P^2-Loc: A Person-2-Person Indoor Localization System in On-Demand Delivery

YI DING, Alibaba Group, China and University of Minnesota, United States
DONGZHE JIANG, Peking University, China
YU YANG, Lehigh University, United States
YUNHUAI LIU, Peking University, China
TIAN HE, University of Minnesota, United States
DESHENG ZHANG, Rutgers University, United States

On-demand delivery is a fast developing business where gig couriers deliver online orders within a short time from merchants to customers. Couriers' accurate indoor locations play an essential role in the business. Most of the existing indoor localization methods cannot be applied in practice due to the high cost or data unavailable on off-the-shelf smartphones. This paper explores a new angle to solve the problem in a *relative* and *infrastructure-free* fashion. We design a person-to-person localization system that can (1) detect encounter events via Bluetooth on couriers' smartphones, and (2) infer couriers' relative locations to all the indoor merchants via deep learning on a graph neural network. The system is infrastructure-free, map-free, and compatible for off-the-shelf devices. We deploy the system on a real-world industry platform. The system runs on the smartphones of 4,075 couriers around 79 merchants for a month. The evaluation in a mall area shows that P^2-Loc improves the mean average error compared with state-of-art infrastructure-based, report-based, and encounter-based methods. We also use an application analysis based on real-world orders and trajectory data to show that the P^2-Loc can save around $40,000 for the platform every day with improved indoor localization results.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing design and evaluation methods**.

Additional Key Words and Phrases: Indoor Localization, Graph Learning, On-Demand Delivery

ACM Reference Format:
Yi Ding, Dongzhe Jiang, Yu Yang, Yunhuai Liu, Tian He, and Desheng Zhang. 2022. P^2-Loc: A Person-2-Person Indoor Localization System in On-Demand Delivery. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 1, Article 9 (March 2022), 24 pages. https://doi.org/10.1145/3517238

1 INTRODUCTION

Nowadays, on-demand delivery [14, 16, 79, 86] is an emerging business for Gig Economy [33] where gig workers deliver orders (e.g., food) within a short time (e.g., 30 minutes) from merchants to customers. This business grows rapidly with several on-demand delivery platforms worldwide (e.g., DoorDash [17] and Eleme [16]).

To achieve timely delivery, couriers' real-time localization is one of the indispensable supporting services involving all the stakeholders including couriers, merchants, customers, and platforms such as courier navigation [85],

Authors' addresses: Yi Ding, dingx447@umn.edu, Alibaba Group, Shanghai, China and University of Minnesota, Minneapolis, United States; Dongzhe Jiang, dongzhe.jiang@pku.edu.cn, Peking University, Beijing, China; Yu Yang, yuyang@lehigh.edu, Lehigh University, Bethlehem, United States; Yunhuai Liu, yunhuai.liu@pku.edu.cn, Peking University, Beijing, China; Tian He, tianhe@umn.edu, University of Minnesota, Minneapolis, United States; Desheng Zhang, desheng.zhang@cs.rutgers.edu, Rutgers University, United States.

2474-9567/2022/3-ART9 $15.00
https://doi.org/10.1145/3517238

merchants' order preparation [18], status tracking for customers [19], and platform order dispatching [86]. While outdoor locations can be obtained by smartphone GPS accurately [51], indoor locations are difficult to acquire due to the weak GPS signal. Given many merchants' shops are located in multi-story malls in urban areas (e.g., 9,562 shops located in 576 the malls in Shanghai City), couriers' indoor localization becomes the bottleneck of improving the user experience and operational efficiency in on-demand delivery.

The state-of-art indoor localization solutions can be organized using the following two-dimensional taxonomy:
• *Absolute* Versus *Relative*. Absolute localization refers to localization in a single predetermined coordinate system (e.g., GPS) or map (e.g., floor plan) with concrete coordinates. Relative localization refers to localization in the context of one's neighbors or local environment [44, 60], usually with additional ranging or odometry information.
• *Infrastructure-based* Versus *Infrastructure-free*. In infrastructure-based solutions, some infrastructures such as Wi-Fi APs [7, 39, 41, 63, 75, 76], LED fixtures [45, 67], RFID tags [71], and PIR sensors [47, 50] are used as *anchors infrastructures* to localize the nearby *target* devices but introduce a high cost for deployment in multiple environments. Particularly, a Bluetooth beacon system, aBeacon [16], was build by Alibaba, which cost more than $100K and retired within two years. In infrastructure-free solutions, landmarks such as acoustic [53, 66], light [84], magnetic [70], and electromagnetic [48] are mapped to fixed locations on floor plans. These methods usually introduce a high cost of maintaining a fingerprints database.

Given the practical limitations, absolute localization is inapplicable due to no accurate GPS or sufficient floor plans (details in Section 2.1.), and infrastructure-based methods are inapplicable due to high cost across multiple environments. Admittedly, relative infrastructure-free localization has been studied (e.g., TransLoc [79]), where they only use couriers' reporting at merchants to obtain anchor information. However, the sparsity and uncertainty of couriers' reporting behavior lead to unsatisfactory performance in accuracy and robustness, severely restricting the upper-layer applications.

We explore *couriers' indoor encounters* as an opportunity to advance the state-of-practice. The encounters can play a key role in couriers' indoor localization for two reasons (details in Section 2.2): (i) frequent indoor encounters convey comprehensive spatial-temporal information that can be aggregated and shared between present and subsequent encounters; (ii) encounter events among couriers can be detected at low cost via Bluetooth Low Energy (BLE) advertising and scanning on couriers' smartphones under couriers' consents (See Section 7 for ethics and privacy protection).

Admittedly, although utilizing encounter information for localization is not new, existing solutions [10, 27, 36, 46, 74] are not applicable due to the following practical challenges in on-demand delivery. **(i) Lack of odometry information.** Most solutions rely on IMU-based dead reckoning to measure the relative distance to anchor infrastructure. However, IMU data on couriers' phones might be difficult to use due to accuracy (calibration lacking for 672 smartphone models of 52 brands used) and privacy (gait may leak identity information [2, 40]) issues. **(ii) Limited anchor information.** Even for encounter-based solutions, a certain amount of anchor infrastructure or semantic anchors with known locations are needed to provide initial location information. However, utilizing infrastructures limits the scale-up ability of the solution. Couriers' reporting events at merchants are used as semantic anchors in TransLoc [79], but the reporting event is sparse (two per delivery order), limiting the performance.

To tackle the challenges, we design, prototype, and evaluate $\texttt{P}^2\texttt{-Loc}$, a localization system based on **p**eer to **p**eer encounters that only use couriers' accounting data and encounter data to infer their relative locations. Specifically, we build a graph neural network (GNN) [73, 83] and utilize the idea of link prediction [21] to infer the couriers' relative locations (i.e., travel time) to all the indoor merchants from sparse reporting events and massive encounter events. In doing so, we build on recent progress in graph-based deep learning to solve the domain-specific problem in a data-driven fashion. Particularly, historical encounter and travel time data are used as labels hence no additional efforts are needed to collect labels. The contributions of this work are three-fold.

• To the best of our knowledge, we are the first to use encounters to build, prototype, deploy, and evaluate a relative, infrastructure-free indoor localization system in a real-world application, i.e., on-demand delivery. Based on the ubiquitous encounter events and couriers' indoor mobility preference, we infer the couriers' real-time locations without extra infrastructure costs, making our solution scalable for a large-scale commercial setting. We also show that the idea of utilizing encounter information for relative localization can be generalized to other problems. We share the source code [15] and one month of the data we collected in P^2-Loc [4] for the research community to validate our results and conduct further research.

• We build a GNN model to aggregate the information in couriers' encounters and infer their real-time locations. To tackle the odometry lacking, we build a graph to implicitly learn the topology of indoor merchants from couriers' historical indoor travel data; to tackle the anchor information lacking, we use GNN to integrate node information and topological structure of the graph and use link prediction to predict couriers' travel time to all the merchants. Unlike GNNs in recommendation systems where the graph is static with binary edge features, the graph in P^2-Loc is temporally correlated with multiple features on heterogeneous edges. We design an embedding network to embed the edge information to the same space with nodes and a recurrent module to utilize short-term memory.

• We prototype and implement P^2-Loc on a commercial on-demand delivery platform, and evaluate P^2-Loc in a mall with 4,075 couriers and 79 merchants for a month. The results show that P^2-Loc outperforms methods based on Wi-Fi, GPS and reporting by 9%, 19%, and 51%, respectively, and outperforms other encounter-based methods (i.e., MDS-based and statistical) by 8%and 31%. As a concrete application of P^2-Loc, we show that the same delivery order scheduling algorithm with better localization results from P^2-Loc can reduce the platform's overdue rate to save $ 40,000 every day via an offline analysis on real-world data. The evaluation and application results lead to some key lessons learned on the trade-off between the performance and model complexity.

2 MOTIVATION

2.1 ETA for Relative Localization

Unlike applications that need targets' *absolute* locations on a predefined map, on-demand delivery only needs targets' *relative* locations to the indoor merchants, hence offering a design opportunity for a map-free localization system. Relative localization works for couriers' indoor localization because it can support multiple upper-layer applications. For example, Yang *et al.* [79] shows relative localization can be used to reduce indoor walking time in order dispatching.

To obtain the relative locations, we need to infer the distance between locations, which can be measured by travel time (i.e., an estimated time of arrival (ETA) problem). Compared to the existing work [57, 86] of estimating the end-to-end delivery time that mainly depends on the couriers' road travel time, our problem is focused on more granular time in the indoor environment, which depends on the "unobservable" indoor environment setting.

2.2 Encounters among Couriers

The value of the couriers' encounters is two-fold: (i) couriers indoor mobility is patterned, so the encounters contain spatial-temporal information. For example, when two couriers encounter, we can use the time between a courier reporting "Departure" at a merchant and the encounter time as the travel time between the merchant and the encounter location because couriers usually take the shortest paths and there are no detours in between [79]. (ii) the encounters are dense; hence subsequent encounters have spatial-temporal connections. In an in-field experiment, we found that 79 couriers move around 37 merchants in a mall in rush hour (11 am), and more than 2,000 times of encounters are recorded, which introduces much more spatial-temporal information compared to reporting events only (around 200 reporting events). An illustrative example of the encounter event between two couriers is shown in Fig. 1.

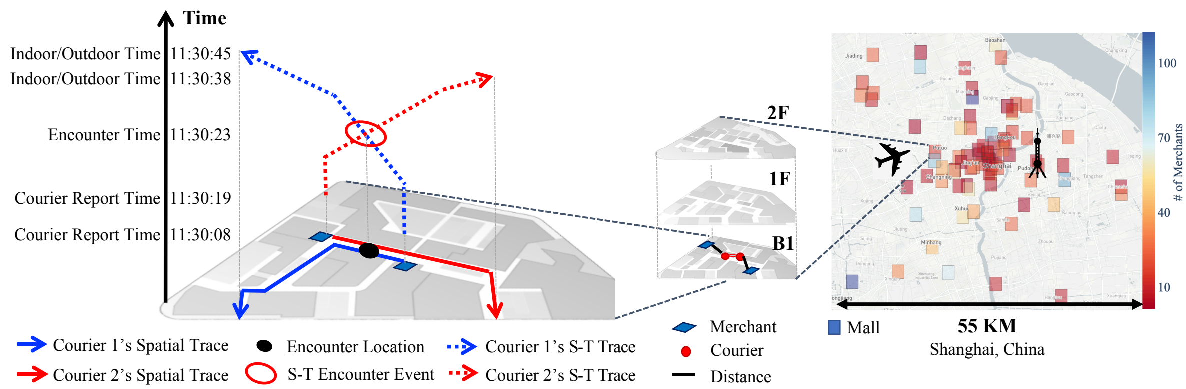

Fig. 1. Illustration of Two Courier's Encounter. (1) Two couriers encountered on their way out of the mall after picking the order on the B1 floor. The couriers' relative locations to the two merchants can be measured as the travel time between their departure and the encounter based on the shortest-path observation. (2) The right map shows all the 116 malls in Shanghai.

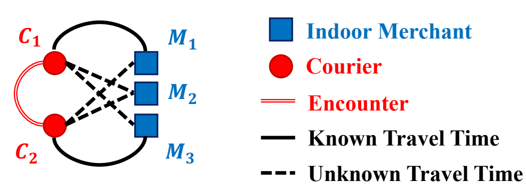

Fig. 2. Problem Setting

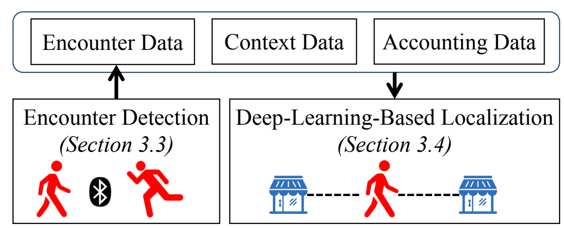

Fig. 3. System Overview

3 DESIGN

3.1 Problem Definition

The setting of the problem is shown in Fig. 2. In a time-varying graph, the input includes the real-time encounter events between couriers (double red line between courier C_1 and C_2), and the travel time between couriers and merchants (solid black line between C_1-M_1 and C_2-M_3). The output is the couriers' real-time relative locations indicated by the travel time between the couriers and merchants (dashed black line between courier C_1, C_2, and merchants M_1, M_2, M_3).

3.2 Overview

Fig. 3 shows the P^2−Loc design with two modules.

Encounter Detection (Section 3.3). In this module, we detect the couriers' encounter events by (1) developing a BLE advertising and scanning module on couriers' smartphones; (2) mining the BLE data to extract encounter events. This mechanism is simple but robust given the real-world constraints, including privacy and security concerns (for both the platform and couriers), hardware compatibility (for both iOS and Android), robustness (fault-tolerant), and non-intrusive working manner (energy and data efficiency).

Deep-Learning-Based Localization (Section 3.4). We build a heterogeneous graph and conduct deep learning on the graph using courier-merchant embedding and merchant embedding to transform the heterogeneous nodes and edges into a unified space. The idea of link prediction in the recommendation system is used to predict

Table 1. Encounter Data

Field	Value
Courier 1/2 ID	C001/C002
Enc. Start Time	7/1/20 12:11:00
Enc. End Time	7/1/20 12:11:20
Min./Max. RSSI	-90dB /-70dB
Var./Avg. RSSI	5.2 / -85dB

Table 2. Accounting Data

Field	Value
Order/Mer. ID	O001/M001
Accepting	7/1/20 12:00:00
Arrival	7/1/20 12:10:00
Departure	7/1/20 12:10:10
Delivery	7/1/20 12:25:00

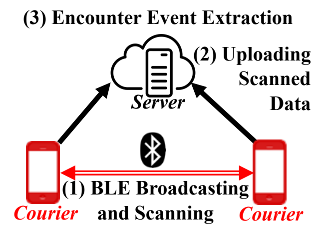

Fig. 4. Encounter Detection

the unknown travel time between couriers and merchants based on some known travel time and encounter information. Note that the model is dynamic based on periodically training with recent data.

Data. As indicated in Fig. 3, three data-sets are collected to build the graph, encounter data, accounting data, and context data. In **encounter data** (Table 1), we calculate the statistics of the encounter events extracted from encounter detection. The **accounting data** (Table 2) logs the time and locations of four primary states of each order, i.e., accepting an order, arrival at the merchant, departure from the merchant (with the order), and final delivery to the customer. The state data are from couriers' manual reporting on their APPs. The accounting data are significant to the platform because they (1) are used for the platform's new order scheduling; and (2) are shown to customers in real-time to improve customers' experiences. The **context data** record some environmental information such as weather, date, and time.

3.3 Encounter Detection

Encounter detection, or proximity detection, has been studied using Wi-Fi [36, 54] and acoustic signals [42, 58]. These solutions, however, are not applicable in our setting. Wi-Fi is not applicable due to the scanning unavailability to non-iOS APPs in off-the-shelf iOS devices [5]. Acoustic is not applicable because of couriers' frequent use of microphones (to contact merchants and customers).

BLE Advertising and Scanning. We use the iBeacon protocol [32, 37] for BLE advertising and scanning, which is a connection-less protocol that does not need a pairing process. There are three parameters in the advertising `ID tuples`, a 16-byte `UUID`, a 2-byte `Major`, and a 2-byte `Minor`. As shown in Fig. 4, the mechanism is as follows: (0) ask for couriers' consents that we can use their smartphones for encounter detection; (1) the consented couriers' smartphones conduct continuous BLE advertising and scanning at the same time in their working hours; (2) the couriers' smartphones upload the received `ID tuples` to a server in real-time by Internet connection (e.g., 4G); (3) the server extracts the encounter events from uploaded data.

The technical details in implementing the system are discussed in Section 4. A straightforward mechanism is used for advertising and scanning because (1) no additional configuration is needed after the couriers' initial consent, i.e., P^2-Loc is transparent and non-intrusive to couriers; (2) APIs provided by Android [24] and iOS [6] are used to guarantee the compatibility of P^2-Loc, which leaves little design space for setting parameters such as transmission power and advertising cycle.

3.4 Deep-Learning-Based Localization

Motivation and Challenges of Using GNN. GNNs have made great success in graph learning tasks such as node presentation and link prediction. The strength of GNNs is in the ability to learn the structure information of graphs. It provides great potential to solve our problem because the links between couriers' locations and merchants' locations are unknown and different links are also strongly related. However, building P^2-Loc based on GNNs has three challenges:
(i) the graph is heterogeneous (i.e., two types of nodes and edges, each with different features);
(ii) the graph is time-vary and temporally correlated because the edges are changing when couriers move around;

(iii) courier-courier encounter edges have multiple features such as encounter duration and RSSI statistics.

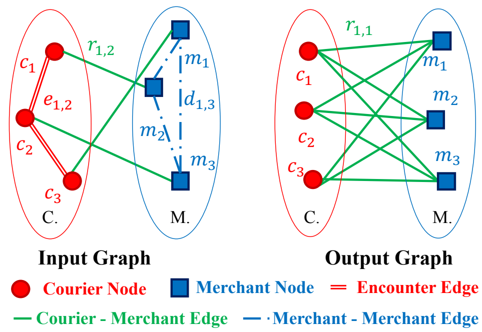

Fig. 5. Input and Output

To address the challenges, we present a novel GNN framework with graph embedding to model heterogeneous nodes and edges, and a recurrent module to consider temporal correlations.

Notations. As shown in Fig. 5, let $C = \{c_1, c_2, ..., c_{n_c}\}$ and $M = \{m_1, m_2, ..., m_{n_m}\}$ be the sets of couriers and merchants respectively, where n_c is the number of couriers, and n_m is the number of merchants. $E = \{e_{1,2}, ..., e_{n_c,n_c-1}\}$ is the set of courier-courier encounter event edges, and the data is collected from couriers' encounters. $R = \{r_{1,2}, ..., r_{n_c,n_m}\}$ is the set of courier-merchant travel time edges, and the data is collected from courier' reports and encounters when they travel from the merchants to the encounter locations, or from the encounter locations to the merchants; $D = \{d_{1,2}, ..., d_{n_m,n_m-1}\}$ is the set of merchant-merchant travel time edges, and the data is collected from couriers' reports when they travel among merchants.

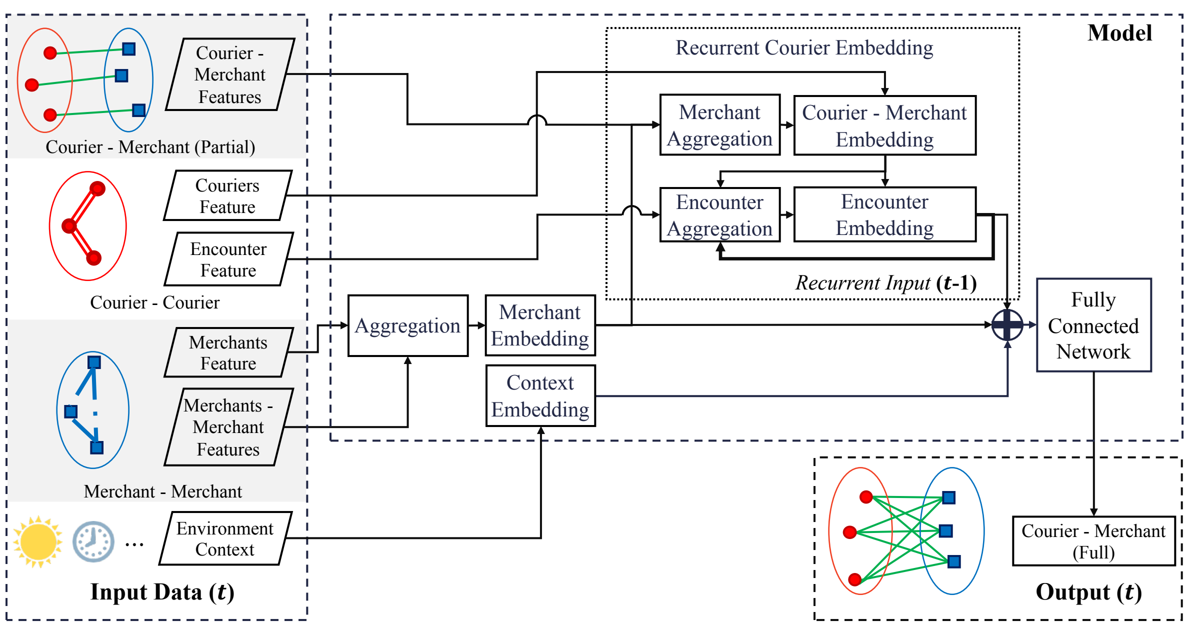

Fig. 6. Deep-Learning-Based Localization Framework

Framework Overview. The architecture of the proposed model is shown in Fig. 6. The model consists of four components: a recurrent courier graph embedding network, a merchant graph embedding network, a context embedding network, and a fully connected network. The input is a heterogeneous graph composed of three sub-graphs, i.e., an incomplete courier-merchant travel time graph, a courier-courier encounter graph, and a merchant-merchant travel time graph. We list all the node features and edge features in Table 3. The output graph is a complete courier-merchant travel time graph. Note that both the input and the output are time-based sequential data. In offline training, we use a time step t (e.g., 10s) to split the input data; in online predicting, we conduct the prediction at each time step.

Table 3. Features Used in Learning

Feature Category	Feature Name	Sample Value	Feature Category	Feature Name	Sample Value
Node Feature (Courier, C.)	Working Experience	2 years	Edge Feature (C.-C. Encounter)	Encounter Start Time	2020/07/01 12:00:00
	# of Picking Up Orders	2		Encounter End Time	2020/07/01 12:00:15
Node Feature (Merchant, M.)	Merchant Floor	1F		Encounter Duration	15 seconds
	Merchant Number	1F-23		Max. & Min. RSSI	-70 dB & -90 dB
	Merchant Type	Fast food		Avg. RSSI	-83 dB
	# of Preparing Orders	4		RSSI Variance	5.2 dB
Edge Feature (C.-M.)	Travel Time	12 seconds	Context Feature	Weather	Rainy
Edge Feature (M.-M.)	Travel Time	15 seconds		Rush Hour	Yes

Recurrent Courier Graph Embedding. The recurrent courier graph embedding aims to learn the latent factors and the temporal connection of couriers' relative locations. The challenge is how to combine the partial courier-merchant graph and courier-courier graph inherently. To address the challenge, we (1) conduct courier-merchant embedding with *Merchant Aggregation* for couriers to incorporate the travel time between the encounter location and merchant; (2) conduct encounter embedding with *Encounter Aggregation* for couriers to incorporate encounter information. Note that the output courier-courier embedding of the last time step is used as a recurrent input of the encounter aggregation to incorporate temporal connections.

In merchant aggregation, for a courier node c_i, we apply an aggregation function θ_R to its neighbors in the courier-merchants subgraph to generate the aggregated node feature $\mathbf{h}_i^R$. A merchants node is the neighbour of a courier node if it is the couriers' last-departed or next-arrived merchant, because they are on the short path to the courier's encounter location, and the travel time is known. Formally, we denote it as the following function:

$$\mathbf{h}_i^R = \sigma\left(\mathbf{W} \cdot \theta_R(\{\mathbf{x}_{i,j}, \forall r_{i,j} \in R\}) + \mathbf{b}\right) \tag{1}$$

where $r_{i,j}$ is an edge from courier c_i to a neighbour merchant m_j, $\mathbf{x}_{i,j}$ is a representation vector to denote the courier-merchant edge $r_{i,j}$. $\mathbf{x}_{i,j}$ is formulated as the concatenation of courier-merchant edge features (i.e., travel time) and merchant node embedding (after aggregation with its neighbors' information in the merchant-merchant graph); θ_R is the merchant aggregation function, and σ denotes the non-linear activation function (i.e., a rectified linear unit). $\mathbf{W}$ and $\mathbf{b}$ are the weight and bias. One popular aggregation function for θ_R is the mean operator where we take the element-wise mean of the vectors in $\{\mathbf{x}_{i,j}, \forall r_{i,j} \in R\}$. This mean-based aggregator assigns equal weight to the couriers encountered, and we adopt a one-layer weighted mean-based aggregator in our implementation of θ_R [38].

$$\mathbf{h}_i^R = \sigma\left(\mathbf{W} \cdot \left\{\sum_{r_{i,j} \in R} \alpha_{i,j} \mathbf{x}_{i,j}\right\} + \mathbf{b}\right) \tag{2}$$

where $\alpha_{i,j}$ is the weight of the encounter $e_{i,j}$, which can be an equal weight $\frac{1}{|E|}$ or an attention-based weight.

After the merchant aggregation, the courier-merchant embedding pipeline is used for concatenating merchant aggregation vector ($\mathbf{h}_i^R$) and courier features (C. in Table 3)) and feeding them into encounter aggregation and encounter embedding.

In the encounter aggregation, for a courier node c_i, we apply an aggregation function θ_E to its neighbors in the courier-courier subgraph to generate the aggregated node feature $\mathbf{h}_i^E$. Formally,

$$\mathbf{h}_i^E = \sigma\left(\mathbf{W} \cdot \theta_E(\{\mathbf{y}_{i,j}, \forall e_{i,j} \in E\}) + \mathbf{b}\right) \tag{3}$$

where $e_{i,j}$ is an edge (i.e., an encounter event) between courier c_i and c_j, $\mathbf{y}_{i,j}$ is a representation vector to denote the courier-courier edge $e_{i,j}$. $\mathbf{y}_{i,j}$ is formulated as the concatenation of the merchant aggregation vector ($\mathbf{h}_i^R$), courier-courier encounter information vector (C.-C. Encounter in Table 3), couriers attribute features vector (C. in Table 3), and the encounter embedding results from the last timestamp together as the input of the courier's embedded vector in the encounter aggregation.

After the encounter aggregation, we concatenate the output of courier-merchant embedding and the output of encounter aggregation ($\mathbf{h}_i^E$) as the final output of the recurrent counter embedding.

Merchant Graph Embedding. The merchant graph embedding is used to learn the latent relationshop between the merchants (e.g., topology and relative locations) in a transformed spatial-temporal space. There are two steps: aggregation and merchant embedding.

In the aggregation, we generate the aggregated node feature for each merchant node by applying the aggregation to its neighbors in the merchant-merchant graph. Two merchants nodes are neighbors if a courier has traveled from one merchant to the other. The input includes two parts: merchants' feature set and merchant-merchant feature set. Merchants' feature set contains merchant-related features, such as the merchant floor, the merchant number, the merchant type, and the number of preparing orders, as shown in Table 3. Specifically, we feed sparse features of merchants to an embedding layer (the same as entity embedding, a simple lookup table). Then we concatenate the embedded sparse feature and dense features of merchants together as merchants' vector. The merchant-merchant feature set contains the travel time between merchants collected from couriers' historical travel time between merchants.

In the merchant embedding, we concatenate the output of the aggregation and the merchant feature (M in Table 3) as the output of the recurrent counter embedding.

Context Embedding. Context embedding takes context information such as weather and time information as the input. Specifically, as the preprocessing of merchants features in merchant embedding, the context's representation is learned based on the entity embedding [28]. Then we fuse the entity embedding results of categorical features and the numerical features using concatenation operation.

Link Prediction. In the prediction part, the input is an arbitrary courier-merchant pair instance, and the output is the travel time between the courier-merchant pair. For a courier-merchant pair instance, we feed the concatenation of the courier embedding vector, the merchant embedding vector, and the context embedding vector to the fully-connect network. Then we get the regression result that stands for the travel time between the courier and merchant. We use ReLU as the activation function in the whole network architecture.

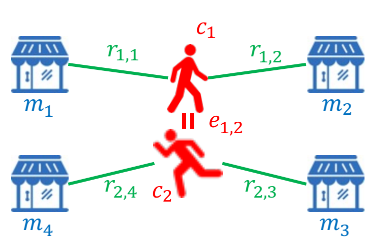

Fig. 7. Training Illustration

Model Training. We adopt a common objective function

$$\text{Loss} = \frac{1}{|O|} \sum_{r_{i,j} \in O} (r'_{i,j} - r_{i,j})^2 \tag{4}$$

where O is a set of edges between couriers and merchants, $r'_{i,j}$ and $r_{i,j}$ are the predicted and labeled travel time between the couriers' encounter locations and merchants. A part (usually 20%) of the collected courier-merchant travel time data is used as labels, which is a common practice in the link prediction. Note that the labeled travel time is collected from couriers' manual reports when they arrive or leave the merchants (Table. 2). Therefore, no additional infrastructures are needed for collecting the label. We use a small example with two couriers and four merchants to show the process in Fig. 7. When two couriers encounter on their ways among merchants, there are four courier-merchants edges and one courier-courier edge, and the corresponding data can be collected. In the training process, $r_{1,2}$ and $r_{2,4}$ can be used as labels since they are the travel time we want to predict (i.e., $r_{i,j}$ in (4)). Although couriers' reports only provide spares anchor information (i.e., two reports per order), we show that one-month data are enough to train the model with impressive performance. Admittedly, couriers' reports may have inaccurate data due to couriers' early or late reports, we use the Bluetooth beacons deployed in the mall to verify that couriers' reports work well as travel time labels.

Dynamic Environment. P^2-Loc is supposed to work in a dynamic environment where merchants' shops may open and close and couriers may come and leave. For the merchants' dynamics, since the model relies on the merchant-merchant graph embedding to learn the topology and relative locations of merchants in a mall, the merchants' information needs to be known in advance. Therefore, data need to be collected to train the model for new malls. The evaluation shows that one-month data are enough to train a satisfactory model that outperforms the baselines, which is marginal compared to the lifetime of a mall. The model also needs to be re-trained periodically to incorporate the new merchants and the floor plan changes (e.g., new elevators). For the couriers' dynamics, the courier's information does NOT need to be known in advance because the inherent information learned in the courier-courier graph and the courier-merchant graph is the relative distance between the merchants' locations and the encountered locations. When new couriers come, the prediction can be directly conducted as long as there are encounter events between the new courier and other couriers (i.e., the new courier is linked to the graphs).

Travel Time Asymmetry. In our design, we use travel time as a metric to measure the distance between a courier and a merchant, but in some extreme cases, the travel time between two locations is asymmetrical. For example, there might be a queue waiting for the upward elevator when the mall opens around 9 am, while the downward elevator is empty, which will lead to asymmetrical travel time between merchants on the ground floor and higher floors. In our graph embedding, we consider the asymmetry implicitly by assigning a direction for each edge.

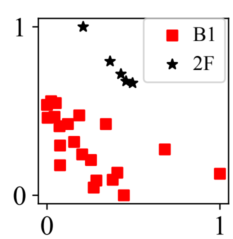

Fig. 8. Embedding Visualization

Model Interpretation. To better understand the model, we visualize the merchant embedding. Fig. 8 shows the projection of the embedding in a 2-D plane. The embedding learns both the floor information and distance information. The merchants on the same floor are nearby and the distance in the embedding is consistent with the travel time in between.

4 IMPLEMENTATION

The P^2-Loc system was designed and implemented in the real world as a core component in the [anonymous] platform, a large-scale on-demand delivery company in China. In the section, we introduce some practical issues rarely discussed or studied in a controlled environment.

4.1 Pilot Study and Real-World Deployment

In-Lab Pilot Study. We first conduct a feasibility study in the lab environment. We use five Android phones in the test and emulate the encounter events at five distances, i.e., 5m, 15m, 20m, 25m, 50m. We found that when the APP is active (either in the foreground or background), the advertising signal is stable within 15m with 90% encounter events captured, but degrades dramatically beyond 25m.

Large-Scale Real-World Deployment. After the in-lab study, we embedded the P^2-Loc into the couriers' APP in the [anonymous] platform. We developed two individual software development kits (SDK) for BLE advertising and scanning in the APP. We set some primary configurations of the P^2-Loc SDK as parameters for further developing, e.g., scanning duration and intervals, and data upload cycles. Note that the SDK and the back-end server developing were also the primary works in the implementation, but we omit them in this paper because they are standard.

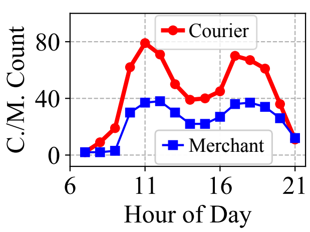

Fig. 9. C./M. Node

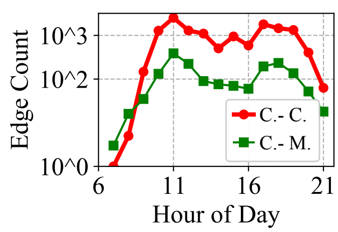

Fig. 10. C.-C./C.-M. Edge

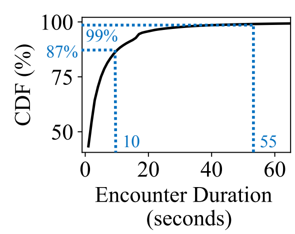

Fig. 11. Duration CDF

Real-World Encounter Facts. The P^2-Loc was embedded in the couriers' APP in June 2020, serving 4,008 couriers and 10,520 merchants. We use the data collected around a normal mall area to illustrate the statistics of the implementation. The average number of active couriers and merchants in different hours is shown in Fig. 9, where we found that the number of couriers is twice of the number of merchants during the day. The average numbers of courier-courier encounters and courier-merchant interactions (i.e., picking up orders) are shown in Fig. 10, where we found that the number of courier-courier encounters is ten times of the number of courier-merchant interactions during all the day. 87% of encounter events last less than 10 seconds (Fig. 11), and almost all the encounter events (99%) last less than 55 seconds.

The courier-courier encounter and courier-merchant graph in the rush hour (11 am) in a single day are shown in Fig. 12. Red circles stand for couriers, and blue squares stand for merchants. Red lines are encounter events between couriers, and green lines are known travel time between couriers and merchants. In the rush hour, 79 couriers move around 37 merchants, making 389 courier-merchants interactions and 2,534 courier-courier encounter events. Intuitively, the encounter density impacts the localization performance, and we will show its impact in Sec. 5 (Fig. 22).

4.2 Reliability of Encounter Detection

In the implementation, we found that not all encounter events can be detected by our encounter detection module, hence we conducted some studies to find out the reasons. Specifically, we define

$$\text{Encounter Detection Reliability} = \frac{\text{\# of Detected Encounters}}{\text{\# of Total Encounters}} \tag{5}$$

where the number of total encounters is estimated based on the BLE beacons we deployed at merchants (Fig. 17) and in-field observation (we spent two days in the mall and record the encounters of couriers).

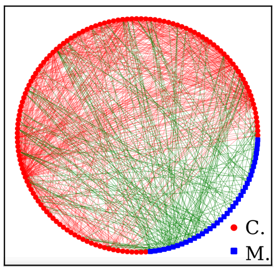

Fig. 12. Input Graph at 11 am

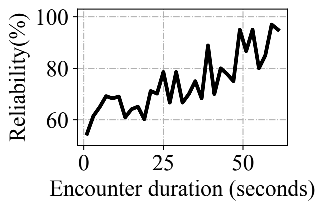

Fig. 13. Duration Impact

Table 4. Device Impact

Broadcast Device (shares) \ Scan Device (shares)	HUAWEI (35%)	Xiaomi (17%)	OPPO (12%)	Vivo (32%)	Others (4%)
HUAWEI (44%)	33%	38%	43%	34%	53%
Xiaomi (23%)	64%	77%	79%	78%	76%
OPPO (17%)	78%	77%	86%	82%	76%
Vivo (16%)	70%	75%	64%	75%	64%

Impact of Encounter Duration. The encounter duration, calculated as the first and last BLE advertising timestamp, has a significant impact on the encounter detection reliability. As shown in Fig. 13, it can be observed that reliability is less than 80% when the encounter duration is less than 25s and is greater than 90% when the encounter duration is longer than 50s.

Impact of Device Hardware. Because there are 52 brands and 672 phone models used by 300K daily active couriers, the system must be compatible with most (if not all) devices. We illustrate the reliability between four major brands as advertising and scanning device pairs in Table 4. Different devices show significant differences as advertising devices. For example, HUAWEI shows inefficiencies compared with other brands when advertising, possibly due to hardware or software differences. We show that P^2-Loc works robustly given undetected encounter events (Fig. 21).

4.3 Privacy in BLE Advertising

Potential Privacy Weakness. In the iBeacon protocol [32] we used, an `ID tuple` is fixed for each device, and the advertising is in *cleartext*. It leads to courier privacy and platform security issues under potential attacks [30, 34, 56]: (1) an adversary can replicate some courier `ID tuples` and advertise them in some other locations, which can lead to wrong encounter detection and problematic order assignment; (2) an adversary can deploy some devices to eavesdrop on the couriers' `ID tuples` and record their advertising locations as "side information" through war-driving [68]. The side information is used to attack an anonymous open data-set (e.g., anonymous online reviews or leaked data from the platform) to re-identify certain couriers [12].

Privacy Protection Mechanism. To address the potential privacy problems, we augment our advertising with the SM3 algorithm, i.e., a public Time-based One-Time Password (TOTP) [72] algorithm to encrypt the `ID tuples`, similar to Google Authenticator. Specifically, the server assigns a seed ID to a courier's phone when he logs in to the platform using the smartphone for the first time. For every duration of K, the server conducts the following three steps: (1) calculating an encrypted `ID tuples` for each smartphone based on its seed ID and current time; (2) updating the mapping of the courier's identity and its newly encrypted `ID tuples`; (3) sending the encrypted `ID tuples` to the smartphone for advertising in the following duration K.

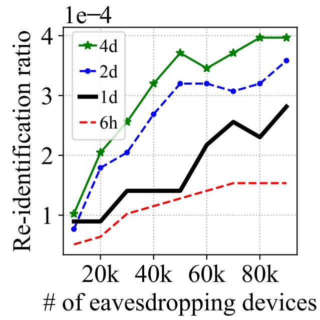

Fig. 14. Privacy Risk

Analysis Results. We evaluate the performance of our location privacy protection with offline analysis on real-world Bluetooth and trajectory data. In the attacking model, a group of adversary devices is randomly deployed at known locations to eavesdrop on couriers' advertising messages to collect side information. The side information is used to attack a supposedly-leaked anonymous data-set with all couriers' traces. In the experiments, we assume adversary devices were deployed in a group of merchants with known locations, then we find the couriers around the merchants as the eavesdropped couriers based on couriers' trajectory. We use 78.1K couriers' trajectory data in one day in Shanghai with anonymous ID as the supposedly-leaked data, and compare it with the couriers' location we eavesdropped around the merchants in a brute-force way. Based on the [12], four spatial-temporal points are enough to identify most individuals. In Fig. 14, we show that the re-identification ratio increases as the adversary utilizes more eavesdropping devices. The re-identification ratio is defined as how many couriers we can identify successfully among all couriers. When we set the ID update cycle as one day (default setting), we found that the possibility of a courier getting re-identified is less than 0.03% even 80k eavesdropping devices are deployed. The risk ratio is still below 0.04% when we use four days as ID update cycles.

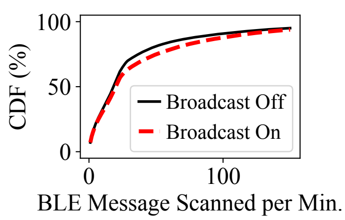

Fig. 15. Impact on BLE Scanning

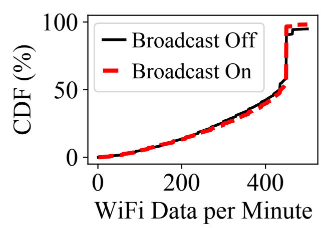

Fig. 16. Impact on WiFi Scanning

4.4 Other Real-World Issues

BLE Advertising - Scanning Interference and BLE - Wi-Fi Interference. One potential problem in implementing $\mathsf{P^2}$-Loc is that there might be conflicts between BLE advertising and scanning. We test the interference by comparing the number of BLE advertising messages scanned when the BLE advertising is on and off. The data indicates that the advertising and scanning module does not conflict because the number of BLE messages scanned per minute does not change much when the advertising is on and off (Fig. 15). We also test the interference between BLE and Wi-Fi by comparing the number of Wi-Fi data scanned when BLE advertising is on and off, and no interference is observed (Fig. 16).

APP and Service Lifetime. We also find a key factor that impacts the encounter detection performance (and possibility all the APP-level mobile computing applications) is that the operating system may kill the BLE advertising service due to resource or energy reasons. In most cases, multiples SDKs are packaged in an APP, and

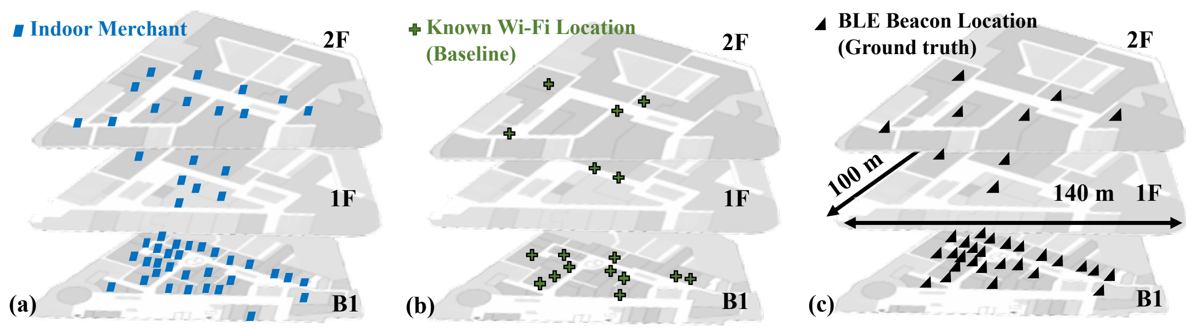

Fig. 17. The three-floor mall in the evaluation. The goal is to infer couriers' relative locations to all the merchants (Fig.(a)). Wi-Fi APs with known locations are used in the Wi-Fi-based baseline (Fig.(b)). Ground truth is collected via BLE beacons deployed at some merchants (Fig.(c)).

each is operating different tasks. Android [26] suggests that consistent tasks without user intervention should be implemented using "service", so our SDK registers a service whenever the APP initiates. However, the registered service might be killed by the OS without notification; hence the SDK is unaware when the BLE advertising or scanning fails. One solution is to re-start the service periodically to avoid unknown failures, which is simple but cannot adapt to different cases. We implement an adaptive strategy that dynamically re-start the service according to the courier's status (indoor/outdoor, order status, etc.). We omit the details here due to space limitations.

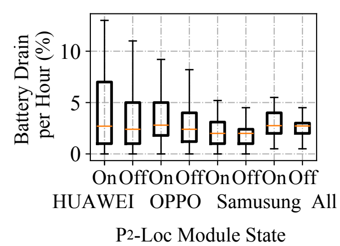

Fig. 18. Energy Consumption

Energy Consumption. In a small-scale test, we asked 62 couriers' permission for their smartphone battery information to analyze the impact of P^2-Loc on smartphone energy consumption. We collected the battery level data from the Android API [25] and removed the corresponding data if the phone is charging. Then we grouped the data by P^2-Loc states (on/off) and brands and calculated the battery drain per hour. The median battery drain per hour is 2.72% when P^2-Loc is off and 2.75% when P^2-Loc is on. The absolute additional energy consumption is 0.03% per hour, and the relative additional energy consumption is 1%. The boxplots of battery drain per hour of all smartphones and some brands are shown in Fig. 18. The results show that different smartphones show different energy consumption, but the additional energy consumption of P^2-Loc is marginal.

5 EVALUATION

5.1 Methodology

Settings. We conducted a thorough evaluation of P^2-Loc in a mall area in Shanghai in Fig. 17 for one month (June 2020). There are 294 active couriers, 51 active merchants, and 2427 delivery orders each day. We deploy some BLE beacon devices in the merchants to get ground truth.

Metrics. In P^2-Loc, we achieve indoor relative localization from the couriers to the merchants. To evaluate the relative location inferred by P^2-Loc, we compare the estimated travel time between the encounter location and a merchant with the ground truth that the courier visited after. We define an absolute time error (ATE) as follows

$$\text{ATE} = |t_i - \hat{t}_i| \tag{6}$$

where $|\hat{t}_i|$ and t_i are predicted travel time and the ground truth respectively.

Ground Truth. Although the training labels can be collected from couriers' reports, a question remains that whether the collected labels work well in a real-world application given potential inaccuracies due to couriers' early or late reports. Therefore, we use the BLE beacons deployed at the merchants to get the ground truth for the couriers' travel time among merchants. We obtained 34 merchants' consents to deploy BLE beacon devices, and each beacon device is bound to one merchant, as shown in Fig. 17. We follow the idea in [61] to find the accurate time of couriers' arrival and departure at each merchant; hence we know the ground truth travel time from the encounter location to the merchants. The training process is infrastructure-free since only couriers' report is used, and we use the ground truth data in the testing phase not only to evaluate the model but also the effectiveness of couriers' report as labels. Since we have verified the effectiveness of couriers' reports as labels in the work, no hardware or infrastructure when we apply P^2-Loc in real-world applications.

Baselines. We choose the following baselines and group them into two categories: *anchors baselines* and *model baselines*. We use anchors baselines including Wi-Fi, GPS, TransLoc, and P^2-Loc- to show the effectiveness of utilizing encounters as additional information. We use model baselines including MDS and Statistical methods to show the effectiveness of GNN. We do not include the methods that required sophisticated devices or extra fingerprinting effort, considering their practical constraints in large-scale deployment.

• *Wi-Fi-based Methods (WiFi).* To show P^2-Loc's superiority compared to state-of-art Wi-Fi-based methods, we embedded a Wi-Fi scanning module in some couriers' Android phones APP under their consent. The module periodically scans the Wi-Fi signals around and returns the Wi-Fi list with a timestamp. The scanning cycle is set as one minute, but the scanning would be throttled when the APP runs in the background [13]. We follow the idea in [35], where we use the order of Wi-Fi RSSI values to find the courier's location and calculate the $\hat{t}_i$ of Wi-Fi based on the couriers' historical travel time. We conducted a wardriving process and collected the location of 18 Wi-Fi APs (Fig. 17).

• *GPS-based Methods (GPS).* To show P^2-Loc's superiority compared to state-of-practice methods. We utilize received GPS signals to localize couriers. The $\hat{t}_i$ of GPS is calculated using the distance between the courier and the merchant, and the average speed variable depends on the area and time.

• *Report-based Methods (TransLoc).* To show the advantages of combining encounter data and accounting data with using accounting data only, we implement TransLoc [79] where they build a symbolic graph from couriers' report data to predict the courier's arrival time at the indoor shops.

• *No-Encounter (*P^2-Loc-*).* To show the effect of encounter information, we implement a deep learning model without encounter data. That is, we only use the courier-merchant distance and the merchant-merchant distance to build the graph in Fig. 5 and Fig. 6.

• *Encounter-based Localization Methods (Enc-MDS, Enc-Stat.)* To show the effectiveness of using GNN in encounter-based localization, we consider two encounter-based works using multidimensional scaling (MDS) [64] and statistical method [65]. The statistical method (Enc-Stat.) is a hybrid solution where Wi-Fi is used for localization and Bluetooth is used for encounter detection.

Hardware and Parameter Settings. The detailed parameter settings for GNN are provided in Table 5. The parameter settings are based on the tradeoff of couriers' indoor mobility, order batching and scheduling cycle, and computation cost. The following hardware and software configurations are used in the evaluation: CentOS, NVIDIA GeForce RTX 2080 Ti, and 78G memory.

Table 5. Parameter Settings.

Hyper-parameter	Setting	Hyper-parameter	Setting
GNN hidden dims.	64	Batch size	512
Embedding dims.	3	Learning rate	0.001
FC network hidden dims.	64	Optimizer	'RMSprop'
Predict time step	10 seconds	Activation	'Relu'

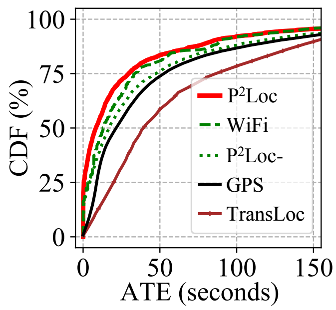

Fig. 19. Performance of Different Types of Anchor Info.

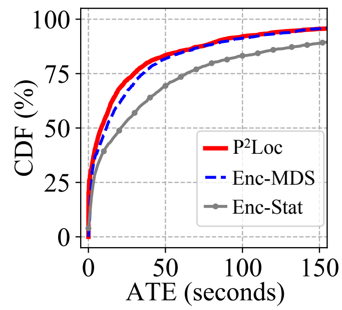

Fig. 20. Performance of Different Models

Performance Compared with Different Types of Anchor Information. Fig. 19 shows the performance of P^2-Loc compared with anchor baselines, where P^2-Loc performs better than all the baselines consistently. Quantitatively, the mean absolute error (MAE) in seconds are 29.39s for P^2-Loc, 32.41s for Wi-Fi, 39.65s for P^2-Loc-, 36.13s for GPS, and 60.40s for TransLoc. It shows that P^2-Loc improves methods based on Wi-Fi, GPS and reporting methods by 9%, 19%, and 51%, respectively, bearing out the advantage of utilizing encounter information compared to other anchor-only solutions. The improvement compared to P^2-Loc- (26%) also verifies the value of encounters in a deep-learning-based method.

Performance Compared with Different Models. Fig. 20 shows the performance of P^2-Loc compared with model baselines, where the MAE are 32.04s for Enc-MDS, and 42.61s for Enc-Stat. The improvement introduced by P^2-Loc is 8% for Enc-MDS, and 31% for Enc-Stat, bearing out the advantage of using GNN.

Performance versus Cost Tradeoff Analysis. We analyze performance (i.e., accuracy) versus cost based on different system design choices: (i) exploring both new hardware and new software (i.e., aBeacon [16]); (ii) new software only (i.e., our P^2-Loc); (iii) neither new hardware nor new software (i.e., TransLoc [79]). Here the hardware and software are additionally deployed or developed instead of existing components such as smartphones or the delivery APP. We argue P^2-Loc can achieve better accuracy and cost tradeoff compared to other design choices. Compared to aBeacon with 80% accuracy on average [16], P^2-Loc can achieve the same accuracy if we consider 46 seconds as the threshold (shown as in Fig. 19), but much less costly than aBeacon (i.e., $10 for hardware only for each shop excluding installation). Even we set the threshold to be 30 seconds (i.e., the pooling time for order dispatching), the accuracy is 74% with a limited decrease compared to aBeacon. Compared to TransLoc with the lowest cost, P^2-Loc doubles the accuracy on average (shown as in Fig. 19) but only has a nearly neglectable cost of software development.

Robustness. Given the reliability issues in encounter detection (Section 4.2), we realize that not all encounter events can be captured. We evaluate the robustness of P^2-Loc by setting a part of encounters as "undetected" (i.e., corresponding encounter data not used), and show the performance at different missing ratios in Fig. 21. Note that the "undetected" encounter data are selected based on a uniform distribution to mimic the real-world setting. Compared with real-world cases (around 25% missing in dashed box), the median ATE (brown bar in the box) does not change much when more encounters are missing, but the mean ATE (green triangle) increases

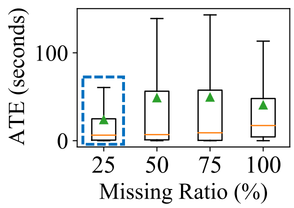

Fig. 21. Robustness

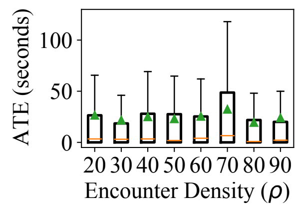

Fig. 22. Encounter Density

when more than 50% of encounter events are NOT detected. This suggests the importance of a reliable encounter detection mechanism. We envision that better localization performance can be achieved with higher encounter detection recall brought by updated smartphone hardware and encounter detection modules.

Impact of Encounter Density. We also evaluate the impact of the density of couriers, merchants, and their interactions by comparing the performance at different hours. The density ρ is defined as the number of average encounter events per courier per hour. Note that unlike the missing encounters in robustness evaluation, when the density decreases, the number of encounter events, the number of couriers, merchants, and their interactions also decreases. As shown in Fig. 22, MAE varies between 19.5s ($\rho = 80$) and 32.3s ($\rho = 30$) when density varies between 20 and 90. A less apparent observation is that MAE decreases when the density increases, but the MAE when $\rho = 30$ and 70 deviates from this trend due to the limited evaluation scale (i.e., one mall). Note that the overall density is around 40, where the MAE is around 27s.

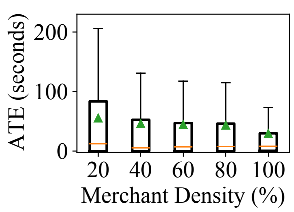

Fig. 23. Merchant Density

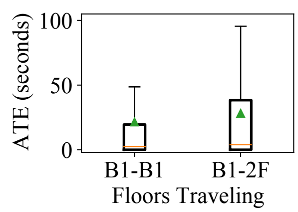

Fig. 24. Impact of Floors

Impact of Merchant Density. The density of merchants in the mall also impacts the performance, because the couriers' reports at the merchants are used as anchor information. Therefore, the merchant density can also be viewed as the anchor density in the evaluation. We do the evaluation by manually selecting certain ratio of the merchants in the training and testing. As shown in Fig. 23, the x-axis is the ratio of merchants we used in the evaluation, where 100% means all the merchants are used. It can be observed that the performance degrades when there are fewer merchants in a mall. The degradation is limited when the density varies between 40% and 80%, but will drop significantly when the density decreased to 20%. Note that the mall in the evaluation has 51 merchants located in three floors where each floor cover an area of 8000 square meters.

Impact of Encounter Merchant Floor. Unlike other indoor localization problems, one challenge in our setting is that couriers travel between indoor merchants on different floors. We evaluate the impact of cross-floor travel by extracting the same-floor data (B1-B1) and the different-floor data (B1-2F). As shown in Fig. 24, the MAE for the same floor and different floors are 21.26 seconds and 27.91 seconds, respectively. It indicates that cross-floor localization is more challenging than same-floor localization, but we can still provide better performance than Wi-Fi and GPS.

Feature Sensitivity Analysis. To evaluate the feature importance, we conduct a sensitivity analysis via the leave-one-out method (i.e., training the model without a feature). The complete model (i.e., P^2–Loc) with all

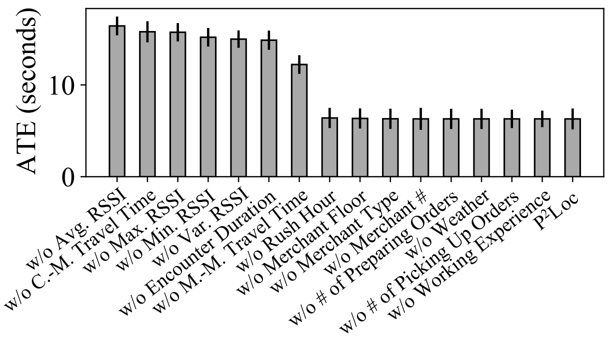

Fig. 25. Feature Sensitivity Analysis

features is used as the baseline. As shown in Fig. 25, edge features such as RSSI statistics, encounter duration, C.-M. travel time, and M.-M. travel time contribute mostly. The contribution of context features and node features is limited due to the limited evaluation environment (one mall) and time (one month). We envision these features will make more contributions in a more extended time scope and extensive area.

Model Latency and Update, Scalability, Online Deployment, and Model Generalization. For the learning model, the offline training takes 4 hours and the online prediction takes 5 seconds. Although the travel time among shops does not vary much day by day, it varies slowly when couriers enter and leave a delivery team, shops open and close in a mall, floorplan changes, and dispatching policy changes. Therefore, the models are updated periodically (e.g., weekly) with new encounters and couriers' report data. Although our work is based on an offline experiment, the model can be directly deployed online given the cloud services (e.g., Tensorflow on Google Cloud [9]). The scalability of the model can be achieved by parallel training and prediction in different malls or districts. Note that because the graph model relies on the merchants' locations and couriers' mobility, it is inappropriate to use an existing model on new malls and couriers. However, we have shown that one-month data is enough to build a decent model, and one month is a short time compared with the alteration of merchants and couriers.

6 APPLICATION

Among all the applications of couriers' indoor localization (e.g., navigation for couriers, demonstration for merchants and customers, order scheduling for the platform), we show P^2-Loc's performance in order scheduling because the benefits can be directly measured by the monetary savings. We conduct an offline analysis on real-world *order and trajectory data* to compare the scheduling performance based on P^2-Loc GPS and Enc-MDS. It is estimated that the improved couriers' indoor locations can save \$ 40,000 for the platform every day nationwide.

Order Scheduling Background. A widely used scheduling strategy is that when an order is placed, it is assigned to a nearby courier with close destinations. As shown in Fig. 26, when an order is placed in merchant m_0, the platform will check all the nearby couriers (i.e., c_1, c_2, and c_3). c_2 is the closest but c_2's destination d_2 is in the opposite direction. Both c_1's and c_3's destination (d_1 and d_3) are close to the destination of the new order (d_0). c_1 is closer to the merchant so the order will be assigned to c_1. This scheduling relies on the estimated distance between the merchants' GPS and couriers' GPS, which is inaccurate when indoor.

Analysis Settings. Because a badly dispatched order brings a terrible experience to both real couriers and customers, it is very challenging to conduct online *in-situ* experiments. Therefore, we obtain some real-world order scheduling data from the cooperating delivery platform, including the reporting timestamps (Table. 2), promised delivery time (different for each order), actual delivery time (from customers' feedback). The orders are considered as GPS-based. We select two subsets of the orders as P^2-Loc-based and Enc-MDS-based. Specifically,

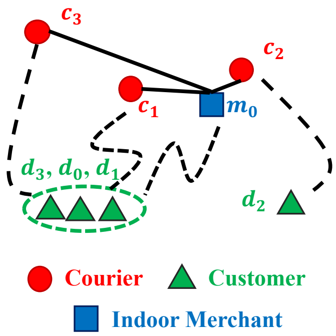

Fig. 26. Order Scheduling

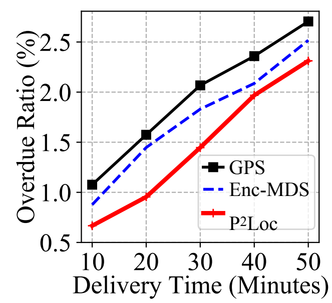

Fig. 27. Application Results

the orders that the same scheduling choice shall be made based on localization results provided by $\texttt{P}^2\texttt{-Loc}$ and Enc-MDS-based method are considered as $\texttt{P}^2\texttt{-Loc}$-based and Enc-MDS-based. We use the overdue ratio (i.e., number of overdue orders among all orders) as a metric because it can (1) measure the performance of order scheduling, (2) measure the monetary benefit of $\texttt{P}^2\texttt{-Loc}$ because the platform needs to compensate customers for each overdue order. The experiment is based on one month of orders in the mall area in Fig. 17, including 33,915 orders and 3,701 couriers.

Results. From Fig. 27 we observe that $\texttt{P}^2\texttt{-Loc}$ reduces the overdue ratio by 0.5% compared with state-of-practice GPS method, and 0.3% compared with state-of-art encounter-based localization method. We think the reason that the seconds of improvement in travel time estimation can improve the whole delivery process is that there around 600 orders to be picked up by 100 riders nearby in the mall in the rush hour (11:00am 12:00pm), and the scheduling strategy is highly relied on estimated travel time of different courier-merchant pairs [78]. Therefore, even small improvement in travel time estimation will be amplified by the order scheduling process and impact on all the orders. Although the overdue reduction brought by $\texttt{P}^2\texttt{-Loc}$ looks marginal (around 4‰), the total volume is enormous (more than 10 million daily orders of a single platform), and our method is infrastructure-free. Therefore it can be easily expanded nation-wide. Given that the platform covers $1 of the overdue penalty for each overdue order, it is estimated that $40,000 can be saved for the platform every day based on $\texttt{P}^2\texttt{-Loc}$.

7 DISCUSSION

Key Lesson Learned: Trade-off Between Performance and Model Complexity. The results (Fig. 19, 20) suggest that the deep learning based approaches outperform naive GPS and simple mechanisms (i.e., MDS) moderately, leading to a 7-second and 3-second performance gain, respectively. This leads to a set of important lessons learned, especially for industry applications, that (1) a simple approach (e.g., GPS) could be adopted first because it is easy and cheaper to deploy; (2) some advanced approaches (e.g., MDS) can improve the performance given additional information that can be obtained with a low cost fashion as inputs (e.g., encounter data); (3) an approach based on complex deep learning may further improve the performance with little performance gain but requires massive training data with accurate labels (that might not be easy to collected at a large scale) and fine-tuned parameters (that may fail when the environment changes). Thus, how to achieve a balanced tradeoff is based on specific business requirements: when low-cost additional information (e.g., encounter data) is available, some simple data-driven approaches can be used; when labeled data can be obtained at low cost and even a moderate improvement is also important (in our case the localization service helps the platform schedule 10 million orders each day), some deep learning approaches could be considered to push the performance to the limit.

Other Lessons Learned. (1) The courier's indoor encounter events offer a great opportunity to achieve relative infrastructure-free indoor localization (Fig. 1). (2) Graph learning works well for the indoor localization problem because the indoor topology and spatial-temporal connections between couriers and merchants can be represented and learned effectively and efficiently (Fig. 8). (3) We show that the system is non-intrusive, energy-efficient, and privacy-preserving in the large-scale real-world implementation (Fig. 14, 15, 16, 18). (4) Although the reliability of the BLE-based encounter detection is impacted by multiple factors (Fig. 13, 4), we show that P^2-Loc is robust (Fig. 21), and potential improvement is expected with hardware updating. (5) We show that P^2-Loc outperforms anchor-based methods (i.e., GPS, Wi-Fi, and TransLoc), and encounter-based methods without deep learning (Fig. 19, 20). We expect the improved couriers' indoor locations can save $ 40,000 for the platform every day (Fig. 27).

Further Explanation on "Anchors". "Anchor information" in this paper means location information from any source, and it is indispensable in a localization system. While for existing "anchor-free" works, they either provide "infrastructure-free" solutions [77] as in our paper or build a "target-relative" coordinate system where only relative locations are needed [55], which does not work in on-demand delivery since we need to know couriers' relative locations to the merchant shops, instead of to other couriers. In P^2-Loc, we used couriers' reports at the merchants as semantic anchor information. Compared to infrastructures, the major drawback of couriers' reports as semantic anchors is that the report is sparse because couriers only report twice (arrival/departure) during the whole delivery process. Therefore, we explore the additional spatial-temporal information in couriers' encounters for localization.

More Applications based on Encounters. Encounter-based indoor localization is our first application, and more applications are envisioned based on encounter detection in on-demand delivery. For example, in the current delivery scheme, an order is delivered by a single courier. This strategy is simple for scheduling and accounting but may have lower efficiency, especially when multiple couriers are waiting at the same merchant or traveling between similar routes. One potential improvement is that we check if we can swap their orders or put all the orders to one courier and free another courier when two couriers encounter.

Generalization to other Applications. Although P^2-Loc is designed and implemented in on-demand delivery, we believe the system and the underlying ideas work in generic scenarios where relative locations are needed based on a few known spatial-temporal points and some encounters. For example, warehouse robots [69] have been envisioned for many years but are still not widely applied due to the high cost. One of the costly parts is the onboard sensors for localization. We believe P^2-Loc, with some modifications, is a potential solution to provide an accurate yet low-cost localization service. Another potential application is vehicle-to-vehicle communication, such as dedicated short-range communications (DSRC) [44]. We can infer all the vehicles' locations given their short-range communication and a few vehicles with known locations.

Limitations. (1) *No Absolute Indoor Localization.* In this work, we infer a courier's relative distances to the indoor merchants as the courier's relative location. We argue that this "relative location" is enough for the on-demand delivery because knowing the distance between the couriers and the merchants is enough for order scheduling and time estimation. For some other applications, absolute locations are needed, so we usually need the floorplan or the absolute location of the anchors (e.g., Wi-Fi AP, BLE Beacon, LED light) to acquire the absolute locations of the target users. However, the floorplan is not always available for all the malls in a city. (2) *Cross-Mall Application.* Since the merchant topology is learned in the merchant embedding, the model only works in the mall where the labeled data are collected. Data collection and model training are needed when we want to apply the model in a new mall. (3) *Inaccurate Courier Report.* It has been shown in [79] that couriers' reports are prone to errors. This paper adopts similar pre-processing as [79] on the report data to filter out errors. The results (Fig. 19) show that solution solely relying on reporting data performs badly. However, integrating massive encounter data with reporting data can significantly improve localization results (Fig. 19).

Ethics, Privacy, and Data Release. All the data are collected under the explicit consent of the couriers. The couriers were informed that their encounters would be logged for localization and order dispatching (The privacy policy will be included in the Camera Ready version due to anonymity but similar to this one [20]). The couriers and merchants ID are anonymous keys to join different data sets. As indicated in Table 3, we did not use personal information, e.g., name, work ID, age, gender, to protect the couriers' privacy. Hence IRB is exempted. We will release one month of our data collected (Table 1, 2) for the research community to validate our results and conduct further research. The release process will be similar as a previously-released data set from the "aBeacon" platform [3] to guarantee privacy and usability.

8 RELATED WORKS

Relative Encounter-based Localization. Relative localization and encounter-based localization are closely related, because relative localization usually relies on communication among neighboring targets. The related works can be categorized into *target-relative* and *anchor-relative*.

• In target-relative works, the targets only need to decide their relative locations to other targets. Graph realization methods (e.g., MDS) based on ranging are used to calculate the relative distance directly [49, 59], or correct the dead reckoning results [64]. EASE [27] propose a distributed algorithm to infer distance among all nodes in the setting that each node knows its own location and encounter information with other nodes. The idea of relative localization is also adopted in the Internet system to assign coordinates to the hosts given the round trip time [11]. Bluetooth and Wi-Fi data from smartphones are used in [44] to decide the relative locations of vehicles. Spatial temporal phase information is used in [60] to decide the relative locations of RFID tags.

• In anchor-relative works, the targets need to decide their relative locations to some anchors with the help of nearby targets. A hybrid solution is proposed in [1] to select some anchor nodes and use them to localize the others. Encounter information from acoustic sensing is used in [10] to localize and navigate users in the indoor environment with pre-placed beacons, and in [46] to refine Wi-Fi localization results. GPS errors are modeled and eliminated in [29] based on the raw GPS data from nearby smartphones. Social-Loc [36] uses the encounter information between smartphones to improve the Wi-Fi-based localization. CoSMiC [62] use the encounters information from Wi-Fi to recover the trace of lost children. Encounter information from Bluetooth is used in [65] to localize devices relative to Wi-Fi APs.

Compared with the existing relative encounter-based works, the contribution of our works is that we use graph learning techniques to aggregate the heterogeneous spatial-temporal information from couriers' encounters, and build an infrastructure-free, and odometry-free localization system that works on off-the-shelf smartphones in a sparse-anchor environment.

Deep Learning for Mobile Applications. Deep learning is becoming increasingly important for mobile applications with the availability of large-scale data from mobile devices [23, 43]. Early works mostly focus on city-wide applications such as traffic forecasting [52, 82], bike mobility modeling [80], and ride-hailing [22], while recent works start to focus finer-grained applications such as gesture recognition [81], health monitoring [31], and indoor localization [7, 8].

9 CONCLUSION

In this work, we perform the first study on couriers' indoor encounters for the localization purpose in on-demand delivery services. We found that couriers follow specific mobility patterns (such as the shortest path between merchants) in an indoor environment, which offers us the opportunity to use dense encounter events to infer the couriers' real-time locations. To tackle some practical challenges, we design a novel GNN to aggregates the encounter events and couriers' reports to infer couriers' relative locations to all the indoor merchants in the mall. Based on the experiment results, our system can improve the localization accuracy significantly compared to the state-of-art anchor-based and encounter-based methods. We also show that our system can help the delivery platform reduce the overdue rate and save money with a case study.

REFERENCES

[1] Ahmed A Ahmed, Hongchi Shi, and Yi Shang. 2005. Sharp: A new approach to relative localization in wireless sensor networks. In *25th IEEE International Conference on Distributed Computing Systems Workshops*. IEEE, 892–898.

[2] Heikki J Ailisto, Mikko Lindholm, Jani Mantyjarvi, Elena Vildjiounaite, and Satu-Marja Makela. 2005. Identifying people from gait pattern with accelerometers. In *Biometric Technology for Human Identification II*, Vol. 5779. International Society for Optics and Photonics, 7–14.

[3] Alibaba Group. 2021. aBeacon Data-set. Webpage.

[4] Alibaba Group, Local Service BU. 2022. P2Loc Data-set Link. https://tianchi.aliyun.com/dataset/dataDetail?dataId=119097.

[5] Apple. 2020. Apple Developer Forums. Webpage.

[6] Apple. 2021. Turning an iOS Device into an iBeacon Device. Webpage.

[7] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. 2020. Deep learning based wireless localization for indoor navigation. In *ACM MobiCom*. 1–14.

[8] Li-Xuan Chuo, Zhihong Luo, Dennis Sylvester, David Blaauw, and Hun-Seok Kim. 2017. Rf-echo: A non-line-of-sight indoor localization system using a low-power active rf reflector asic tag. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 222–234.

[9] Google Cloud. 2021. Deploying models. Webpage.

[10] Ionut Constandache, Xuan Bao, Martin Azizyan, and Romit Roy Choudhury. 2010. Did you see Bob? Human localization using mobile phones. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. 149–160.

[11] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. 2004. Vivaldi: A decentralized network coordinate system. *ACM SIGCOMM Computer Communication Review* 34, 4 (2004), 15–26.

[12] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. 2013. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports* 3 (2013).

[13] Google Developers. 2020. Wi-Fi scanning overview. Webpage.

[14] Xuetao Ding, Runfeng Zhang, Zhen Mao, Ke Xing, Fangxiao Du, Xingyu Liu, Guoxing Wei, Feifan Yin, Renqing He, and Zhizhao Sun. 2020. Delivery Scope: A New Way of Restaurant Retrieval for On-demand Food Delivery Service. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3026–3034.

[15] Yi Ding and Dongzhe Jiang. 2022. P2̂-Loc Source Code. https://github.com/P2Loc/P2Loc.

[16] Yi Ding, Ling Liu, Yu Yang, Yunhuai Liu, Desheng Zhang, and Tian He. 2021. From Conception to Retirement: a Lifetime Story of a 3-Year-Old Wireless Beacon System in the Wild. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. USENIX Association, Boston, MA. https://www.usenix.org/conference/nsdi21/presentation/ding

[17] DoorDash. 2020. DoorDash. Webpage.

[18] DoorDash. 2021. How do scheduled orders work? Webpage.

[19] DoorDash. 2021. Where is my order? Webpage.

[20] Eleme. 2021. Privacy Policy and User Agreement for Couriers' APP. https://logisticsapp.ele.me/static/zb-h5/dist/protocol.html\#/?type=secret\&opt_type=0.

[21] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference*. 417–426.

[22] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3656–3663.

[23] Taesik Gong, Yeonsu Kim, Jinwoo Shin, and Sung-Ju Lee. 2019. Metasense: few-shot adaptation to untrained conditions in deep mobile sensing. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 110–123.

[24] Google. 2021. Get Started with Beacons. Webpage.

[25] Google. 2021. Monitor the Battery Level and Charging State. Webpage.

[26] Google Developers. 2020. Services overview. Webpage.

[27] Matthias Grossglauser and Martin Vetterli. 2003. Locating nodes with EASE: Last encounter routing in ad hoc networks through mobility diffusion. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, Vol. 3. IEEE, 1954–1964.

[28] Cheng Guo and Felix Berkhahn. 2016. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737* (2016).

[29] Will Hedgecock, Miklos Maroti, Janos Sallai, Peter Volgyesi, and Akos Ledeczi. 2013. High-accuracy differential tracking of low-cost GPS receivers. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. 221–234.

[30] Jun Huang, Wahhab Albazrqaoe, and Guoliang Xing. 2014. Blueid: A practical system for bluetooth device identification. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2849–2857.
[31] Sinh Huynh, Rajesh Krishna Balan, JeongGil Ko, and Youngki Lee. 2019. VitaMon: Measuring heart rate variability using smartphone front camera. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 1–14.
[32] Apple Inc. 2020. iBeacon. Webpage.
[33] Jason T Jacques and Per Ola Kristensson. 2019. Crowdworker economics in the Gig Economy. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–10.
[34] Markus Jakobsson and Sid Stamm. 2007. Web camouflage: Protecting your clients from browser-sniffing attacks. *IEEE Security & Privacy* 5, 6 (2007), 16–24.
[35] Yifei Jiang, Yun Xiang, Xin Pan, Kun Li, Qin Lv, Robert P Dick, Li Shang, and Michael Hannigan. 2013. Hallway based automatic indoor floorplan construction using room fingerprints. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. 315–324.
[36] Junghyun Jun, Yu Gu, Long Cheng, Banghui Lu, Jun Sun, Ting Zhu, and Jianwei Niu. 2013. Social-Loc: Improving indoor localization with social sensing. In *ACM SenSys*. 1–14.
[37] Kleomenis Katevas, Hamed Haddadi, Laurissa Tokarchuk, and Richard G Clegg. 2016. Detecting group formations using iBeacon technology. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. 742–752.
[38] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[39] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. 269–282.
[40] Frauke Kreuter, Georg-Christoph Haas, Florian Keusch, Sebastian Bähr, and Mark Trappmann. 2020. Collecting survey and smartphone sensor data with an app: Opportunities and challenges around privacy and informed consent. *Social Science Computer Review* 38, 5 (2020), 533–549.
[41] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. 2014. Accurate indoor localization with zero start-up cost. In *ACM MobiCom*. 483–494.
[42] Nicholas D Lane, Petko Georgiev, and Lorena Qendro. 2015. DeepEar: robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 283–294.
[43] Taegyeong Lee, Zhiqi Lin, Saumay Pushp, Caihua Li, Yunxin Liu, Youngki Lee, Fengyuan Xu, Chenren Xu, Lintao Zhang, and Junehwa Song. 2019. Occlumency: Privacy-preserving remote deep-learning inference using SGX. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–17.
[44] Dong Li, Tarun Bansal, Zhixue Lu, and Prasun Sinha. 2012. Marvel: multiple antenna based relative vehicle localizer. In *Proceedings of the 18th annual international conference on Mobile computing and networking*. 245–256.
[45] Liqun Li, Pan Hu, Chunyi Peng, Guobin Shen, and Feng Zhao. 2014. Epsilon: A visible light based positioning system. In *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*. 331–343.
[46] Hongbo Liu, Yu Gan, Jie Yang, Simon Sidhom, Yan Wang, Yingying Chen, and Fan Ye. 2012. Push the limit of WiFi based localization for smartphones. In *Proceedings of the 18th annual international conference on Mobile computing and networking*. 305–316.
[47] Xuefeng Liu, Tianye Yang, Shaojie Tang, Peng Guo, and Jianwei Niu. 2020. From relative azimuth to absolute location: Pushing the limit of pir sensor based localization. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–14.
[48] Chris Xiaoxuan Lu, Yang Li, Peijun Zhao, Changhao Chen, Linhai Xie, Hongkai Wen, Rui Tan, and Niki Trigoni. 2018. Simultaneous localization and mapping with power network electromagnetic field. In *ACM MobiCom*. 607–622.
[49] David Moore, John Leonard, Daniela Rus, and Seth Teller. 2004. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*. 50–61.
[50] Sujay Narayana, R Venkatesha Prasad, Vijay S Rao, Tamma V Prabhakar, Sripad S Kowshik, and Madhuri Sheethala Iyer. 2015. PIR sensors: Characterization and novel localization technique. In *Proceedings of the 14th international conference on information processing in sensor networks*. 142–153.
[51] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. 2010. Energy-efficient rate-adaptive GPS-based positioning for smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*. 299–314.
[52] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1720–1730.
[53] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. 2007. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th international conference on Embedded networked sensor systems*. 1–14.
[54] Timothy J Pierson, Travis Peters, Ronald Peterson, and David Kotz. 2019. Proximity detection with single-antenna IoT devices. In *ACM MobiCom*. 1–15.

[55] Nissanka B Priyantha, Hari Balakrishnan, Erik Demaine, and Seth Teller. 2003. Anchor-free distributed localization in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*. 340–341.

[56] Bahman Rashidi, Carol Fung, Anh Nguyen, Tam Vu, and Elisa Bertino. 2017. Android user privacy preserving through crowdsourcing. *IEEE Transactions on Information Forensics and Security* 13, 3 (2017), 773–787.

[57] Sijie Ruan, Zi Xiong, Cheng Long, Yiheng Chen, Jie Bao, Tianfu He, Ruiyuan Li, Shengnan Wu, Zhongyuan Jiang, and Yu Zheng. 2020. Doing in One Go: Delivery Time Inference Based on Couriers' Trajectories. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2813–2821.

[58] Hiroyuki Satoh, Makoto Suzuki, Yukou Tahiro, and Hiroyuki Morikawa. 2013. Ambient sound-based proximity detection with smartphones. In *ACM SenSys*. 1–2.

[59] Yi Shang and Wheeler Ruml. 2004. Improved MDS-based localization. In *IEEE INFOCOM 2004*, Vol. 4. IEEE, 2640–2651.

[60] Longfei Shangguan, Zheng Yang, Alex X Liu, Zimu Zhou, and Yunhao Liu. 2015. Relative localization of {RFID} tags using spatial-temporal phase profiling. In *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*. 251–263.

[61] Guobin Shen, Zhuo Chen, Peichao Zhang, Thomas Moscibroda, and Yongguang Zhang. 2013. Walkie-Markie: Indoor pathway mapping made easy. In *Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*. 85–98.

[62] Hyojeong Shin, Taiwoo Park, Seungwoo Kang, Bupjae Lee, Junehwa Song, Yohan Chon, and Hojung Cha. 2014. CoSMiC: Designing a mobile crowd-sourced collaborative application to find a missing child in situ. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. 389–398.

[63] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. 2018. Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver. In *Proceedings of the 16th annual international conference on mobile systems, applications, and services*. 376–388.

[64] Andrew Symington and Niki Trigoni. 2012. Encounter based sensor tracking. In *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*. 15–24.

[65] Viet-Cuong Ta, Trung-Kien Dao, Dominique Vaufreydaz, and Eric Castelli. 2018. Smartphone-based user positioning in a multiple-user context with Wi-Fi and Bluetooth. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 206–212.

[66] Stephen P Tarzia, Peter A Dinda, Robert P Dick, and Gokhan Memik. 2011. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*. 155–168.

[67] Zhao Tian, Yu-Lin Wei, Wei-Nin Chang, Xi Xiong, Changxi Zheng, Hsin-Mu Tsai, Kate Ching-Ju Lin, and Xia Zhou. 2018. Augmenting Indoor Inertial Tracking with Polarized Light. In *ACM MobiSys*. 362–375.

[68] Arvin Wen Tsui Tsui, Wei-Cheng Lin, Wei-Ju Chen, Polly Huang, and Hao-Hua Chu. 2010. Accuracy performance analysis between war driving and war walking in metropolitan Wi-Fi localization. *IEEE Transactions on Mobile Computing* 9, 11 (2010), 1551–1562.

[69] Kelen CT Vivaldini, Jorge PM Galdames, Thales S Bueno, Roberto C Araújo, Rafael M Sobral, Marcelo Becker, and Glauco AP Caurin. 2010. Robotic forklifts for intelligent warehouses: Routing, path planning, and auto-localization. In *2010 IEEE International Conference on Industrial Technology*. IEEE, 1463–1468.

[70] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. 197–210.

[71] Ju Wang, Liqiong Chang, Omid Abari, and Srinivasan Keshav. 2019. Are RFID Sensing Systems Ready for the Real World?. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. 366–377.

[72] Wikipedia. 2020. Time-based One-Time Password. Webpage.

[73] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[74] Henk Wymeersch, Jaime Lien, and Moe Z Win. 2009. Cooperative localization in wireless networks. *Proc. IEEE* 97, 2 (2009), 427–450.

[75] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. 2019. mD-Track: Leveraging multi-dimensionality for passive indoor Wi-Fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–16.

[76] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. 2015. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 537–549.

[77] Yurong Xu, Yi Ouyang, Zhengyi Le, James Ford, and Fillia Makedon. 2007. Mobile anchor-free localization for wireless sensor networks. In *International Conference on Distributed Computing in Sensor Systems*. Springer, 96–109.

[78] Zhe Xu, Zhixin Li, Qingwen Guan, Dingshui Zhang, Qiang Li, Junxiao Nan, Chunyang Liu, Wei Bian, and Jieping Ye. 2018. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 905–913.

[79] Yu Yang, Yi Ding, Dengpan Yuan, Guang Wang, Xiaoyang Xie, Yunhuai Liu, Tian He, and Desheng Zhang. 2020. TransLoc: transparent indoor localization with uncertain human participation for instant delivery. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–14.

[80] Zidong Yang, Ji Hu, Yuanchao Shu, Peng Cheng, Jiming Chen, and Thomas Moscibroda. 2016. Mobility modeling and prediction in bike-sharing systems. In *Proceedings of the 14th annual international conference on mobile systems, applications, and services*. 165–178.

[81] Yinggang Yu, Dong Wang, Run Zhao, and Qian Zhang. 2019. RFID based real-time recognition of ongoing gesture with adversarial learning. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 298–310.
[82] Chaoyun Zhang and Paul Patras. 2018. Long-term mobile traffic forecasting using deep spatio-temporal neural networks. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 231–240.
[83] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 793–803.
[84] Chi Zhang and Xinyu Zhang. 2016. LiTell: Robust indoor localization using unmodified light fixtures. In *ACM MobiCom*. 230–242.
[85] Yan Zhang, Yunhuai Liu, Genjian Li, Yi Ding, Ning Chen, Hao Zhang, Tian He, and Desheng Zhang. 2019. Route prediction for instant delivery. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–25.
[86] Lin Zhu, Wei Yu, Kairong Zhou, Xing Wang, Wenxing Feng, Pengyu Wang, Ning Chen, and Pei Lee. 2020. Order Fulfillment Cycle Time Estimation for On-Demand Food Delivery. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2571–2580.