Talent Demand-Supply Joint Prediction with Dynamic Heterogeneous Graph Enhanced Meta-Learning

Zhuoning Guo
Harbin Institute of Technology & Baidu Research, Baidu Inc. & Baidu Talent Intelligence Center, Baidu Inc.
1183710109@stu.hit.edu.cn

Hao Liu*
The Hong Kong University of Science and Technology (Guangzhou) & The Hong Kong University of Science and Technology
liuh@ust.hk

Le Zhang
Baidu Research, Baidu Inc
zhangle0202@gmail.com

Qi Zhang
University of Science and Technology of China & Baidu Talent Intelligence Center, Baidu Inc.
zq26@mail.ustc.edu.cn

Hengshu Zhu
Baidu Talent Intelligence Center, Baidu Inc.
zhuhengshu@baidu.com

Hui Xiong*
The Hong Kong University of Science and Technology (Guangzhou) & The Hong Kong University of Science and Technology
xionghui@ust.hk

ABSTRACT

Talent demand and supply forecasting aims to model the variation of the labor market, which is crucial to companies for recruitment strategy adjustment and to job seekers for proactive career path planning. However, existing approaches either focus on talent demand or supply forecasting, but overlook the interconnection between demand-supply sequences among different companies and positions. To this end, in this paper, we propose a *Dynamic Heterogeneous Graph Enhanced Meta-learning* (DH-GEM) framework for fine-grained talent demand-supply joint prediction. Specifically, we first propose a Demand-Supply Joint Encoder-Decoder (DSJED) and a Dynamic Company-Position Heterogeneous Graph Convolutional Network (DyCP-HGCN) to respectively capture the intrinsic correlation between demand and supply sequences and company-position pairs. Moreover, a Loss-Driven Sampling based Meta-learner (LDSM) is proposed to optimize long-tail forecasting tasks with a few training data. Extensive experiments have been conducted on three real-world datasets to demonstrate the effectiveness of our approach compared with five baselines. DH-GEM has been deployed as a core component of the intelligent human resource system of a cooperative partner.

CCS CONCEPTS

• **Information systems → Data mining**.

KEYWORDS

labor market forecasting, demand-supply modeling, sequential modeling, heterogeneous graph neural network, meta-learning

ACM Reference Format:
Zhuoning Guo, Hao Liu*, Le Zhang, Qi Zhang, Hengshu Zhu, and Hui Xiong*. 2022. Talent Demand-Supply Joint Prediction with Dynamic Heterogeneous Graph Enhanced Meta-Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539139

1 INTRODUCTION

Recent years have witnessed the increasingly competitive war on talent acquisition [4]. Organizations and companies continuously review and adapt their recruitment strategies to align with the radically varied labor market, which raises an urgent need for labor market forecasting. As an essential block of labor market analysis, labor market forecasting aims to model the landscape of time-evolving labor market, including both talent demand [37, 39] and supply [16, 36] variation. Indeed, timely and accurate forecasting of the labor market trend not only helps the government and companies for policy and recruitment strategy readjustment but is also beneficial for job seekers to plan their career path proactively [7].

Extensive studies have been made for labor market forecasting from different perspectives. Conventional heuristic methods mainly focus on coarse-grained labor market analysis (*e.g.*, industry-specific demand trend [19] and geographic-occupational labor market concentration [2]) based on survey data [27]. Such methods rely on classic statistical models and domain expert knowledge, but more sophisticated latent data dependencies have not been considered. The new emerging data-driven methods utilize machine learning techniques to exploit large-scale data acquired from online professional platforms. For example, TDAN [37] leveraged attention mechanism to forecast the talent demand value of next time interval with observed data and Ahead [36] constructed Dual-GRU with the heterogeneous graph embedding to predict the next moving

* Corresponding authors.

KDD '22, August 14–18, 2022, Washington, DC, USA.

ACM ISBN 978-1-4503-9385-0/22/08...$15.00
https://doi.org/10.1145/3534678.3539139

company, position and working duration from the supply perspective. The above methods formulate the talent demand or supply forecasting task as a time series prediction problem, and various sequential deep learning models have been proposed to capture the latent temporal correlation of the market trend variations.

However, after analyzing large-scale real-world data, we identify two important labor market variation characteristics, which have been rarely considered in previous studies. On the one hand, talent demand and supply are intrinsically correlated with each other. For example, the emerging demand of a rising company will attract more talents, and the oversupply of a position may curb the demand for an extended period to resolve the excessive talents. Modeling the interconnection between talent demand and supply variation can provide extra information for both tasks to predict more precisely. On the other hand, the demand-supply variation of different companies and positions are correlated yet diversified. Companies and positions in the same industry may follow similar co-evolvement patterns [1, 6], e.g., GE and Toyota may recruit many computer vision engineers in the trend of self-driving. However, even subsidiaries of the same company may have very different talent demand requirements at different times. Distilling and incorporating the shared knowledge between related companies and positions can further improve the effectiveness of both talent demand and supply forecasting. Inspired by the above characteristics, in this work, we study the problem of *Talent Demand-Supply Joint Prediction* (TDSJP), where the talent demand and supply of positions in every company are predicted simultaneously.

Three major challenges arise toward TDSJP. First, existing labor market forecasting methods either focus on talent demand or supply prediction but overlook the intrinsic correlation between talent demand and supply variation. It is challenging to incorporate the interconnection between two different tasks in a mutually reinforcing way. Second, the correlation between different companies and positions may vary. Collectively sharing information between all companies and positions may introduce unexpected noise and degrade the prediction performance. Prior studies mainly focus on the company- or position-level trend analysis. How to distill commonly shared knowledge and reduce potential noise information for fine-grained company-position demand-supply forecasting is another challenge. Third, the volume of talent demand and supply timely varies, and forecasting the fine-grained talent demand and supply for multiple companies further strengthens the sparsity issue. Many companies only have demand and supply records in short periods. The last challenge is accurately predicting talent demand and supply variation based on a few instances.

To address the aforementioned challenges, in this paper, we propose the *Dynamic Heterogeneous Graph Enhanced Meta-learning* (DH-GEM) framework. Specifically, we first construct fine-grained talent demand-supply sequences and a time-evolving company-position graph to encode the co-evolve patterns of demand-supply sequences and company-position pairs. We devise a *Demand-Supply Joint Encoder-Decoder* (DSJED) to attentively capture the intrinsic correlation between demand and supply variation. Moreover, to incorporate the time-evolving relationship between companies and positions, we propose the *Dynamic Company-Position Heterogeneous Graph Convolutional Network* (DyCP-HGCN) to selectively

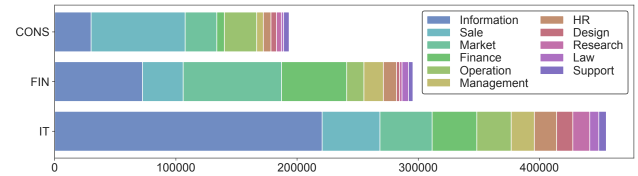


Figure 1: Positions distribution of three real-world datasets.

preserve common knowledge between company and position representations for more effective demand-supply prediction. Finally, a *Loss-Driven Sampling based Meta-learner* (LDSM) is proposed to train the prediction framework, in which companies with fewer data are optimized with a higher learning priority to obtain better initial parameters. In this way, the long-tail demand-supply prediction tasks can absorb high-level knowledge from companies with sufficient training data to achieve better prediction performance. DH-GEM has been deployed as a core functional component of the intelligent human resource system of a cooperative partner, providing timely insights and guidance for users.

The major contributions of this paper are summarized as follows. 1) We formulate fine-grained talent demand and supply forecasting tasks as a joint prediction problem, and a graph-enhanced meta-learning framework is proposed based on in-depth data-driven analysis. 2) We design an attentive joint encoder-decoder module and a dynamic graph representation learning module to extract shared knowledge between demand-supply sequences among different companies and positions. 3) We propose a meta-learner with loss-driven sampling strategy to learn transferable prior knowledge to optimize predictions for companies with insufficient data. 4) We conduct extensive experiments on real-world datasets to demonstrate the superiority of the proposed approach.

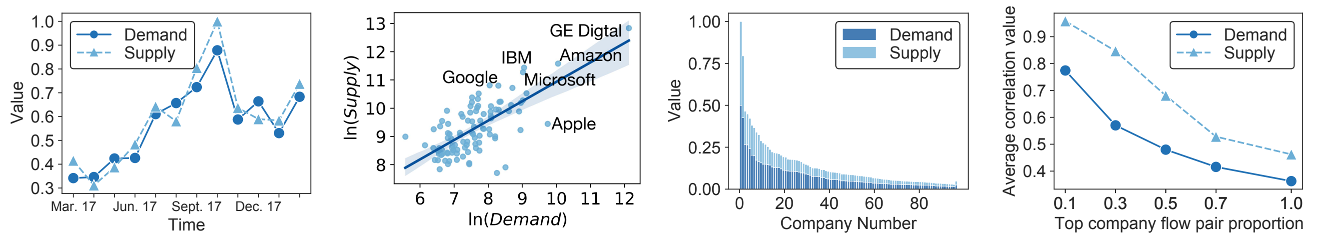
2 TALENT TREND MODELING WITH DYNAMIC COMPANY-POSITION GRAPH

In this section, we present three real-world datasets used in our study, extract talent demand and supply sequences from the raw data, and detail the construction of dynamic company-position heterogeneous graph with a preliminary data analysis.

2.1 Data Collection and Description

We collect the real-world data from LinkedIn, one of the largest online professional networks (OPNs), where companies can publish job postings for talent hunting and employees can create their own profiles of work experiences.

Specifically, we construct large-scale datasets from three major industries, *i.e.*, *Information Technology* (IT), *Finance* (FIN) and *Consuming* (CONS). All three datasets are ranged from March 2016 to March 2019. Particularly, there are $455,192$ job postings and $2,004,973$ work experiences in IT, $295,651$ job postings and $1,787,386$ work experiences in FIN, and $193,481$ job postings and $1,237,048$ work experiences in CONS. Following the official position titles for job hunting on LinkedIn and existing techniques [17], we categorize and align raw positions in job posting and work experience data into 11 classes, including *Information*, *Sale*, *Market*, *Finance*, *Operation*, *Management*, *HR*, *Design*, *Research*, *Law* and *Support*. The distribution of each position is shown in Figure 1.



(a) Overall distribution over time.

(b) Joint talent demand-supply distribution.

(c) Normalized volume distribution.

(d) Correlation of connected companies in DyCP-HG.

Figure 2: Distributions of the IT dataset: (a) the overall talent demand and supply distribution, (b) joint talent demand-supply distribution of companies, (c) normalized volume distribution of talent demand and supply, (d) correlation of talent demand and supply between connected companies and positions in DyCP-HG.

2.2 Talent Demand and Supply Quantification

In practice, companies are usually unwilling to publish their talent data to maintain the competitive edge in the war of talent acquisition. Measuring the precise volume of talent demand and supply becomes a challenging task. Fortunately, the public available job posting and employee work experience data can be adopted as the proxy to reflect the labor market variation.

Talent demand is the quantity that a company requires for a position in a period aiming at business expansion or filling the gap of negative turnover. Usually, when talent demand for a position occurs in a company, they will release job postings calling for suitable and intended candidates, leading a positive correlation between talent demand and the corresponding job posting [34]. And talent supply is the number of candidates that a company provides for the labor market [1, 22], *i.e.*, the leaving employees. We use the job hopping in work experiences data to estimate the amount that a company supplies talents for other ones. In this work, we use the collected both job posting and job hopping data to quantify talent demand and talent supply as follows.

Definition 1. ***Talent Demand and Talent Demand Sequence.*** *Talent demand $D^t_{c,p}$ is defined as the number of job postings published by company c for position p at timestamp t. Correspondingly, the talent demand sequence is defined as a time-series $D^{t_s,t_e}_{c,p} = \{D^t_{c,p} | t_s \leq t \leq t_e\}$, where t_e and t_s are start and end time of the sequence.*

Definition 2. ***Talent Supply and Talent Supply Sequence.*** *Talent supply $S^t_{c,p}$ is defined as the number of job hopping from company c and position p at timestamp t. Correspondingly, the talent supply sequence is defined as a time-series $S^{t_s,t_e}_{c,p} = \{S^t_{c,p} | t_s \leq t \leq t_e\}$, where t_e and t_s are start and end time of the sequence.*

Note following previous studies [37], we discretize the continuous-time into a sequence of equal-length time intervals (*i.e.*, one month), and align talent demand and supply sequences. Moreover, we augment the uni-variate sequence by incorporating sequential segmentation, value normalization and trend types labeling. Please refer to Appendix A.2 for details. The enhanced talent demand-supply sequences describe the fine-grained labor market trend variation and can be adopted for subsequent talent demand-supply forecasting.

We conduct the preliminary analysis of the talent demand and supply sequences on IT dataset. Note the distributions on other datasets are similar, and we omit them due to the page limit. Figure 2(a) and Figure 2(b) report the overall talent demand and supply trend over time and the demand-supply correlation of each company, respectively. Obviously, the talent demand and supply are time-varying and positively correlated, motivating us for joint prediction. Figure 2(c) further depicts normalized volume distribution of talent demand and supply. We observe the highly synchronized long-tail distribution of talent demand and supply, where the demand-supply volume of over 80% companies is less than 0.25. In fact, the long-tail distribution inspires us to explore advanced meta-learning techniques for companies with a few data samples, detailed in Section 3.5.

2.3 Dynamic Company-Position Heterogeneous Graph Construction

By extracting the job-hopping information from the work experience data, we construct the dynamic company-position heterogeneous graph to capture the co-evolving patterns and relationship between companies and positions.

Definition 3. ***Company-Position Heterogeneous Graph*** *(CP-HG). The Company-Position Heterogeneous Graph is defined as $G = (V, E)$, where $V = V_C \cap V_P$ and $E = E_{c,c} \cap E_{p,p} \cap E_{c,p}$. V_C and V_P are nodes of all companies and positions. $E_{c,p}$ indicates demand and supply values of the company-position pair. $E_{c,c}$ and $E_{p,p}$ indicates the job hopping between two companies and two positions.*

Definition 4. ***Dynamic Company-Position Heterogeneous Graph*** *(DyCP-HG). The Dynamic Company-Position Heterogeneous Graph is defined as $\mathcal{G}^{t_s,t_e} = (G^{t_s}, \cdots, G^{t_e})$ where t_s and t_e are the start and end timestamp, and G^t is a CP-HG at timestamp t satisfying $t_s \leq t \leq t_e$.*

On the one hand, the frequent job-hopping between companies and positions describes a high relevance between company- and position- pairs, which can positively influence the labor market trend [5]. On the other hand, the preserved heterogeneous relationship between companies and position described by the edge connection also provides extra information for prediction [12]. Figure 2(d) reports the averaged Pearson correlation of connected companies in DyCP-HG. By varying k, the portion of company-pairs with highest edge weights (*i.e.*, the number of job hopping between the company-pair), from 100% to 10%, the pair-wise correlation of talent demand and supply sequences increases significantly.

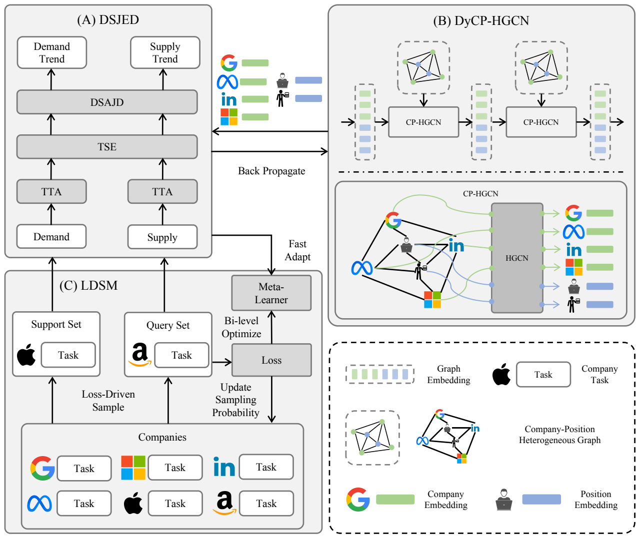


Figure 3: An overview of the DH-GEM framework.

Such results demonstrate the effectiveness of DyCP-HG for linking highly correlated nodes, which can be exploited to provide extra knowledge for talent demand and supply prediction.

3 TALENT DEMAND-SUPPLY JOINT PREDICTION

In this section, we formulate the *Talent Demand-Supply Joint Prediction* (TDSJP) task and present the DH-GEM framework in detail.

3.1 Problem Formulation

PROBLEM 1. *Given the talent demand and supply sequences $D_{c,p}^{t_s,t_e}$ and $S_{c,p}^{t_s,t_e}$ of the company-position pair (c, p), as well as the corresponding dynamic company-position heterogeneous graph $\mathcal{G}^{t_s,t_e}$, we aim to simultaneously predict demand and supply for pair (c, p) in the next timestamp, as*

$$y_D^{t_e+1}, y_S^{t_e+1} \leftarrow \mathcal{F}(D_{c,p}^{t_s,t_e}; S_{c,p}^{t_s,t_e}; \mathcal{G}^{t_s,t_e}), \tag{1}$$

where $y_D^{t_e+1}$ and $y_S^{t_e+1}$ are the estimated company-position wise talent demand and supply trend in the next timestamp, and $\mathcal{F}(\cdot)$ is the joint prediction function we aim to learn.

3.2 Framework Overview

Figure 3 overviews the proposed DH-GEM framework, which includes three essential tasks: (1) modeling joint dependency between demand-supply sequences, (2) capturing fine-grained correlation among companies and positions, and (3) learning transferable knowledge from data-sufficient tasks to improve the long-tail prediction performance. Specifically, for the first task, we propose a *Demand-Supply Joint Encoder-Decoder* (DSJED) to attentively capture the intrinsic correlation between demand and supply variation. After that, we devise a *Dynamic Company-Position Heterogeneous Graph Convolutional Network* (DyCP-HGCN) to incorporate the time-varying correlation between companies and positions. Finally, we propose a *Loss-Driven Sampling based Meta-learner* (LDSM) to optimize the overall framework by shedding more light on long-tail prediction tasks with limited data.

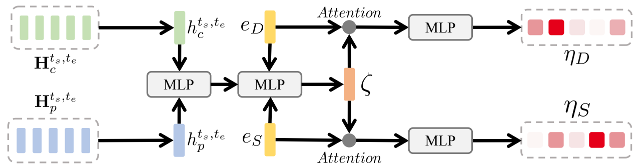


Figure 4: The architecture of DSAJD.

3.3 Demand-Supply Joint Encoder-Decoder

Considering talent demand and supply are intrinsically correlated with each other, we thus design the *Demand-Supply Joint Encoder-Decoder* (DSJED) to attentively capture the intrinsic correlation between demand and supply for better prediction.

As each element of demand and supply sequences is a real value, to represent them comprehensively for sequential modeling, we propose the *Trend Temporal Amplifier* (TTA). Specifically, we first map these scalars to high-dimensional vectors by multi-layer perceptron. Further, to reflect the information of the specific company and position, we concatenate these vectors with the corresponding temporal company and position embedding h_c^t and h_p^t, and go through a multi-layer perceptron to obtain the representation of each element in two sequences, *i.e.*, e_D^t and e_S^t. The construction of the temporal company and position embedding h_c^t and h_p^t will be introduced in the next subsection.

Afterwards, to obtain the representation of two sequences, we leverage the encoder of Transformer [30] with sinusoidal positional encoding to design the *Trend Sequential Encoder* (TSE). We apply TSE for demand and supply sequence respectively. Particularly, the TSE shares parameters for two sequences, which is beneficial to catch common evolving patterns of both demand and supply sequences. In such a way, TSE encodes two sequences to represent the trend embedding of demand and supply as

$$e = TransformerEncoder(\{e^{t_s}, \cdots, e^{t_e}\}), \tag{2}$$

where e can represent the trend embedding of demand e_D or supply e_S, and e^t can represent e_D^t or e_S^t.

As discussed in Section 2.2, demand and supply have a strong connection reflected by sequential values. So we propose the *Demand Supply Attentive Joint Decoder* (DSAJD) shown in Figure 4, to decode demand and supply sequential encoding with consideration of mutual relationships. First, we generate the time-evolving embedding for company c and position p respectively

$$h^{t_s,t_e} = \mathbf{M}_{t_s,t_e} \cdot \mathbf{A} \cdot \mathbf{H}^{t_s,t_e}, \tag{3}$$

where h^{t_s,t_e} can be $h_c^{t_s,t_e}$ or $h_p^{t_s,t_e}$, $\mathbf{M}_{t_s,t_e}$ is a 0-1 vector that indicates whose indices between t_s and t_e as one, $\mathbf{A}$ is a learnable attentive vector and $\mathbf{H}^{t_s,t_e}$ is the list of $(h_{c_1}^{t_s}, h_{c_2}^{t_e}, \cdots)$ or $(h_{p_1}^{t_s}, h_{p_2}^{t_e}, \cdots)$. Then, to devise the company-position-aware demand-supply joint sequential feature ζ, we fuse e_D, e_S, $h_c^{t_s,t_e}$ and $h_p^{t_s,t_e}$ as follows

$$\zeta = MLP(e_D || e_S || MLP(h_c^{t_s,t_e} || h_p^{t_s,t_e})), \tag{4}$$

where $MLP(\cdot)$ represents the multi-layer perceptron, the $||$ represents the concatenation operation. Furthermore, to achieve information sharing, two attentive modules merge ζ with e_D and e_E

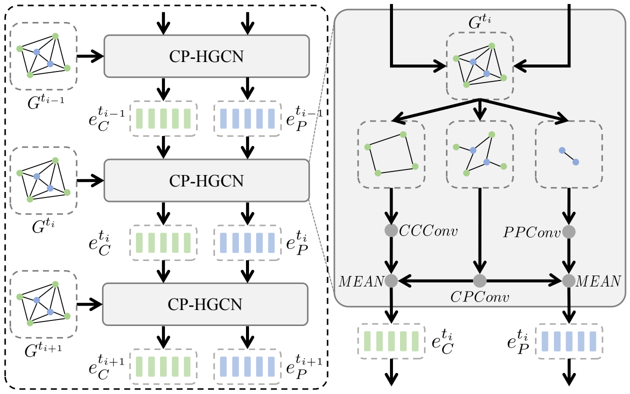


Figure 5: The architecture of DyCP-HGCN.

respectively as the new feature $\hat{e_D}$ and $\hat{e_E}$, which is defined as

$$\hat{e} = \mathbf{w}[e;\zeta], \tag{5}$$

where e can be e_D or e_S, $\hat{e}$ can be $\hat{e_D}$ or $\hat{e_E}$, $\mathbf{w}$ denotes learnable parameters. Finally, $\hat{e_D}$ and $\hat{e_E}$ are fed to two independent multi-layer perceptrons. And the output will be operated by $LogSoftMax$ as vector η_D and η_S. Specifically, the dimension of output vector equals to the number of trend types, and the i-th element is the predicted probability of the trend type i. Moreover, we use *ArgMax* to transform η_D and η_S into the trend type y_D and y_S as follows

$$y = \arg\max_{i\in[1,N_y]}\eta^i, \tag{6}$$

where y is y_D or y_S, η is η_D or η_S, η^i is the i-th element of η, and N_y denotes the number of trend types.

3.4 Dynamic Company and Position Representation Learning

To incorporate the dynamic relationship between companies and positions, we propose the *Dynamic Company-Position Heterogeneous Graph Convolutional Network* (DyCP-HGCN) to preserve common knowledge between companies and positions representations. As shown in Figure 5, the DyCP-HGCN leverages dynamic recurrent process to encode the DyCP-HG with the output of node embedding for each timestamp. We denote DyCP-HGCN as Φ and define it as

$$(\mathbf{H}_C^{t_s},\cdots,\mathbf{H}_C^{t_e}),(\mathbf{H}_P^{t_s},\cdots,\mathbf{H}_P^{t_e}) = \Phi(\mathcal{G}^{t_s,t_e}), \tag{7}$$

where $\mathbf{H}_C^t$ and $\mathbf{H}_P^t$ are the company and position node embedding at timestamp t.

3.4.1 Company-Position Heterogeneous Graph Convolutional Network. As a cell of DyCP-HGCN, *Company-Position Heterogeneous Graph Convolutional Network* (CP-HGCN) is designed to learn the static company and position embedding h_c and h_p. We generally define CP-HGCN $\phi(\cdot)$ as follows

$$\mathbf{H}_C,\mathbf{H}_P = \phi(G;\mathbf{H}_C';\mathbf{H}_P'), \tag{8}$$

where $\mathbf{H}_C'$ and $\mathbf{H}_P'$ are input company and position node embedding, $\mathbf{H}_C$ and $\mathbf{H}_P$ are output ones. Specifically, $\phi(\cdot)$ contains three steps.

Firstly, to handle the heterogeneity, we separate a CP-HG G into three sub-graphs according to their types of edges, *i.e.*, $G(V_c,E_{c,c})$, $G(V_p,E_{p,p})$ and $G(V,E_{c,p})$ respectively.

Secondly, for three sub-graphs of demand-supply edges, *i.e.*, $E_{c,p}$, company-hopping edges $E_{c,c}$, and position-hopping edges $E_{p,p}$, we adopt three graph convolutional operation $CPConv(\cdot)$, $CCConv(\cdot)$ and $PPConv(\cdot)$ respectively to generate the node representation by aggregating the neighboring information. Three convolutional operation can be generally defined as

$$h_u = \sigma(b + \sum_{v\in N_u}\frac{w_{uv}}{\sqrt{|N_u|}\sqrt{|N_v|}}\cdot(h_v\cdot\mathbf{W})), \tag{9}$$

where we denote h_u as the embedding of node u, σ as ReLU activation function, b and $\mathbf{W}$ as learnable parameters, w_{uv} as the edge weight between node u and v, and $|N_v|$ as the number of neighbors of node v.

Thirdly, considering the company node embedding $\mathbf{H}_C$ is produced by $CPConv(\cdot)$ and $CCConv(\cdot)$, while position node embedding $\mathbf{H}_P$ by $CPConv(\cdot)$ and $PPConv(\cdot)$, we leverage the mean updating operation for company and position embedding respectively to obtain the final output embedding of $\phi(\cdot)$.

3.4.2 Dynamic Recurrent Process. To continuously learn the temporal pattern of company and position embedding and generate representations for company and position at each timestamp, we adopt the recurrent process as shown in Figure 5. Specifically, we use the learned embedding of the previous timestamp $t-1$ as the input of a recurrent cell (*i.e.*, CP-HGCN $\phi(\cdot)$) and output the new embedding at the current timestamp t. Based on the single cell function $\phi(\cdot)$, the recurrent process at timestamp t is defined as

$$\mathbf{H}_C^t,\mathbf{H}_P^t = \phi(G^t;\mathbf{H}_C^{t-1};\mathbf{H}_P^{t-1}), t_s\le t\le t_e. \tag{10}$$

The $\mathbf{H}_C^{t_s-1}$ and $\mathbf{H}_P^{t_s-1}$ are initialized randomly. In this way, we can get the list of company and position embedding from t_s to t_e orderly, *i.e.*, $(\mathbf{H}_C^{t_s},\cdots,\mathbf{H}_C^{t_e}),(\mathbf{H}_P^{t_s},\cdots,\mathbf{H}_P^{t_e})$.

3.5 Loss-Driven Sampling based Meta-learner

According to preliminary analysis in Section 2.2, the demand and supply of different companies follow the long-tail distribution. End-to-end models naturally perform well for companies with massive training data, but worse for these long-tail companies with "few-shot" instances. Therefore, we introduce the *Loss-Driven Sampling based Meta-learner* (LDSM) to train the overall end-to-end prediction framework, with a special strategy on long-tail tasks.

First, we optimize model by Negative Log-Likelihood Loss with Poisson distribution (*i.e.*, $P(Y=y)=\frac{\eta^y}{y!}\exp(-\eta)$) as

$$\mathcal{L} = \sum -\log P(Y=y) = \sum \exp(-\eta) - y\eta + \log y!, \tag{11}$$

where y represents y_D or y_S and η represents η_D or η_S. To simplify calculation, the last term can be approximated according to the *Stirling's Formula*, $\log y! \approx y\log y - y + \frac{1}{2}\log(2\pi y)$. The overall objective is the combination of both the demand and supply prediction loss $\mathcal{L}_{overall} = \mathcal{L}_D + \mathcal{L}_S$, which is optimized via back-propagation.

Furthermore, inspired by the recent success of Model-Agnostic Meta-Learning (MAML) [10] on learning good parameter initialization for few-shot tasks, we formulate the TDSJP as a meta-learning problem to alleviate the long-tail distribution issue. The goal is to extract globally shared meta-knowledge from diverse companies to enable fast adaptation and more accurate predictions when forecasting demand-supply for companies with limited data.

In particular, we formulate demand-supply prediction of each company as individual tasks and construct the *Taskset* as below.

Table 1: The overall performance of talent demand-supply joint prediction on three real-world datasets.

	IT			FIN			CONS		
	Accuracy	Weighted-F1	AUROC	Accuracy	Weighted-F1	AUROC	Accuracy	Weighted-F1	AUROC
LV	0.3750	0.3019	0.6690	0.3906	0.3331	0.6831	0.3739	0.2999	0.6767
LR	0.5099	0.4776	0.8011	0.5250	0.5078	0.8100	0.4962	0.4812	0.7899
GBDT	0.6134	0.6083	0.8778	0.5981	0.5938	0.8683	0.5469	0.5398	0.8344
LSTM	0.6034	0.5995	0.8732	0.6001	0.5860	0.8697	0.5632	0.5581	0.8458
Transformer	0.6343	0.6375	0.8950	0.6191	0.6180	0.8842	0.5737	0.5726	0.8551
DH-GEM	**0.6813**	**0.6840**	**0.9168**	**0.6791**	**0.6825**	**0.9155**	**0.6230**	**0.6249**	**0.8883**

Definition 5. ***Taskset.*** *Denote $\mathcal{T}_i$ as the demand-supply prediction task for company $c_i \in C$, the taskset is defined as the set of all company-specific demand-supply prediction tasks $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_i\}_{i=1}^{|C|}$, where $|C|$ is the number of companies.*

Different from existing meta-learning methods that sample tasks with equal probability, we devise the loss-driven sampling strategy during meta-learning to enforce the model to shed more light on long-tail tasks. Intuitively, according to Equation 11, the task with higher training loss indicates the larger prediction error, requiring additional learning efforts. We detail the LDSM below.

Initially, we randomly initialize model parameters θ and set equal probability for each task $\mathcal{T}_i$ as $p_i^{(0)} = \frac{1}{|C|}$. In the j-th epoch, we run several steps model adaption. In each adaptation step, according to the sample probabilities, we sample a task $\mathcal{T}_i$ as the support set, evaluate its loss $\mathcal{L}_{\mathcal{T}_i}^{(j)}(f_\theta)$ and derive the parameter update by

$$\theta' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{(j)}(f_\theta), \tag{12}$$

where θ' is the updated parameters, $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{(j)}(f_\theta)$ is the gradient of j-th epoch loss for learner f_θ of task $\mathcal{T}_i$. Correspondingly, we sample a task $\mathcal{T}_i'$ as the query set in the same way and evaluate its gradient $\nabla_\theta \mathcal{L}_{\mathcal{T}_i'}^{(j)}(f_\theta)$. In the end of j-th epoch, we leverage bi-level optimization to update θ by

$$\theta \leftarrow \theta - \beta \nabla_{\theta'} \sum \mathcal{L}_{\mathcal{T}_i'}^{(j)}(f_{\theta'}), \tag{13}$$

where $\sum \mathcal{L}_{\mathcal{T}_i'}^{(j)}(f_{\theta'})$ is evaluated in meta-training steps before. Before the next epoch, we update the sampling probability by

$$p_i^{(j+1)} = \frac{\mathcal{L}_{\mathcal{T}_i'}^{(j)}}{\sum_{\mathcal{T}_k' \in \boldsymbol{\mathcal{T}}} \mathcal{L}_{\mathcal{T}_k'}^{(j)}}, \tag{14}$$

where $p_c^{(j+1)}$ is the sampling probability for $(j+1)$-th epoch, $\mathcal{L}_{\mathcal{T}_i}^{(j)}$ is the validated loss in j-th epoch. Please refer Appendix A.3 for the complete algorithm.

4 EXPERIMENTS

4.1 Experimental Setup

In this subsection, we introduce the metrics of our experiments[1] and the baselines we compared. The hyper-parameters and implementation details are provided in Appendix A.4.

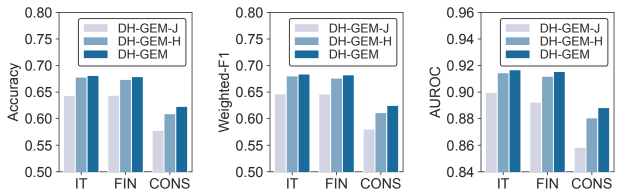
[1] Complete code are available on https://github.com/gzn00417/DH-GEM.

4.1.1 Metrics. In our experiments, the prediction of demand and supply is a multiclass classification task. Therefore, we mainly adopt Accuracy to evaluate the overall performance of models. Besides, we also use weighted F1 score (Weighted-F1) and area under receiver operating characteristic (AUROC) for evaluation.

4.1.2 Baselines. We compare DH-GEM with the following baselines, including statistic based method, traditional machine learning methods, and deep learning methods.
1) **LV** (Last Value) is a statistical classifier only using the last trend value of talent demand or supply.
2) **LR** (Logistic Regression) is a linear machine learning model.
3) **GBDT** (Gradient Boosting Decision Tree) is an additive model in a forward stage-wise fashion.
4) **LSTM** [14] (Long Short-Term Memory) is a typical recurrent neural network for time series prediction.
5) **Transformer** [30] is an attention mechanism based model which is very popular for modeling various sequence data.

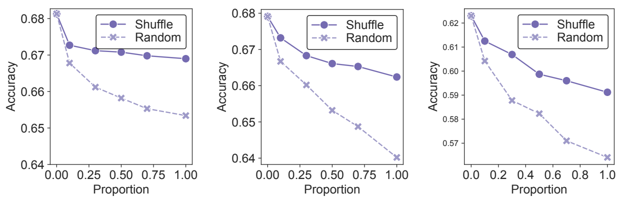
4.2 Overall Results

The overall results of DH-GEM and all baselines on all three datasets (*i.e.*, IT, FIN and CONS) are reported in Table 1. To summarize the overall joint prediction performance, the reported Accuracy, F1 and AUROC value in Table 1 are the total average of both demand and supply prediction. Detailed results of the separate demand and supply prediction for each dataset are shown in Table 3, Table 4 and Table 5 of Appendix A.5, which present consistent conclusion with the averaged results. Obviously, DH-GEM significantly outperforms all baseline models in terms of all three metrics on all three datasets. Specifically, with the dynamic heterogeneous graph enhanced meta learning, the accuracy of DH-GEM achieves at least (7.4%, 9.69%, 8.59%) improvements on IT, FIN and CONS compared with all baselines. Moreover, we discover that Transformer, an attention based deep learning model, outperforms the linear model (LR), tree-based model (GBDT) and recurrent neural network (LSTM), which consistently verifies the advantages of the attention architecture we adopted in our framework. Furthermore, compared with GBDT which only applies observed demand-supply values as input features, DH-GEM achieves significant improvement by capturing the co-evolving pattern of the demand-supply sequence via joint sequential modeling. Finally, compared with LSTM and Transformer which only incorporate temporal dependencies, DH-GEM achieves better performance by absorbing common knowledge between companies and positions through graph representation learning.



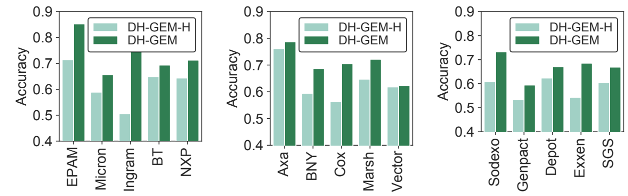
(a) Accuracy (b) Weighted-F1 (c) AUROC

Figure 6: Ablation study on TDSJP task.



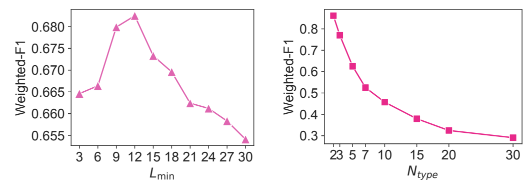
(a) IT (b) FIN (c) CONS

Figure 7: Effectiveness of graph embeddings.



(a) IT (b) FIN (c) CONS

Figure 8: The performance of long tail companies.



(a) $L_{\min}$ (b) N_y

Figure 9: Parameter sensitivity of $L_{\min}$ and N_y.

4.3 Ablation Study

To evaluate the effectiveness of DyCP-HGCN module and meta learning strategy, we conduct ablation study with two variant DH-GEM models. **DH-GEM-J** is a variant of DH-GEM without the DyCP-HGCN module. **DH-GEM-H** is another variant of DH-GEM without the LDSM module. As shown in Figure 6, removing DyCP-HGCN or LDSM leads to remarkable performance degradation, which verifies the effectiveness of these modules. Specifically, the accuracy of DH-GEM-J decreases 5.87% on IT, 5.47% on FIN and 7.80% on CONS. The most important reason is that DyCP-HGCN brings rich common information by connecting companies and positions. Besides, the accuracy of DH-GEM-H decreases 0.46% on IT, 0.78% on FIN, and 2.20% on CONS. Despite the performance gain of LDSM is not as large as DyCP-HGCN, LDSM effectively helps the prediction model on forecasting long-tail data. We provide additional experiments on long-tail companies in Section 4.4.

To further validate the effectiveness of DyCP-HGCN, we evaluate it with two embedding modification strategies. Specifically, **Shuffle** permutes the learned graph embeddings between different companies. **Random** replaces the learned graph embeddings with random values from the Gaussian distribution $\mathcal{N}(0, 1)$.

Figure 7 shows the impact of different proportion of modified embeddings (*e.g.*, 0.7 Shuffle means 70% embeddings are shuffled). We can make two observations, 1) modifying more embeddings leads to more significant prediction performance degradation. 2) DH-GEM performs much worse with larger proportion of random modified embeddings. This is because shuffle still keeps the general market information. The above results demonstrate the learned representation plays a crucial role in demand-supply prediction.

4.4 Effectiveness on Handling Long-tail Tasks

To show the effectiveness of the LDSM, we further compare DH-GEM with DH-GEM-H (*i.e.*, the model variant without LDSM) on five long-tail companies in three datasets. As shown in Figure 8, DH-GEM achieves remarkable accuracy improvements on long-tail tasks in IT (*e.g.*, Ingram 48.48%), FIN (*e.g.*, Cox 25.16%) and CONS (*e.g.*, Exxon 26.09%). Looking into results, we find the performance of companies with a handful of data has been greatly improved by more frequent training, which validates effectiveness of our LDSM module. Together with the results reported in Figure 6, we can conclude that DH-GEM achieves more robust and bias-free prediction results for companies with a varying number of instances.

4.5 Parameters Sensitivity

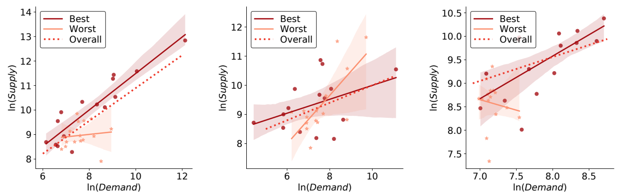
In this subsection, we first analyse the joint sensitivity of temporal embedding dimension $dim_t = |e^t|$ and graph node embedding dimension $dim_g = |h_c| = |h_p|$ by using accuracy on IT. The best parameter combination for DH-GEM is $dim_t = 16$ and $dim_g = 4$, which achieves accuracy of 0.6813. The detailed experiment results are shown in Appendix A.6. The potential reason is that the sequential data provides more information for predicting, and too large embedding dimension may introduce unnecessary training difficulties for the model to converge.

Besides, Figure 9(a) reports the influence of the input length. As can be seen, the model achieves optimal results with $L_{\min} = 12$. Too long or too short input sequence degrade the performance. As too long sequences may introduce more noise, while too short sequences can not provide sufficient sequential dependent information. We choose $L_{\min} = 12$ in the rest evaluations.

Moreover, we report the sensitivity of the number of trend types N_y in Figure 9(b). In fact, as N_y increases, the model has to classify more types of talent trend, and intuitively it needs to capture more complex patterns to recognize the difference between types. Therefore, the performance will accordingly decrease.

4.6 Qualitative Study

In Figure 2(c), we have uncovered the correlation between talent demand and supply. In this subsection, we further conduct a qualitative study to demonstrate the effectiveness of talent demand-supply joint prediction. Specifically, we choose the top 15 companies with the best performance (*i.e.*, highest accuracy) and the bottom 15



(a) IT (b) FIN (c) CONS

Figure 10: Talent demand-supply distribution of companies with the best and worst performance.

companies with the worst performance (*i.e.*, lowest accuracy). We visualize the correlation between demand and supply of these two groups with the overall dataset, as shown in Figure 10. As can be seen, the regression line of top companies is more reasonable, *i.e.*, positively upgrading, and close to the overall regression line. In contrast, the regression line of bottom companies shows a diverged distribution with the overall dataset. The above observations validate our key assumption that the talent demand and supply variation are interconnected and can be incorporated to improve the prediction effectiveness. Moreover, further optimization on these companies with diverged demand and supply variation can be applied in the future to improve the overall performance.

4.7 System Deployment

DH-GEM has been deployed in the intelligent human resource system of a cooperative partner. For human resource users in companies, we design the system shown in Figure 11 that present both detailed talent demand and supply historic values and the forecasted future trend, and select the top demand or supply of companies or positions with detailed information. These give guidance for employers to grasp the future demand and supply to adjust recruitment strategy and chances to poach talents from supplying companies for our demanding positions. Besides, more examples can be referred to Appendix A.7 including views for the government and talents.

5 RELATED WORK

Overall, the related works of this paper can be summarized into four parts, namely *labor market trend forecasting*, *time series prediction*, *graph neural network*, and *meta learning*.

Labor Market Trend Forecasting. As talent becomes the significant competitiveness between companies, growing attention has been paid to labor trend analysis [6]. For example, a Generalized Least Squares based model was built in [19] by heuristic methods on a governmental realistic data, and analysis on labor market collapse, recovery and evidence of policy response at COVID-19 onset have been studied [3]. With the technology of machine learning, new emerging methods gradually substitute the traditional ones. MTLVM [39] is a sequential latent variable model for learning the labor market trend. Focusing on talent demand forecasting, TDAN [37] is a data-driven neural sequential approach targeting on fine-grained talent demand and its sparsity issue. NEMO [16] is designed for job mobility, *i.e.*, talent supply prediction using contextual embedding. Ahead [36] aims at talent next career move forecasting with a tailored heterogeneous graph neural network and Dual-GRU. Fortune Teller [21] predicts upgrading career paths

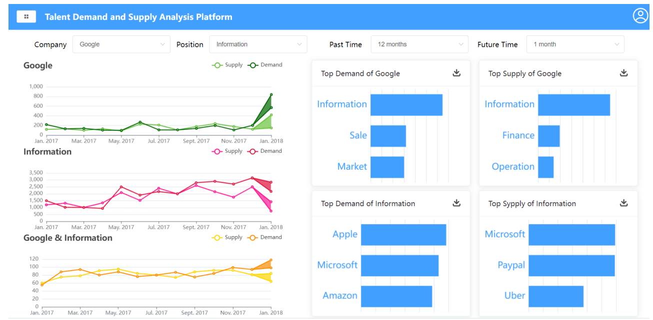


Figure 11: The screenshot of our deployed system.

through fusing information on social networks. However, these works either focus on talent demand or supply prediction but overlook the intrinsic correlation between the talent demand and supply variation. In addition to demand and supply prediction, other topics of labor market skill validation [26, 29] have been extensively studied in recent years and deployed in human resource systems.

Time Series Prediction. Recently, deep learning based sequential approaches such as Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM) [14] have gained unprecedented popularity, due to its capability on learning effective feature representations from complex time series. Along this line, Transformer [30] is a novel Encoder-Decoder architecture solely based on attention mechanisms, which shows a unique superiority on parallel processing and has been adopted as a workhorse for various sequential forecasting tasks. Meanwhile, talent analysis has been formulated as time series problem, while sequential models have been applied in labor market trend forecasting. For example, TDAN [37] leveraged attention based sequential model to forecast talent demand, and Ahead [36] used GRU to predict company, position and duration of the next career move.

Graph Neural Network. Graph Neural Network (GNN) has been demonstrated as a powerful tool for modeling non-Euclidean relational data structures. As a basic variant, Graph Convolutional Network (GCN) [15] preserves structural proximity for homogeneous graph nodes with a graph convolutional operator. Besides, Heterogeneous Graph Attention Network (HAN) [31] captures sophisticated node-level and semantic-level dependencies for heterogeneous nodes via a hierarchical attention module. In this work, we propose a dynamic heterogeneous graph neural network to capture the complicated correlation between companies and positions.

Meta-learning. As an emerging learning paradigm, meta-learning has been recognized as a promising way for handling few-shot learning tasks. As one of the most representative meta-learning approaches, Model-Agnostic Meta-Learning (MAML) [10] learns optimal initial parameters of neural networks to transfer globally shared knowledge to new tasks with limited data. As another example, Prototypical network [28] achieves better classification accuracy by computing distances between new data and prototype representations. In this work, we leverage MAML based meta-learning to improve the prediction performance for long-tail companies.

6 CONCLUSION

In this paper, we presented DH-GEM, a dynamic heterogeneous graph enhanced meta-learning framework to cope with the talent demand-supply joint prediction (TDSJP) problem. We first constructed fine-grained demand-supply sequences with a dynamic company-position graph to represent the co-evolve patterns. Then we devised the demand-supply joint encoder-decoder (DSJED) to mine shared information between demand and supply sequences implicitly. After that, we proposed the dynamic company-position heterogeneous graph convolutional network (DyCP-HGCN) to extract dynamic company and position representation for better fine-grained demand-supply prediction. Finally, a Loss-Driven Sampling based Meta-learner (LDSM) was devised to train the prediction framework to transfer knowledge for long-tail tasks from companies with sufficient data. Extensive experiments on three real-world datasets demonstrated that DH-GEM achieves the best performance compared with five baselines. Importantly, we have deployed DH-GEM as a core functional component of the intelligent system of a cooperative partner.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 62102110, 61836013) and Foshan HKUST Projects (FSUST21-FYTRI01A, FSUST21-FYTRI02A). And this work was done when the first author interned in Baidu Research.

REFERENCES

[1] Yossi Aviv. 2001. The effect of collaborative forecasting on supply chain performance. *Management science* 47, 10 (2001), 1326–1343.

[2] José Azar, Ioana Marinescu, and Marshall Steinbaum. 2022. Labor market concentration. *Journal of Human Resources* 57, S (2022), S167–S199.

[3] Alexander W Bartik, Marianne Bertrand, Feng Lin, Jesse Rothstein, and Matt Unrath. 2020. *Measuring the labor market at the onset of the COVID-19 crisis*. Technical Report. National Bureau of Economic Research.

[4] J Stewart Black and Patrick van Esch. 2021. AI-enabled recruiting in the war for talent. *Business Horizons* 64, 4 (2021), 513–524.

[5] Ron Boschma, Rikard H Eriksson, and Urban Lindgren. 2014. Labour market externalities and regional growth in Sweden: The importance of labour mobility between skill-related industries. *Regional Studies* 48, 10 (2014), 1669–1690.

[6] Peter Cappelli. 2008. Talent management for the twenty-first century. *Harvard business review* 86, 3 (2008), 74.

[7] Peter Cappelli and JR Keller. 2014. Talent management: Conceptual approaches and practical challenges. *Annu. Rev. Organ. Psychol. Organ. Behav.* 1, 1 (2014), 305–331.

[8] Di Chai, Leye Wang, and Qiang Yang. 2018. Bike flow prediction with multi-graph convolutional networks. In *Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems*. 397–400.

[9] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. 2019. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems* 21, 11 (2019), 4883–4894.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR, 1126–1135.

[11] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining*. 2221–2231.

[12] Omar A Guerrero and Robert L Axtell. 2013. Employment growth through labor flow networks. *PloS one* 8, 5 (2013), e60808.

[13] Qi He, Jaewon Yang, and Baoxu Shi. 2020. Constructing knowledge graph for social networks in A deep and holistic way. In *Companion Proceedings of the Web Conference 2020*. 307–308.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[15] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[16] Liangyue Li, How Jing, Hanghang Tong, Jaewon Yang, Qi He, and Bee-Chung Chen. 2017. Nemo: Next career move prediction with contextual embedding. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 505–513.

[17] Shan Li, Baoxu Shi, Jaewon Yang, Ji Yan, Shuai Wang, Fei Chen, and Qi He. 2020. Deep job understanding at linkedin. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2145–2148.

[18] Hao Lin, Hengshu Zhu, Yuan Zuo, Chen Zhu, Junjie Wu, and Hui Xiong. 2017. Collaborative company profiling: Insights from an employee's perspective. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[19] Lin Lin, Jeou-Shyan Horng, Yi-Chen Chen, and Chang-Yen Tsai. 2011. Factors affecting hotel human resource demand in Taiwan. *International Journal of Hospitality Management* 30, 2 (2011), 312–318.

[20] Yan Liu, Bin Guo, Daqing Zhang, Djamal Zeghlache, Jingmin Chen, Sizhe Zhang, Dan Zhou, Xinlei Shi, and Zhiwen Yu. 2021. MetaStore: a task-adaptative meta-learning model for optimal store placement with multi-city knowledge transfer. *ACM Transactions on Intelligent Systems and Technology (TIST)* 12, 3 (2021), 1–23.

[21] Ye Liu, Luming Zhang, Liqiang Nie, Yan Yan, and David Rosenblum. 2016. Fortune teller: predicting your career path. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[22] Erin E Makarius and Mahesh Srinivasan. 2017. Addressing skills mismatch: Utilizing talent supply chain management to enhance collaboration between companies and talent suppliers. *Business horizons* 60, 4 (2017), 495–505.

[23] Qingxin Meng, Hengshu Zhu, Keli Xiao, Le Zhang, and Hui Xiong. 2019. A hierarchical career-path-aware neural network for job mobility prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 14–24.

[24] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1720–1730.

[25] Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Chao Ma, Enhong Chen, and Hui Xiong. 2020. An enhanced neural network approach to person-job fit in talent recruitment. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–33.

[26] Baoxu Shi, Jaewon Yang, Feng Guo, and Qi He. 2020. Salience and market-aware skill extraction for job targeting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2871–2879.

[27] Joseph Kee Ming Sia. 2013. University choice: Implications for marketing and positioning. *Education* 3, 1 (2013), 7–14.

[28] Jake Snell, Kevin Swersky, and Richard S Zemel. 2017. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175* (2017).

[29] Ying Sun, Fuzhen Zhuang, Hengshu Zhu, Qi Zhang, Qing He, and Hui Xiong. 2021. Market-oriented job skill valuation with cooperative composition neural network. *Nature communications* 12, 1 (2021), 1–12.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[31] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*. 2022–2032.

[32] Xiao Yan, Jaewon Yang, Mikhail Obukhov, Lin Zhu, Joey Bai, Shiqi Wu, and Qi He. 2019. Social skill validation at LinkedIn. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2943–2951.

[33] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The World Wide Web Conference*. 2181–2191.

[34] Yuliang Yao, Rajiv Kohli, Susan A Sherer, and Jerold Cederlund. 2013. Learning curves in collaborative planning, forecasting, and replenishment (CPFR) information systems: An empirical analysis from a mobile phone manufacturer. *Journal of Operations Management* 31, 6 (2013), 285–297.

[35] Le Zhang, Tong Xu, Hengshu Zhu, Chuan Qin, Qingxin Meng, Hui Xiong, and Enhong Chen. 2020. Large-scale talent flow embedding for company competitive analysis. In *Proceedings of The Web Conference 2020*. 2354–2364.

[36] Le Zhang, Ding Zhou, Hengshu Zhu, Tong Xu, Rui Zha, Enhong Chen, and Hui Xiong. 2021. Attentive heterogeneous graph embedding for job mobility prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2192–2201.

[37] Qi Zhang, Hengshu Zhu, Ying Sun, Hao Liu, Fuzhen Zhuang, and Hui Xiong. 2021. Talent demand forecasting with attentive neural sequential model. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3906–3916.

[38] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.

[39] Chen Zhu, Hengshu Zhu, Hui Xiong, Pengliang Ding, and Fang Xie. 2016. Recruitment market trend analysis with sequential latent variable models. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 383–392.

A APPENDIX

A.1 Mathematical Notations

Table 2: Key mathematical notations.

Symbol	Description
N_y	The number of trend types
C	The companies set
P	The positions set
h_c^t	The embedding of companies c at timestamp t
h_p^t	The embedding of positions p at timestamp t
$\mathbf{H}_C^t$	The embedding matrix of companies at timestamp t
$\mathbf{H}_P^t$	The embedding matrix of positions at timestamp t
L_T	The length of a sequence
N_u	The neighbors of node u on graph
$\mathcal{L}$	The loss value of predictive model
$\mathcal{T}_i$	The task of i-th company in Taskset $\mathcal{T}$
θ	The prediction model parameters
dim_t	The embedding dimension of e^t
dim_g	The node embedding dimension of *DyCP-HGCN*

A.2 Sequence Augmentation Algorithm

Algorithm 1: Sequence Augmentation Algorithm.

Input: All demand or supply sequences, δ_L, δ_t, $L_{\min}$, N_y
Output: Augmented demand or supply dataset
/* $Seq_{c,p}^{t_s,t_e}$ is $D_{c,p}^{t_s,t_e}$ or $S_{c,p}^{t_s,t_e}$. */
/* Sequence segmentation. */
for $i = 0, 1, 2, \cdots$ *and* $L_{\min} + i\delta_L < N_t$ **do**
  $L = L^{\min} + i\delta_L$ // Sequence length
  for $j = 1, 2, \cdots$ *and* $j\delta_t + L - 1 \leq N_t$ **do**
    $t_s = j\delta_t$ // Start time
    $t_e = t_s + L - 1$ // End time
    $x = Seq_{c,p}^{t_s,t_e-1}$ // Observed trend
    $y = Seq_{c,p}^{t_e}$ // Next trend
    $Seq_{c,p}^{t_s,t_e} \leftarrow (x, y)$ // Add splitted data
    dataset $\leftarrow Seq_{c,p}^{t_s,t_e}$;
  end
end
/* Normalize trend values in $[0, 1]$. */
Trend value set $\leftarrow Norm(\{y | for\ (x, y)\ in\ dataset\})$;
/* Labeling trend. */
Sorted trend value set $\leftarrow Sort(Trend\ values\ set)$;
$n \leftarrow len(Sorted\ trend\ value\ set)$;
$\epsilon = \frac{n}{N_y}$;
for $v_i \in Sorted\ trend\ value\ set$ **do**

$$y(v_i) = \begin{cases} 1, & 0 \leq i < \epsilon \\ 2, & \epsilon \leq i < 2\epsilon \\ \cdots & \\ N_y, & (N_y - 1)\epsilon \leq i \leq 1.0 \end{cases}$$

end

A.3 Meta Learning with Loss-Driven Sampling

Here we present the algorithm of meta learning with loss-driven sampling.

Algorithm 2: Meta learning with loss-driven sampling.

Input: All *Taskset* $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_i\}_{i=1}^{|C|}$, fast adaption learning rates α and bi-level optimization learning rate β
Output: Initial model parameters θ
Random initialize model parameters θ;
Initialize sampling probability $\{p_i^{(0)} = \frac{1}{|C|}\}$;
while $j = 1, 2, \cdots, N_{epoch}$ **do**
  for $k = 1, 2, \cdots, N_{step}$ **do**
    Sample a task $\mathcal{T}_i$ from $\mathcal{T}$ with probability $\{p_i^{(j)}\}$;
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{(j)}(f_\theta)$;
    Compute the adapted parameters with gradient descent $\theta' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{(j)}(f_\theta)$;
    Sample a new task $\mathcal{T}_i{}'$ from $\mathcal{T}$ for meta-update;
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i'}^{(j)}(f_\theta)$;
  end
  Update $\theta \leftarrow \theta - \beta \nabla_{\theta'} \sum \mathcal{L}_{\mathcal{T}_i'}^{(j)}(f_{\theta'})$;
  Set sampling probability $\{p_i^{(j+1)} = \frac{\mathcal{L}_{\mathcal{T}_i'}^{(j)}}{\sum_{\mathcal{T}_k' \in \mathcal{T}} \mathcal{L}_{\mathcal{T}_k'}^{(j)}}\}$;
end

A.4 Implementation Details

A.4.1 DH-GEM. For hyper-parameters, we choose the number of trend types $N_y = 5$, the minimum length of fine-grained sequences $L_{\min} = 12$, the embedding dim of e_D^t $dim_t = 16$, the embedding dim of graph node representation $dim_g = 4$, the head number of multi-head attention in TSE as 4, the feed forward dimension as 16, the number of layers in TSE as 2 and the output dimension of any other multi-layer perceptron as 4. We use Adam Optimization with learning rate as 0.01, learning rate scheduler reducing rate as 0.9, step as 4, weight decay as $1e-6$. The DH-GEM is run on the machine with Intel Xeon Gold 6148 @ 2.40GHz, V100 GPU and 64G memory.

A.4.2 Baseline. For traditional models, talent demand and supply input length of LV, LR and GDBT are fixed length of 5, and we also set company and position index as feature for LR and GBDT. For LSTM and Transformer, they follow the structure of DSJED and substitute the TSE as specific encoders, and the DSAJD as two independent 2-layer multi-layer perceptron. The input and output dimension of encoders keep consistent with DH-GEM. Specifically,

- **LV** is implemented by a simple one-layer perceptron.
- **LR** is implemented by a linear regression module and the loss is calculated with $L2$ penalty.
- **GBDT** is implemented by a gradient boosting decision tree with 10 estimators, 0.1 learning rate.

- **LSTM** substitutes the TSE of DSJED as two parameter-independent 2-layer long short-term memory.
- **Transformer** substitutes the TSE of DSJED as two parameter-independent 2-layer Transformer encoder with sinusoidal positional encoding.

A.5 Detailed Demand-Supply Evaluation Result

The detailed performance of separate demand and supply prediction on three datasets are reported in Table 3, Table 4 and Table 5.

Table 3: Detailed evaluation result on IT dataset.

	Demand			Supply		
	Accuracy	F1	AUROC	Accuracy	F1	AUROC
LV	0.2912	0.3087	0.5807	0.4578	0.3950	0.7973
LR	0.4001	0.3312	0.7285	0.6196	0.6239	0.8736
GBDT	0.5970	0.5881	0.8659	0.6298	0.6286	0.8898
LSTM	0.5767	0.5682	0.8628	0.6301	0.6261	0.8842
Transformer	0.6208	0.6175	0.8899	0.6478	0.6492	0.8997
DH-GEM-J	0.6395	0.6351	0.8959	0.6475	0.6541	0.9026
DH-GEM-H	0.6882	0.6893	0.9166	**0.6681**	**0.6700**	0.9109
DH-GEM	**0.6981**	**0.6983**	**0.9199**	0.6645	0.6674	**0.9119**

Table 4: Detailed evaluation result on FIN dataset.

	Demand			Supply		
	Accuracy	F1	AUROC	Accuracy	F1	AUROC
LV	0.2887	0.1756	0.5644	0.4925	0.4917	0.8019
LR	0.4712	0.4377	0.7502	0.5788	0.5780	0.8697
GBDT	0.6024	0.5917	0.8591	0.5937	0.5960	0.8775
LSTM	0.6000	0.5739	0.8602	0.6001	0.5932	0.8794
Transformer	0.6192	0.6088	0.8800	0.6190	0.6237	0.8890
DH-GEM-J	0.6583	0.6584	0.8910	0.6295	0.6325	0.8931
DH-GEM-H	0.7005	0.7010	0.9199	0.6472	0.6513	0.9022
DH-GEM	**0.7025**	**0.7064**	**0.9255**	**0.6556**	**0.6588**	**0.9039**

Table 5: Detailed evaluation result on CONS dataset.

	Demand			Supply		
	Accuracy	F1	AUROC	Accuracy	F1	AUROC
LV	0.2702	0.1958	0.5601	0.4777	0.4041	0.7933
LR	0.4171	0.4005	0.7190	0.5753	0.5618	0.8609
GBDT	0.5167	0.5139	0.8056	0.5771	0.5657	0.8631
LSTM	0.5230	0.5039	0.8204	0.6033	0.5900	0.8711
Transformer	0.6030	0.5966	0.8285	0.5444	0.5451	0.8762
DH-GEM-J	0.5510	0.5510	0.8336	0.6048	0.6054	0.8785
DH-GEM-H	0.5964	0.5988	0.8694	0.6222	0.6204	0.8878
DH-GEM	**0.6192**	**0.6229**	**0.8801**	**0.6268**	**0.6260**	**0.8919**

A.6 Parameter Sensitivity Experiment Result

Table 6: Parameter sensitivity of dim_t and dim_g.

dim_t \ dim_g	2	4	8	16	32
2	0.2430	0.2528	0.3183	0.3064	0.3119
4	0.6227	0.6432	0.6490	0.6420	0.6340
8	0.6262	0.6579	0.6620	0.6567	0.6493
16	0.6321	**0.6813**	0.6790	0.6782	0.6681
32	0.6411	0.6795	0.6782	0.6523	0.6495

A.7 Detailed System Deployment Demonstration

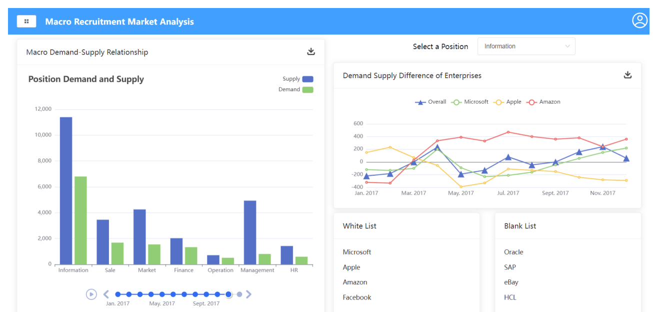


Figure 12: Deployed system: government view.

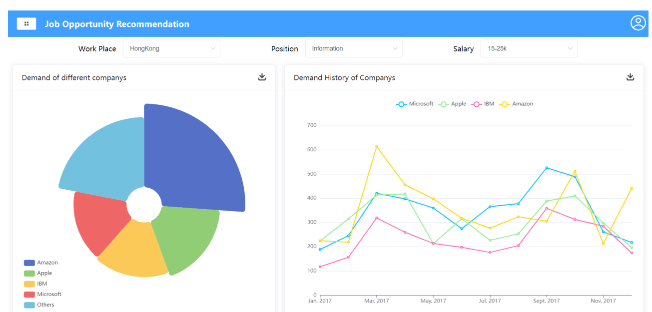


Figure 13: Deployed system: talent view.