

Who Will Travel With Me? Personalized Ranking Using Attributed Network Embedding for Pooling

Lei Tang^{id}, Member, IEEE, Zihang Liu, Rongguo Zhang, Zongtao Duan^{id}, and Yunji Liang

Abstract—In ride matching, the search results can be personalized for a particular driver. Given a query with trip plans, it is advantageous to rank potential riders in terms of who are most appealing to the driver for increasing occupancy rates. While personalized ranking approaches such as collaborative filtering and factorization are available, they are not suitable for pooling because candidate riders are associated with different preferences, and their travel is sparsely distributed with a long tail of users for a few popular destinations. The user embedding method is a good candidate in terms of alleviating data sparsity, but it has issues such as difficulty encoding user preferences from rich information. In this study, we explore user embedding techniques for the purposes of short-term personalized rider ranking, where the aim is to present to drivers a set of potential riders who share similar itineraries with them and can be picked up on their current route. Considering trip requests, along with the preferences issued in advance, this study uses attribute representations to rank the riders based on the higher-order similarities in the participants' itineraries in a three-step manner: (i) start with a distributed representation of the riders' preference regarding the cost of extra distance, (ii) generate user embeddings in a heterogeneous network with the meeting points and associated waiting times, and (iii) match and rank riders for drivers depending on an attribute fusion operation by adopting a personal route and schedule. Our proposed method performs well in an offline estimation on a huge dataset from DiDi in Chengdu, China. Experimental results indicate that with the learned embeddings, we can obtain statistically significant advancements (e.g., 4.6–29.5% increase in mean reciprocal rank (MRR); 2.8–17.4% in normalized discounted cumulative gain (nDCG)) over current methods for pooling ranking. Furthermore, we implement the proposed method on our simulated pooling system. These results validate that personalized ranking can undoubtedly boost the number of trips served, and reduce the total trip distance and waiting time.

Index Terms—Attributed network representation, personalized ranking, pooling, SUMO.

I. INTRODUCTION

POOLING is an emerging mode of transport that enables individuals with similar itineraries and schedules to share

Manuscript received 27 January 2021; revised 11 July 2021 and 1 September 2021; accepted 15 September 2021. Date of publication 27 September 2021; date of current version 9 August 2022. This work was supported in part by the Key Research and Development Plan Project of Shaanxi Province, China, under Grant 2019ZDLGY17-08 and Grant 2019ZDLGY03-09-01. The Associate Editor for this article was M. Mesbah. (Corresponding author: Lei Tang.)

Lei Tang, Zihang Liu, Rongguo Zhang, and Zongtao Duan are with the School of Information Engineering, Chang'an University, Xi'an 710064, China (e-mail: tanglei24@chd.edu.cn; lzhhollyhang@chd.edu.cn; rongguozhang@chd.edu.cn; ztduan@chd.edu.cn).

Yunji Liang is with the School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: liangyunji@nwpu.edu.cn).

Digital Object Identifier 10.1109/TITS.2021.3113661

a journey in an individual vehicle. By increasing the occupancy of vehicles, the driver and rider(s) naturally divide the costs of fuel and tolls so that all benefit from the shared ride [1]. In China, for instance, challenges regarding heavy traffic and poor parking lots are acute problems with far-reaching consequences. In this context, pooling is attractive and offers immense benefits, including opportunities to increase the driver's income and reduce their down time [2], [3]. With our previous data study of real-world trips (Fig. 1), the potential for increased mobility sharing is clear [4]. A fraction of trips, ranging between 15% and 25%, can be shared by a simple model for trip aggregation. Furthermore, on average, over 22% of trips may potentially be shared with at least one other trip when two riders are in close proximity to each other in time and space (walking distance from one another is less than 0.5 km, and the time difference is less than 300 seconds). This percentage increases to over 45% if a car can transport up to four passengers, which implies that the number of carpoolers is a determinate of trip sharing. However, a lower utilization of vehicles occurring under 200 trips is probably related to the sparse spread of trip requests; this is highlighted via weekly patterns in mobility sharing (see Fig. 1(b)). The scaling law relating the daily number of shared trips with total trips has an overall R^2 value of 0.93. The shared trips are especially ubiquitous on weekends compared with the routine weekly trip requests. Thus, trip density and spatiotemporal patterns are likely to play a large role in trip sharing. Companies such as Didi have introduced pooling to their platforms. By the end of 2017, there were over 21 million DiDi drivers in China, serving more than 450 million passengers from 400 cities daily [5]. It is reported that due to pooling, the cost of detours were reduced by an average of 30%, 1.05 billion seats were shared by DiDi's pooling services, and the traffic congestion index decreased by 10% to 20%.

In the context of this study, multiple riders can be matched with a driver whose vehicle capacity is not exceeded. For an area with high passenger volume, drivers often prefer pooling services that can selectively provide passengers for them quickly. Effective ride-matching depends not only on the orientation of drivers and riders under time and distance constraints, but more substantially on personal preferences, adaptable idling time, pooling trip types [6], and social preferences [7]. Therefore, personalizing matching results for a particular driver based on time schedules as well as preferences, and further ranking riders is the most attractive and effective type of solution. With the one million drivers and one billion riders on platforms like Uber, the most critical problem is how to personalize and quickly rank

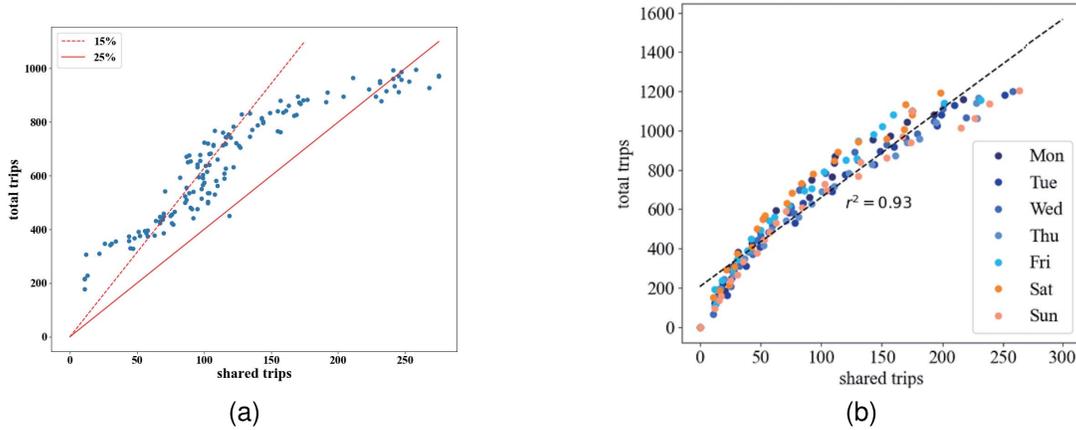


Fig. 1. Mobility sharing analysis (Chengdu DiDi Express trip data from November 2016 were used). (a) The interplay between the daily number of total DiDi trips and combinable trips with the same origin. (b) Weekly patterns by clustering points with the same color. The colors of the dots correspond to different weekdays.

riders whose time schedule and social interaction preferences are in line with the driver's desires for trip cost and duration.

There are three major technical challenges facing personalized ranking in ride-matching:

Attribute: Pooling can be understood as a sociological phenomenon due to the rich information about participants, ranging from gender, age, and income level [8], to hobbies and talkativeness [9]. We term all such auxiliary information as attributes, which have a large influence on preference elicitation and ranking when jointly considering attributes and schedule proximity. A deep challenge for integrating personal preferences is to take in different underlying attributes embedded within the raw trajectory data.

Sparsity: Due to the fact that a large number of passengers tend to travel to a relatively small number of popular sites [10], there is a long tail in the destination-frequency distribution (Fig. 2(a)). This leads to sparsity in the trajectory data. It is thus difficult to learn the proximity in both spatial and temporal dimensions, and train an accurate ranking model; this is especially true for passengers with destinations that are quite isolated. Fitting $\log(\text{frequency})$ and $\log(\text{rank})$ using linear regression yields an R-squared value of 0.88259, and a p-value of $4.775e-17$. Fig. 2(a) shows that the rank-frequency follows a power law distribution in which a few destinations dominate the data.

Scalability: Despite the fact that many existing approaches for ride-matching work well on a first-come-first-served basis in the general spatial extent [11], [12], they ignore the diverse relationships embedded within other information indicators. For example, the greater the utilization rate of vehicles for serving, the less time a vehicle will have to pick up a passenger (see Fig. 2(b) and 2(c)). This law provides a means of ranking riders by considering decreasing the average time needed to be en route to next passengers. Such indicators should thus be represented efficiently for good ride-matching; however, this poses difficulties when dealing with a much larger-scale travel dataset.

Considering trip requests with preferences that are issued in advance, in this study, we address the challenges of calculating similarities between driver and candidate riders that need to be ranked for further personalization. Existing works have very rarely explicitly considered personal preferences for carpoolers as the objective. Here, we jointly optimize ranking for both preference and schedule similarity. To achieve this, we construct an attributed pooling network (ARN) derived from the participants' travel behavior, and then apply network embedding methods to learn the user embeddings. Since drivers are typically eager to pick up and drop off riders en route [13], there is a cost to the rider due to increased walking and waiting. We can use travel cost signals, e.g., walking toward the meeting point and waiting for a ride, to show rider preferences in real-time personalization.

We then propose to leverage the attribute embeddings, which are low-dimensional vector representations learned from these signals, to generate a set of candidate riders based on the similarities of both preference and schedule. The candidate riders are jointly computed via these embedding vectors. Note that in earlier works, spatiotemporal optimality is used to estimate these similarities. However, this method explicitly views the co-occurrence of participants in spatial and temporal dimensions [12], [14], [15]. In our work, using a meta-path-based walk in an ARN, we can obtain higher-order closenesses between drivers and riders.

The novel contributions of this paper are as follows:

- (1) We formally define the problem of personalized ranking for pooling, which demonstrates the importance of attributes for learning participant preferences in benefiting from the shared ride. We leverage distance saving to create personalization attributes for our ranking model, and to enhance our ride-matching.
- (2) We design a powerful method to create the attributed pooling network from the travel behavior of two million passengers of Didi Chuxing. Efficient and scalable embedding for an ARN are developed to alleviate the sparsity problem. Our method can preserve both the

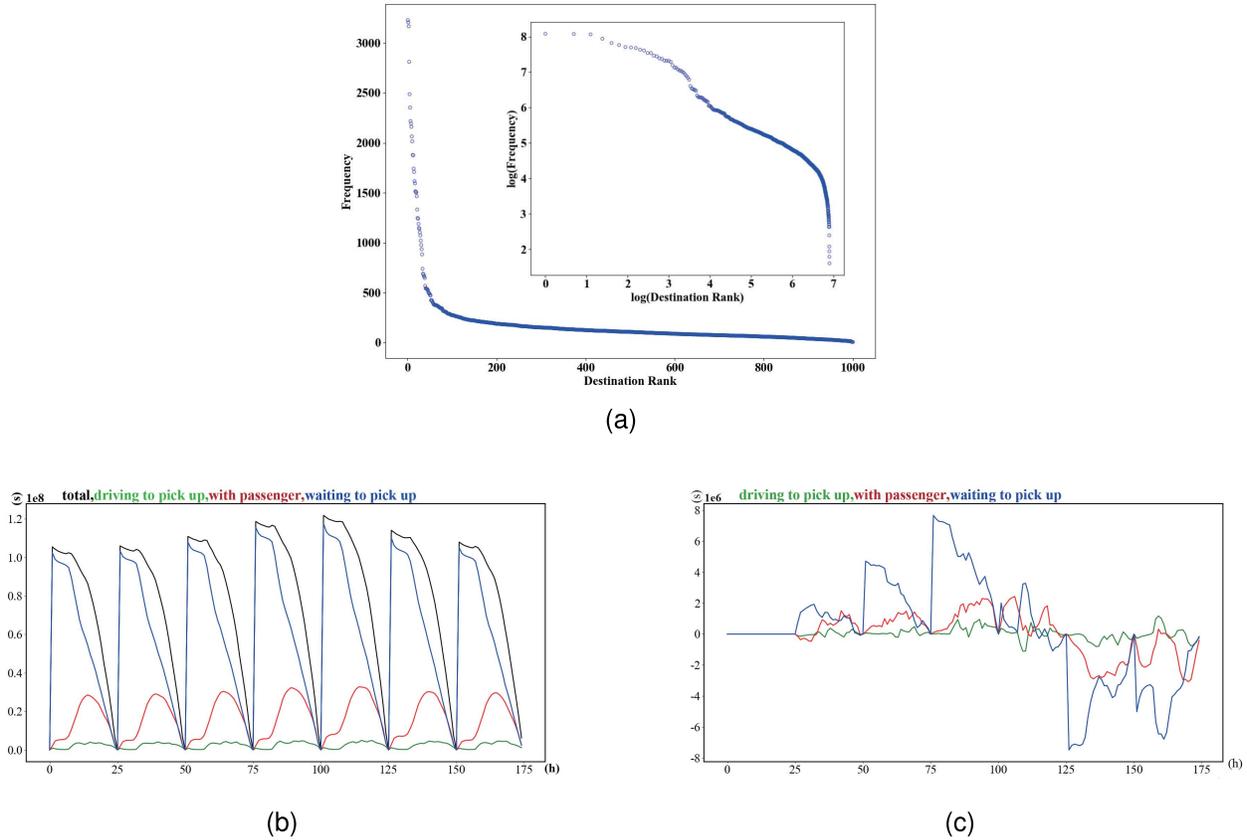


Fig. 2. (a) Destination rank-frequency and log-log plot. (b) The daily changes in vehicles' operations modes (i.e., on the way to pick up; serving; cruising). (c) The differences from the last day's modes.

schedule and preference proximity of each driver-rider pair.

- (3) We carry out experiments on real-world datasets for the two tasks of pooling prediction and ride ranking. Our results demonstrate the capability of the proposed method. We also evaluate our model by developing a realistic pooling simulator that provides a platform for comparing various matching policies on relevant metrics.

II. RELATED WORK

A. Matching in Pooling

Driver-rider matching services predict personal willingness to share a ride conforming to travel characteristics [16] such as mode of transport and trip purpose [17]. There is a large collection of studies on stable matches in two-sided markets. Typical applications include the centralized matching of a set of passengers and drivers participating in shared taxi rides [15], [18]. Earlier work on pooling matching typically considers both the spatial and temporal dimensions of exact pick-up locations, and typically involve matching a driver with the nearest rider on a first-come-first-served basis [13], [19], [20]. Researchers have proposed various algorithms focused on the discovery of a rider's trip information to optimize system efficiency, such as reduction in total travel time and cost [4], [21]. Furuhata *et al.* [22] consider the shared operating cost as the optimization objective to detect the routes

and schedules of the vehicles in real time. Such vehicles can service the ride requests that appear dynamically over time at different locations. A new rider can be matched by minimizing the total cost of other carpoolers when receiving the request. Stiglic *et al.* [13] introduce meeting points to make a driver pick up multiple riders without making other stops. By studying the participant's time flexibility on trip duration, Stiglic *et al.* [23] also demonstrates the efficiency gains of different objectives to achieve an optimal matching.

These typical paradigms of pooling matching rarely take into account the traveler's attributes, such as gender, transportation mode, carry-on baggage, and payment. The underlying assumption is that no rejection would arise after the matching is performed. In practice, a rejection can happen if the rider finds the assigned driver unsuitable, which may result from the rider's preference attribute, including travel distance, time requirements, destination types [24], and characteristics of the driver. Disregarding such attributes can lead to a poor system acceptance [12] and matching rate [23]. Existing methods for this matching problem are primarily developed for graphs with known edge weights [25], [26]; however, this is not suitable for providing solutions with personalized matching, where we would like to leverage as many fine-grained attributes as possible. Zheng and Chen [27] addressed the optimal assignment problem by assigning users with the proper tasks to maximize their acceptance. They also developed a pruning strategy for obtaining the optimal

solution. Moreover, Berlingerio *et al.* [28] investigated the importance of social attributes in constructing an enjoyable pooling experience. Our previous work [7] measured the bias of a rider to group with others via social media data, and relied on the destinations derived to identify passengers who could travel together. Wang *et al.* [29] discussed pooling with different types of social contact, and imposed a matching constraint using the trust level between participants. Unsurprisingly, it has been concluded that aside from schedule proximity, preference attributes also provide important factors for the pooling matching procedure [30]. Although Zhang and Zhao pointed out the social benefits of pooling, and developed matching according to the preference rank orders for fellow passengers, they neither addressed the issues of real-valued preference representation nor utilized the preferences to offset the weaknesses of efficiency-based matching models. Thus, to obtain more informative representations for participants, we should model information on attributes to complement the OD-based matching, thereby enabling the two parts to closely collaborate with each other.

B. Similarity Join in Matching

Typically, ride-matching refers to a search for a group of individual travelers who have similar itineraries and time schedules. In this context, the similarity join is a major operation for matching [31]. Similarity join and search were widely studied [32]–[36], and many studies have been conducted to support proximity-based trajectory processing [37], [38], group rides [15], [39], and order dispatching [13], [19], [40]. However, most existing methods only focus on leveraging the explicit trip information [41], [42], such as extra locations and expected arrival time [43]. In reality, multiple types of supplementary data (e.g., spatial distribution, temporal period, and moving path) are available to enrich trip data [44]. This rich attribute information also reveals the participant’s preferences, thus exerting a huge impact on the formation of pooling. In addition, the riders are often connected to multiple typed objects, such as a special location (e.g., work place), or a fellow rider (e.g., friend) when traveling, which correspond to various kinds of relations. It is thus possible for network-based methods [45] to represent complicated travel features and the connections between them, such as when and where the participants should meet to minimize their trip costs [46].

Many advanced similarity join algorithms have emerged for information networks, where given two sets of nodes in a network, a similarity join returns several pairs of nodes, which are ranked based on the neighborhood or structural proximities [35]. One mainline work leverages meta-path-based contexts for preserving semantic information [47]. Xiong *et al.* considered various semantic meanings underlying links, and addressed the problem of the similarity join while considering the heterogeneity and diversity of the networks [31]. These methods were widely applied in various fields, including feature prediction [48], [49] and object recommendation [50]. Note that these applications deepened domain knowledge to prefer meta-paths while ignoring the attributes of nodes, and thus they cannot catch the rich semantics in a real travel

TABLE I
SYMBOLS

Symbol	Description
u_i	pooling participant
$\mathcal{N}_{\mathcal{P}}^n(u_i)$	Neighbors of u_i along with meta-path \mathcal{P}
$d_i, r_j, o_i, d_i, o_j, d_j$	Driver and rider, their origin, and their destination
$M_j^{(i)}$	For rider r_j , the feasible meeting points on the original route of driver d_i
m_α, m_β	Pick-up meeting point, drop-off meeting point
t_m	Waiting time at m
σ_j	Maximum walking distance for r_j
$t_{o_j}, t_{o_j m_\alpha}$	Arrival time at o_j , time period from departing o_j to arriving at m_α
$\delta^{(i,j)}$	System-wide distance savings for a match of d_i with r_j

environment. Moreover, a few studies [51]–[53] leverage the network-based similarity joins to achieve pooling matching. Some apply social ties data to nodes to improve the robustness of the matching. Nonetheless, these methods discount the heterogeneity of travel-related information, and hence cannot model diverse relations with a network structure, nor can they search the node pairs in terms of structure similarity.

III. PRELIMINARIES

In the following we provide a set of definitions to systematically formulate the problem of personalized ranking as an embedding learning problem on the ARN.

Given a heterogeneous network [54] $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, each node $v_i \in \mathcal{V}$ is correlated with types of attributes $a_i \in \mathcal{A}$. Each edge $e_{ij}^r = (v_i, v_j) \in \mathcal{E}_r$, $\mathcal{E} = \cup_r \mathcal{E}_r$ is classified into a particular edge type r . In practice, \mathcal{G} could be undirected, and we have $e_{ij}^r = e_{ji}^r$. Our goal is to learn a low-dimensional representation for each node (participant) in the space \mathcal{R}_d ($d \ll |\mathcal{V}|$). Since the spatiotemporal patterns and attributes provide distinct sources of proximities, we are inclined to obtain a comprehensive representation of the participants. In our work, we strive to develop a ranking method that learns user embedding in terms of schedules and preferences. Notations are summarized in Table I.

A. Problem Statement

We consider a pooling setting in which a sequence of trip requests over time from potential riders is issued in advance. Each request designates an origin and a destination location, as well as auxiliary information that implies its potential timing [14]. Referring to the “cap of maximal detour” rule [55], the difference between the departure time and the actual pick-up time of shared trip should be considered. For any driver–rider pair, if one’s actual arrival time to the pick-up location is greater than a predetermined time flexibility, the pair is considered non-shareable [30]. In other words, riders to be matched, the two must be able to board at the same location within a particular time period.

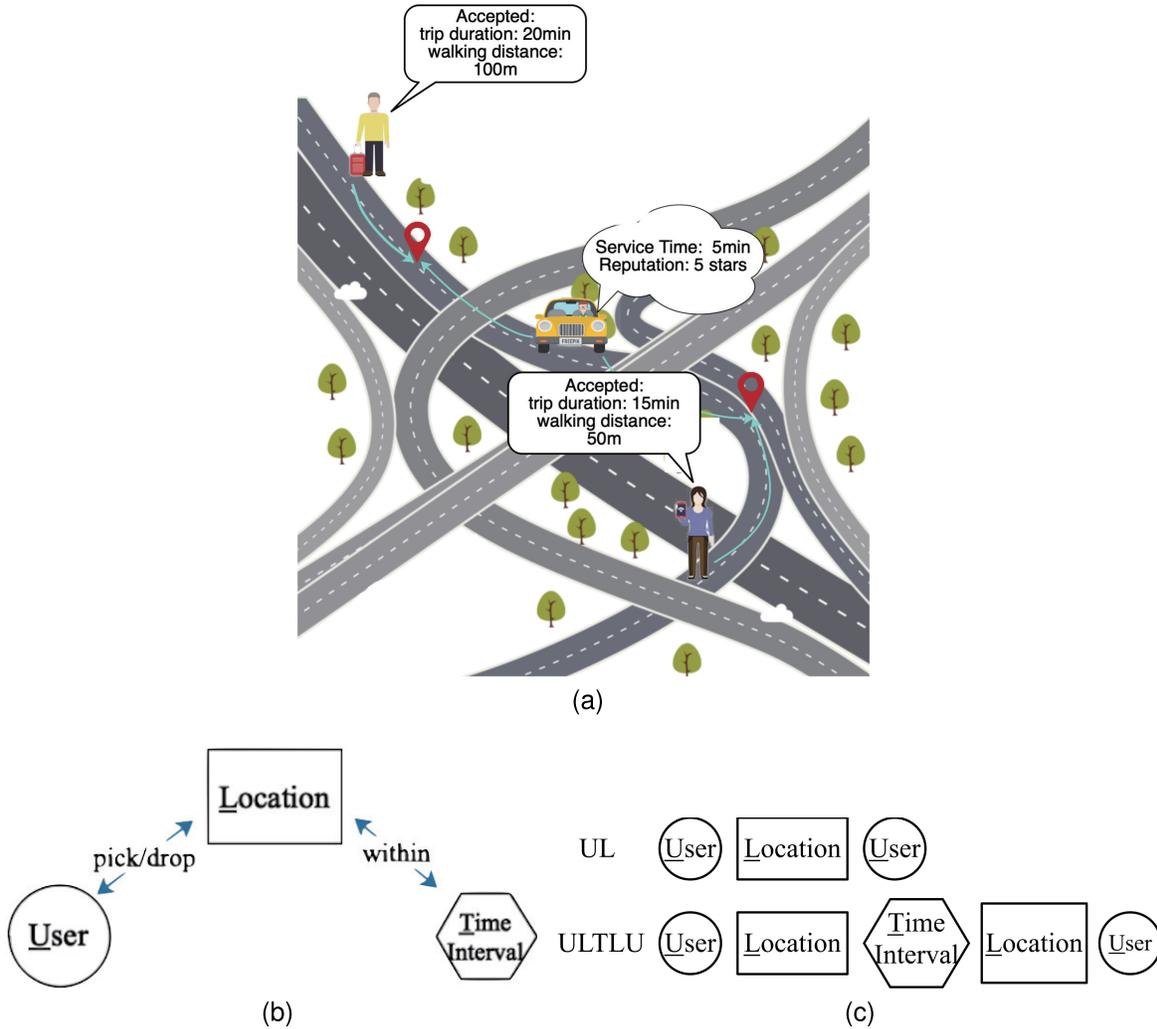


Fig. 3. ARN and network schema. (a) Toy example. (b) Network schema. (c) Symmetric meta-paths.

A similar work can be found in [43], where a vehicle-shareability network is proposed to link multiple shareable trips (denoted as nodes), and thus reduce fleet size. The network assumes that the driver makes a detour for the next trip, but ignores the possible increase in travel time, leading to the driver possibly not arriving at the destination on time. Moreover, the minimum fleet problem is NP-hard and has to be optimized, which makes it difficult to solve efficiently. However, in the present work, a shared trip is represented as an attributed heterogeneous network, in which nodes contain user types \mathcal{U} (i.e., drivers and riders), location types \mathcal{L} (i.e., pick-up points), and time types \mathcal{T} (i.e., time differences). The relations of nodes among \mathcal{U} , \mathcal{L} , and \mathcal{T} are three-dimensional, and each relation represents the various spatiotemporal patterns of the participant’s travel behavior. Each user type is associated with a certain attribute \mathcal{A} referring to their preferences for the shared ride, such as “gender,” “reputation,” and “trip costs.” Fig. 3 shows that the ARN can describe pooling behaviors of participants via considering extra meeting points on the routes.

Definition 1 (Meta-Path-Guided Schedule Proximity): denotes the proximity of nodes in terms of the participant’s schedules. For u_i , there exists a visited node u_j when u_i

walks along the given symmetric meta-path \mathcal{P} (starting from u_i), which indicates the direct proximity. We employ the word “neighbors” to indicate the $n - th$ step nodes of u_i as $\mathcal{N}_{\mathcal{P}}^n(u_i)$.

Taking Fig. 3(c) as an example, given the symmetric meta-path “User - Location - User(ULU)” and a user u_1 , we can obtain neighbors $\mathcal{N}_{\mathcal{P}}^1(u_1) = l_1, l_2$ and $\mathcal{N}_{\mathcal{P}}^2(u_1) = u_2$. Then, all the neighbors of u_1 are denoted as $\mathcal{N}_{ULU}(u_1) = u_1, l_1, l_2, u_2$. Likewise, the neighbors of a location node l_1 are $\mathcal{N}_{ULU}(l_1) = u_1, u_2$. The meta-path “ULU” means that we can assign a driver to a passenger if they are connected by a path containing a shared location. Details can be found in our previous work [46], which focuses more on striving for optimal matches via the spatiotemporal relationships between riders and drivers, with no consideration for personal preferences on shared trips.

Definition 2 (Trip Attribute): Represents the participant preferences in each shared trip. We use the distance savings as the measure of attributes. Consider the example where for each driver $d_i \in D$ and rider $r_j \in R$, if there is a possible match between d_i and r_j (with a pick up at m_α and/or a drop off at m_β), edge connecting nodes d_i and r_j arise. We use o_i and d_i to indicate the origin and destination of a driver,

and o_j and d_j to indicate a rider. The system-wide savings in total trip distance due to a match is

$$\begin{aligned} \delta^{(i,j)} &= (d_{o_i d_i} + \sum_{j \in R} d_{o_j d_j}) - (d_{o_i m_\alpha} + d_{m_\beta d_i} + d_{m_\alpha m_\beta} \\ &\quad + \sum_{j \in R} (d_{o_j m_\alpha} + d_{m_\beta d_j})) \\ &= \sum_{j \in R} (d_{o_j d_j} - (d_{o_j m_\alpha} + d_{m_\beta d_j})) \end{aligned} \quad (1)$$

We denote the distance from a rider's origin to their destination as $d_{o_j d_j}$. We note that a pick up or drop off can occur at the rider's origin and destination as well, i.e., $m_\alpha = o_j$ and $m_\beta = d_j$.

In this work, we investigate the problem of the personalized ranking of riders using a sequence of trip announcements. The profile vectors can be automatically learned for producing the schedule patterns of a participant. We then specify the problem as a task of calculating similarities between drivers and candidate riders that need to be ranked with deep representations. An ARN presents the overall travel profile, and a rider's personal preferences are usually implied by their attributes. Hence, this task has a combined objective of keeping both the total schedule patterns and rider attributes in representation learning.

Formally, given

- An ARN, i.e., $\mathcal{G} = (\mathcal{U}, \mathcal{L}, \mathcal{T}, \mathcal{E}, \mathcal{A}, \mathcal{P})$;
- Two disjoint sets of user-type nodes, i.e., a set of riders $R = \{r_1, r_2, \dots, r_j\} \subset \mathcal{U}$, and a set of drivers $D = \{d_1, d_2, \dots, d_i\} \subset \mathcal{U}$;
- Meta-path-based relations among the nodes, i.e., $\mathcal{P} \in \mathcal{R}^{\mathcal{U} \times \mathcal{L}} \cup \mathcal{R}^{\mathcal{U} \times \mathcal{L} \times \mathcal{T}}$;
- K different sets of attributes of riders r_j for driver d_i , i.e., $\mathcal{A}_i = \{\delta^{(i,1)}, \dots, \delta^{(i,j)}, \dots\} (i = 1, \dots, K) (j = 1, \dots, M)$.

We aim to rank riders listed in R based on embedding similarities between representations of d_i and r_j , i.e.,

$$\mathcal{Z} = \left(\bigcup_{i=1}^K \mathcal{Z}^{(i)}, \bigcup_{j=1}^M \mathcal{Z}^{(j)} \right), \mathcal{Z}^{(j)} = \text{aggregator}(\mathcal{Z}_{\mathcal{A}}^{(j)}, \mathcal{Z}_{\mathcal{S}}^{(j)})$$

$$(r_j \in \mathcal{N}_p^n(d_i))$$

where $\mathcal{Z}^{(i)}$ denotes the representation of driver nodes. The embedding $\mathcal{Z}^{(j)}$ of node r_j is aggregated from its attribute embedding $\mathcal{Z}_{\mathcal{A}}^{(j)}$ and structural embedding $\mathcal{Z}_{\mathcal{S}}^{(j)}$ on \mathcal{G} .

B. Framework Overview

The basic idea of the proposed framework is to design an attributed network representation for enriching the personalized ranking in ride-matching. With the help of the ARN, our framework leverages meta-paths to guide the selection of different "neighbors" and obtain the rich embeddings of drivers and riders. Moreover, we represent the rider preferences regarding cost of trip distance via attribute embedding.

Fig. 4 exhibits a sketch of the proposed framework that contains four tasks: (i) constructing the ARN and aggregating the information of meta-path-guided neighbors to represent the profiles of pooling; (ii) measuring the distance savings to

indicate the personal attribute; (iii) fusing the embeddings of attributes and network structure to develop a jointly learned participant representation from the ARN; and (iv) measuring the similarity between embeddings to rank candidate riders. We illustrate these tasks in detail in the following sections.

IV. ENCODING ATTRIBUTES

A. Determining Meeting Points

Stiglic *et al.* demonstrated the benefits of introducing meeting points for increasing the number of matched participants [13]. We view a ride-share setting where riders are inclined to walk to and from different meeting points to facilitate easy trip sharing. In this paper, we discuss the personal distance savings when matching drivers and riders with meeting points. The first focus is to identify the best pickup meeting points as soon as the trip announcements has been received.

Each rider r_j defines a maximum distance σ_j within which they can reach a meeting point. R^* is the set of candidate riders who may be picked up and dropped off at the points are σ_j distant from their origin or destination. The set of possible meeting points along the original route $Route(d_i)$ is $M_j^{(i)} := \{m \in Route(d_i) | d_{m o_j} \leq \sigma_j \text{ or } d_{m d_j} \leq \sigma_j\}$ for r_j . With a larger set of trip announcements, the efficiency issue becomes huge when searching $M_j^{(i)}$ because most existing methods are iterative and have a high computational cost. Our approach applies an R-tree for the spatial indexing. We organize a hierarchy of nested 2d rectangles to index the driver's trajectory data. Each node represents the minimum bounding rectangle (MBR) of its children or, for leaf nodes, a set of sampling points on $Route(d_i)$. By computing the distance between r_j 's origin and all the points, searching in the R-tree runs from the root to the leaf nodes to find $M_j^{(i)}$ within a fixed-radius search region.

Once we obtain the set of $M_j^{(i)}$ for a rider, we need to ensure the time feasibility of a matching with $M_j^{(i)}$. While a pooling trip is feasible if all participants can arrive the pickup location, it may still be inconvenient for a driver and/or rider as it may involve some waiting time. The acceptable waiting time of a rider at each $m \in M_j^{(i)}$ is denoted as t_m , i.e., the time window between the earliest arrival times from riders' origins to m , and the latest departure times from m to their destinations. Here, t_m is different for different riders and meeting points. Referring to Stiglic *et al.*'s work [13], a driver can be matched with a rider if there is an overlapping interval within their time windows. In other words, a match is time-feasible if all participants can reside at m during an interval overlapping within the time window of $r_j \in R^*$ and d_i , i.e., either $\bigcap_{m \in M_j^{(i)}} (t_m \subset \tau_m)$ or $\bigcap_{m \in M_j^{(i)}} (\tau_m \subset t_m)$, where τ_m is the time window of the driver.

Consider the example for driver d_i and rider r_j with a pick-up meeting point m_α and a drop-off meeting point m_β . We define t_{m_α} for r_j and τ_{m_α} for d_i to be $[t_{o_j} + t_{o_j m_\alpha}, t_{d_j} - (t_{d_j m_\beta} + t_{m_\alpha m_\beta})]$ and $[t_{o_i} + t_{o_i m_\alpha}, t_{d_i} - (t_{d_i m_\beta} + t_{m_\alpha m_\beta})]$. We denote the arrival time at o_j as t_{o_j} , and the time period from departing o_j to arriving at m_α as $t_{o_j m_\alpha}$. The candidate meeting points $M_j^{(i)}$ and associated overlapping intervals are

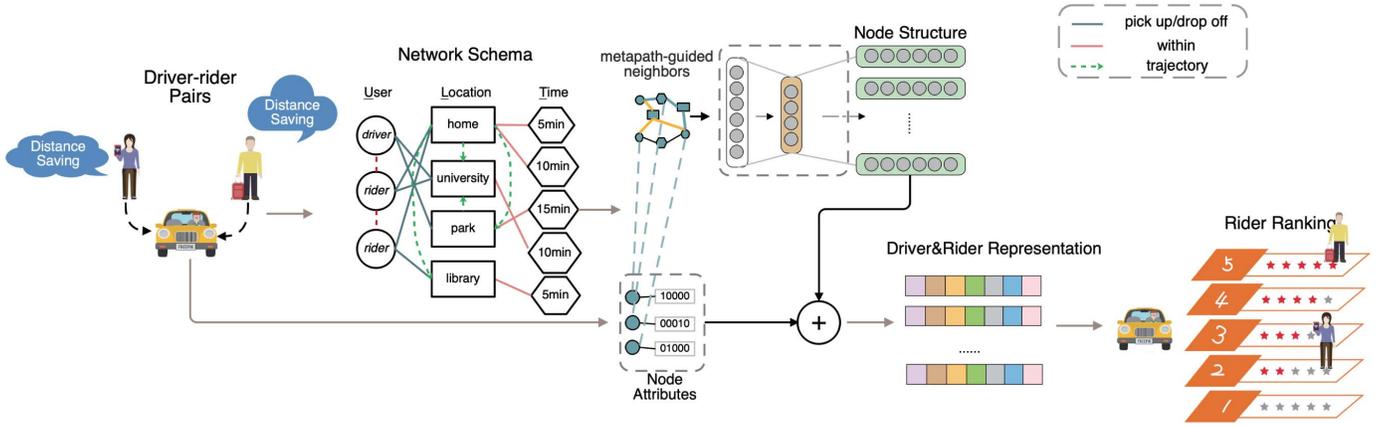


Fig. 4. An outline of rider ranking via joint embedding.

then pruned if both $t_{d_j} - (t_{d_j m_\beta} + t_{m_\alpha m_\beta}) < t_{o_i} + t_{o_i m_\alpha}$ and $t_{d_i} - (t_{d_i m_\beta} + t_{m_\alpha m_\beta}) < t_{o_j} + t_{o_j m_\alpha}$ are not satisfied. Therefore, a pick-up meeting point $m \in M_j^{(i)}$ for a driver-rider pair (d_i, r_j) is optimal when the associated driving distance savings $\delta^{(i,j)}$ in (1) is maximized.

B. Distance Savings Embedding

In an ARN, the distance savings are treated as the discrete rider attributes. Given a driver-rider pair, we observe that each pair of pick-up and drop-off meeting points brings a distance savings. Searching $M_j^{(i)}$ generates multiple candidate meeting points and associated multiple disjoint sets of distance savings for each r_j . We consider the attributes to be categorical, and convert them to a series of binary features by way of one-hot encoding. Specifically, in consideration of each driver-rider pair (d_i, r_j) , we distribute the distance savings value into λ sections (e.g., 600 meters per section) and associate the maximum value with a low-dimensional attribute embedding vector $\mathcal{Z}_A^{(j)}$. For example, we express a value of 240 as the vector $\mathcal{Z}_A^{(j)} = \{1, 0, 0, \dots, 0\} \in \mathcal{R}^\lambda$, where the feature of value 1 implies the achieved maximum distance savings for rider r_j .

V. THE PROPOSED FRAMEWORK

Assume that a set of time schedules from N participants is given, where each schedule specifies a driver-rider pair with a best meeting point and associated overlapping interval. We then aim to learn a d -dimensional real-valued representation of each participant by preserving the structure of the ARN, where similar participants are placed close together in the embedding space. In this section, we illustrate the proposed framework: Joint Embedding for personalized Ranking in pooling (JERR).

A. User Embedding

To be able to calculate the similarities of schedules between drivers and riders that need to be ranked, every participant should have a unique embedding. We first propose an ARN, based on which we leverage meta-paths to obtain different-step

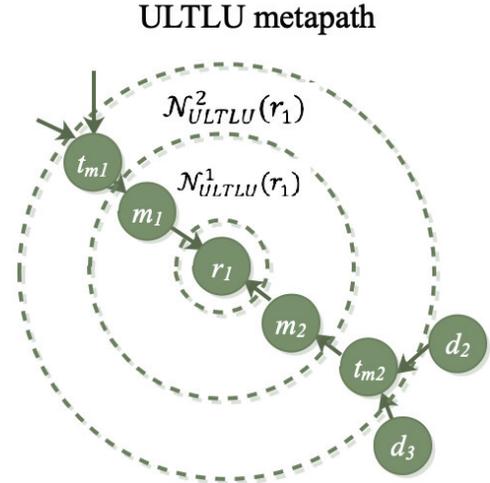


Fig. 5. An example of meta-path-guided neighbors generation. Nodes in this example are from Fig. 3.

neighbors of a node. The embeddings of drivers and riders are the aggregation of their neighbors under meta-paths.

We show an example in Fig. 5 to illuminate the meta-paths of an ARN. We have ULU and ULTLU, and aggregate the meta-path-guided neighbors to obtain the embedding $\mathcal{Z}_S^{(1)}$ for rider r_1 . According to the network schema in Fig. 3(b), we extract all N -step neighbors of r_1 as $\bigcup_{n \in N} \mathcal{N}_{ULU}^n(r_1) = \{m_1, m_2, d_1, d_2\}$, $\bigcup_{n \in N} \mathcal{N}_{ULTLU}^n(r_1) = \{m_1, m_2, t_{m1}, t_{m2}, d_1, d_2\}$, and construct a walk path to predict the probability distribution of r_1 's neighbors in terms of the local network structure. In this example, under the meta-path 'ULTLU', the walker at node m_1 transferred from r_1 can arrive at time-type nodes: t_{m1} . Following this process, we can obtain the ULTLU-guided embedding of r_1 , such as $\mathcal{Z}_S^{(1)}$.

We develop a way to search the different-step neighbors $\mathcal{N}_P^n(v)$ of node v along the meta-path, and aggregate the walk path, which is then fed to a skip-gram model. We define a set of neighbors surrounding node v along an l -length path as the

“context” of v :

$$N_n = \bigcup_{n=1}^c \mathcal{N}_P^n(v) (c \in [\lceil \frac{l-1}{2} \rceil, l-1]) \quad (2)$$

The proximity probability of each node v and $u \in N_n$ can be generated via the Softmax function as follows:

$$p(u|v; \theta) = \frac{e^{\mathcal{Z}_u \cdot \mathcal{Z}_v}}{\sum_{u \in N_n} e^{\mathcal{Z}_u \cdot \mathcal{Z}_v}} \quad (3)$$

where \mathcal{Z}_u and \mathcal{Z}_v denote the embeddings of u and v , respectively.

To learn the t -dimensional representation of a user-type node \mathcal{Z}_S (i.e., $\mathcal{Z}_S^{(i)}$ for driver d_i or $\mathcal{Z}_S^{(j)}$ for rider r_j) that captures the structural properties of the ARN, we consider the problem as a maximum likelihood problem. The probability that u is related to all nodes $v \in V_P$ should be maximized, i.e.,

$$\sum_{v \in V_P} \sum_{u \in N_n} p(u|v; \theta) \quad (4)$$

where V_P is a set of nodes on a meta-path \mathcal{P} starting from a user-type node. For illustration, consider the ARN's structure in Fig. 3(a). The neighborhood of a node r_1 can be structurally close to another location (e.g., m_1, m_2), time (e.g., $t_{m_1} & t_{m_2}$), and drivers ($d_1 & d_2$). From (3) and (4), we observe that the proposed framework models the spatiotemporal contexts of driver–rider pairs, where pairs with a similar context (i.e., arriving at the same meeting point within a time interval) have similar representations.

We leverage the negative sampling approach proposed in [56] to optimize our model:

$$\arg \max_{\theta} \log \sigma(\mathcal{Z}_u \cdot \mathcal{Z}_v) + \sum_{u' \in N_e(v)} \log \sigma(-\mathcal{Z}_{u'} \cdot \mathcal{Z}_v) \quad (5)$$

where $\sigma(x)$ is a sigmoid function, and $N_e(v)$ is the set of negative samples that are randomly sampled from the whole node set in the ARN in accordance with $p(v) \sim d_v^{3/4}$, where d_v is the in-degree of v . The stochastic gradient descent can facilitate the optimization.

B. Enhanced User Embedding With Attributes

To blend the advantage of both structure and attribute modeling, we concatenate their embeddings using early fusion, and jointly optimize all parameters. Since rider preferences can complement the learning of schedule proximity, it allows a real-time personalization in ride-matching. To address the challenge of attribute fusion in the ARN, we propose a user-level attention network to immediately learn the importance of attributes on rider ranking. In what follows, the design of the JERR is elaborated on layer by layer.

1) *Input Layer*: Taking a group of rider embeddings as input, the learned weights can be expressed as follows:

$$(\omega_{\phi_1}, \omega_{\phi_2}) = att(\mathcal{Z}_S, \mathcal{Z}_A) \quad (6)$$

where *att* denotes the encoder–decoder neural network, which shows that the user-level attention can learn both the structure and attribute information of each rider.

Due to the different dimensions of the learned embeddings, we first transform the set of embeddings through a nonlinear transformation. Let the adjacency matrices of $\mathbf{P} = (\mathcal{Z}_S)_t$ and $\mathbf{A} = (\mathcal{Z}_A)_\lambda$ be $\mathbf{W}_S \in \mathcal{R}^{t \times k}$ and $\mathbf{W}_A \in \mathcal{R}^{\lambda \times k}$. We have $\mathbf{F} = \tanh(\mathbf{W}_S \cdot \mathbf{P} + b_s)$, $\mathbf{G} = \tanh(\mathbf{W}_A \cdot \mathbf{A} + b_a)$ where $\mathbf{F}, \mathbf{G} \in \mathcal{R}^{|R| \times k}$, b_s and b_a form a bias vector, and $|R|$ is the number of riders.

2) *Attention Layer*: This layer provides the decoder with information from every pair's hidden representations, which correspond to a row in matrices \mathbf{F} and \mathbf{G} , denoted by $\mathbf{q}_{\phi_1}, \mathbf{q}_{\phi_2}$. We then obtain the score of these representations via the dot product between the hidden states. To learn the importance of each representation, we put the scores into a *softmax* function so that the weight, denoted as ω_{ϕ_i} , can be normalized across all representations:

$$\omega_{\phi_i} = \frac{\exp(\mathbf{q}_{\phi_1}^T \mathbf{q}_{\phi_i})}{\sum_{i=1}^2 \exp(\mathbf{q}_{\phi_1}^T \mathbf{q}_{\phi_i})} \quad (7)$$

If $\mathbf{q}_{\phi_1}^T$ and \mathbf{q}_{ϕ_i} have a large dot product, the structure is an informative representation, and the weight of the structure embedding will be large.

3) *Output Layer*: Finally, using the learned weights as coefficients, we are able to merge the hidden representations by multiplying each \mathbf{q}_{ϕ_i} by its weight. We obtain the final embedding X^j for rider r_j as follows:

$$\mathcal{Z}^{(j)} = \sum_{i=1}^2 \omega_{\phi_i}^{(j)} \cdot \mathbf{q}_{\phi_i}^{(j)}, \quad (8)$$

To better understand the user-level aggregation, we provide a short description in Fig. 6. The final embedding is combined with structure- and attribute-specific embeddings. We can then employ the final embedding to rank riders by their preferences.

4) *Optimization*: We specify the link probability in the ARN between the driver and riders as the objective function. In this way, the JERR framework is jointly trained to minimize the deviation over all user-type nodes regarding all parameters $\Theta = \{\mathbf{W}_S, \mathbf{W}_A, b_s, b_a\}$:

$$\arg \min_{\Theta} \sum_{d_i \in D, r_j \in R} \cos(\mathcal{Z}^{(i)} - \mathcal{Z}^{(j)}) \quad (9)$$

We employ the widely used mini-batch Adaptive Moment Estimation (Adam) optimizer [57] to adapt the learning rate for each parameter [58]. In particular, we initialize the parameters in Θ to optimize the JERR. Then, we sample a batch size of N embeddings $(\mathcal{Z}_S, \mathcal{Z}_A)$ of structure and attributes at random, retrieve their representations after $L(L = \frac{|R|}{N})$ steps of propagation, and then update the parameters by using the gradients of the loss (i.e. (9)). After optimization, we obtain the fusion representation $\mathcal{Z}^{(j)}$ for each rider r_j , which leads to better performance.

We sum up our algorithm in Algorithm 1. Since the JERR is an encoder–decoder-based solution, and assuming that we have k embedding dimensions and a batch size of N , each iteration of JERR takes $O(|R| \cdot k^2)$ operations. The global time complexity of JERR is $O(n_s \cdot |R| \cdot k^2)$ when we run n_s iterations of training. The memory complexity is $O(n_s \cdot k \cdot |R|)$. Since we treat k as a small constant, the complexity of each

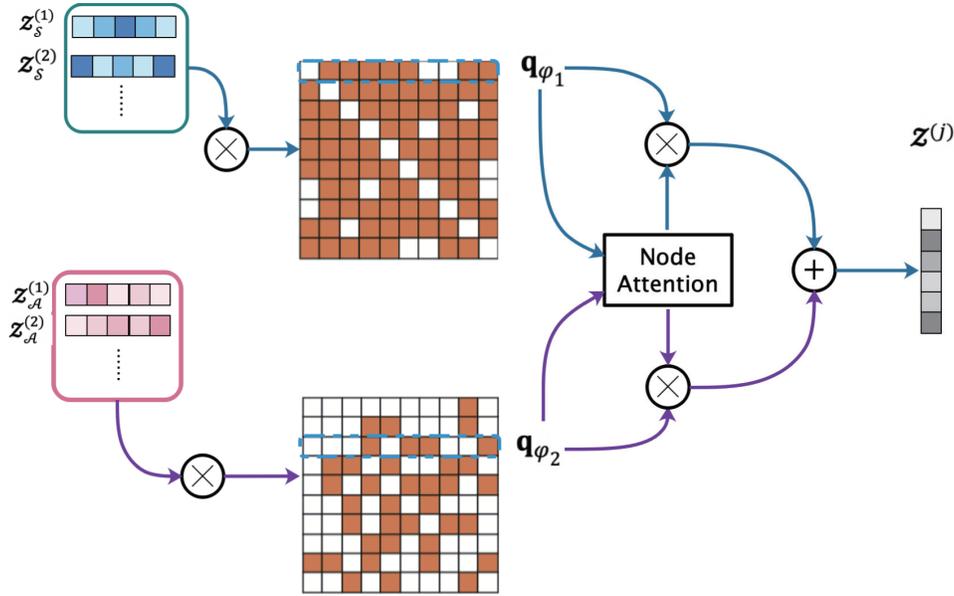


Fig. 6. Explanation of user-level aggregation. In the input layer, two types of embeddings of riders are projected onto the unified matrix space. In the attention layer, joint learning of the weights of each type and fusion of the node embeddings takes place.

iteration increases linearly with the number of nodes, i.e. $|R|$, in the ARN, which implies the efficiency and scalability of JERR.

C. Candidates Search for Ranking

The proposed method helps incorporate personal preferences into the driver-to-rider similarity computation. Specifically, for each pair (d_i, r_j) , we exploit our JERR to learn the representations $\mathcal{Z}^{(i)}$ and $\mathcal{Z}^{(j)}$ of the pair. Next, we train a neural network with $\mathcal{Z}_S^{(j)}$ and $\mathcal{Z}_A^{(i)}$ as input, in order to preserve the preferences on trip cost. Eventually, we rank riders via their average cosine similarities to generate a candidate list:

$$\cos(d_i, r_j) = \frac{\mathcal{Z}^{(i)} \cdot \mathcal{Z}^{(j)}}{\|\mathcal{Z}^{(i)}\| \cdot \|\mathcal{Z}^{(j)}\|} \quad (10)$$

VI. SIMULATION

We evaluate the proposed JERR using a realistic pooling simulator. The simulation is designed to enable a fair comparison between various matching policies.

A. Base Case Settings

We generated a dataset of daily driver’s trips and rider’s requests within the Chengdu Er’Huan region via Grid2Demand.¹ Most of the demands were related to the zones, and arise around points of interest(POI). One hundred random streams of driver’s trips with an origin–destination pair were deployed at the beginning of the simulation. For each rider’s announcement, we generated the origin and destination points randomly around the center of FuLi plaza.

¹Grid2Demand: a tool for generating zone-to-zone travel demand based on grid zones, <https://github.com/Anjun93/grid2demand>

TABLE II
TYPICAL FEATURES OF THE BASE CASE INSTANCES

Trip pattern	Setting
Number of drivers, riders	100, 40
<i>dflex</i>	0.005
Waiting time	300 s
Average driving speed	30 km/h
Average walking speed	1.22 m/s
Max. number of meeting points for a driver	30

The departure times we in intervals of 60 seconds. Afterward, we measured the expected arrival time by adding trip duration flexibility and direct travel time to the departure time; this flexibility was considered to be 300 seconds for all riders. Duration flexibility refers to the maximum extra trip time that riders accept, which indicates the “waiting time” from departing.

Subsequently, the state of the riders is determined as either waiting for rides at the origin under a certain matching policy or starting to walk toward a meeting point. We adopt a walking speed of 1.22 m/s [59]. The distance flexibility *dflex* defines the possibility for riders to walk to a meeting point. *dflex* can be fine-tuned when probing for a possible meeting point in the R-tree. With the help of a particular matching policy, the simulator periodically detects the position of vehicles and riders. After each ride match, drivers who are assigned orders will pick up riders and transport them to their drop-off locations or destinations. We consider that after a driver finishes an order, he would be available to be reallocated to a new trip request starting from another meeting point. Table II summarizes the features of the base case instances, and the workflow of the simulator is depicted in Fig. 7.

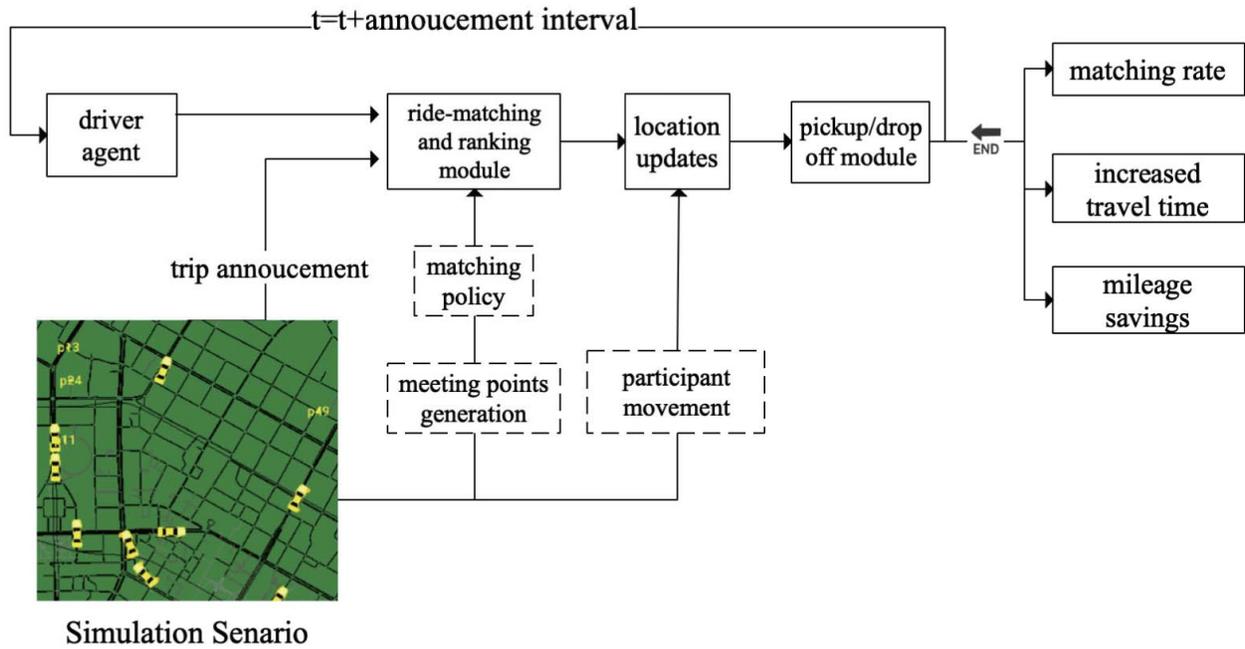


Fig. 7. Workflow of the rider ranking simulator.

We compare solutions using the following metrics: (1) the matching rate for riders; (2) the distance savings, i.e., the fraction of the system-wide vehicle-miles if all participants drove alone; and (3) the trip time increase, i.e., the average relative increase in the trip duration, as a fraction of original travel time.

B. Impact of Pooling With Multiple Riders

Fig. 8 shows distance savings for different vehicle capacities. The ' n/POI ' denotes the number of trip requests submitted at each POI. To reduce the influences of POI on matching, we compare the savings under the same POI distribution. The results suggest that the primary benefit of carpooling is an increase in distance saving. When the number of drivers and riders in the system is roughly different, it is more desirable to have single driver-multiple riders matches for maximizing the utilization of vehicles (from 1 to 2.9 when pooling four riders, in Table III). We see that the number of vehicles served in matches with one rider is quite small in different vehicular distributions when 4 riders per vehicle are allowed at most. This is due to an increase in the number of single driver-multiple rider matching opportunities. That is, if it is possible to match four riders with the same driver, but it is also possible to match the four riders with different drivers, then the former option is preferred as it results in assigning 1 driver, while the latter results in 4 vehicles served.

C. Impact of Unbalanced Supply and Demand

We conduct some experiments with 150 participants, and using driver-to-rider ratios of 1:1, 2:1, and 1:2. The results can be found in Table IV. We focus on the matching rates in the unbalanced scenarios.

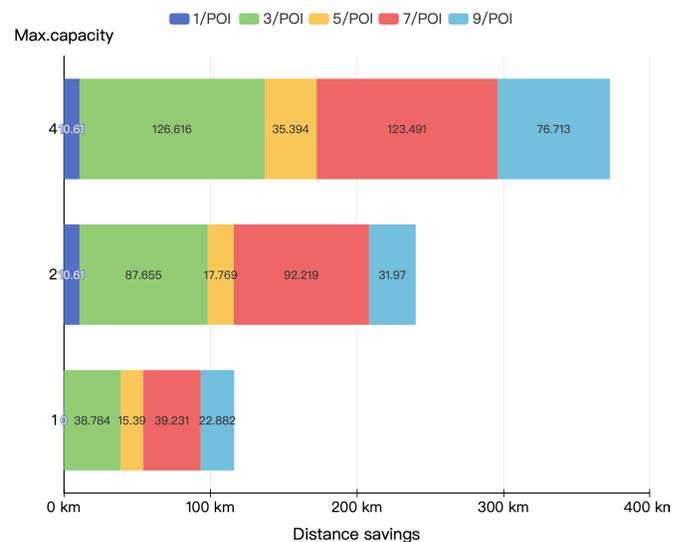


Fig. 8. Distance savings when single, double, and quadruple matches for different trip-request distributions.

Inspecting the differences from the base cases tells us that trip density plays a large role in matching. For the ratio 1:2, a high system density (100 trips) restricts the capability of the system to establish matches. However, since the riders have to wait much longer for success pickup, which consequently increases the trip time.

D. Impact of Duration Flexibility

We investigate the impact of duration flexibility on the matching rate with different numbers of trip announcements. We compare the system performance when the waiting time shorter, i.e., $t_m = 300$ s, and when the waiting time is

Algorithm 1 Embedding Generation

Input: meta-path scheme \mathcal{P} , rider embeddings $\mathcal{Z}^{(j)}$; driver embeddings $\mathcal{Z}^{(i)}$

```

1: build R-tree with drivers' trajectories;
2: for each rider do
3:   find meeting points based on R-tree;
4:   filter meeting points with time window;
5:   for each driver do
6:     if driver and rider have feasible meeting points then
7:       calculate distance savings according to (1);
8:       encode maximum distance savings using one-hot
       encoding as  $\mathcal{Z}_A$ ;
9:     end if
10:  end for
11: end for
12:  $\mathcal{Z}^{(i)}, \mathcal{Z}_S^{(j)} = UserEmbedding(\mathcal{P})$  using (2)-(5);
13: function FUSION( $\mathcal{Z}_S^{(j)}, \mathcal{Z}_A^{(j)}$ )
14:    $\mathbf{F} = \tanh(\mathbf{W}_S \cdot \mathbf{P} + b_s)$ ;
15:    $\mathbf{G} = \tanh(\mathbf{W}_A \cdot \mathbf{A} + b_a)$ ;
16:   for each row  $\mathbf{q}_{\phi_1}^{(j)}, \mathbf{q}_{\phi_2}^{(j)}$  in  $\mathbf{F}, \mathbf{G}$  do
17:      $\omega_{\phi_1}^{(j)}, \omega_{\phi_2}^{(j)} = softmax(\mathbf{q}_{\phi_1}^{(j)}, \mathbf{q}_{\phi_2}^{(j)})$ ;
18:     generate embedding  $\mathcal{Z}^{(j)}$  according to (7) and (8);
19:   end for
20:   optimize parameters by Adam with (9);
21: end function
22: return embedding results

```

TABLE III

EFFECT OF CARPOOLING ON VEHICLE SERVED, WHERE n/POI MEANS TO DEPLOY n VEHICLES AT EACH POINTS

Max.capacity	1	2	4			
			1	2	3	4
1vPOI	4	7	1	1	2	2
2vPOI	11	13	0	1	3	7
3vPOI	9	10	1	1	0	7
4vPOI	10	12	0	1	3	6
5vPOI	4	5	0	0	0	5
No.riders matched	38	81	119			
No.vehicles served	38	47	41			
No.carpoolers per vehicle	1	1.7	2.9			

longer, i.e., $t_m = 900$ s. Fig. 9 illustrates that increasing the flexibility from 300 to 900 s contributes to a 9.9% growth in average matching rate for 45 trips. Moreover, a low flexibility of 300 s heavily reduces the matching rate. Even at the higher density (45 trips), the average matching rate is only 51.1%. We also can conclude in Figure 8 that the distributions of riders are critical in matching. When most of trip requests (e.g., 5 trips) arise around the POIs, that will make more matches. Instead, the trip density increases to 45, part of which is far away from POIs, but the matching rate reduces.

VII. EVALUATION

In this section, we give the experimental details to demonstrate the effectiveness of the proposed framework. We first

TABLE IV
EFFECTIVENESS OF THE DRIVER-TO-RIDER RATIOS

Driver-rider ratio	1:1	2:1	1:2
System			
Matching rate (%)	56	66	49
Distance savings (%)	91.55	92.84	91.81
Rider			
Trip time increase (%)	62.30	72.44	58.47
Average trip time (s)	787	836	736
Average trip distance (km)	11.94	6.90	5.43
Driver			
Trip time increase (%)	75.95	82.16	76.74
Average trip time (s)	572	626	531
Average trip distance (km)	6.51	6.44	6.03

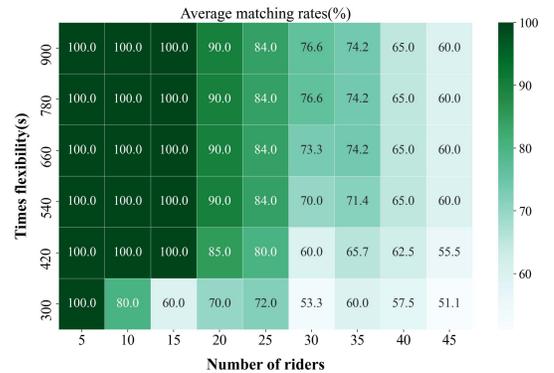


Fig. 9. Matching rates for different duration flexibilities and number of trip announcements.

cover the details of training user embeddings, and their offline evaluation. We then compare performance of link prediction with other advanced embedding methods. Finally, we make comparisons of our ranking model to validate the capabilities of leveraging user embeddings to implement features for personalization in rider ranking.

A. Datasets and Baselines

We use a real travel dataset, collected in the city of ChengDu and generated by DiDi chuxing, which is one of the largest pooling service companies. It contains 1,070,404,834 highly sampled trajectories (about every 2–4 s) of trip requests received by nearly 1,142,717 private drivers within one month. We first extract 50,000 trajectories of drivers, the feasible meeting points, and features associated with distance savings. Then, we construct the ARN, which consists of three types of nodes: users, locations (i.e., meeting points), and time periods (i.e., waiting times). Since users can share rides to locations, we collect time-limited pickup behavior to construct the relations between users and locations. Some statistics of the ARN are shown in Table V.

To decide on the ideal method for rider ranking, we test how good trained user embeddings are for producing the candidate rider listings based on their time schedules and attributes. Let

TABLE V
STATISTICS OF THE ARN

Type	Number
No. users	23,896
No. locations	4,723
No. periods	9,468
No. edges between “user” and “locations”	574,337

us assume we are given some driver and rider candidates that need to be ranked. By measuring the cosine-based similarities between embeddings of drivers and candidate riders, we can list the candidates and identify the position of the selected rider in the list.

-*metapath2vec* [60]. *metapath2vec* employs the heterogeneous Skip-Gram model on the created meta-path-based node sequences. We followed the meta-path settings, including window size, walk length, walks per node, and the number of negative samples.

-*MNE* [61]. MNE adopted one popular embedding and several supplementary embeddings for each edge type, which are jointly learned by a unified network embedding model. The supplementary embedding is initialized to zero with the size of 10.

-*ASNE* [58]. ASNE gains node representations by connecting deep neural network models with social information to develop the network structure and heterogeneous node attributes together. ASNE uses an early fusion model to preserve the nodes’ structural and attribute proximity in social networks.

Note that *metapath2vec* and MNE are devised to adopt only the structural information. We then extend them to incorporate attributes for a fair comparison with JERR. Following [58], we name these variants as *metapath2vec** and *MNE**.

B. Experimental Setups

We choose the link prediction task for user embedding evaluation. The reason for this is that a better embedding should produce better user representations from both the network structure and user attributes, which will lead to a superior prediction of the occurrence of links between driver–rider pairs. Our implementation of JERR is based on TensorFlow. To ensure a fair test, for all the embedding-based methods, we set the number of structural embedding dimensions to be 128. For MNE, we use the implementation released by the original authors. Since ASNE has two kinds of embeddings (the structural proximity and attribute proximity), we set the embedding dimension for the network structure and attributes to be 128 and 20, respectively, and then concatenate them together. For all methods, in the training stage, we initialize the model parameters with a Gaussian distribution (with a mean of 0.0 and standard deviation of 0.01), and optimize the model with mini-batch Adam. For our method, the mini-batch size is 64, the initial learning rate is 0.005, and the regularization parameter is 0.001. The hyper-parameter controlling the walking process is set to be the same as that of *metapath2vec*.

We also use a hyperbolic tangent function (\tanh) for all experiments.

For all the ride-matching methods, we set the driving speed to be 30 km/h, and walk speed to/from a meeting point is set to 1.22 m/s [59]. The maximum allowable walking duration and distance are 900 s and 3 km, respectively. The service time related to the pickup and drop off of a rider is negligible. For searching meeting points by the R-tree, we define the maximum number of entries stored $N = 32$ in each node. All our experiments are conducted on an Intel Core i7-6700 CPU @3.4 GHZ with 16 GB RAM.

C. Performance Comparison

1) *Characteristics of JERR: Link Prediction.* We randomly offer 20% of the links as the testing set, and train JREE on remaining links. Following [54], we randomly select the same number of positive and negative edges for each edge type, and list both positive and negative instances in accordance with the prediction results. We employ the area under the ROC curve (AUC) as the metric to judge the ranking quality. Fig. 10 lists the experimental results obtained on the DiDi dataset. JERR reaches the best performance among all the methods, with at least a 7.16% performance increase in AUC compared with the best results of other methods. Meanwhile, JERR exhibits more stability when we use fewer links for training. Especially, compared with the pure structure-based methods, *metapath2vec* and MNE, JREE performs better with only half the number of links. Besides, *metapath2vec** (MNE*) is slightly improves over *metapath2vec* (MNE). This illustrates the efficiency of attributes for finding out missing links, along with the advantage of JREE in using attributes for achieving better matching. We can see that JERR consistently outperforms ASNE, which also combines attributes in the network embedding. Even though ASNE compresses the node representation by allowing hidden layers to learn the interactions between structure and attribute, it fails to obtain sufficient information for weighting the proximity of the structure and attributes. In this way, nodes with a lower structural proximity as well as a large attribute proximity may be mistakenly ranked.

Network Sparsity Analysis. We set up 3 groups of experiments, where 10%, 30%, and 50% of the user–user edges are removed, respectively. The connections of any two users (i.e., a driver–rider pair) under the 80% train ratio are identified firstly, and we put all the remaining user–user edges (of which the proximities are up to 0.7) into the training set. For each edge in the training set, we randomly sample a rider that is not linked to a driver, and regard this pair as the negative instance in the training set. The testing set is formed using the same way. We make the following observations from Table VI:

- JERR gains the best performance. For instance, as 10%–50% edges are moved, JERR improves the performance by 0.4%–6.5% compared to the baseline. The reason for this is that JERR captures the relations in personal preferences and the shared schedule.
- JERR and ASNE show better performance than MNV and *Metapath2Vec*, which further shows the significance of attribute information.

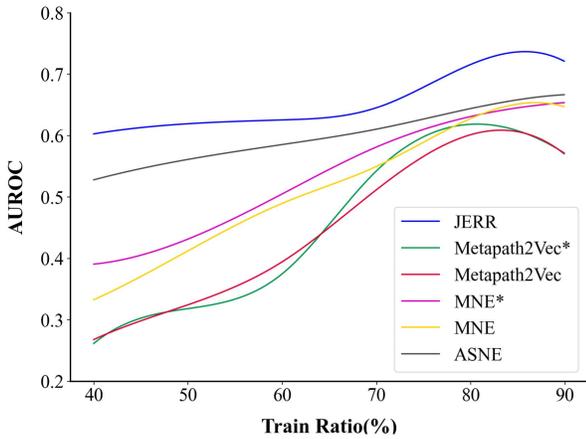


Fig. 10. Comparison of different methods for link prediction task on DiDi dataset.

TABLE VI
PREDICTION PERFORMANCE WITH EDGES REMOVED

	% Removed edges	10%	30%	50%
F1(%)	JERR	77.03	75.54	70.42
	ASNE	74.41	70.81	65.25
	MNE	76.87	72.3	61.68
	Metapath2Vec	73.45	69.09	64.45
	JERR	70.22	67.90	64.64
AUC	ASNE	69.79	66.43	63.22
	MNE	68.45	64.38	62.09
	Metapath2Vec	67.47	64.05	60.93

- The performance of Metapath2Vec is worst. Metapath2Vec can only work for structure unweighted networks, and cannot take advantage of additional information such as edge weights.

Parameter Sensitivity. We now test the effect of the training ratio q on performance when other parameters are fixed. The results are shown in Fig. 11(a). We find that the performance of JERR increases with the training ratio, and achieves its maximum when q is 90%. Fig. 11(b) illustrates the accuracy when changing the base embedding dimension d from the default $d = 128$. With increasing embedding dimension, the performance first improves, and then slowly starts to drop. This is because a larger dimension may introduce sparsity in riders and drivers' representations, making it difficult to encode their relations accurately. To check the effect of negative sampling, we study the accuracy of JERR with various lengths of contextual window c . The result is shown in Fig. 11(c). Fixing $d = 128$, the value increases when c increases to 3, and tends to be stable when $c > 5$, thus indicating that more context nodes related to the given meta-paths (i.e. ULTLU) are sampled, and that the JERR has reached its best solution. We also discover that negative sampling can drive the training process to be more stable.

2) *Numerical Results for Ranking Riders:* To validate the effectiveness of rider ranking, we use some popular models. Note that some matching models are not included, such as [4],

[12], [22], because they are not suitable for our problem. For each driver in the test set, we treat all the riders independent of drivers as being negative. Then, each method outputs the rider's ranking.

We give a brief description below of the policies used to compute the final ranking. Variations arise from the way the optimization operates under different assumptions.

-*Baseline.* A Hungarian method, (also known as the KM algorithm) that solves the weighted minimum matching problem in a complete bipartite graph, was used for a multiple rider/single driver setting. There are some similar matching methods [26] designed for graphs with known edge weights, but they focus more on improving the searching efficiency. We class these algorithms together.

-*Stiglic et al.'s Work [13].* This considers a pooling setting where one pick up and drop off are supported per shared ride. A multi-rider match is feasible when it can satisfy the time constraints, i.e., more than one riders can traverse to the meeting point simultaneously. In all experiments, we match one driver and multiple riders by solving a weighted bipartite matching problem.

-*Shareability Network-Based Methods [43], [62].* These approaches match multiple shareable trips via a network, and models the trips as nodes. By assessing the link possibility of these nodes in terms of extra travel time between any two trips, a vehicles can be assigned to serve several trips sequentially or simultaneously.

Efficiency of Ride-Matching. In this work, we use the number of participants matched and the mileage savings as measures of matching efficiency. In Fig. 12, we present the result obtained in the base case setting with embedding to that obtained in the case without embedding; we also increase the number of trips from 800 to 2,000. We can see that the introduction of embedding produces a substantial increase in matching rates. The matching rate using JERR increases by 15.4% (from 56.9% to 72.5%), and is slightly larger than that of the KM algorithm (the weights of the edges in KM were set to be 1) because of matches consider more road information. Besides, the KM algorithm is widely used for optimal matching by seeking a one-to-one relationship, but fails to match a single driver and multiple riders, and thus degrades efficiency. In comparison with Stiglic's work, we notice that JERR maintains an improvement in mileage savings, with an averaged improvement from 0.8% to 50.8%. We suspect this is due to preserving the attribute information on distance savings. Stiglic *et al.* determined the meeting points from the exact locations, without considering the latent spatiotemporal proximities. Our results also reveal that benefits can be reached when the meeting points are within the limits of a rider's origin and destination. In Fig. 13, we see that for 2,000 participants, when the batch size is 64 in Stiglic's work and the KM algorithm, the running time is 112.39% and 32.61% longer, respectively, than that of JERR. Therefore, we believe that JERR is efficient enough to handle matching tasks on large car-hailing systems like DiDi and Uber.

We also compare our method with the online model from Kondor *et al.*'s work [62], where the trips connected via the shareability network are treated as successful matches. For

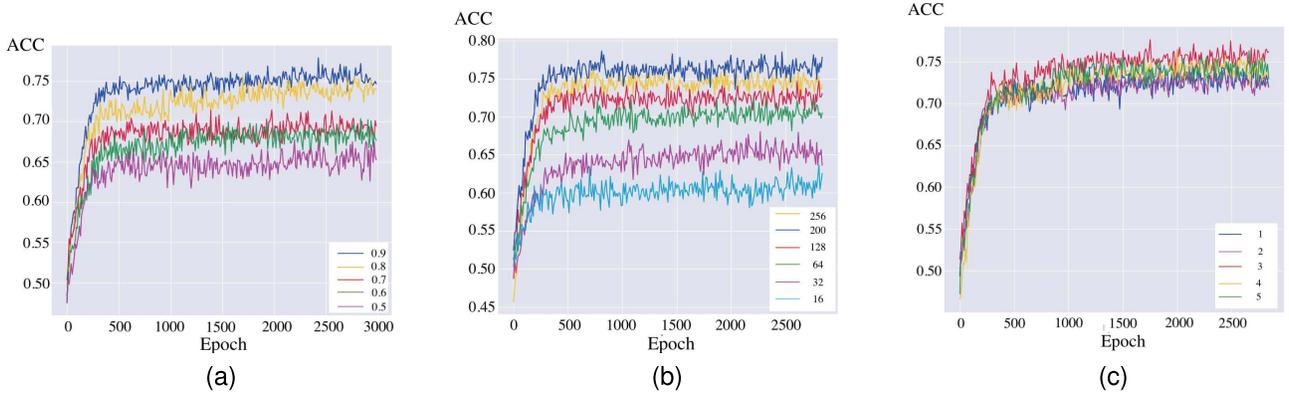


Fig. 11. Parameter sensitivity of JERR w.r.t. training ratio, dimension, and length of windows.

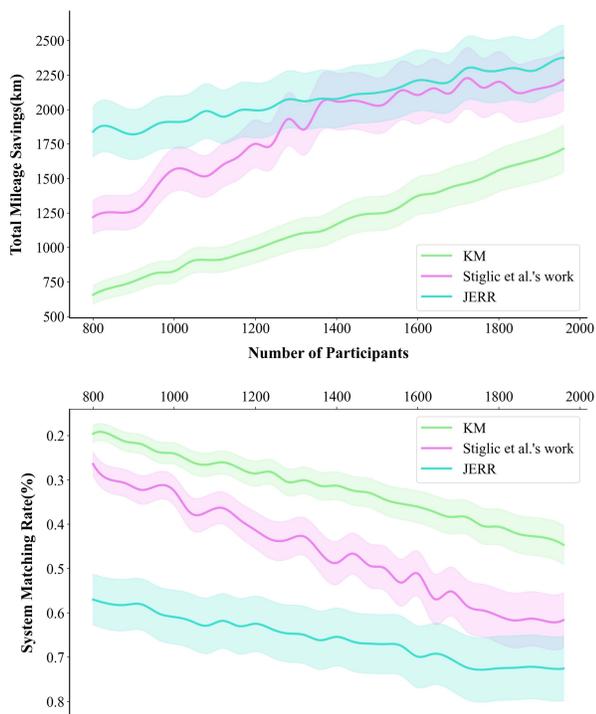


Fig. 12. Effects of trip densities.

a fair comparison with JERR, which additionally exploits attributes, we extend the shareability network [43] to incorporate attributes. Specifically, the detour tolerances for connecting trips are randomly generated as individual attributes of each trip. We dub the variant Shareability+attr. We calculate the average waiting time of riders unserved in JERR, or of trip requests unconnected, as done in Kondor *et al.*'s work. We see in Fig. 14 that these results offer significant improvements over the compared approaches via the shareability network. On the DiDi dataset, where the trip requests are quite dispersed, vehicles have to detour to probe for and connect to the shareable trips' OD, thus leading increased travel time. However, the flexible matching with available meeting points in JERR can minimize potential risks to travel time,

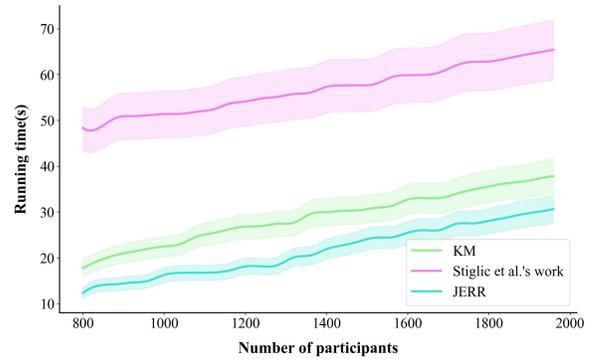


Fig. 13. Running time with different number of participants.

allowing drivers to pick up the riders en route. Thus, we can conclude that proximity modeling via the ARN is essential for making significant improvements in ride-matching and vehicle utilization.

Efficiency of Ranking. To validate the effect of proximity modeling on matching, we evaluate each method's capability of finding and ranking riders. We apply two common ranking metrics: mean reciprocal rank (MRR), and normalized discounted cumulative gain (nDCG). Matches are considered as positive labels. Since the shareability network-based approaches search for a trip satisfying certain conditions each time, without a candidate list, these ranking metrics do not apply to them. Let S be the set of drivers for the evaluation, and let $r_{s,i}$ be equal to 1 when the rider at position i is recommended to be matched, and 0 otherwise. For each metric, the larger the better. The metrics are formulated below:

$$MRR = \frac{1}{|S|} \sum_{s \in S} \frac{1}{\min_{r_{s,i}=1} i}$$

$$nDCG = \frac{1}{|S|} \sum_{s \in S} \frac{\sum_i \frac{r_{s,i}}{\log_2(i+1)}}{j \sum_i \frac{r_{s,j}}{\log_2(j+1)}} \quad (11)$$

Table VII lists all metrics used in the experiments. We divide the test dataset into ten sub-datasets, and calculate the averages for each metric per sub-dataset. JERR achieves the best results for all metrics. Meanwhile, Metapath2Vec is worse than MNE and ASNE. This suggests that the relationship between drivers and riders cannot be expressed with simple network

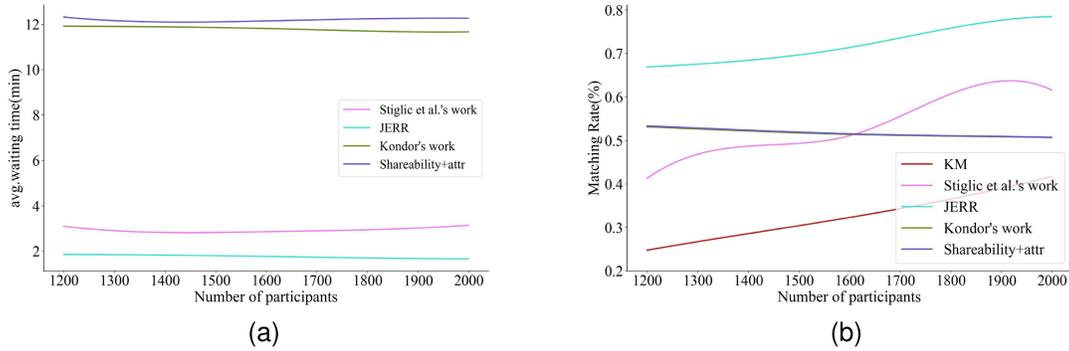


Fig. 14. Performance of matching using different network models.

TABLE VII
RESULTS FROM RANKING EXPERIMENTS

	MRR	nDCG
JERR	0.483	0.437
Metapath2Vec	0.381	0.379
MNE	0.432	0.388
ASNE	0.407	0.409
KM	0.188	0.263
Stiglic <i>et al.</i> 's	0.266	0.310

relationships. JERR is significantly better than the three other models. We conclude that this is because it can express more complex relations of drivers and riders by combining the attributes of riders. Because Stiglic *et al.*'s work exhibits an improvement over the KM algorithm, implying that the meeting points should work better. However, it calculates the distance savings without leveraging the network structure, which makes the ranking only rely on the distance savings. KM performs worst of all the methods, since it allows only one driver to be assigned to a rider, and thus it is unable to express more relations between each driver–rider pair and make more matchings.

VIII. DISCUSSION AND CONCLUSION

A better understanding of individual behaviors and preferences is fundamental when developing a pooling system. Unfortunately, ranking that is done via preference closeness rather than exact trip locations may be problematic, partly due to the challenge of representing personalized preferences when there is a high degree of heterogeneity across individuals in travel preference [63]. This includes the pick-up and drop-off times, travel duration, financial benefits, and person-specific information, such as gender, age, professional profile, feedback, and reliability scores [64]. In contrast to traditional pooling matching, such as Zhang and Zhao's work [30], we have modeled pooling preferences as node attributes in a network. To learn descriptive representations for pooling participants, it is necessary to clarify the relationships implied in the network structure. To this end, we proposed a generic framework for embedding pooling networks and preserve the closeness between nodes with attributes. We extended a deep neural network via an attention mechanism to model the latent interre-

lations between network structures and attributes. Remarkable performances on link prediction and rider ranking also shows the effectiveness of incorporating attributes in the model.

While a network for ranking riders provides abundant sources of information, e.g., individual schedules and demographics, we will examine the following in future work. First, JERR will be strengthened by considering the dynamics of participant preferences. It might need to be noted here that the preference we used is independent of travels, which drives our design of modeling the proximity via attribute embedding. However, we have observed that preference for travel attributes in pooling may vary from one day to the next. As an illustration, a rider's waiting time may depend on the time of a day. Although we consider the changing preference where the distance savings is different for different riders and meeting points, continuous change is not supported. In such conditions, focusing on offline evaluation of attribute proximity might induce unacceptable results. Some recent studies contribute to predictive modeling via time-series analysis [65]. These models can be seen as special cases of recurrent neural networks that do not treat time itself as a feature, and typically assume that inputs are synchronous. In practice, RNNs often fail to effectively make use of time as an attribute. Thus, to feed the time-dependent preferences as an input dimension, and concatenating the representations of time and attributes to form multiple types of vector embeddings can be a good way forward. A network can then learn which embeddings it prefers by predicting a weight for each embedding type, depending on the context.

Second, we are interested in considering trust in pooling, and investigating how to model the evolution of pooling networks, which contain rich multi-modal social data. For instance, the trust in pooling links develops over time; strangers who share the same routes a few times in their daily commute can gain trust and be connected in a network [51]. Based on several snapshots of networks in everyday life, and reflecting the trust level at a given moment, we can predict the positive or negative input of participant behavior on the trust level. Then, the dynamic co-evolution of networks can be modeled by adding dynamic information into the attributes of nodes and edges. Lastly, we will deal with boosting the efficiency of our JERR by developing better indexing structures to make it more suitable for use in a large-scale commercial environment.

REFERENCES

- [1] W. J. Mitchell, C. E. Borroni-Bird, and L. D. Burns, *Reinventing the Automobile: Personal Urban Mobility for the 21st Century*. Cambridge, MA, USA: MIT Press, 2010.
- [2] V. Handke and H. Jonuschat, "Mobility," in *Flexible Ridesharing: New Opportunities and Service Concepts for Sustainable Mobility*. Berlin, Germany: Springer, 2012, pp. 5–11.
- [3] Shared and Digital Mobility Committee. (Sep. 24, 2018). *Taxonomy and Definitions for Terms Related to Shared Mobility and Enabling Technologies*. [Online]. Available: https://www.sae.org/standards/content/j3163_201809/
- [4] L. Tang, Z. Duan, Y. Zhu, J. Ma, and Z. Liu, "Recommendation for ridesharing groups through destination prediction on trajectory data," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1320–1333, Feb. 2021, doi: [10.1109/TITS.2019.2961170](https://doi.org/10.1109/TITS.2019.2961170).
- [5] D. Chuxing. (2017). *Corporate Citizenship Report*. (Apr. 18, 2018). [Online]. Available: <https://www.didiglobal.com/news/NewsDetail?id=384&type=news>
- [6] M. Furuhashi, M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig, "Ridesharing: The state-of-the-art and future directions," *Transp. Res. B, Methodol.*, vol. 57, pp. 28–46, Nov. 2013, doi: [10.1016/j.trb.2013.08.012](https://doi.org/10.1016/j.trb.2013.08.012).
- [7] L. Tang, D. Cai, Z. Duan, J. Ma, M. Han, and H. Wang, "Discovering travel community for POI recommendation on location-based social networks," *Complexity*, vol. 2019, pp. 1–8, Feb. 2019, doi: [10.1155/2019/8503962](https://doi.org/10.1155/2019/8503962).
- [8] J. D. Burger, J. Henderson, G. Kim, and G. Zarrella, "Discriminating gender on Twitter," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, Edinburgh, U.K., 2011, pp. 1301–1309.
- [9] M. Pennacchiotti and A.-M. Popescu, "Democrats, republicans and starbucks aficionados: User classification in Twitter," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Diego, CA, USA, 2011, pp. 430–438.
- [10] X. Qian, X. Zhan, and S. V. Ukkusuri, "Characterizing urban dynamics using large scale taxicab data," in *Engineering and Applied Sciences Optimization (Computational Methods in Applied Sciences)*, vol. 38, N. Lagaros and M. Papadrakakis, Eds. Cham, Switzerland: Springer, 2015, pp. 17–32.
- [11] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: A queueing-theoretical perspective," *Int. J. Robot. Res.*, vol. 35, nos. 1–3, pp. 186–203, Jul. 2015, doi: [10.1177/0278364915581863](https://doi.org/10.1177/0278364915581863).
- [12] L. Zhang *et al.*, "A taxi order dispatch model based on combinatorial optimization," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Halifax, NS, Canada, Aug. 2017, pp. 2151–2159.
- [13] M. Stiglic, N. Agatz, M. Savelsbergh, and M. Gradisar, "The benefits of meeting points in ride-sharing systems," *Transp. Res. B, Methodol.*, vol. 82, pp. 36–53, Dec. 2015, doi: [10.1016/j.trb.2015.07.025](https://doi.org/10.1016/j.trb.2015.07.025).
- [14] X. Wang, N. Agatz, and A. Erera, "Stable matching for dynamic ride-sharing systems," *Transp. Sci.*, vol. 52, no. 4, pp. 850–867, Aug. 2018, doi: [10.1287/trsc.2017.0768](https://doi.org/10.1287/trsc.2017.0768).
- [15] X. Qian, W. Zhang, S. V. Ukkusuri, and C. Yang, "Optimal assignment and incentive design in the taxi group ride problem," *Transp. Res. B, Methodol.*, vol. 103, pp. 208–226, Sep. 2017, doi: [10.1016/j.trb.2017.03.001](https://doi.org/10.1016/j.trb.2017.03.001).
- [16] Z. Wang, B. Guo, Z. Yu, and X. Zhou, "Wi-Fi CSI-based behavior recognition: From signals and actions to activities," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 109–115, May 2018, doi: [10.1109/MCOM.2018.1700144](https://doi.org/10.1109/MCOM.2018.1700144).
- [17] L. Tang, Z. Duan, and Y. Zhao, "Toward using social media to support ridesharing services: Challenges and opportunities," *Transp. Planning Technol.*, vol. 42, no. 4, pp. 355–379, Apr. 2019, doi: [10.1080/03081060.2019.1600242](https://doi.org/10.1080/03081060.2019.1600242).
- [18] S. Ma, Y. Zheng, and O. Wolfson, "Real-time city-scale taxi ridesharing," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 7, pp. 1782–1795, Jul. 2015, doi: [10.1109/TKDE.2014.2334313](https://doi.org/10.1109/TKDE.2014.2334313).
- [19] X. Tang *et al.*, "A deep value-network based approach for multi-driver order dispatching," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Anchorage, AK, USA, Jul. 2019, pp. 1780–1790.
- [20] S. Shaheen and A. Cohen, "Shared ride services in North America: Definitions, impacts, and the future of pooling," *Transp. Res.*, vol. 39, no. 4, pp. 427–442, Jul. 2018, doi: [10.1080/01441647.2018.1497728](https://doi.org/10.1080/01441647.2018.1497728).
- [21] M. Rigby, S. Winter, and A. Kruger, "A continuous representation of Ad Hoc ridesharing potential," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2832–2842, Oct. 2016, doi: [10.1109/TITS.2016.2527052](https://doi.org/10.1109/TITS.2016.2527052).
- [22] M. Furuhashi *et al.*, "Online cost-sharing mechanism design for demand-responsive transport systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 692–707, Apr. 2015, doi: [10.1109/TITS.2014.2336212](https://doi.org/10.1109/TITS.2014.2336212).
- [23] M. Stiglic, N. Agatz, M. Savelsbergh, and M. Gradisar, "Making dynamic ride-sharing work: The impact of driver and rider flexibility," *Transp. Res. E, Logistics Transp. Rev.*, vol. 91, pp. 190–207, Jul. 2016, doi: [10.1016/j.trre.2016.04.010](https://doi.org/10.1016/j.trre.2016.04.010).
- [24] A. K. M. M. R. Khan, O. Correa, E. Tanin, L. Kulik, and K. Ramamohanarao, "Ride-sharing is about agreeing on a destination," in *Proc. 25th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Redondo Beach, CA, USA, Nov. 2017, pp. 1–10.
- [25] R. Duan and S. Pettie, "Linear-time approximation for maximum weight matching," *J. ACM*, vol. 61, no. 1, pp. 1–23, Jan. 2014, doi: [10.1145/2529989](https://doi.org/10.1145/2529989).
- [26] N. Ta, G. Li, T. Zhao, J. Feng, H. Ma, and Z. Gong, "An efficient ride-sharing framework for maximizing shared route," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 2, pp. 219–233, Feb. 2018, doi: [10.1109/TKDE.2017.2760880](https://doi.org/10.1109/TKDE.2017.2760880).
- [27] L. Zheng and L. Chen, "Maximizing acceptance in rejection-aware spatial crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 1943–1956, Sep. 2017, doi: [10.1109/TKDE.2017.2676771](https://doi.org/10.1109/TKDE.2017.2676771).
- [28] M. Berlingerio, B. Ghaddar, R. Guidotti, A. Pascale, and A. Sassi, "The GRAAL of carpooling: Green and social optimization from crowd-sourced data," *Transp. Res. C, Emerg. Technol.*, vol. 80, pp. 20–36, Jul. 2017, doi: [10.1016/j.trc.2017.02.025](https://doi.org/10.1016/j.trc.2017.02.025).
- [29] Y. Wang, S. Winter, and N. Ronald, "How much is trust: The cost and benefit of ridesharing with friends," *Comput., Environ. Urban Syst.*, vol. 65, pp. 103–112, Sep. 2017, doi: [10.1016/j.compenvurbysys.2017.06.002](https://doi.org/10.1016/j.compenvurbysys.2017.06.002).
- [30] H. Zhang and J. Zhao, "Mobility sharing as a preference matching problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2584–2592, Jul. 2019, doi: [10.1109/TITS.2018.2868366](https://doi.org/10.1109/TITS.2018.2868366).
- [31] Y. Xiong, Y. Zhu, and P. S. Yu, "Top-K similarity join in heterogeneous information networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1710–1723, Jun. 2015, doi: [10.1109/TKDE.2014.2373385](https://doi.org/10.1109/TKDE.2014.2373385).
- [32] Y. Jiang, G. Li, J. Feng, and W.-S. Li, "String similarity joins: An experimental evaluation," *Proc. VLDB Endowment*, vol. 7, no. 8, pp. 625–636, Apr. 2014, doi: [10.14778/2732296.2732299](https://doi.org/10.14778/2732296.2732299).
- [33] M. Yu, G. Li, T. Wang, J. Feng, and Z. Gong, "Efficient filtering algorithms for location-aware publish/subscribe," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 4, pp. 950–963, Apr. 2015, doi: [10.1109/TKDE.2014.2349906](https://doi.org/10.1109/TKDE.2014.2349906).
- [34] C. Yang, M. Liu, F. He, X. Zhang, J. Peng, and J. Han, "Similarity modeling on heterogeneous networks via automatic path discovery," in *Machine Learning and Knowledge Discovery in Databases (Lecture Notes in Computer Science)*, vol. 11052, M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, and G. Ifrim, Eds. Cham, Switzerland: Springer, 2019, pp. 37–54.
- [35] J. Zhang *et al.*, "Fast and flexible top-K similarity search on large networks," *ACM Trans. Inf. Syst.*, vol. 36, no. 2, pp. 1–30, Sep. 2017, doi: [10.1145/3086695](https://doi.org/10.1145/3086695).
- [36] L. B. Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski, "Matching the blanks: Distributional similarity for relation learning," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, 2019, pp. 2895–2905.
- [37] X. Li, V. Ceikute, C. S. Jensen, and K. Tan, "Effective online group discovery in trajectory databases," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2752–2766, Dec. 2013, doi: [10.1109/TKDE.2012.193](https://doi.org/10.1109/TKDE.2012.193).
- [38] C. Chen, Y. Ding, X. Xie, S. Zhang, Z. Wang, and L. Feng, "TrajectoryCompressor: An online map-matching-based trajectory compression framework leveraging vehicle heading direction and change," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 2012–2028, May 2020, doi: [10.1109/TITS.2019.2910591](https://doi.org/10.1109/TITS.2019.2910591).
- [39] M. T. Mahin and T. Hashem, "Activity-aware ridesharing group trip planning queries for flexible POIs," *ACM Trans. Spatial Algorithms Syst.*, vol. 5, no. 3, pp. 1–41, Sep. 2019, doi: [10.1145/3341818](https://doi.org/10.1145/3341818).
- [40] C. Chen *et al.*, "Crowddeliver: Planning city-wide package delivery paths leveraging the crowd of taxis," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1478–1496, Jun. 2017, doi: [10.1109/TITS.2016.2607458](https://doi.org/10.1109/TITS.2016.2607458).
- [41] W. Huang, Y. Zhang, Z. Shang, and J. X. Yu, "To meet or not to meet: Finding the shortest paths in road networks," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 4, pp. 772–785, Apr. 2018, doi: [10.1109/TKDE.2017.2777851](https://doi.org/10.1109/TKDE.2017.2777851).
- [42] X.-Y. Yan, W.-X. Wang, Z.-Y. Gao, and Y.-C. Lai, "Universal model of individual and population mobility on diverse spatial scales," *Nature Commun.*, vol. 8, no. 1, p. 1639, Nov. 2017, doi: [10.1038/s41467-017-01892-8](https://doi.org/10.1038/s41467-017-01892-8).
- [43] M. M. Vazifeh, P. Santi, G. Resta, S. H. Strogatz, and C. Ratti, "Addressing the minimum fleet problem in on-demand urban mobility," *Nature*, vol. 557, no. 7706, pp. 534–538, May 2018, doi: [10.1038/s41586-018-0095-1](https://doi.org/10.1038/s41586-018-0095-1).

- [44] X. Wang, X. Zheng, Q. Zhang, T. Wang, and D. Shen, "Crowdsourcing in ITS: The state of the work and the networking," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1596–1605, Jun. 2016, doi: [10.1109/TITS.2015.2513086](https://doi.org/10.1109/TITS.2015.2513086).
- [45] X. Yin, X. Hu, Y. Chen, X. Yuan, and B. Li, "Signed-PageRank: An efficient influence maximization framework for signed social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 2208–2222, May 2021, doi: [10.1109/TKDE.2019.2947421](https://doi.org/10.1109/TKDE.2019.2947421).
- [46] L. Tang, Z. Liu, Y. Zhao, Z. Duan, and J. Jia, "Efficient ridesharing framework for ride-matching via heterogeneous network embedding," *ACM Trans. Knowl. Discovery Data*, vol. 14, no. 3, pp. 1–24, May 2020, doi: [10.1145/3373839](https://doi.org/10.1145/3373839).
- [47] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-K similarity search in heterogeneous information networks," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, Aug. 2011, doi: [10.14778/3402707.3402736](https://doi.org/10.14778/3402707.3402736).
- [48] Y. Xiao, J. Zhang, and L. Deng, "Prediction of lncRNA-protein interactions using HeteSim scores based on heterogeneous networks," *Sci. Rep.*, vol. 7, no. 1, pp. 1–12, Jun. 2017, doi: [10.1038/s41598-017-03986-1](https://doi.org/10.1038/s41598-017-03986-1).
- [49] J. G. Hoskins, C. Musco, C. Musco, and C. E. Tsourakakis, "Inferring networks from random walk-based node similarities," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, Montréal, QC, Canada, 2018, pp. 3708–3719.
- [50] Z. Jiang, H. Liu, B. Fu, Z. Wu, and T. Zhang, "Recommendation in heterogeneous information networks based on generalized random walk model and Bayesian personalized ranking," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Marina Del Rey, CA, USA, Feb. 2018, pp. 288–296.
- [51] Y. Wang, R. Kutadinata, and S. Winter, "The evolutionary interaction between taxi-sharing behaviours and social networks," *Transp. Res. A, Policy Pract.*, vol. 119, pp. 170–180, Jan. 2019, doi: [10.1016/j.tra.2018.10.043](https://doi.org/10.1016/j.tra.2018.10.043).
- [52] Y. Li, R. Chen, L. Chen, and J. Xu, "Towards social-aware ridesharing group query services," *IEEE Trans. Services Comput.*, vol. 10, no. 4, pp. 646–659, Jul./Aug. 2017, doi: [10.1109/TSC.2015.2508440](https://doi.org/10.1109/TSC.2015.2508440).
- [53] X. Geng *et al.*, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 3656–3663, Jul. 2019, doi: [10.1609/aaai.v33i01.33013656](https://doi.org/10.1609/aaai.v33i01.33013656).
- [54] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, and J. Tang, "Representation learning for attributed multiplex heterogeneous network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Anchorage, AK, USA, Jul. 2019, pp. 1358–1368.
- [55] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 37, pp. 13290–13294, 2014, doi: [10.1073/pnas.1403657111](https://doi.org/10.1073/pnas.1403657111).
- [56] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2013, pp. 3111–3119.
- [57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [58] L. Liao, X. He, H. Zhang, and T.-S. Chua, "Attributed social network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2257–2270, Dec. 2018, doi: [10.1109/TKDE.2018.2819980](https://doi.org/10.1109/TKDE.2018.2819980).
- [59] J. N. Laplante and T. P. Kaeser, "The continuing evolution of pedestrian walking speed assumptions," *ITE J.*, vol. 74, no. 9, pp. 32–40, Sep. 2004.
- [60] Y. Dong, N. V. Chawla, and A. Swami, "Metapath2vec," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Halifax, NS, Canada, Aug. 2017, pp. 135–144.
- [61] H. Zhang, L. Qiu, L. Yi, and Y. Song, "Scalable multiplex network embedding," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, Jul. 2018, pp. 3082–3088.
- [62] D. Kondor, X. Zhang, M. Meghji, P. Santi, J. Zhao, and C. Ratti, "Estimating the potential for shared autonomous scooters," *IEEE Trans. Intell. Transp. Syst.*, early access, Jan. 28, 2021, doi: [10.1109/TITS.2020.3047141](https://doi.org/10.1109/TITS.2020.3047141).
- [63] B. Mo, Z. Cao, H. Zhang, Y. Shen, and J. Zhao, "Competition between shared autonomous vehicles and public transit: A case study in Singapore," *Transp. Res. C, Emerg. Technol.*, vol. 127, Jun. 2021, Art. no. 103058, doi: [10.1016/j.trc.2021.103058](https://doi.org/10.1016/j.trc.2021.103058).
- [64] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *Eur. J. Oper. Res.*, vol. 223, no. 2, pp. 295–303, Dec. 2012, doi: [10.1016/j.ejor.2012.05.028](https://doi.org/10.1016/j.ejor.2012.05.028).
- [65] Z. C. Lipton, D. Kale, and R. Wetzell, "Directly modeling missing data in sequences with RNNs: Improved classification of clinical time series," in *Proc. 1st Mach. Learn. Healthcare Conf.*, Los Angeles, CA, USA, 2016, pp. 253–270.



Lei Tang (Member, IEEE) received the Ph.D. degree in computer science and technology from Northwestern Polytechnical University in 2012. From 2009 to 2010, she was a Visiting Researcher at the Chair of Information Systems, Mannheim University, Germany. She is currently working with the School of Information Engineering, Chang'an University, China. Her research interests include the area of intelligent transportation systems and network representation. She is a member of ACM, China Computer Federation (CCF), and Pervasive Computing Technical Committee. She received the Distinguished Young Scholars Award from the Universities of Shaanxi.



Zihang Liu received the M.S. degree from Chang'an University, in 2021. His research interests include data mining and network embedding. He is a Student Member of CCF.



Rongguo Zhang is currently pursuing the degree with the School of Computer Science, Chang'an University, China. His research interests include traffic simulation and data mining.



Zongtao Duan received the Ph.D. degree in computer science from Northwestern Polytechnical University, China, in 2006. From 2009 to 2010, he was a Post-Doctoral Research Fellow with the University of North Carolina at Chapel Hill, USA. He is currently a Professor with the School of Information Engineering, Chang'an University, China. His research interest includes context-aware computing in transportation. He is a member of CCF, CCF High Performance Computing, and Pervasive Computing Technical Committee. He received the Science and Technology Innovation Leading Talent of Shaanxi Province.



Yunji Liang received the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2016. During 2012–2017, he worked at the University of Arizona, USA, as a Visiting Scholar and a Post-Doctoral Researcher, respectively. He is currently an Associate Professor with Northwestern Polytechnical University. His research interests include social computing and crowdsensing.