Appendix A. Method Supplement

A.1 Mathematical Notations

Symbol	Description
$\mathcal{O}$	Occupation set;
$\mathcal{S}$	Skill set;
o	An occupation, $o \in \mathcal{O}$;
s	A skill, $s \in \mathcal{S}$;
t	The timestamp;
$\mathcal{P}^t$	Job description set at timestamp t;
Δt	The number of timestamps for pre-training;
$\mathcal{R}^t$	The skill demand of occupation at timestamp t;
$\mathcal{G}^t$	The OSD graph at timestamp t;
$\mathcal{V}$	The node set of occupations and skills, $\mathcal{V} = \mathcal{O} \cup \mathcal{S}$;
$\mathcal{E}^t$	The edge set of $\mathcal{G}^t$;
A^t	The weight adjacency matrix of $\mathcal{G}^t$;
e^t	An edge in $\mathcal{G}^t$, $e^t \in \mathcal{E}^t$;
E	The primary embeddings of occupations and skills;
Z	The bias cross-attention enhanced embeddings of occupations and skills;
B^{bias}	The learnable bias in bias cross-attention;
μ^t	The mean in the Gaussian distribution generated by the encoder at timestamp t;
σ^t	The variance in the Gaussian distribution generated by the encoder at timestamp t;
ϕ^t	The mean in the Gaussian distribution of temporal features at timestamp t;
δ^t	The variance in the Gaussian distribution of temporal features at timestamp t;
H^t	The latent variable generated by the Gaussian distribution (μ^t, σ^t) at timestamp t;
A^a	The learnable graph in adaptive temporal encoding unit;
α^t	The stable term of temporal features;
β^t	The trend term of temporal features;
ω^t	The evolved trend term of temporal features.

Table S1: Mathematical Notations.

A.2 Algorithm Procedure

In this paper, we designed a novel algorithm to implement the edge prediction problem on dynamic graphs. In order to make our algorithm procedure easier to understand, the pseudocode for the training pipeline of pre-training and fine-tuning is available in Algorithm 1.

A.3 Discussion on Advantages of PEFT

In the realm of dynamic graph learning, traditional research has predominantly focused on spatio-temporal forecasting for nodes and dynamic link prediction. Rarely have efforts been directed toward resolving issues about edge regression within dynamic graphs.

Our approach deviated by leveraging past dynamic graph sequences to forecast forthcoming changes in the overall

Algorithm 1: Training Procedure.

Input: Given a training dataset with a job description set sequence $(\mathcal{P}^1, \mathcal{P}^2, ...\mathcal{P}^T)$, and the devised model **DyGAE** with initial parameters Θ.

Output: The **DyGAE** for OSD forecasting with optimal parameters $\Theta*$.

Function PRE-TRAINING(**GAE**, $\mathcal{P}^0$)**:**
- $\Theta_{GAE} \leftarrow$ Parameters of **GAE** for pre-training;
- Construct Graph $\mathcal{G}^0$ based on $\mathcal{P}^0$;
- Initial E with Bert and P^0 by Eq.1;
- **while** *not convergence* **do**
  - $\mu^0, \sigma^0 \leftarrow$ Encoder Outputs, $H^0 \sim (\mu^0, \sigma^0)$;
  - $P^{\mathcal{E}}, \hat{\mathcal{R}}^0 \leftarrow$ Decoder Outputs;
  - compute the loss $\mathcal{L}^0_{GAE}$ by Eq.10;
  - update Θ_{GAE} using $\nabla_\theta \left(\mathcal{L}^0_{GAE}\right)$;
- **end**
- **return** The optimal $\Theta_{GAE}*$, μ^0, σ^0;

end

Function ATU-OPT(**ATU**, $\mathcal{G}^t$**,** t)**:**
- $\Theta^t_{ATU} \leftarrow$ Parameters of **ATU** at timestamp t;
- **while** *not convergence* **do**
  - $\phi^t, \delta^t \leftarrow$ **ATU** outputs;
  - $\mu^t = \mu^0 + \phi^t, \sigma^t = \sigma^0 + \delta^t$, $H^t \sim (\mu^t, \sigma^t)$;
  - $P^{\mathcal{E}^t}, \hat{\mathcal{R}}^t \leftarrow$ Decoder Outputs;
  - compute the loss $\mathcal{L}^t_{GAE}$ by Eq.10 with H^t as the input of decoders and $\mathcal{G}^t$ as the target graph;
  - compute the loss $\mathcal{L}^t_{trend}$ by Eq.13;
  - $\mathcal{L}^t_{tem} = \lambda_{trend}\mathcal{L}^t_{trend} + \mathcal{L}^t_{GAE}$;
  - update Θ^t_{ATU} using $\nabla_\theta \left(\mathcal{L}^t_{tem}\right)$;
- **end**
- **return** The optimal $\Theta^t_{ATU}*$, ϕ^t, δ^t;

end

Function TSM-OPT(**TSM**, $(\phi^{\Delta t+1}, \phi^{\Delta t+2}, ..., \phi^T)$)**:**
- $\Theta_{TSM} \leftarrow$ Parameters of **TSM**;
- **while** *not convergence* **do**
  - **for** $t \in [\Delta t + 1, T]$ **do**
    - $\alpha^t = \mathrm{FC}_8(\phi^t)$, $\beta^t = \mathrm{FC}_9(\phi^t)$;
    - $\omega^t = \mathrm{MLP}(\beta^t)$;
    - $\hat{\phi}^t = \alpha^t + \beta^t$, $\hat{\phi}^{t+1} = \alpha^t + \omega^t$;
  - **end**
  - compute the loss $\mathcal{L}_{shift}$;
  - update Θ_{TSM} using $\nabla_\theta \left(\mathcal{L}_{shift}\right)$;
- **end**
- return The optimal $\Theta_{TSM}*$;

end

Procedure TRAIN(**DyGAE**, $(\mathcal{P}^1, \mathcal{P}^2, ..., \mathcal{P}^T)$)**:**
- $(\mathcal{P}^0, \mathcal{P}^{\Delta t+1}, ...\mathcal{P}^T) \leftarrow$ Reconstruct $\{\mathcal{P}^t | t \in [1, T]\}$;
- The **DyGAE** consists of the backbone **GAE**, the adaptive temporal coding unit **ATU**, and the temporal shift module **TSM**;
- Construct Graph $\mathcal{G}^0$ based on $\mathcal{P}^0$;
- $\Theta_{GAE}*, \mu^0, \sigma^0$ = PRE-TRAINING(**GAE**, $\mathcal{G}^0$);
- **for** $t \in [\Delta t + 1, T]$ **do**
  - Construct Graph $\mathcal{G}^t$ based on $\mathcal{P}^t$;
  - $\Theta^t_{ATU}*, \phi^t, \delta^t$ =ATU-OPT(**ATU**, $\mathcal{G}^t, t$);
- **end**
- $\Theta_{TSM}* =$ TSM-OPT(**TSM**, $(\phi^{\Delta t+1}, \phi^{\Delta t+2}, ..., \phi^T)$);
- $\Theta* = \Theta_{GAE}* + \sum_{t\in[\Delta t+1,T]} \Theta^t_{ATU}* + \Theta_{TSM}*$;
- **return DyGAE** with optimal parameters $\Theta*$;

end

Date	Dai	Fin	IT	Man
2020/01-06	3,806,127	1,494,318	1,500,300	964,823
2020/07-12	8,015,781	2,567,487	2,675,593	2,044,281
2021/01-06	9,168,875	2,936,613	3,662,220	1,913,414
2021/07-12	13,042,287	3,970,267	4,999,137	3,640,062
2022/01-06	12,503,386	4,102,521	5,531,762	3,167,375
2022/07-12	11,541,187	3,492,851	5,264,994	2,572,476
2023/01-06	11,655,335	3,368,295	3,838,109	3,276,155
2023/07-12	12,610,429	3,236,394	4,522,106	3,766,562

Table S2: # of JDs in each timestamp on Daily (Dai), Finance (Fin), IT, and Manufacturing (Man) datasets.

Data	Dai	Fin	IT	Man
# of occupations	244	93	193	155

Table S3: # of occupations of Daily (Dai), Finance (Fin), IT, and Manufacturing (Man) datasets.

graph structure. This not only entails predicting edge existence but also involves forecasting specific edge weights. To address this, our model must adeptly capture the evolving trends in dynamic graph sequences and, crucially, discern the correlations between graph nodes. Consequently, we introduced a pre-training and parameter-efficient fine-tuning paradigm, enabling the model to discern stable node correlations and adapt to the latest trend changes. This comprehensive approach facilitated the joint prediction of future edge existence and weights.

The utilization of PEFT was motivated by the necessity to avoid rendering the model incapable of maintaining stable representations through full fine-tuning. Instead, our approach exclusively focused on evolving temporal characteristics, enabling the model to adeptly capture the dynamic evolution of representations over time. This approach disentangled the representation, enhancing the model's ability to perceive temporal dynamics comprehensively. Simultaneously, it optimized the efficiency of model training.

Appendix B. More Details on Experiment Settings and Results

B.1 Data Description

In this study, we have collected JDs from four distinct industries: Daily, Finance, IT, and Manufacturing.

- **Daily**: The Daily dataset predominantly encompasses occupations related to daily life, such as store managers, Chinese chefs, and custodial staff.
- **Finance**: The Finance dataset includes some securities brokers, fund managers, and other occupations closely related to the financial industry.
- **IT**: The IT dataset contains occupations vital to the realms of Artificial Intelligence, Internet technology, and hardware development like algorithm engineers, data annotators, and JAVA developers.
- **Manufacturing**: The Manufacturing dataset comprises a spectrum of traditional industrial occupations, including material engineers, welding engineers, and electromechanical engineers.

The quantities of JDs and occupational categories within the datasets are presented in Table S2 and Table S3, respectively. It is noteworthy that all datasets were characterized by a common skill set comprising 1030 skills.

B.2 More Implementation Details

The experiments utilized PyTorch-1.13.1 on a single A800 GPU. We employed the Adam optimizer with an initial learning rate of 0.0001 during both the pre-training and fine-tuning stages. The hyperparameters λ_{nll}, λ_{con}, λ_{rg}, λ_{rank}, λ_{trend}, λ_{re}, λ_{ne}, and λ_{com} were set to 1, 0.05, 100, 0.1, 1, 1, 1, and 1, respectively. During the pre-training phase, GAE underwent training for 1000 epochs. Subsequently, in the adaptive temporal unit optimization, fine-tuning was performed for 1000 epochs. In the optimization of the temporal shift module, fine-tuning spanned 500 epochs. To ensure robustness, all experiments were conducted with a repetition of 4 trials.

B.3 Baseline Descriptions and Implement Details

To validate the accurate OSD forecasting, we selected several dynamic graph learning methods and divided them into three classes: Spatio-Temporal Forecasting, Dynamic Link Prediction, and Skill Demand Forecasting. * indicates the suboptimal baselines, due to the uncompetitive performance and page limitation, we have not presented the corresponding experimental results in the Experiment Section, but in Table S4.

Spatio-Temporal Forecasting The spatio-temporal forecasting method bears close relevance to our problems. These baseline models also construct graph on the nodes and tackle regression prediction challenges amidst continuous temporal changes. Nevertheless, a key distinction lies in the traditional spatio-temporal forecasting's emphasis on regressing nodes, whereas our study centers around the regression task involving edges. Consequently, we have adapted the baseline models to shift from a node regression task to an edge regression problem. We achieved these baselines based on the PyTorch Geometric Temporal[1].

- **DCRNN** [Li *et al.*, 2018]: DCRNN employs bidirectional random walks on the graph to capture spatial dependencies, and an encoder-decoder architecture with scheduled sampling to model temporal dependencies.
- **GConvGRU*** [Seo *et al.*, 2018]: The proposed model integrates convolutional neural networks (CNN) on graphs for identifying spatial structures and recurrent neural networks (RNN) to detect dynamic patterns. Two architectures, GConvGRU and GConvLSTM, are explored for the Graph Convolutional Recurrent Network (GCRN).
- **EvolveGCN*** [Pareja *et al.*, 2019]: EvolveGCN addresses the challenges of weight learning and graph evolution at each timestamp. Two methods are introduced: EvolveGCNH, which learns the weight matrix of the graph at each time as a hidden state, and EvolveGCNO, which directly employs the weight evolution as a hidden state output, decoupled from node embedding.

[1]https://github.com/benedekrozemberczki/pytorch_geometric_temporal.git

Data	Metric	GConvGRU	EvolveGCNH	EvolveGCNO	GCLSTM	MPNNLSTM	DHGEM	Pre-DyGAE
Dai	AUC%↑	51.20±0.35	49.40±0.40	49.50±0.39	51.34±1.06	50.43±0.81	51.48±0.93	96.96±0.04
	Hits@1%↑	0.32±0.09	0.16±0.01	0.16±0.00	0.37±0.16	0.18±0.04	0.42±0.21	42.97±0.33
	MRR%↑	3.04±0.29	1.14±0.01	1.15±0.01	3.11±0.46	2.53±0.13	3.26±0.51	51.92±0.19
	EGM%↓	0.18±0.01	0.07±0.00	0.07±0.00	0.10±0.01	2.15±0.12	0.11±0.01	0.06±0.00
	MAE%↓	33.31±3.26	3.42±3.31	1.83±0.60	8.73±2.54	2378.55±123.06	10.69±2.97	0.30±0.01
	RMSE%↓	4.21±0.02	4.32±0.03	4.30±0.00	4.26±0.01	43.01±1.91	4.24±0.03	0.66±0.27
Fin	AUC%↑	51.73±0.78	48.62±1.71	49.36±0.43	51.55±0.51	50.48±0.66	51.71±1.41	96.16±0.09
	Hits@1%↑	0.41±0.14	0.15±0.04	0.17±0.01	0.49±0.03	0.22±0.06	0.57±0.10	48.59±0.34
	MRR%↑	3.54±0.43	1.36±0.15	1.43±0.04	3.76±0.20	2.91±0.19	4.06±0.13	57.89±0.37
	EGM%↓	0.27±0.02	0.13±0.01	0.12±0.00	0.18±0.03	2.51±0.32	0.23±0.04	0.09±0.02
	MAE%↓	38.61±5.54	7.82±8.65	3.98±0.24	16.13±4.78	1905.69±583.84	24.17±9.71	0.34±0.02
	RMSE%↓	4.71±0.03	4.93±0.03	4.88±0.00	4.80±0.03	40.91±2.24	4.73±0.05	0.47±0.24
IT	AUC%↑	50.85±0.65	49.55±0.36	56.26±8.63	51.54±1.14	50.56±0.89	52.18±1.27	96.04±0.06
	Hits@1%↑	0.40±0.05	0.07±0.00	0.81±1.01	1.00±0.37	0.26±0.05	0.95±0.29	44.34±0.67
	MRR%↑	3.96±0.11	1.30±0.01	5.10±5.17	5.40±0.80	3.39±0.13	5.39±0.57	52.59±0.45
	EGM%↓	0.33±0.01	0.16±0.00	0.36±0.28	0.26±0.02	2.87±0.08	0.35±0.04	0.11±0.02
	MAE%↓	22.06±1.53	2.86±0.21	17.38±22.41	14.38±3.19	1140.97±38.53	23.79±3.18	0.35±0.02
	RMSE%↓	5.23±0.02	5.39±0.00	5.27±0.16	5.18±0.08	42.02±2.38	5.15±0.05	0.73±0.34
Man	AUC%↑	51.25±1.30	49.48±0.38	49.64±0.11	51.66±1.37	50.61±0.82	52.08±1.11	96.92±0.07
	Hits@1%↑	0.33±0.03	0.19±0.01	0.19±0.01	0.35±0.08	0.25±0.04	0.68±0.22	45.62±0.63
	MRR%↑	3.18±0.06	1.49±0.05	1.51±0.03	3.35±0.18	2.89±0.15	4.11±0.55	53.85±0.46
	EGM%↓	0.23±0.01	0.11±0.00	0.11±0.00	0.15±0.02	2.55±0.22	0.17±0.02	0.07±0.02
	MAE%↓	20.97±1.21	2.48±1.56	2.76±0.32	8.30±4.98	1328.72±178.34	12.12±4.14	0.30±0.03
	RMSE%↓	4.19±0.02	4.30±0.01	4.29±0.00	4.24±0.03	41.52±1.31	4.16±0.06	0.68±0.18

Table S4: Performance comparisons of uncompetitive baselines for OSD forecasting on four datasets.

Data	Metric	P3a	RP3b	NGCF	LightGCN	MultVAE	w-o con	w-o rank	w-o bias	L1	our
Dai	AUC%↑	60.28±0.00	60.82±0.00	62.07±0.08	77.48±1.69	81.42±0.00	93.72±0.02	94.82±0.01	94.62±0.01	94.78±0.01	**94.92±0.02**
	Hits@1%↑	13.36±0.00	12.87±0.00	12.25±0.19	13.58±0.64	13.07±0.00	15.89±0.15	19.14±0.07	18.88±0.29	19.58±0.20	**21.58±0.24**
	MRR%↑	16.82±0.00	16.52±0.00	18.50±0.09	20.56±0.69	19.76±0.00	23.30±0.11	26.76±0.14	26.17±0.17	26.75±0.21	**27.39±0.02**
	EGM%↓	0.03±0.00	0.03±0.00	0.03±0.00	0.34±0.03	0.02±0.00	0.04±0.01	0.02±0.00	0.55±0.31	0.06±0.06	**0.02±0.05**
	MAE%↓	382.64±0.00	94.13±0.00	22.31±0.94	164.75±2.22	7.81±0.00	1.48±0.18	0.93±0.15	14.34±8.33	1.99±1.27	**1.33±1.33**
	RMSE%↓	140.49±0.00	26.72±0.00	2.50±0.03	2.82±0.15	2.90±0.00	0.31±0.05	0.19±0.04	1.65±0.69	0.36±0.28	**0.26±0.22**
Fin	AUC%↑	58.43±0.00	58.85±0.00	70.15±0.09	76.45±1.10	77.22±0.00	93.38±0.06	94.52±0.03	94.37±0.11	94.44±0.15	**95.56±0.05**
	Hits@1%↑	13.21±0.00	12.08±0.00	14.23±0.10	15.74±1.00	12.08±0.00	19.46±0.25	21.19±0.33	21.16±0.31	21.54±0.21	**22.64±0.27**
	MRR%↑	16.12±0.00	15.18±0.00	20.70±0.09	23.43±0.96	18.69±0.00	26.73±0.27	28.60±0.15	28.37±0.25	28.84±0.17	**29.02±0.17**
	EGM%↓	0.04±0.00	0.03±0.00	0.03±0.00	0.34±0.05	0.02±0.00	0.09±0.04	0.09±0.07	1.09±0.62	0.06±0.03	**0.02±0.02**
	MAE%↓	10.72±0.00	3.62±0.00	0.51±0.20	0.86±6.40	0.51±0.00	0.22±0.59	0.27±0.88	1.49±7.94	0.12±0.39	**0.09±0.05**
	RMSE%↓	119.65±0.00	30.73±0.00	2.63±0.02	2.73±0.05	3.12±0.00	0.78±0.26	1.45±0.28	2.46±0.93	0.54±0.17	**0.16±0.07**
IT	AUC%↑	58.47±0.00	58.95±0.00	71.86±1.37	69.79±0.07	74.30±0.00	92.67±0.41	93.93±0.10	93.98±0.06	93.30±0.07	**94.42±0.12**
	Hits@1%↑	11.30±0.00	11.23±0.00	12.75±1.21	14.35±0.28	13.30±0.00	16.65±0.83	16.56±0.29	18.37±0.39	19.75±0.34	**20.14±0.07**
	MRR%↑	14.86±0.00	14.96±0.00	18.65±1.20	19.75±0.23	18.88±0.00	23.03±0.93	23.23±0.31	24.97±0.36	25.42±0.41	**26.79±0.13**
	EGM%↓	0.06±0.00	0.05±0.00	0.34±0.03	0.07±0.00	0.07±0.00	0.06±0.01	0.11±0.07	0.15±0.07	0.13±0.03	**0.05±0.04**
	MAE%↓	391.10±0.00	92.55±0.00	97.30±5.42	19.44±0.96	19.43±0.00	0.32±0.08	0.43±0.20	0.73±0.23	0.47±0.09	**0.22±0.13**
	RMSE%↓	216.36±0.00	44.41±0.00	3.34±0.08	2.78±0.02	3.34±0.00	0.66±0.28	0.59±0.16	0.76±0.23	0.58±0.08	**0.26±0.21**
Man	AUC%↑	60.15±0.00	61.31±0.00	66.66±0.10	75.35±2.27	79.62±0.00	93.19±0.08	93.77±0.07	93.67±0.01	92.83±0.01	**93.94±0.06**
	Hits@1%↑	11.14±0.00	10.93±0.00	11.02±0.05	11.22±0.61	11.27±0.00	13.87±0.24	14.08±0.32	13.55±0.23	13.61±0.21	**14.66±0.08**
	MRR%↑	14.80±0.00	14.86±0.00	16.34±0.07	17.36±0.65	16.45±0.00	20.58±0.21	20.81±0.33	20.13±0.12	20.27±0.12	**21.34±0.11**
	EGM%↓	0.06±0.00	0.05±0.00	0.05±0.00	0.32±0.03	0.03±0.00	0.04±0.01	0.04±0.00	0.58±0.34	0.07±0.04	**0.03±0.01**
	MAE%↓	318.73±0.00	79.49±0.00	27.10±0.68	101.97±4.07	5.74±0.00	0.47±0.01	0.56±0.11	4.41±2.33	0.51±0.13	**0.37±0.04**
	RMSE%↓	119.02±0.00	26.40±0.00	2.36±0.01	2.58±0.03	2.95±0.00	0.32±0.11	0.48±0.16	1.29±0.52	0.35±0.12	**0.12±0.08**

Table S5: Performance comparisons on OSD graph completion on four datasets.

- **DyGrEncoder** [Taheri *et al.*, 2019]: This approach combines a sequence-to-sequence encoder-decoder model with gated graph neural networks (GGNNs) and long short-term memory networks (LSTMs). The encoder captures temporal dynamics in an evolving graph, and the decoder reconstructs the dynamics using the encoded representation.

- **AGCRN** [Bai *et al.*, 2020]: AGCRN proposes two adaptive modules to enhance Graph Convolutional Network (GCN): 1) Node Adaptive Parameter Learning (NAPL) captures node-specific patterns, and 2) Data Adaptive Graph Generation (DAGG) infers inter-dependencies among different traffic series automatically.

- **A3TGCN** [Bai *et al.*, 2021]: A3T-GCN learns short-time trends in time series using gated recurrent units, incorporates spatial dependence based on the road network's topology through graph convolutional networks, and introduces an attention mechanism to adjust the importance of different time points for improved prediction accuracy.

- **GCLSTM*** [Chen *et al.*, 2022]: GCLSTM is an end-to-end model integrating a Graph Convolution Network

(GCN) embedded Long Short-Term Memory network (LSTM) for dynamic network link prediction. GCN captures local structural properties, while LSTM learns temporal features across snapshots of a dynamic network.

- **MPNNLSTM*** [Panagopoulos *et al.*, 2021]: To investigate the impact of population movement on COVID-19 spread, MPNNLSTM utilizes graph neural networks (GNNs) and employs a transfer learning approach based on Model-Agnostic Meta-Learning (MAML) to leverage knowledge from models of other countries.

Dynamic Link Prediction We utilized DygLib[2], a publicly available library, to implement a set of baseline models for dynamic graph link prediction. To tailor these baselines to our specific scenario, we incorporated an additional Multi-layer Perceptron (MLP) to forecast specific skill demand and employed an MAE Loss for regression tasks.

- **TGAT** [da Xu *et al.*, 2020]: TGAT introduces the Temporal Graph Attention layer, designed to adeptly aggregate temporal-topological neighborhood features simultaneously capturing time-feature interactions efficiently.
- **CAWN** [Wang *et al.*, 2021]: This work proposes a novel approach for representing temporal networks, termed CAW-N, which utilizes Contextualized Attention Walks (CAWs) to encode temporal network motifs. This method captures network dynamics while maintaining full inductiveness.
- **DyGFormer** [Yu *et al.*, 2023]: DyGFormer stands as a state-of-the-art method, characterized by its conceptual simplicity. It exclusively learns from nodes' historical first-hop interactions through a neighbor co-occurrence encoding scheme that explores correlations between source and destination nodes based on their historical sequences and a patching technique that divides each sequence into multiple patches and feeds them to a Transformer. This approach allows the model to effectively and efficiently benefit from longer historical contexts.

Skill Demand Forecasting To substantiate the superiority of our model in predicting skill demand at the occupational level, we conducted a comprehensive comparison with several established baselines about overall skill demand. During the implementation of the baseline, we further disaggregated our skills at the occupational level for skill prediction. Notably, the following baselines undertake modeling and forecasting for both the supply and demand aspects. In our problem, we specifically focused on the modeling and forecasting related to skill demand, omitting the corresponding aspects related to the supply side.

- **DHGEM** [Guo *et al.*, 2022][3]*: DHGEM presents a framework designed for fine-grained joint prediction of talent demand and supply. The architecture encompasses a transformer-based encoder and a dynamic heterogeneous Graph Convolutional Network (GCN). This combination is tailored to capture correlations between demand and supply sequences, complemented by an RNN for modeling dynamic recurrent processes.
- **CHGH** [Chao *et al.*, 2024][4]: CHGH introduces an encoder-decoder network featuring: i) a cross-view graph encoder to grasp interconnections between skill demand and supply; ii) a hierarchical graph encoder for modeling the co-evolution of skills from a cluster-wise perspective; and iii) a conditional hyperdecoder. The conditional hyperdecoder enables the joint prediction of demand and supply variations by incorporating historical demand-supply gaps.

B.4 More Details for Experiment Analysis

Due to page limitations in the body of the paper, we presented the experimental results on OSD forecasting of uncompetitive baselines in Table S4. Notably, nearly every baseline exhibited suboptimal performance in link prediction metrics, with significant disparities observed in metrics such as MAE.

Among these less successful baselines, a majority employed methodologies akin to the end-to-end integration of GCN and RNN. This observation underscored the inherent challenges of relying solely on such end-to-end approaches to capture trend changes in dynamic edge regression problems. It further underscored the significance of incorporating pre-training on stable structures to effectively capture dependencies between occupations and skills.

B.5 More Details for OSD Graph Completion

We presented the complete OSD Graph Completion experiments on four datasets in Figure 1(a). In addition to the compared backbones discussed in the Experiment section, we also evaluated two less competitive baselines, namely P3a [Cooper *et al.*, 2014] and RP3b [Paudel *et al.*, 2016]. The findings exhibited a consistent trend with the observations discussed in the experiments, underscoring the robustness of the learned representations by GAE across all datasets.

B.6 More Details for Ablation Studies

The ablation studies on four datasets are summarized in Figure S1. "Next" and "mean" refer to updating temporal features with the most recent ones and computing mean temporal features from past timestamps, respectively. The results align with study observations, confirming the effectiveness of the designed module across all datasets.

B.7 More Details for Parameter Sensitivity Analysis

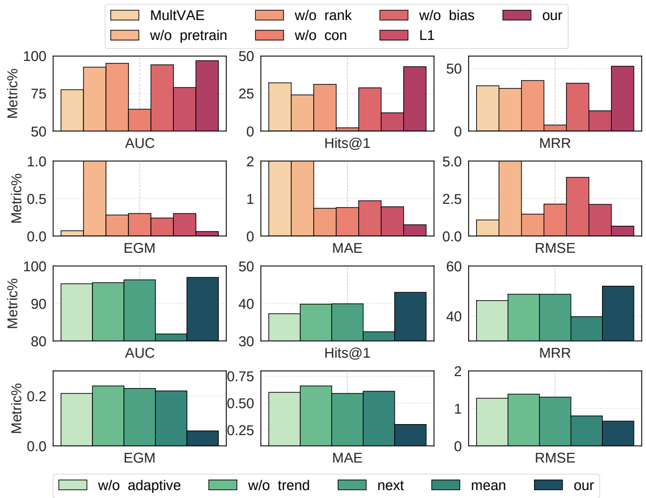
OSD Graph Completion We conducted experiments on the Fin dataset for OSD graph completion to assess the impact of different hyperparameters, namely λ_{rg}, λ_{rank}, and λ_{con}, introduced to predict OSD, differentiate feature spaces for various skills, and address sparse skills, respectively.

Our model was trained by varying λ_{rg} from 10.0 to 500.0, λ_{con} from 0.01 to 0.1, and λ_{rank} from 0.02 to 0.20. As depicted in Figure S2, GAE demonstrated optimal performance when λ_{rg}, λ_{con}, and λ_{rank} are set to 100, 0.05, and 0.1, respectively. Importantly, the parameter experiments revealed consistent parameter selections across all metrics, indicating

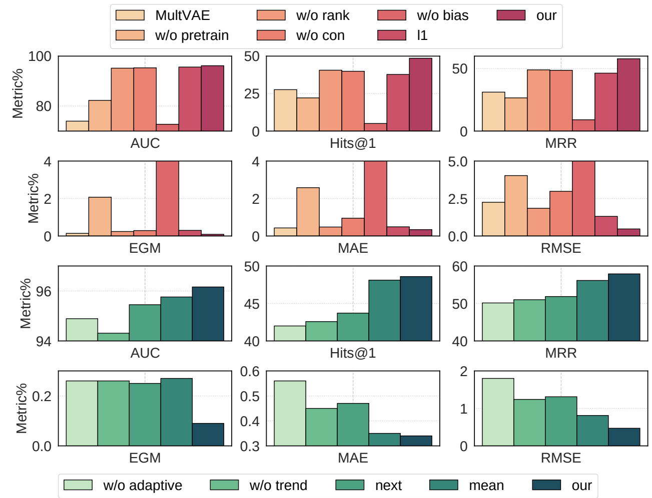
[2]https://github.com/yule-BUAA/DyGLib.git

[3]https://github.com/gzn00417/DH-GEM

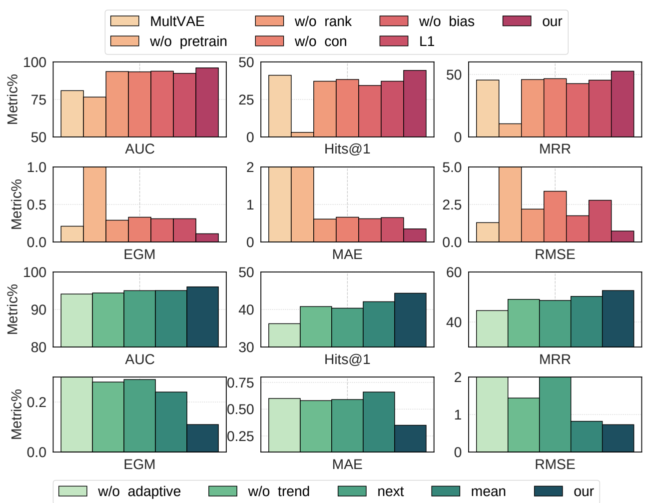
[4]https://github.com/youngfish42/Awesome-FL.git



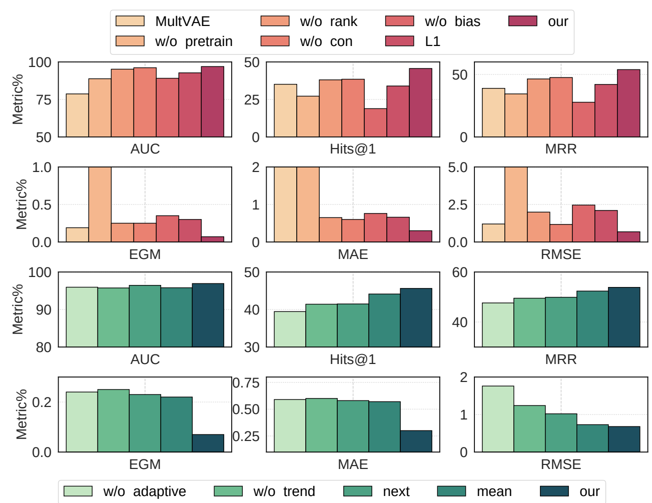
(a) OSD forecasting ablation studies on Dai dataset.



(b) OSD forecasting ablation studies on Fin dataset.



(c) OSD forecasting ablation studies on IT dataset.



(d) OSD forecasting ablation studies on Man dataset.

Figure S1: OSD forecasting ablation studies on four dataset.

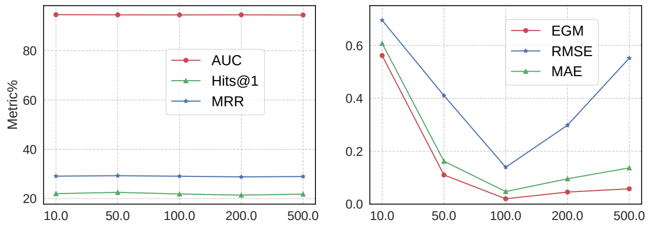
a strong correlation between the tasks. This highlighted the beneficial impact of multi-task learning, facilitating concurrent improvement across various tasks.

Adaptive Temporal Encoding Unit Optimization We conducted experiments on the Fin dataset to assess the impact of different values for λ_{trend}. The term $\mathcal{L}_{trend}$ was introduced to explicitly enable dynamic representations to discern changes in trends.

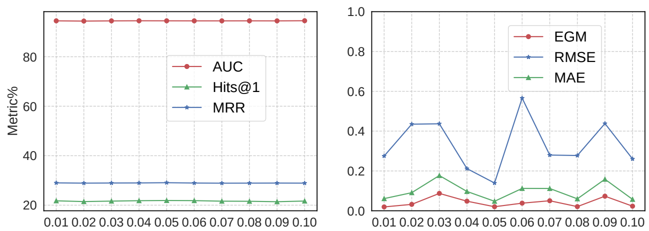
We optimized the parameters of the adaptive temporal encoding unit by varying λ_{trend} from 0.2 to 2.0. As illustrated in Figure S3, the majority of metrics exhibited stability under different parameter values, with a slight elevation observed around 1.0. However, beyond 1.8, the effectiveness showed a declining trend. These findings underscored the significance of trend-aware loss as an auxiliary task, emphasizing the importance of maintaining a moderate weight, as excessively high weights may impact the primary task.

Temporal Shift Module Optimization In the optimization of the temporal shift module, we maintained λ_{re} at a fixed value of 1 while adjusting the other two hyperparameters, namely λ_{ne} and λ_{com}, to examine their impact. As depicted in Figure S3, diverse indicators exhibited relative stability under different parameter settings, indicating weak sensitivity to parameter variations in this context. The concurrent training of the three hyperparameters proved advantageous for PreDyGAE in capturing the patterns associated with the migration of time series.

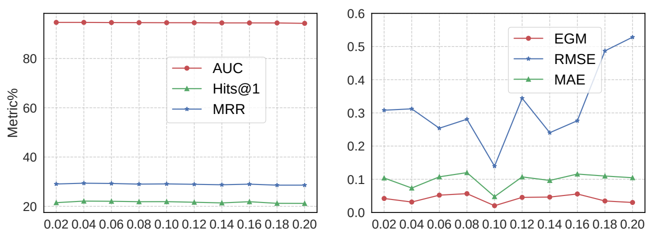
The Influence of Δt We have already discussed the impact of Δt on experimental results. The specific experimental results are shown in Table S6.



(a) λ_{rg}



(b) λ_{con}



(c) λ_{rank}

Figure S2: Performance of OSD graph completion with different parameter settings of λ_{rg}, λ_{con} and λ_{rank}.

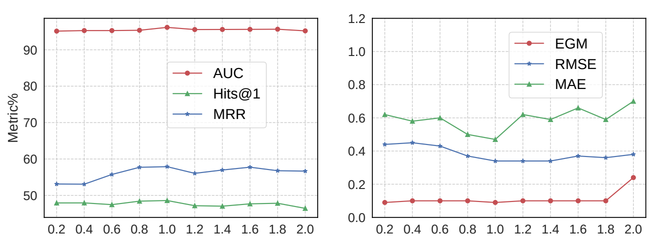


Figure S3: Parameter Sensitivity Analysis of λ_{trend} in adaptive temporal encoding unit optimization on Fin dataset.

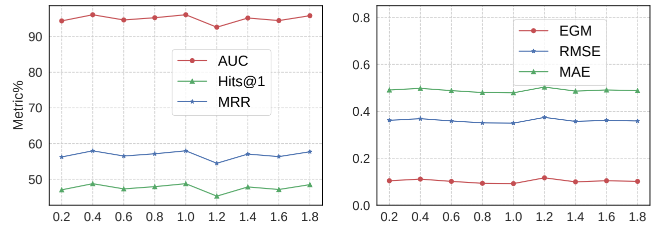
Δt	AUC%↑	Hits@1%↑	MRR%↑	EGM%↓	MAE%↓	RMSE%↓
1	$88.02_{\pm 6.20}$	$29.19_{\pm 16.84}$	$35.35_{\pm 18.24}$	$0.13_{\pm 0.00}$	$0.75_{\pm 0.35}$	$3.25_{\pm 0.06}$
2	$93.76_{\pm 1.62}$	$43.07_{\pm 1.01}$	$50.79_{\pm 0.03}$	$0.14_{\pm 0.02}$	$0.72_{\pm 0.48}$	$2.22_{\pm 0.13}$
3	$93.98_{\pm 1.69}$	$45.00_{\pm 1.33}$	$52.40_{\pm 2.00}$	$0.11_{\pm 0.01}$	$0.68_{\pm 0.11}$	$0.96_{\pm 0.10}$
4	$96.16_{\pm 0.09}$	$48.59_{\pm 0.34}$	$57.89_{\pm 0.37}$	$0.09_{\pm 0.02}$	$0.34_{\pm 0.02}$	$0.47_{\pm 0.24}$
5	$95.44_{\pm 0.35}$	$46.64_{\pm 0.27}$	$54.43_{\pm 0.01}$	$0.10_{\pm 0.00}$	$0.50_{\pm 0.23}$	$1.25_{\pm 0.02}$
6	$95.83_{\pm 0.03}$	$44.07_{\pm 0.30}$	$52.32_{\pm 0.41}$	$0.10_{\pm 0.00}$	$0.60_{\pm 0.03}$	$1.92_{\pm 0.0}$

Table S6: Parameter Experiment on Δt on Fin dataset.

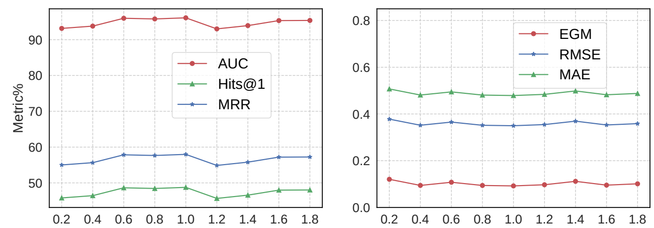
Appendix C. Case study

Utilizing the OSD from January 2022 to December 2023 and our predicted OSD for July to December 2023, we computed the OSD differences between consecutive timestamps. Subsequently, we extracted specific occupations and skills, presenting the results visually in Figure S5.

The figure illustrated varying trends in skill demand changes at different times, with the same skill exhibiting different evolving patterns across diverse occupations. This observation aligned with the underlying motivation of our research problem. Moreover, upon evaluating the prediction outcomes, our model demonstrated a high degree of concordance with actual results, affirming the efficacy of our model.



(a) λ_{ne}



(b) λ_{com}

Figure S4: Parameter Sensitivity Analysis of λ_{ne} and λ_{com} in temporal shift module optimization on Fin dataset.

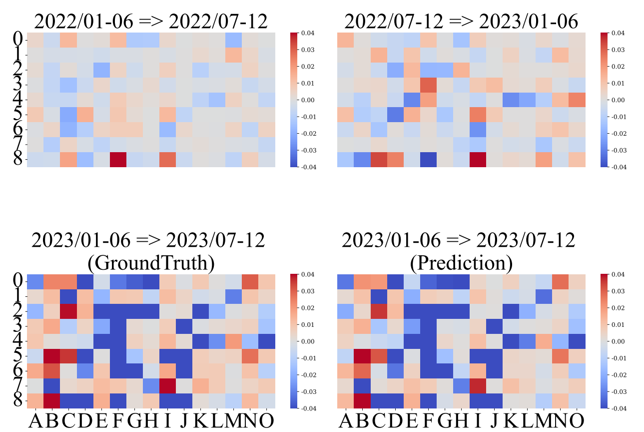


Figure S5: A heatmap illustrating the disparity in OSD between the two timestamps. Each row in the heatmap corresponds to an occupation, listed from top to bottom as follows: data product manager (0), product manager (1), data analyst (2), image algorithm (3), data annotation/AI trainer (4), recommendation algorithm (5), algorithm researcher (6), interaction designer (7), and search algorithm (8). Each column denotes a skill, with skills listed as follows: SQL (A), writing (B), coding (C), big data (D), tidy (E), model (F), decision making (G), visualization (H), computer repair (I), PyTorch (J), office (K), Python (L), coordination (M), English (N), and typing (O). Red represents growth and blue represents decline, with darker colors representing greater degrees of growth or decline respectively.

References

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

Jiandong Bai, Jiawei Zhu, Yujiao Song, Ling Zhao, Zhixiang Hou, Ronghua Du, and Haifeng Li. A3t-gcn: Attention temporal graph convolutional network for traffic forecasting. *ISPRS International Journal of Geo-Information*, 10(7), 2021.

Wenshuo Chao, Zhaopeng Qiu, Likang Wu, Zhuoning Guo, Zhi Zheng, Hengshu Zhu, and Liu Hao. A cross-view hierarchical graph learning hypernetwork for skill demand-supply joint prediction. In *Proceedings of the 33th International Joint Conference on Artificial Intelligence*, AAAI'24. AAAI Press, 2024.

Jinyin Chen, Xueke Wang, and Xuanheng Xu. Gc-lstm: Graph convolution embedded lstm for dynamic network link prediction. *Applied Intelligence*, pages 1–16, 2022.

Colin Cooper, Sang Hyuk Lee, Tomasz Radzik, and Yiannis Siantos. Random walks in recommender systems: Exact computation and simulations. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion. Association for Computing Machinery, 2014.

da Xu, chuanwei ruan, evren korpeoglu, sushant kumar, and kannan achan. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations (ICLR)*, 2020.

Zhuoning Guo, Hao Liu, Le Zhang, Qi Zhang, Hengshu Zhu, and Hui Xiong. 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, page 2957–2967. Association for Computing Machinery, 2022.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR '18)*, 2018.

George Panagopoulos, Giannis Nikolentzos, and Michalis Vazirgiannis. Transfer Graph Neural Networks for Pandemic Forecasting, April 2021. arXiv:2009.08388 [cs, stat].

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs, November 2019. arXiv:1902.10191 [cs, stat].

Bibek Paudel, Fabian Christoffel, Chris Newell, and Abraham Bernstein. Updatable, accurate, diverse, and scalable recommendations for interactive applications. 2016.

Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured Sequence Modeling with Graph Convolutional Recurrent Networks. In Long Cheng, Andrew Chi Sing Leung, and Seiichi Ozawa, editors, *Neural Information Processing*, pages 362–373, Cham, 2018. Springer International Publishing.

Aynaz Taheri, Kevin Gimpel, and Tanya Berger-Wolf. Learning to represent the evolution of dynamic graphs with recurrent models. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, New York, NY, USA, 2019. Association for Computing Machinery.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. In *International Conference on Learning Representations*, 2021.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 2023.