Submitted to *Information Systems Research*
manuscript (Please, provide the manuscript number!)

(Authors' names blinded for peer review)

Text of your abstract

Key words:

---

1. Spatial-Temporal Job Mobility Predictor

The proposed job mobility predictor models how an employee chooses the next job when considering a career move. At any decision time, the predictor summarizes the employee's career development and consults potential jobs from social networks to infer the most probable job transition via a spatial-temporal graph. The key components of our model are

- *Dynamic Typicality Encoder:* a path-dependent encoder to embed employees' career progression. It extends the standard sequence model by accounting for the recency and tenure of each job.
- *Dual Temporal Graph Encoder:* a pair of temporal graph encoders, one tracks the evolving association among job titles, and the other recommends potential job titles via employees having similar career development.
- *Bi-level Probabilistic Decoder:* a probability generator to yield the likelihood of obtaining each job in the job title universe. It incorporates a filtering layer over the coarse job title groups to help the predictor to focus on the most promising candidate job titles.

Figure 1 presents the overall architecture of the proposed spatial-temporal job mobility predictor, and Table 1 includes the main notations in this paper. Formally, the job mobility prediction problem can be cast to the form below:

Problem: For an employee $e \in E$, at any career decision time t_n, we aim to find a conditional probability distribution $\mathbb{P}(X_{n+1} \mid \mathcal{F}_n)$ over title universe T, with $\mathcal{F}_n$ represents the information set available to e at t_n.

We organize this section as follows: subsection 1.1, 1.2 build up information set $\mathcal{F}_n$ incrementally; subsection 1.3 maps the embedding derived from $\mathcal{F}_n$ to a probability distribution; finally, we present implementation details and algorithm in subsection 1.4.

Figure 1 **Overall Architecture of the Proposed Spatial-Temporal Job Mobility Predictor**

NOT AVAILABLE

1.1. Component 1: Dynamic Typicality Encoder

In this subsection, the discussion is centered around a single employee $e \in E$, to ease notation, we suppress the dependence on e everywhere. The typicality of employee e is an evolving quantity mainly determined by his/her career experiences, each of which is defined via the

Table 1 Notation Summary

Symbol	Description
E, T	The set of all employees and all job titles.
$\vec{\omega}(e)$	The career trajectory of employee e, where $w_i(e)$ is e's i-th job transition with $i \in \{1, ..., N(e)\}$.
$X_.$	A T-valued random process. Projected to interval $[t_{n-1}, t_n]$, it defines a job transition $w_n(e) = (t_{n-1}, t_n, X_n)$.
$z(\cdot)$	The embedding operator, for instance, $z(x)$ is the embedding of title $x \in T$.
$\tilde{c}$, c	The candidate/ultimate memory captures the employee's evolving typicality.
$\mathcal{G}$	The spatial-temporal graph that consists of all employees' job transition records.
$\mathcal{V}_E(t)$, $\mathcal{V}_T(t)$	The temporal employee/title nodes observed at time t.
$\mathcal{E}_{TT}(t)$, $\mathcal{E}_{ET}(t)$	The temporal title-title edge and employee-title edge observed at time t.

job title X_n and the corresponding tenure Δ_n, i.e., $\Delta_n \equiv t_n - t_{n-1}$. We model the employee's *dynamic typically* $\big\{c(t); t \in \{t_0, ..., t_N\}\big\}$ as a payoff of his/her career trajectory, which takes the following form,

$$c(t_n) = \widetilde{c}(X_n, \Delta_n) + \mathtt{df}(t_{n-1}, t_n) \cdot c(t_{n-1}), \; n = 1, ..., N. \tag{1}$$

Here, $\tilde{c}$ encodes the update from the current job X_n and $\mathtt{df}(\cdot, t_n)$ accounts for the time decay of the past experiences. For the initial typicality $c(t_0)$, we use the employee's educational background because it is the most relevant static feature to e's first job. Albeit its simple form, Eqn. (1) yields a desired typicality representation as a tenure-recency-adjusted aggregation of all past job titles. Indeed, by unrolling Eqn. (1), we found each job is re-weighted by its tenure, and $\mathtt{df}$ has a cascading effect that discounts each job based on its temporal distance from the current time. In addition, Eqn. (1) is compatible with a wide range of sequence encoders, in particular, most variants of RNN (recurrent neural network model). We detail a two-step approach to customize RNN to be time-aware.

Firstly, we compose $\mathtt{df}(t_{n-1}, t_n)$ with a multiplier f and a discount factor p,

$$\mathtt{df}(t_{n-1}, t_n) = f(t_n) \cdot p(t_{n-1}, t_n) \tag{2}$$

The multiplier f is usually called a "gate" that controls the amount of past memory $c(t_{n-1})$ flowing through, e.g., the forget gate in LSTM. The addition of p makes the forgetting mechanism dependent on the recency of the past events. A typical choice of p would be $\exp(-r\Delta_n)$ with discount rate $r \geq 0$. In general, p shall satisfy the telescope property, i.e., $p(a,c) = p(a,b) \cdot p(b,c)$, in order to attach proper discounting to each job title.

Secondly, we incorporate the tenure of each job in generating the candidate typicality $\tilde{c}$,

$$\tilde{c}(X_n, \Delta_n) = i(t_n) \cdot \tanh\Big(q(\Delta_n) \cdot W_c\, z(X_n)\Big) \tag{3}$$

Here, W_c is a learnable parameter, gate i determines the contribution from the current job title, e.g., the input gate in LSTM, and $z(\cdot)$ is the semantics embedding of job title X. We set q to be a non-decreasing convex function $q(\Delta) \equiv 1 - 1/(\Delta+1)^{\lambda}$, with $\lambda > 0$. Undoubtedly, the typicality of a job title increase as one works longer at the very job. However, it might be arguable to assume a uniform growth rate. Similar to the learning curve for a new skill, there should be a marginal effect, that is, the development of expertise slows down after passing a threshold work length. This is achieved with $q(\cdot)$, where a large λ implies the need for less time for an individual to be proficient in a job.

The dynamic typicality encoder $c(\cdot)$ models the information set via the focal employee's career trajectory, $\mathcal{F}_n = \{w_1, ..., w_n\}$. If we feed $c(t_n)$ to a decoder, the derived probability distribution over job titles depends only on e's career progression until t_n.

1.2. Component 2: Dual Temporal-Spatio Graph Encoders

Subsection 1.1 takes a vertical view that solely focuses on an individual's own career development. However, treating everyone's career orthogonal to each other seems to be a strong assumption and restriction. On the macro level, the job mobility pattern, from one job title to another, is an indication of the association among job titles. On the micro level, the social network discovers new career directions for individuals seeking career movement. The necessity of combining vertical view and horizontal view motivates a dynamic graph framework.

We introduce a *spatial-temporal graph* $\mathcal{G}$ as a collection of time-stamped job transition events, $\mathcal{G} = \{\, \omega_i(e)\,;\, \forall e \in E,\ \forall i \,\}$. The occurrence of a job transition event, from title x to x', at time t has two consequences:

1. Introduce a job title node $x' \in \mathcal{V}_T(t)$ if not already exist, and an edge $a_{xx'} \in \mathcal{E}_{TT}(t)$ or increment the weight by 1 if already exists.
2. Introduce an employee node $e \in \mathcal{V}_E(t)$ if not already exist, and an edge $a_{ex'} \in \mathcal{E}_{ET}$ if not already exist.

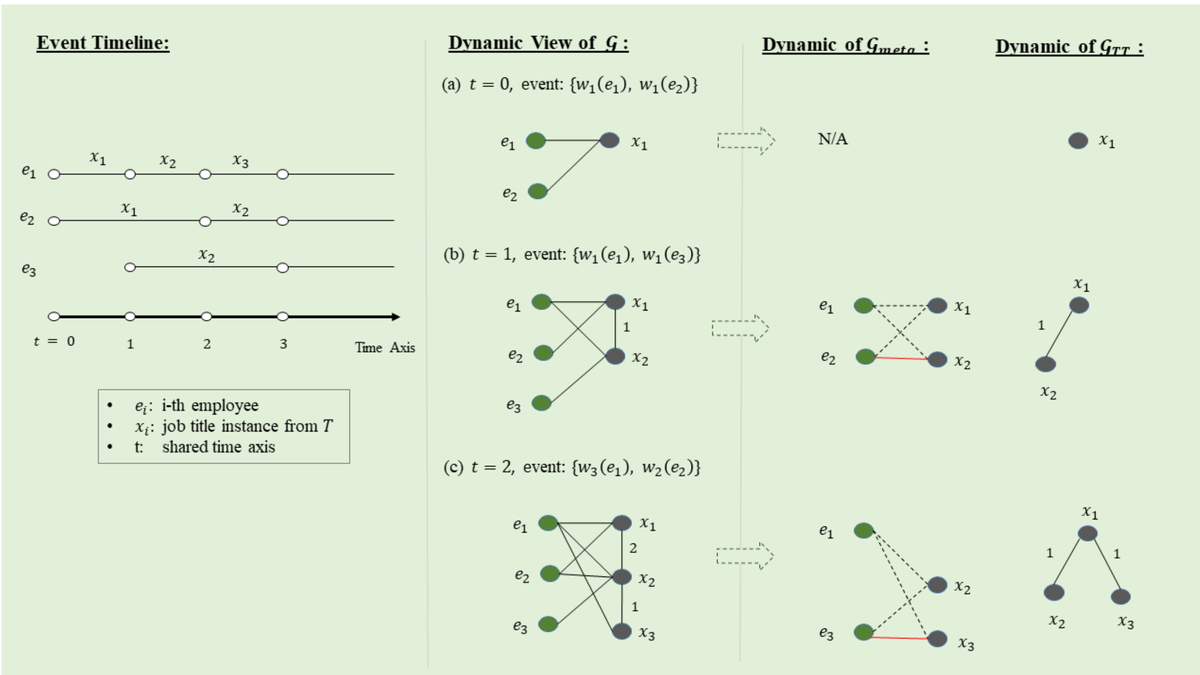
A snapshot of the temporal-spatial graph $\mathcal{G}$ at time t is a heterogeneous bipartite graph $\mathcal{G}(t) \equiv \big(\{\mathcal{V}_E, \mathcal{V}_T\}[. \wedge t], \{\mathcal{E}_{ET}, \mathcal{E}_{TT}\}[. \wedge t]\big)$, where $. \wedge t$ stands for the accumulated nodes or edges up to time t (the second column of Figure 2 is a series of consecutive graph snapshots). Next, we derive two sub-graphs from $\mathcal{G}(t)$, the job title graph and job title-employee graph, which extend the information set $\mathcal{F}_t$ from the previous section.

Job Title Graph

The intention of building a job title graph is to enhance the semantic-based similarity with evolving job market information. One way to achieve this is to track the inter-job title conversions dynamically. At every timestamp t, we extract a job-title graph $\mathcal{G}_{TT}(t)$ from $\mathcal{G}(t)$ with node set $\mathcal{V}_T[t]$ and edge set $\mathcal{E}_{TT}[t]$. Each edge is weighted by the transfer rate computed as the number of employees being converted between the connected job titles. A high transfer rate might signal the convergence of their nature, e.g., the skills required for both jobs have a big overlap, while a small transfer rate implies the difficulty in making a career move between these two titles. The rightmost strip of Figure 2 showcases step-wise change of inter-title association. It's worth mentioning that the $\mathcal{G}_{TT}$ graph does not persist the title interactions from the previous step, as we try to make it the most up-to-date.

In order to systematically fuse the semantic similarity and the transfer rate-based distance, we leverage graph representation learning. There is a large amount of literature on designing

Figure 2 Dual Temporal Spatial Graph Encoder. The graphic representation of career transition events are displayed in the center column. The graph keeps accumulating new titles, employees and their interactions. The right column demonstrates how the dual graph encoders interpret the comprehensive $\mathcal{G}$ at each timestamp. In sub-graph $\mathcal{G}_{meta}$, the edges in red are the actual edges, which are connected via ETET metapath, i.e., the dashed lines. In sub-graph $\mathcal{G}_{TT}$, the edge weights reflect the title conversion in the current period only.



aggregation functions. For simplicity, we use the convolution architecture proposed by Kipf and Welling. The dynamic job title embedding matrix at time t is denoted by $\mathbf{Z}_T(t)$,

$$\mathbf{Z}_T^{(l+1)}(t) = \texttt{relu}(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}\mathbf{Z}_T^{(l)}(t)W^{(l+1)}) \tag{4}$$

The initial embedding $Z_T^{(0)}$ are stacked semantic embeddings via language models, such as BERT. In Eqn 4, $\widetilde{A} = A + \mathbf{I}$ is a weighted adjacency matrix with self-connections, $\widetilde{D}$ is the degree matrix, and $W^{(l+1)}$ is a parameter matrix for the $(l+1)^{\texttt{th}}$-layer.

Employee-Title Graph

Besides self-career development, the employee's job transition is also influenced by others' career choices. The spatial-temporal graph $\mathcal{G}$ allows modeling such social exposure, where potential career opportunities are introduced via reference employees that share similar experiences as the focal employee. We follow an *E-T-E-T metapath* to derive an employee-job title sub-graph $\mathcal{G}_{meta}(t)$ at any time t. The metapath is written expressively as,

$$\mathtt{E} \xrightarrow{\mathtt{workAs}} T \xrightarrow{\mathtt{workAs}^{-1}} E \xrightarrow{\mathtt{workAs}} T$$

To elaborate, let us suppose employee e_1 currently holds $x =$ *software engineer*, which employee e_2 used to work as. We link e_1 and e_2 via the shared job x. e_2 also worked as $x' =$ *data science* after x that further connects e_2 to x'. The path $e_1 \to x \leftarrow e_2 \to x'$ is therefore an instance of metapath ETET. It recommends e_1 a new job x' that might appear atypical based on e_1's own experience.

Due to the dynamic nature of the graph $\mathcal{G}_{meta}(\cdot)$, we impose the following constraints:

- *Limited-Memory:* An employee shall not refer to the "outdated" job titles of the reference employee. It requires specifying a *lookback window size* γ. For instance, in Figure 2, we set $\gamma = 1$, then $\mathcal{G}_{meta}$ derived from (c) ignores the metapath in (b) where e_1 and e_2 share the same job title x_2.
- *Non-Anticipativity:* An employee cannot refer to the titles of the reference employee that occurs in the future. For instance, in Figure 2, (b) is the observed graph for $t \in [1, 2)$, when employee e_1 and e_2 possess(ed) the same job title x_1, later on, e_1 started to worked on job x_3. However, we cannot link e_1 to x_3, as x_3 is not observable by e_2 until passing $t = 2$.

Having set up the ETET graph, we next proceed to its representation learning. Fix a focal employee e, we construct e's time t_n embedding in 3 steps,

1. Given an reference employee e' connected to e via shared title $x_{s'}$, we compute their career similarity as,

$$\alpha_{ee'} \equiv \sigma\Big(c_e(t_n)\, W_c^\top\, c_{e'}(t')\Big) \tag{5}$$

   Here, t' stands for when e' finishes $x_{s'}$, and W_c is a learnable square matrix.

2. Denote the temporal neighborhood of the focal employee as[1]

$$\mathcal{N}_e(t_{n-1}) \equiv \{e' \in \mathcal{G}_{meta}(t_{n-1}),\ e' \in \mathcal{V}_E(t_{n-1})\},$$

   and the set of job titles introduced by reference employee e' as $x(e')$. To calculate the normalized weight to a potential title x' introduced by e',

$$\beta_{ex'} \equiv \frac{\exp\left(\alpha_{ee'}\right)}{\sum_{e' \in \mathcal{N}_e} |x(e')| \exp\left(\alpha_{ee'}\right)}$$

   Here $|\cdot|$ is the cardinality of the set.

3. The aggregation of e's potential job titles via reference employees,

$$d(t_n) \equiv \sigma\Bigg(\sum_{e' \in \mathcal{N}_e} \sum_{x^* \in x(e')} \beta_{ex^*} \cdot z_T(t_{n-1})\Bigg),$$

   where $z_T(t_{n-1})$ is the row corresponding to title x^* in dynamic title embedding matrix Z_T evaluated at time t_{n-1}.

The ultimate embedding for focal employee is just the concatenation of c_n and d_n, $z_e(t_n) \equiv c_e(t_n) \,||\, d_e(t_n)$. The integration of self-evolving typicality and the recommended job titles from graph encoders extends the information set $\mathcal{F}_n$ to the graph snapshot $\mathcal{G}(t_{n-1})$. Notice, setting $\mathcal{F}_n = \mathcal{G}(t_n)$ leads to data leakage as it perceives the job title for $[t_n, t_{n+1})$. In subsection 1.3, we details the design of a decoder based on the enlarged information set $\{\mathcal{F}_n\}$.

[1] We suppress time dependence when there is no ambiguity, e.g., $\mathcal{N}_e(t) = \mathcal{N}_e$.

1.3. Component 3: Bi-level Probabilistic Decoder

In general, the job title collection T is quite considerable, which impose challenges for predictor to locate the candidate job title set, let alone further narrowing down to a handful job titles. As discussed in the data collection section (TO WRITE), we, in addition, maintain a coarse classification of job titles, $T = \{T_1, T_2, ..., T_K\}$ with T_i being the job title group. For any job $x \in T$, we have a title group identification function $\mathtt{gp}(\cdot)$ that finds a T_i in T, i.e., $\mathtt{gp}(x) = T_i \subset T$. Based on the two-layer structure, we formulate the probability of getting next job x standing at t_{n-1} as,

$$\begin{aligned} \mathbb{P}(X_{n+1} = x \mid \mathcal{F}_n) &= \mathbb{P}\left(\mathtt{gp}(X_{n+1}) = \mathtt{gp}(x) \,\middle|\, \mathcal{F}_n\right) \cdot \mathbb{P}\left(X_{n+1} = x \,\middle|\, \mathtt{gp}(x)\right) \\ &\equiv p\left(\mathtt{gp}(x)\right) \cdot \frac{s(x)}{\sum_{x' \in \mathtt{gp}(x)} s(x')} \end{aligned} \tag{6}$$

The design of decoder amounts to specifying how p and z decipher the summarized information from encoder.

Distribution over Title Group

The function $p(\cdot)$ computes the probability of being in group $\mathtt{gp}(x)$. Given the concatenated embedding $z_e(t_n)$, we can feed it into a deep neural network connecting softamx to generate probability distribution over job title groups. To make predictor more informative, we can further consider the focal employee's personality. An employee with high devotion usually sticks with a few title groups that is best aligned with his/her initial expectation, while an employee lacks of concentration tends to explore different career directions. In either case, the personality has a direct impact on the shape of distribution, i.e., "unimodal" (concentration) or "uniform" (diversification). To quantify such behavior, we introduce a *dynamic diversity measure* $rho(\cdot)$ that can be applied on the career trajectory up to current time. At time t_n, the diversity is calculated as,

$$\rho(t_n) = -\sum_{x \in U(n)} h(x) \ln h(x), \;\; \textit{where } h(x) = \frac{\sum \Delta_i \cdot \mathbf{1}_{\{x_i = x\}}}{\sum_{i=1}^{n} \Delta_i} \tag{7}$$

where $U(n)$ are set of unique titles up to t_n. Putting all above together, we have

$$p\big(gp(x)\big) = \texttt{softmax} \circ \texttt{ann}\Bigg(W_g,\, z(t_n)||\rho(t_n)\Bigg) \tag{8}$$

Distribution over Job Title

Similar to Eqn (8), we structure scoring function $s(\cdot)$ as

$$s(x) = \texttt{softmax} \circ \texttt{ann}\Bigg(W_s,\, z(t_n)\Bigg) \tag{9}$$

Essentially, $s(\cdot)$ is a conditional distribution over T. To make $s(x)$ further conditioning on the title group it belongs to, we normalize the score of each title x within in $\texttt{gp}(x)$, which yields the second term on the right hand side of Eqn. (6).

1.4. Design of Algorithm

One of the main challenges in implementing the spatial-temporal job mobility predictor is *batching*. All employees are dependent on each other due to social influence graph $\mathcal{G}_{meta}$. If we random select a subset of employees as a batch, it might not include all employees interacting with the subset, which lessens the effectiveness of creating $\mathcal{G}_{meta}(\cdot)$. To address the concern, we add a pre-process step to build $\mathcal{G}_{meta}$ for each timestamp in advance. The information of $\mathcal{G}_{meta}(t)$, $\forall t$, is stored in the following data structure $\mathcal{S}$: for each focal employee e,

```
Focal Employee e :
{
    ref employee e1 : reference title set REF_TITLE(e1)
    ref employee e2 : reference title set REF_TITLE(e2)
    ......
}
```

This is a one-time effort but the stored data can be used across different batches and epochs. It efficiently queries the employees and job titles needed to embed a set of focal employees. When applied on batching,

1. firstly, we randomly select a set of focal employees form E;
2. secondly, we repeatedly query $\mathcal{S}$ to collate all employees needed in order to assemble the embeddings of focal employees at each time stamp.

Along the same line, we store the graph structure of $\mathcal{G}_{TT}(t)$ for all t as well as the semantic embedding of all job titles. In training time, we generate the dynamic title embedding via Eqn. (4), which will be fixed for for a given batch.

Algorithm 1: Encoder Algorithm – Single Batch

Data: a batch of employees $B \subset E$ with their career trajectories $\{\vec{\omega}(e)\}_{e\in B}$, and $\mathcal{G}_{TT}(\cdot)$
Result: the dynamic embedding for all $e \in B$ at their respective transition times.

1. Get dynamic job title embeddings: $z_T(\cdot) = \texttt{GCN}\big(\mathcal{G}_{TT}(\cdot)\big)$;

2. Expand set of employees: $B' = B \cup \texttt{QUERY\_E}(\mathcal{S}, B)$ // find all employees via $\mathcal{S}$;

3. Initialize Career Memory container: `CarMem`;

4. Initialize step-wise embedding container $\texttt{Emb}[\cdot][\cdot]$;

4. **While** $i \le N$:

5. Typicality embedding: $\{c_e(t_i)\}_{e\in B'}$ via (1), using $z_T(i)$;

6. Update `CarMem` with $\{c_e(t_i)\}$; // keep K-step memory **7.** **For** $e \in B$:

8. **If** $\texttt{hasEvent}(\omega_i(e))$:

9. Get ref. employee and titles pairs $\mathcal{P} = \texttt{QUERY\_T}(\mathcal{S}, e)$;

10. Generate $d_e(t_i) = \texttt{Syn}(\texttt{CarMem}, \mathcal{P})$; // 3-step approach

11. Update embedding container $\texttt{Emb}[e][t_i] = c_e(t_i)||d_e(t_i)$;

12. **Return** $\texttt{Emb}[\cdot][\cdot]$