervised Learning for Recommender Systems: A Survey. *arXiv:2203.15876* (2022).
- [48] Yingya Zhang, Cheng Yang, and Zhixiang Niu. 2014. A research of job recommendation system based on collaborative filtering. In *ISCID*.
- [49] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, et al. 2022. RecBole 2.0: Towards a More Up-to-Date Recommendation Library. *arXiv preprint arXiv:2206.07351* (2022).
- [50] Wayne Xin Zhao, Zihan Lin, Zhichao Feng, Pengfei Wang, and Ji-Rong Wen. 2021. A Revisiting Study of Appropriate Offline Evaluation for Top-N Recommendation Algorithms. *TOIS* (2021).
- [51] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *CIKM*.
- [52] Yong Zheng et al. 2021. Multi-Objective Recommendations: A Tutorial. *arXiv:2108.06367* (2021).
- [53] Bin Zhou, Shujia Qin, Xiao-Pu Han, Zhe He, Jia-Rong Xie, and Bing-Hong Wang. 2014. A model of two-way selection system for human behavior. *PLoS one* (2014).
- [54] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM*.
- [55] Chen Zhu, Hengshu Zhu, Hui Xiong, Chao Ma, Fang Xie, Pengliang Ding, and Pan Li. 2018. Person-job fit: Adapting the right talent for the right job with joint representation learning. *TMIS* (2018).