SmartLOC: Indoor Localization with Smartphone Anchors for On-Demand Delivery

YI DING, Alibaba Group, China and University of Minnesota, United States

DONGZHE JIANG, Peking University, China

YUNHUI LIU, Peking University, China

DESHENG ZHANG, Rutgers University, United States

TIAN HE, University of Minnesota, United States

On-demand delivery is a rapidly developing business worldwide, where meals and groceries are delivered door to door from merchants to customers by the couriers. Couriers' real-time localization plays a key role in on-demand delivery for all parties like the platform's order dispatching, merchants' order preparing, couriers' navigation, and customers' shopping experience. Although GPS has well solved outdoor localization, indoor localization is still challenging due to the lack of large-coverage, low-cost anchors. Given the high penetration of smartphones in merchants and frequent rendezvous between merchants and couriers, we employ merchants' smartphones as indoor anchors for a new sensing opportunity. In this paper, we design, implement and evaluate SmartLOC, a map-free localization system that employs merchants' smartphones as anchors to obtain couriers' real-time locations. Specifically, we design a rendezvous detection module based on Bluetooth Low Energy (BLE), build indoor shop graphs for each mall, and adopt graph embedding to extract indoor shops' topology. To guarantee anchors' accuracy and privacy, we build a mutual localization module to iteratively infer merchants' state (in-shop or not) and couriers' locations with transformer models. We implement SmartLOC in a large on-demand delivery platform and deploy the system in 566 malls in Shanghai, China. We evaluate SmartLOC in two multi-floor malls in Shanghai and show that it can improve the accuracy of couriers' travel time estimation by 24%, 43%, 70%, and 76% compared with a straightforward graph solution, GPS, Wi-Fi, and TransLoc.

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, Yunhui Liu, Desheng Zhang, and Tian He. 2021. SmartLOC: Indoor Localization with Smartphone Anchors for On-Demand Delivery. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 4, Article 153 (December 2021), 24 pages. <https://doi.org/10.1145/3494972>

1 INTRODUCTION

With the increasing need for city-wide food and grocery delivery, on-demand delivery is in the rapid expansion where Gig workers deliver online orders (e.g., food, medicine, etc.) [39] within a short time (e.g., 30 minutes) from shops to customers [18, 95]. In fact, several on-demand delivery platforms are growing worldwide, e.g.,

Authors' addresses: Yi Ding, dingx447@umn.edu, Alibaba Group, Shanghai, China, University of Minnesota, Minneapolis, United States; Dongzhe Jiang, dongzhe.jiang@pku.edu.cn, Peking University, Beijing, China; Yunhui Liu, yunhui.liu@pku.edu.cn, Peking University, Beijing, China; Desheng Zhang, desheng.zhang@cs.rutgers.edu, Rutgers University, United States; Tian He, tianhe@umn.edu, University of Minnesota, Minneapolis, United States.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

2474-9567/2021/12-ART153 \$15.00

<https://doi.org/10.1145/3494972>

DoorDash [22], UberEats [71], Instacart [38], and Prime Now [2] in the U.S.; Deliveroo [16] in the U.K.; Meituan [56] and Eleme (Alibaba local service company) [23] in China. On such a platform, a customer uses an APP to place an order at a shop; a courier is assigned by the platform to pick up the order at the shop and then deliver it to the customer. All participating parties need to know all couriers' real-time locations, which is used by the platform to assign new orders to an optimal courier (e.g., closest), by the customer to see the order progress, and by the merchant to estimate the order preparation deadline.

Indoor localization has been well studied. For most solutions, certain equipment calibrated in prior is utilized as *anchors*. *Target* devices can be localized by nearby anchors, such as Wi-Fi APs [76], Bluetooth Low Energy (BLE) beacons [17, 54, 63, 68], LED fixtures [48, 50, 78, 91, 96], natural landmarks [52, 70, 74] with known location. They provide good results in a controlled setting, e.g., airports and museums.

In the real-world environment, however, existing anchor-based solutions present some inherent limitations in either scale, cost, or accuracy for applications requiring large-scale coverage (e.g., city-wide) like on-demand delivery. Indoor maps are unavailable and too costly to build given hundreds of malls in a city, while the cost of an anchor system (e.g., BLE beacons) is prohibitively high at city scale due to deployment and maintenance costs like fingerprinting and calibration. For example, a BLE beacon system, aBeacon [20], was build by Eleme, Alibaba in 2017. aBeacon costs more than \$100K for hardware and maintenance at 12,109 merchants in the Shanghai, and is retired within two years. Wi-Fi-based methods are widely studied, but the high energy consumption and OS restrictions hinder its large-scale application. Recent works [86] adopt users' report (e.g., platform couriers) at certain locations as "semantic" anchors in the indoor environment to achieve localization at low cost, but the models used manual reporting time as the main input, which is sparse (e.g., three reports for a 30-minute order delivery process) and prone to error. Moreover, physical beacons are still deployed for validation and evaluation in the work [86]. Our design goal is to advance the state-of-art [20, 86] in building a low-cost localization system in terms of monetary cost (compared with [20]) and human effort (compared with [86]).

To achieve this goal, we explore high penetration of low-cost smartphones and the merchant APP in merchants (e.g., shop owners), which offers us a unique opportunity to build a new anchor system without dedicated physical devices but using merchants' smartphones under their consent (See our Discussion section for ethics and privacy protection) from hardware, software, and human perspectives. From the hardware perspective, new models of low-cost (e.g., less than \$200), fully-functional (e.g., BLE) smartphones are brought to the market every year; from the software perspective, the percentage of merchants using merchant APP (instead of PC) for order management has been increasing e.g., this percentage increased from 47% to 81% from 2018/08 to 2020/05 for the delivery platform we are working for; from the human perspective, the merchants' willingness to let their phones used as anchors is the key for this participatory anchor system, which is a reasonable assumption because the anchors can help the couriers find the merchants' shops easier, hence deliver the orders faster for a better customer experience, ultimately benefiting merchants themselves.

Although the idea of making merchants' smartphones as anchors is straightforward, there are multiple technical challenges in building a real-world system. The most significant challenge comes from the merchants' mobility uncertainty, i.e., merchants (thus their smartphones) are not always in their shops. In this case, we should turn off the anchor function for both the ethnic reason, i.e., merchants' privacy, and the technical reason, i.e., accuracy as the anchor of shops. However, it is difficult to decide whether a merchant is in her shop or not without additional data collection (e.g., we have no GPS, Wi-Fi, or smartphone sensor data from merchants for ethnic reasons), extra hardware and direct human input.

In this paper, we design and implement SmartLOC, an indoor localization system that employs merchants' smartphones as anchors to localize couriers. The core idea is that we can detect rendezvous events between couriers and merchants in shops (including couriers passing by), which is a spatial-temporal point that merchants' information (e.g., number of orders in preparation) and couriers' information (e.g., GPS, number of orders in delivery) can be fused to infer couriers' real-time locations. Specifically, We (1) design a BLE-based rendezvous

detection module to detect the encounter between merchants and couriers; (2) learn a graph embedding model of the topology and travel time among shops in a mall; (3) build a mutual localization module to infer merchants' state and couriers' locations iteratively with two attention-based transformer models. Specifically, our contributions in this paper are as follows.

- We are the first, to the best of our knowledge, to design, prototype, and evaluate a map-free, smartphone-anchor-based localization system SmartLOC via graph embedding and transformer in a city-wide commercial platform. Both the conceptual novelty of using existing hardware and software as a smartphone anchor and the technical novelty of mutual localization can be generalized to other indoor localization and navigation problems, and help us transfer academic designs into industrial applications. We share the source code [19] and one month of data we used [1, 21] for the community to validate our results and conduct further research.
- We use graph embedding to aggregate select features from historical merchants, couriers, and rendezvous data to learn the topology of shops, merchants' behavior, and couriers' mobility patterns. We concatenate the embedding with real-time inputs from merchants and couriers to infer the merchants' states and couriers' locations. We use the graph isomorphism network (GIN) for embedding because GIN is effective at handling graph topology. We use the transformer for mutual localization for its sequence learning ability under merchants' mobility uncertainty.
- We implement and deploy SmartLOC on a commercial on-demand delivery platform in a large city, Shanghai. We evaluate SmartLOC in two large malls for 3 months. The results show that we outperform GPS-based, report-based, Wi-Fi-based methods, by 43%, 70%, and 76%, respectively. For a fair comparison, we also implement a model with the same input as SmartLOC, but with a MDS-based modeling, and we show that our design introduces 24% improvement. We also show that SmartLOC can help the platform save \$176,800 each year in a single city and help the couriers reduce detour ratio by 12% via a data-driven emulation.

2 MOTIVATIONS FOR SMARTLOC

2.1 Importance of Indoor Localization

Given the urbanization in developing and developed countries, many newly opened shops are located in multi-floor malls [40], letting couriers spend much time in the indoor environment to pick up the orders. Based on the analysis of a real-world city-wide application of "IODetector[94]", a sensor-based indoor/outdoor detection method, couriers spend around 1/3 of total delivery time in the indoor environment. Given the long time that couriers stay indoor, localization's importance comes in three folds:

Time Estimation. Couriers' accurate real-time indoor locations reveal the accurate time of couriers' arrival and departure at indoor shops, which is used to estimate pick-up time, meal preparation time, and delivery time. The above estimated time helps customers choose shops, merchants prepare meals, and couriers deliver orders.

Order Dispatching. Real-time couriers localization helps the platform decide the distance and travel time between different courier-shop pairs and then dispatch the orders to appropriate couriers (e.g., the closest).

Indoor Navigation. Given the large space (multiple floors) and the shop volume in some malls, couriers sometimes need navigation services. Existing indoor navigation services provided by Google Maps and Gaode [27] cannot provide satisfactory results because neither indoor map nor indoor localization is ubiquitously available. In particular, in the Application section, we show that we can provide practical semantic indoor navigation based on the indoor localization and shop graph offered by SmartLOC.

2.2 Existing Real-World Indoor Anchors

A ubiquitous and accurate anchor system for localization is fundamental for location-based services. Popular anchor systems in the real world are GPS, wireless beacons, landmarks, and manual reports, as shown in Table 1.

Table 1. Comparison of Existing Anchors.

Anchors	Coverage	Accuracy	Maintenance Cost
GPS	Global	Low	Low
Wireless Beacons [20, 53, 62, 76]	Low	High	High
Landmarks [52, 70, 74, 96]	Low	High	Low
Manual Report [86]	High	Low	High
SmartLOC	High	High	Low

GPS. GPS is the most widely used localization system. Although GPS performs poorly in an indoor environment due to weak signals blocked by the buildings, it can provide localization in coarse granularity.

Wireless Beacons. Researchers have proposed and implemented many localization systems based on Wi-Fi [76], RFID [75, 84], and BLE beacons [20, 25, 28] with accuracy from centimeters to meters. However, the cost is prohibitively high for a city-wide wireless beacon system due to deployment, fingerprinting, and calibration effort. In particular, *Wi-Fi-based solutions* are mostly advocated, while according to our platform data, Wi-Fi signals only cover 51% of our indoor shop. Wi-Fi-based solutions also have two other limitations: (1) high energy consumption, i.e., continuous Wi-Fi scanning drains the smartphone’s battery quickly [3]; (2) the Wi-Fi scanning (a prerequisite for Wi-Fi localization) is only available to OS-level functions in Apple iOS but unavailable for common APPs in practice[4]. For *BLE-based solutions*, although BLE beacons are cheap ($\leq \$20$ each), \$300 per year per beacon is estimated to guarantee its functionality (replace ones that are missing or broken) [10, 15]. A city-wide beacon system was built by Alibaba in 2017 [20], the system cost \$100K and retired after two years due to the battery running out and hardware broken.

Landmarks. Landmarks like acoustic [70], magnetic [74], electromagnetic [52], and light [96] are also discovered as anchors. These natural anchors share the advantages of zero start-up cost but usually rely on unique building structures like windows or elevators. QR code emerges as a new type of indoor anchors [57], but the QR code is prone to location fraud and needs additional human effort (manual scanning).

Manuel Report. Recent works on couriers’ manual report at shops localize the couriers [86] automatically. Still, the work relies on reliable and consistent reporting behavior, which is not generally available due to couriers’ intentional early report (for avoiding overdue responsibility) or unintentional late report (i.e., forget to report).

2.3 Relative Localization and ETA

The relaxed requirement of “relative localization” instead of “absolute localization” offers us additional design space to localize couriers with merchants’ smartphones. In on-demand delivery, the information that the platform cares about is the distance, or travel time, between the couriers and the shops for order dispatching purposes.

In this work, the relative location is measured as the travel time from the courier’s location to the shops, which is also an Estimated Time of Arrival (ETA) problem in transportation. Our problem differs from normal ETA problems [64, 95] in: (1) finer-grained time scales (seconds level) in indoor environment; (2) no need for “traffic” data. Given the differences, we believe the relationship between location and travel time revealed in this work can provide some insight for both localization and transportation works. Note that travel time estimation is non-trivial because a remarkable variation in travel time – from seconds to minutes of walking – is observed in multi-floor malls and shopping centers. The variation is non-negligible in instant delivery, where more than half of orders complete within 30 minutes.

2.4 Merchants' Mobility

Merchants' mobility is the main challenge in utilizing merchants' smartphone as the anchor for their shops. Although most merchants only use the platform's APP when they are physically in the shop, some may leave the shops without logging off their merchant APP. We show the merchants' mobility in Fig. 1 as the merchants and shops' distance when couriers received the merchants' BLE advertisement message. Couriers' GPS (as proximate merchant locations because we do not collect merchant GPS locations and receiving a BLE message indicates a courier and a merchant are very close) and the shops' latitude/longitude are used to calculate the distance. We found that around 14% of BLE advertisement are scanned at more than 100 meters away from the shop, and 4% more than 500 meters. This fact indicates that we cannot assume the merchants are always in the shop, and we need to infer whether a merchant is in her shop to protect her privacy (e.g., turn off BLE outside the shop) and guarantee the accuracy of the smartphone anchors.

Fig. 1. Merchants' Mobility

Fig. 2. System Overview

3 DESIGN OF SMARTLOC

3.1 Overview: a Three-Module System

Based on three data sources (i.e., BLE data, couriers' data, and merchants' data), we build SmartLOC with three modules as in Fig. 2: (1) a rendezvous detection module uses BLE advertisement/scanning on merchants' and couriers' smartphones to detect their encounters; (2) a shop graph embedding module uses the travel time deduced to build a graph to model the topology and travel time among shops; (3) a mutual localization module infers the merchants' state (i.e., physically in shop or not) and the couriers' locations online on the graph iteratively. Note that the graph embedding is not a one-time process. The mutual localization results are used to update the graph periodically, like daily. The detailed interaction of these three modules is given as follows.

(1) Rendezvous Detection. We let the merchants' smartphones conduct BLE advertisement and the couriers' smartphones scan and upload the received advertisement to the backend server. The information of the detected rendezvous events such as the merchant's and courier's ID, time, duration, and RSSI statistics are used as the input for graph embedding and mutual localization. Details are in Sec.3.2.

(2) Shop Graph Embedding. We build an indoor shop graph to model the topology and different couriers' traveling time among shops. Graph embedding has been studied recently to achieve outdoor travel time estimation [24, 37, 49], given its ability to encode geographic information. We utilize the following information as node features to build the graph: historical traveling information among shops; context information such as order, time, weather; merchants' features such as shop type; and couriers' features, such as experience. Details are in Sec.3.3.

(3) Mutual Localization. We conduct mutual localization to merge the courier's and the merchant's location information given their rendezvous to decide the merchants' states (in her shop or not) and couriers' locations (travel time to all the shops) iteratively. The location and state results calculated are also sent back to the graph embedding process to update the graph model. Details are in Sec.3.4.

Data Flows. As shown in Fig. 2, the data flows are as follows among the modules:

- (1) to (2): the collected rendezvous data are used to generate trip data as the input for graph embedding;
- (1) to (3): the real-time rendezvous data are used as feature for mutual localization;
- (2) to (3): the shop embedding are used as features for mutual localization;
- (3) to (2): the predicted travel time from the localization model are used as the input for graph embedding;
- Inside (3): the outputs of the merchants' state model (e.g., state probability) are used as the input of the couriers' localization model and the outputs of the couriers' localization model (e.g., travel time) are used as the input of the merchants' state model.

3.2 Module 1: Rendezvous Detection

Rendezvous detection has been studied with various solutions, such as acoustic [66, 77, 93], Wi-Fi [59, 65], and ultra-wideband (UWB) radios [11]. These methods have limited applicability in on-demand delivery because couriers need to make phone calls frequently, Wi-Fi scanning is energy-consuming, and no UWB in some smartphone models.

In SmartLOC, we use a *BLE-based advertisement/scanning* mechanism to collect the raw rendezvous data, and a *rendezvous event extraction* process to extract the rendezvous information such as time, duration, and RSSI statistics as the features for graph embedding. We use BLE for proximity detection because it is energy-efficient and does not affect couriers' normal usage [42]. We use the RSSI statistics (e.g., Average RSSI, Max. RSSI, Min. RSSI) as rendezvous features instead of using raw RSSI data to calculate the distance for detection purpose. The reasons are (1) path loss model is unreliable and difficult to calibrate for the heterogeneous environment (i.e., different malls) and hardware (i.e., smartphone brands) [47]; (2) the indoor multi-path environment impacts the RSSI-based signals significantly. However, the RSSI statistics contain useful information of the rendezvous events. For example, Min./Max. RSSI may indicate the shortest/longest distance between the merchants and the couriers in a rendezvous event.

Fig. 3. Merchant BLE

Fig. 4. Courier BLE

BLE-based advertisement/scanning is designed as follows. We let a merchant's smartphone advertise when we believe he is in his shop; we let a courier's smartphone continuously scanning for the merchants nearby in her working hours. We decide to let the merchants advertise and couriers scan, instead of the other way around, for two reasons: (1) to avoid advertisement signal interference when multiple couriers working in the same shop, especially in popular restaurants at peak hour; (2) to be compatible with existing BLE beacon services in public areas like airports and museums where couriers can also acquire their locations. Note that merchants' advertisement function is only enabled when we know that his shop is open and he is logged in our platform, and a couriers' scanning is only enabled when we know that he is log-on and online. Fig. 3 and 4 show the specific mechanism. All the merchants' and couriers' states in the process are available in standard platform accounting and log data except merchants' "In-Shop" state, which are decided by mutual localization results in Sec.3.4. Note that if our model detects a merchant is not in "In-Shop" state, we turn off her BLE advertising for two purposes: (1) protect merchants' privacy so that merchants' trajectories are not tracked outside the shop; (2) guarantee system's accuracy so that couriers will not be localized to wrong places.

Fig. 5. Raw RSSI

Fig. 6. Clustered RSSI

Rendezvous event extraction is designed as follows. We use DBSCAN [67] to extract rendezvous events from raw BLE data (i.e., merchant ID, courier ID, RSSI, and timestamp) based on the data points density. As shown in Fig. 5, the raw data uploaded from couriers only contains the IDs, timestamps, and RSSI. DBSCAN [67] is a widely used algorithm for spatial-temporal clustering to discover clusters with arbitrary shapes. The algorithm visits the whole data-set and tags each point either as core point, border point, or noise based on the number of neighbor points within a distance threshold Eps comparing with a neighbor count threshold $MinPts$. We set $MinPts = 1$ since an encounter event can be composed with as few as one BLE message. We set $Eps = 2$ seconds since both advertising and scanning cycle is less than 1 second; hence the temporal gap between two consecutive BLE messages within an encounter should not be greater than 2 seconds. Fig. 6 shows the results of the clustered encounter events. The extraction can be directly applied in real-time by checking the temporal gap between consecutive BLE messages scanned. Details on implementation is explained in Sec. 4, and the privacy issues and solutions are discussed in Sec. 7.

3.3 Module 2: Shop Graph Embedding

Motivation of Using Graph Isomorphism Network. Graph neural networks (GNNs) have achieved state-of-the-art performance in node classification, link prediction, graph classification [85], and outdoor travel time estimation [24, 37, 49]. In outdoor ETA works, the road network is defined as a directed graph, and the road segments are represented as edges [24] or nodes [37]. Similarly, localizing the couriers in a mall can be modeled as a link prediction problem where nodes are shops and edges are travel time. Although the couriers' traveling pattern is dynamic, the topology of shops and the travel time distribution is relatively fixed at the day level. The graph model is updated daily to consider the dynamic. In this paper, we use *graph isomorphism network* (GIN) because GIN is more powerful in representing graph topology than the widely-used GNNs, such as GraphSAGE [34] and GCN [44].

Background. GNNs broadly follows a recursive neighborhood aggregation scheme, where each node aggregates feature vectors of its neighbors as its embedding [85]. After k iterations of aggregation, a node is represented by its transformed feature vector, capturing the structural information within the node's k -hop neighborhood. Formally, let $G = (V, E)$ denote a graph where V is the node set and E is the edge set, the k -th layer of a GNN is

$$a_v^{(k)} = \text{AGGREGATE}^{(k)} \left(\left\{ h_u^{(k-1)} : u \in \mathcal{N}(v) \right\} \right) \quad (1)$$

$$h_v^{(k)} = \text{COMBINE}^{(k)} \left(h_v^{(k-1)}, a_v^{(k)} \right) \quad (2)$$

where $a_v^{(k)}$ and $h_v^{(k)}$ are aggregating results and the feature vector of node v at the k -th iteration/layer.

Different from traditional GNN, AGGREGATE and COMBINE in GIN based on multi-layer perceptrons (MLP) are

$$h_v^{(k)} = \text{MLP}^{(k)} \left(\left(1 + \epsilon^{(k)} \right) \cdot h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(k-1)} \right) \quad (3)$$

where ϵ is used to distinguish the root/central node. It is proved in [85] that by using MLP, GIN is capable of mapping non-isomorphic graphs into different embeddings, which enables GIN to outperform other GNNs in representing graph structures, motivating our choice for GIN.

Fig. 7. Graph Embedding with GIN

Framework. The framework of GIN-based graph embedding is shown in Fig. 7. The input is the raw shop graph where the nodes are the shops in a mall, and the edges are couriers' travel time among the shops; the output is the embedding of each node in a high-dimensional space. The edges indicate the topology of shops based on sequential rendezvous events; the node feature includes the static merchant feature shown in Table 2. In each GIN layer, we first use the AGGREGATE and COMBINE function indicated in equation (3) to extract the local topology of each node together with their edge information, and then propagate the local topology to its neighboring nodes layer by layer, until all the nodes are aware of the global connectivity in their embedding. Note that the raw graph in Fig. 7 is a directed graph because the travel from node A to B may be different when reverse, especially when A and B are on different floors.

Supervised Embedding. Similar as link prediction [89], we use a supervised graph embedding scheme, where the known historical travel time from the rendezvous results are used to train the embedding. The following smooth L_1 loss is used to train the models because the distribution of the travel time is long-tailed,

$$\text{Loss}(y^i, \theta) = \begin{cases} \frac{1}{2}(y^i - \hat{y}_\theta^i)^2, & \text{if } |y^i - \hat{y}_\theta^i| < 1 \\ |y^i - \hat{y}_\theta^i| - \frac{1}{2}, & \text{otherwise} \end{cases} \quad (4)$$

where y^i is the travel time label; $\hat{y}_\theta^i$ is the predicted travel time under a model parameter θ .

Moving Across Malls. Given that the above graph embedding is conducted separately for each mall, the localization is decomposed into consecutive parts like indoor (BLE based) - outdoor (GPS based) - indoor when couriers move across malls in the delivery process. To switch from indoor localization to outdoor localization, we use the horizontal accuracy [6, 29] and the number of satellites [30] of GPS observations (e.g., horizontal accuracy less than 10 meters, or more than six satellites observed) to decide that the courier has entered the outdoor environment. To switch from outdoor localization to indoor localization, the BLE-based localization will be activated after the first rendezvous event is detected. We are also working on using IODetector [94] to achieve fine-grained indoor/outdoor detection to differentiate the entrances of the malls. We omit the details here since they are not the major part of our design.

Non-Smartphone Merchants. For shops not using smartphones to manage orders, we use the couriers' reported arrival and departure information as rendezvous information to build the graph. Although the reported data introduce inaccurate spatial-temporal information, we show in the evaluation that our method outperforms the pure "report-based method" (i.e., a state-of-the-art TransLoc model [86]).

3.4 Module 3: Mutual Localization

3.4.1 Basic Idea. The core idea of mutual localization is that we can extract *accurate* couriers' locations from a rendezvous event based on *inaccurate* merchants' states (due to merchants' mobility) and *inaccurate* couriers' locations (due to couriers' mobility and GPS inaccuracy) with their complementary nature. Moreover, the couriers' location information and its confidence can propagate on both spatial and temporal dimensions, which further increases the accuracy of our system.

Fig. 8. The idea of Mutual Localization

As shown in Fig. 8, there are three rendezvous events, and the three events occurred in a short time so that the location information propagates with high confidence: (1) in the first rendezvous event, the first courier with a Starbucks' order encountered the Starbucks' merchant; hence the rendezvous may be at the Starbucks. Therefore we update the courier's location and the merchant's state accordingly; (2) in the second rendezvous event, the second courier without any active order encountered the same Starbucks' merchant, we propagate the merchant's location information to this second courier because this courier brings no new location information (e.g., no Starbucks' order); (3) in the last rendezvous event, the second courier encountered a Steak merchant whose Steak shop is far away, and the merchant did not accept new orders for a while. The Steak merchant may be taking a break near the Starbucks, so we update the courier's location and the merchant's state accordingly, and then stop the merchant's BLE advertisement as in Fig. 3.

To achieve mutual localization, we design a merchants' state model to estimate and track the merchants' state (i.e., in her shop or not), and a couriers' localization model to estimate and track the couriers' locations. Please note we do not infer merchant's detailed locations due to privacy and ethics consideration. As shown in Fig. 2, the two models work iteratively where the output of the merchants' state model are used as the input (i.e., couriers' dynamic features) of the couriers' localization model, and *vice versa*.

3.4.2 Merchants' State Model. Motivation of Attention-based Transformer Model. Recurrent neural networks (RNN) such as long short term memory (LSTM) have long been used to tackle the sequential problem such as machine translation [83], speech recognition [32], and mobile applications [90]. However, RNN suffers from the two main issues [72]: (1) The sequential process inhibits parallelization, so the training is slow; (2) The model relies heavily on the previous states, which in our setting could be noises due to merchants' mobility. Recently, a new sequence-to-sequence model, Transformer [72], is proposed to use the attention mechanism to learn the

weights and relation of data points from the complete sequence. In SmartLOC, we adopt the attention-based transformer to take the couriers' previous rendezvous events and other dynamic information into consideration to infer the merchants' states in the presence of merchants' mobility noises.

Fig. 9. Encoder and Decoder

Fig. 10. Merchants' State Model with Transformer

Background. The core part of the transformer model is a stack of encoders and decoders. As shown in Fig. 9, the attention layer is a multi-head self attention network to learn the weight of input data points; the feed forward layer is a simple position-wise fully connected feed-forward network. In the decoder, an additional attention network is used to perform multi-head attention over the encoder stack's output. Residual connections around each sub-layer are used in both encoders and decoders, followed by the layer normalization.

Table 2. Model Features

Feature Category	Feature Name	Sample Value	Feature Category	Feature Name	Sample Value
Rendezvous Events Feature	Rendezvous Start Time	2020/12/20 11:38:40	Environment Feature	Time slot	Morning/Noon/Night
	Rendezvous End Time	2020/12/20 11:38:45		Hour	11
	Rendezvous Event Duration	5 seconds		Weekday	Saturday
	Average RSSI.	-80 dB		Is Business Day?	No
	Max. RSSI.	-76 dB		Weather	Rainy
	Min. RSSI.	-84 dB	Courier Feature (Dynamic)	GPS Latitude	31.2371
Courier Feature (Static)	# of Working Months	6		GPS Longitude	121.4092
Merchant Feature (Static)	Merchant Type	Fast Food		ETA to the Rend. Merchant	10 seconds
	Avg. Order Price	45		# of In-Delivery Orders	3
	Avg. Merchant Confirmation Time	3 seconds	Merchant Feature (Dynamic)	Business State	On/Off
	Avg. Courier Waiting Time	183 seconds		# of Preparing Orders	4
	Avg. Order Delivery Time	1682 seconds		Order Accepting Mode	Auto/Manual
	Avg. Predicted Preparing Time	460 seconds		Last Order Accept Time	68 seconds (ago)
				Last Order Info. Print Time	54 seconds (ago)

Framework. We design the merchants' state model with the attention-based transformer, as shown in Fig. 10. The input embedding of heterogeneous features are first concatenated, and then fed into the encoder after a positional encoding process (i.e., early fusion), which helps the network learn the order of the sequence since the transformer model contains no recurrence and no convolution. Similarly, the outputs embedding, i.e., the merchants' states, is fed into the decoder after a positional encoding process. The output of the transformer is

concatenated with the static courier and merchant features and then fed to a fully-connected layer, which outputs the final result: merchants' state probability. As in Table 2, there are 7 types of features of the model:

- Shop graph embedding represents the shops' topology and the travel time among them, which is the output of the graph embedding module (Sec.3.3).
- Rendezvous events features contain the temporal feature with couriers' behavior information and the signal features with the distance information.
- Couriers' dynamic features imply the couriers' behavior and locations in real-time.
- Merchants' dynamic features imply whether the merchant is in her shop, used to infer the rendezvous' location.
- Environment features indicate the context information like date, time, and weather.
- Couriers' static features like the work experience may reflect couriers' indoor mobility habits.
- Merchants' static features like the couriers' waiting time at the shop may reflect merchants' operational habits.

Training. The merchants' states' labels are collected from a small group of merchants who use Bluetooth printers to print receipt in the shops. Therefore we know that they are in the shops when connected with Bluetooth printers, which is within the platform's normal data collection. We adopt the following cross-entropy loss function for the training as the merchants' state inference is a binary classification problem.

$$\text{Loss}(s^i, \theta) = s^i \log \hat{s}_\theta^i + (1 - s^i) \log(1 - \hat{s}_\theta^i) \quad (5)$$

where s^i is the binary merchant state label; $\hat{s}_\theta^i$ is the predicted merchant state under model parameter θ .

3.4.3 Couriers' Localization Model. The couriers' localization model shares the same transformer structure (Fig. 10) and model input (Table 2) as the merchants' state model. The only difference is the output, the couriers' relative location (i.e., travel time to all the shops). The labels are from the trips where the origin and the destination of rendezvous events are the known shops by the merchants' state model. Similar to that in the graph embedding (equation 4), smooth L_1 loss is used because the couriers' localization is a regression problem with long tails.

4 IMPLEMENTATION OF SMARTLOC

The SmartLOC system was not only designed but also implemented in the real world as a key component in [anonymous], one of the largest on-demand delivery company in China. The section introduces real-world implementation and deployment details, including a pilot study, real-world rendezvous events, energy issues, and the APP's execution in the OS background. The real-world implementation shows our system has the potential to be energy-efficient and robust in a large-scale setting.

Fig. 11. Data Count

Fig. 12. Signal Strength

4.1 In-Lab Pilot Study

Before deploying SmartLOC to the commercial platform, we first conduct an in-lab pilot study to validate the basic functionality of SmartLOC. We let an iOS phone and an Android phone work as merchants' smartphone to conduct BLE advertisement, and let four Android phones work as couriers' smartphones and perform BLE scanning. We use the default settings for BLE advertisement power and frequency. We test the number of BLE

advertisement scanned and the signal strength at different distances (5m, 15m, 20m, 23m) in an indoor open space. As shown in Fig. 11 and 12, iOS performs significantly better than Android in both advertisement count and signal strength, which also conforms to the observations in our large-scale real-world deployment.

Fig. 13. Malls in Shanghai

Fig. 14. Spatial Density

Fig. 15. Temporal Density

4.2 Real-World Commercial Deployment

We embedded SmartLOC into the delivery system of the [anonymous] platform. We developed two software development kits (SDKs), one for BLE advertising on merchants' smartphones, and the other for BLE scanning on couriers' smartphones. Configurations such as scanning duration, scanning intervals, and data upload cycles are set as parameters for further development. Note that the SDK and the back-end server development were also the primary works in the implementation, but we omit them here due to the space limit.

We deployed the merchants' APP and couriers' APP with SmartLOC SDKs in Shanghai, China, one of the world's largest cities. As in Fig. 13, merchants and couriers with SmartLOC SDKs are working in 566 malls and markets in Shanghai. In December 2020, the number of monthly active users (MAU) of the SmartLOC SDK are 191,572 merchants and 66,848 couriers.

Real-World Rendezvous Density. We show impact of the spatial and temporal rendezvous density in Fig. 14 and 15. As shown in Fig. 14, more than 20% of malls have more than 20 shops, and around 10% of malls have more than 40 shops, which makes SmartLOC-based localization possible in these malls. As shown in Fig. 15, a courier will receive more than 100 BLE advertisements within one hour in 70% of cases, and more than 500 BLE advertisements in 20% cases, guaranteeing real-time updating of the couriers' locations.

Energy Consumption. In a small-scale experiment, we collected the smartphones' battery information under the merchants' and couriers' consent to evaluate the energy consumption of the SmartLOC SDKs. We found that the additional battery drain is less than 3% on couriers' smartphones due to BLE scanning, and less than 1% on merchants' smartphones due to BLE advertising.

Background Execution. As a localization system, SmartLOC needs continuously BLE advertising and scanning when the APP is in either the foreground or background as long as the couriers and merchants are logged in the platform to accept or deliver orders. However, BLE resources may be recycled by the operating system when APPs are in the background due to resource competition or energy efficiency reasons. Keeping the SDK alive and maintaining normal sensing is one of the most challenging problems in implementing a sensing-based system in real-world commercial APPs. In our implementation, we use the persistent connections provided in HTTP/1.x [26] to keep the APP alive in normal cases and restart the SDK when we found it is not functioning. The implementation improves the robustness of the SDKs in a heterogeneous OS environment. We omit the details here due to the space limit.

5 EVALUATION OF SMARTLOC

5.1 Methodology

Settings. We conduct an evaluation in two shopping malls in Shanghai. The first mall is Global Harbor [81], a six-floor mall with 480,000 m^2 floor area, where 193 couriers work around 104 shops, and 240,764 orders are delivered from the mall in three months. The second mall is Bailian, a six-floor mall with 430,000 m^2 floor area, where 197 couriers work around 93 shops, and 237,084 orders are delivered in three months. We collected three months of data, where we use the first two months' data for training and the last one month's data for testing. Although the rendezvous detection has been implemented in the 566 malls in Shanghai and the data has been collected, it is challenging to train 566 deep learning models at the current stage as an academic study since training 566 models needs additional hardware and distributed development skills at industrial level. Therefore, we choose two typical malls to train the learning models and evaluate the results. We also use two data-driven experiments in Section 6 to show the expected benefit of fully implementing the system at the city level.

Ground Truth. The ground truth of the couriers' locations and their travel time between indoor shops are collected from a small group of merchants who uses Bluetooth printers to print receipt in the shops. We ask for their permission for the Bluetooth printer connection information on their smartphone, which is part of legitimate business data collection under their consent. Therefore, when rendezvous events occur when the merchants are connected to the printers, we know that the merchants are in the shop with very high probability, and thus the rendezvous occurs in the shops. Although a Bluetooth printer is a perfect indicator to infer whether the merchants are in the shops, only 20% of all the merchants uses Bluetooth printers, so the printer connection information cannot be used for system design, but only for the system evaluation.

Table 3. Parameter Settings.

Hyper-parameter	Setting	Hyper-parameter	Setting
GIN hidden dims.	128	# of Transformer layer	3
Learning rate	5e-4	Transformer dims.	256
Batch size	512	feedforward	
Activation	'ReLU'	Input Seq. Length	360s
Transformer heads	8	Predict Time Step	5s

Hardware and Parameter Setting. The detailed parameter settings for graph embedding and transformer are provided in Table 3. The parameter setting is based on related deep learning studies and obtained from the grid search. The input sequence length and predict time step are tuned based on the tradeoff of couriers' indoor mobility, order batching and scheduling cycle, and computation cost. We use the following hardware configurations in the evaluation, Cent OS, NVIDIA GeForce RTX 2080 Ti, and 78G memory.

5.2 Results

Metrics: ATTE. Since the goal of SmartLOC is to estimate couriers' relative locations to the shops in the mall, we use the difference between the estimated and ground-truth travel time among shops as the metric to measure the performance of SmartLOC. We define the absolute travel time error (ATTE) as follows, similar to that in [86], $ATTE = |t_i - \hat{t}_i|$ where $\hat{t}_i$ and t_i are estimated and ground-truth travel time.

Baselines. To show the effectiveness of SmartLOC, we adopt the following baselines. Note that works requiring additional devices, labor-intensive fingerprinting, or high-sampling-rate IMU data are not adopted as baselines due to practical reasons and delivery platform requirements, e.g., energy efficiency.

- *MDS-based (SmartLOC-)*. To show the effectiveness of graph-embedding and mutual localization in rendezvous-based localization, we consider an encounter-based localization method [69] that use the same input as SmartLOC but a traditional solution, i.e., multidimensional scaling (MDS); hence we call it SmartLOC-. SmartLOC- uses the encounter and trajectory information to build a graph and use MDS for graph realization to localize the targets.
- *GPS-based (GPS)*. GPS is used as a lower-bound baseline where we calculate couriers' travel time between shops according to the shops' latitude and longitude, GPS reported from couriers' smartphones (three points per minute), and the average speed near the mall area.
- *Wi-Fi-based (WiFi)*. To show the effectiveness of BLE-based rendezvous detection and graph-based localization, we adopt a Wi-Fi-based baseline. Many Wi-Fi-based localization works have been proposed [60], and most existing works use Channel State Information (CSI) to obtain the distance and angle information between smartphone and Wi-Fi AP. However, CSI is unavailable on standard commercial smartphones [35]. Therefore, we follow the idea in [41], where we use the order of Wi-Fi RSSI values to find the courier's location and calculate $\hat{v}_i$ from the historical travel time of the courier between merchants. Under the consent of some couriers with Android phones, we collected the Wi-Fi scanning data during their working hours for evaluation only.
- *Report-based (TransLoc)*. TransLoc [86] builds a symbolic graph from couriers' report data to predict the courier's arrival time at indoor shops yet with three limitations: (1) Couriers' manual report is used as input, which is inaccurate and sparse; (2) Couriers' locations are predicted in a pointwise fashion, while the trajectory information is not considered; (3) the symbolic graph used in TransLoc tries to learn the travel time statistically, different from deep-learning-based graph embedding (e.g., our work) with dynamic platform data.

Fig. 16. Overall Results

Overall Results. The overall results are shown in Fig.16. We can observe that SmartLOC and SmartLOC- perform better than Wi-Fi and TransLoc when $ATTE \leq 60s$, and graph learning and transformer introduced in SmartLOC further improves the performance. GPS performs similarly as SmartLOC in the second mall but significantly worse in the first mall, possibly because many shops in the first mall are located on underground floors. Wi-Fi and TransLoc perform disappointingly, possibly due to the sparse Wi-Fi deployment (42 in Mall 1 and 30 in Mall 2) and inaccurate courier report behavior.

Specifically, for the overall result for both malls, the mean absolute error (MAE) of ATTE is 12.51s for SmartLOC, 16.48s for SmartLOC-, 21.89s for GPS, 42.29s for TransLoc, and 52.93s for WiFi in seconds. That is, SmartLOC improves the travel time estimation by 24%, 43%, 70%, and 76% compared with SmartLOC-, GPS, TransLoc, and WiFi, respectively.

Impact of Temporal Rendezvous Density. The density of rendezvous events is one of the major impact factors of relative localization performance. We plot the rendezvous density and the ATTE of different times in a day in Fig. 17, here the density is defined as the ratio of rendezvous events in this hour compared to the total count of rendezvous events in the whole day. Two density peaks can be observed at 11 (lunchtime) and 19 (dinner time). A weak negative correlation can be observed between density and performance, but the average error is less than

17 seconds, even at around 11 pm when the density drops dramatically. Note that in Fig.17 and the follow plots, the error bar indicates the standard error of the ATTE MAE [82].

Fig. 17. Impact of Rendezvous Density

Fig. 18. Impact of Undetected Rendezvous

Fig. 19. Impact of Penetration

Fig. 20. Impact of Device Heterogeneity

Impact of Undetected Rendezvous. Due to the multiple reasons like APPs accidental shutdown and BLE signal blocked by walls, not all rendezvous events can be detected by the rendezvous detection module. Our results show that an average of 20% rendezvous events are missing where couriers picked up orders in a shop with proper-functioning smartphone anchors, but no rendezvous were detected. We evaluate the robustness of SmartLOC by intentionally removing some real-time rendezvous data in the mutual localization module. The results in Fig. 18 show that SmartLOC's performance remains satisfactory (≤ 15 seconds) even 60% of rendezvous data are missing, which shows the robustness of SmartLOC.

Impact of System Penetration. Give that the system relies on merchants' smartphones as BLE advertisers, the penetration of the system (i.e., how many merchants participate among all merchants in a mall) also impacts the system performance. We conduct an evaluation in a mall, where we remove Bluetooth data from certain amount of participating merchants in the training set, and use couriers' manual report at the merchants instead, as discussed in "Non-Smartphone Merchants" paragraph in Section 3.3. When penetration is 0%, the system can be viewed as TransLoc+, where we only use couriers' report data but with advanced learning algorithms. As shown in Fig.19, the real-world penetration in the mall is 85% (the rightmost data point). The performance improves as the penetration increases but become stable and much better than the baselines when the penetration is greater than 35%, which can be achieved in most malls. The results indicate that the penetration has an impact on the performance when penetration is very low (e.g., $<30\%$), but the impact is ignorable for most malls where the penetration is greater than 35%.

Impact of Device Heterogeneity. Given that the device heterogeneity impacts the BLE advertising, we conduct an evaluation to investigate the impact of iOS device ratio among merchants on the system performance. To increase the variety of device heterogeneity, we evaluate the system performance in two more malls (Mall 3 & 4). Based on the results in Fig.20, the portion of merchants using iOS does NOT has a consistent impact on the final performance. One reason is that impacts of iOS ratio on the system are two folds: (1) iOS devices have higher transmission power and frequency (Fig.11, 12), which leads to more detected rendezvous events and then the better localization results; (2) Background Bluetooth advertising is banned in iOS, which leads to less detected rendezvous events (if merchants APP is not running in foreground) and then the worse localization results. Therefore more iOS devices dose not necessarily lead to better localization performance since the merchants' using habit (i.e., let APP running in foreground or background) also impacts the performance, but it's difficult to analyze without further data. Another reasons is that as analyzed in Fig.16, many factors may impact SmartLOC's performance apart from devices heterogeneity (e.g., floorplan, indoor environment), it's difficult to show the impact of device heterogeneity on mall level.

Impact of Label Scarcity. As explained in the evaluation setting, we use the Bluetooth printer to obtain the labels for merchants' states and couriers' localization models. This leads to a question on how to run the system

Fig. 21. Impact of Label Scarcity

when there is few or even no Bluetooth printer in a mall. To test the system's robustness to label scarcity, we conduct an evaluation in a mall where two cases are considered: (1) only 10% of the label is available; (2) no label is available at all. When no label is available, we assume all the merchants are in the shop when rendezvous happens. As shown in Fig. 21, when only 10% of labels are available, the performance decreases but not much (i.e., ATTE increases by 3%), which indicates that the system can perform well with very few labels. When no labels are available, the overall performance decreases significantly (22%) due to some extremely large errors (i.e., merchants are outside the shop when rendezvous happens). However, even no labels are available, the performance is still better than the baselines in Fig. 16. Besides the Bluetooth printers, we are seeking different approaches to collect the labels. Wi-Fi is a promising solution where we can ask merchants' permission to upload the Wi-Fi scanning list, then we can compare the SSIDs with the shop name to guess whether the merchant is in the shop. We don't need many merchants' Wi-Fi data since only a few labels can bring satisfactory performance, as shown in Fig. 21.

Ablation Study for Models. To justify the proposed model architecture (i.e., GIN+Transformer) and show the impact of two models separately, we conducted a model ablation study where we try different model combinations. For graph embedding, we tried GIN, GCN [44], GraphSAGE [34], and GAT [73]. For localization, we tried Transformer and LSTM. The results in Fig. 4 show that (1) GIN+Transformer is the best combination; (2) Graph embedding impacts more on the final results; (3) X+Transformer outperforms X+LSTM consistently; (4) For the performance of graph learning algorithms: GIN>GCN>GraphSage>GAT.

Table 4. Model Ablation

Models	ATTE MAE (s)	Models	ATTE MAE (s)
GIN+Transformer	12.14	GIN+LSTM	12.40
GCN+Transformer	12.50	GCN+LSTM	12.79
GraphSAGE	12.80	GraphSAGE	13.40
+Transformer		+LSTM	
GAT+Transformer	13.22	GAT+LSTM	13.74

Fig. 22. Feature Sensitivity

Ablation Study for Features. We conduct an ablation study for feature sensitivity analysis using the leave-one-out method (i.e., training the model without a feature) to show the importance of the features we used. The model with complete features is used as the baseline in the rightmost of the figure. In Fig. 22, we only plot the 15 most important features due to the space limit. We can find that merchant-related features like “LastAcceptOrderTime”, “PreparingOrderCnt”, and “MerchantType” play more important roles in our system. This result also confirms

the design principle of our mutual localization module, where the merchants' state and couriers' locations are related, and we need to infer them at the same time but iteratively.

Model Latency, Update, and Scalability. The offline training takes 1 minute and 20 minutes for the graph embedding module and the attention-based transformer model respectively, and the online prediction takes 0.26 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, dispatching policy changes, and the weather changes. Therefore, the models are updated on a daily basis with new rendezvous, couriers and merchants data. A daily update frequency also matches the “extract, transform, load (ETL) [80]” processes used in many companies' data warehouses, which are usually conducted on a daily-basis. The scalability of the model can be achieved by parallel training and prediction in different malls or districts. Note that the hardware cost is a one-time investment shared by different services, but the profit is made every day with improved localization.

Fig. 23. Real-Time Settings

Real-Time Settings. As a localization service, real-time performance is also essential. Given that online algorithms only consume little time (e.g., SmartLOC uses less than one second), the real-time performance relies fundamentally on the data collection frequency. The upload frequency of different input data varies due to collection mechanisms and energy consumption. We collect the data updating frequencies of four data types: BLE (for SmartLOC), GPS, Wi-Fi, and couriers' report (for TransLoc). As in Fig. 23, both GPS and BLE have very high data update frequency, where $\geq 50\%$ of new data comes in 2 seconds, and $\geq 95\%$ of new data comes in 10 seconds. GPS updates fast because GPS does not rely on infrastructure, and BLE updates fast due to the high density of indoor BLE advertiser (i.e., merchants' smartphones in SmartLOC). Wi-Fi updates slowly because Android has a scanning throttling strategy for the energy-saving purpose [31]. Couriers' report updates even slower because it relies on human input, and couriers only report once at the merchant for each order. Notably, in a commercial setting like the delivery platform we are working with, GPS updating is implemented every 10 seconds due to energy consumption, data transmission, and storage limitation. While BLE updating can be implemented at every 1 second given its low energy cost and limited data consumption (i.e., no data generated when no advertisers are around). To summarize, SmartLOC and other BLE-based systems are practical solutions for indoor localization in real-time aspect.

6 APPLICATIONS OF SMARTLOC

To show the effectiveness of the SmartLOC in a real-world on-demand delivery application and the potential benefit of citywide implementation, we conducted two data-driven experiments based on one-month data in a mall area. Since a poorly dispatched order brings terrible experiences to both couriers and customers, it is not practical to conduct in-situ online experiments on the platform at the current stage. The data-driven experiments show that SmartLOC can significantly reduce the couriers' indoor detour and can save \$176,800 each year in Shanghai for the platform.

6.1 Semantic Indoor Navigation

The first application is a semantic indoor navigation service. Indoor navigation is important for couriers, especially for new couriers, in large multi-floor malls.

We use the idea of semantic navigation [13]. The indoor shops are used as road signs to navigate couriers from the Origin (O) shop to the Destination (D) shop. We build a hypergraph from the graph embedding results (Sec.3.3) where a threshold is used to decide whether two shops are semantically neighboring. The link prediction results based on the graph embedding are used as the edges' weight in the hypergraph. We collected the couriers' indoor routes from the rendezvous detection (Sec.3.2). A route is considered as "with navigation" when the route is the subset of the shortest path we found on the hypergraph with the correct order, otherwise, the route is considered as "without navigation".

We define a route's detour ratio as the ratio of the route time and average travel time of the route's OD. A detour ratio smaller than one means the route takes less time than average; while greater than one means it takes more time than average. The emulation results are shown in Fig. 24. "Hop of Route" in the x-axis means how many shops the courier passed by on his route. The average detour rate of routes with navigation is smaller than 1 regardless of the hops, while the ones without navigation grows greater than 1 when the routes get longer than 4 hops. The average detour ratio is reduced by 12% from 1.05 to 0.92 with the help of SmartLOC. The results show that couriers' detours can be significantly reduced if they follow the navigation offered by SmartLOC.

Fig. 24. Navigation

Fig. 25. Dispatching

6.2 Order Dispatching

The second application is the location-based order dispatching. In most state-of-practice systems, the platform dispatch a newly placed order to a nearby courier based on the shop's and the courier's location. GPS is used to acquire couriers' locations, but GPS is inaccurate around the mall area, as discussed in the Introduction. With SmartLOC, we can improve couriers' location accuracy for dispatching by using graph-based localization results when rendezvous was detected.

We collected the couriers' order data and localization results from SmartLOC. An order is considered as dispatched "with SmartLOC" if the order is dispatched to a courier where we know she is around the shop from SmartLOC otherwise, the order is considered as dispatched "without SmartLOC".

We use a key business indicator, i.e., timeout rate, to show the effectiveness of SmartLOC in order dispatching. An order is labeled as "timeout" if the order is delivered after the promised delivery time. Since compensation are made to the customers for timeout orders, we want to reduce the timeout rate to minimize the compensation.

The emulation results are shown in Fig. 25. Both timeout rates grow with the delivery time, but the timeout rate of orders dispatched with SmartLOC is steadily 0.4% less than that without SmartLOC. Based on our emulation, the overall order timeout rate is reduced from 2.97% to 2.58%. We also find that 93% of total orders in the mall are eligible for SmartLOC-based dispatching. If we assume the results hold for all malls in Shanghai and the platform pays \$1 for each timeout order, it is expected to save \$176,800 each year in Shanghai for the platform.

7 DISCUSSION

Lessons Learned. (1) Compared with existing anchors (Table 1), merchants' smartphones are potential ubiquitous anchors for couriers, but also bring challenges due to merchants' mobility (Fig. 1). (2) BLE-based rendezvous detection showed satisfactory performance in controlled environment (Fig. 11, 12), but around 20% of rendezvous events are undetected in real-world environment. (3) In spite of the various rendezvous density (Fig. 14, 15) and uncontrolled environment factors, the real-world evaluation in two malls have shown that our system outperforms multiple state-of-art and state-of-practice baselines (Fig. 16, 17, 18). In particular, SmartLOC outperforms TransLoc suggested the advantage of automatic sensing compared to human reporting; SmartLOC outperforms SmartLOC-suggested the benefit of GIN and deep learning. Our ablation study (Fig. 22) suggested the importance of taking merchants' mobility into consideration. Finally, we found SmartLOC is robust to rendezvous density (Fig. 17) and undetected rendezvous (Fig. 18). (4) Data-drive emulations (Fig. 24, 25) show that SmartLOC has the potential to reduce couriers' indoor detour rate by 12%, and save \$176,800 each year in Shanghai for our platform.

Why Improvement at "Seconds" Level Matters? Although the absolute improvement introduced by SmartLOC is not significant (around 10-second improvement each trip in time estimation, 0.13 smaller in detour rates (1.05-0.92), and 0.4% reduction in timeout rate), the aggregated improvement is significant given the orders' large volume in our platform. For example, 0.4% reduction in timeout rate means that \$176,800 can be saved each year in a single city, which is significant for a platform operating in 300 cities in China.

Merchant Privacy Protection. We are aware that merchants' BLE advertisements may leak merchants' identity and location privacy. To address the potential privacy problems, we modified the advertisement with the SM3 algorithm, i.e., a public Time-based One-Time Password (TOTP) [79] algorithm to encrypt the advertised data, similar to Google Authenticator. Specifically, the server assigns a seed ID to a merchant's phone when she first-time logs in the platform using the smartphone. An encrypted ID tuple for each smartphone based on its seed ID and the current time is calculated periodically hence that the merchants' locations are protected from eavesdropping or wardriving. Moreover, the assigned seed ID is stored in the cloud instead of in the smartphone to avoid being attacked on the smartphone side.

Ethics, Privacy, and Data Release. All the data sets are collected under the consent of the couriers. The couriers were informed their GPS, BLE data will be logged for localization and order dispatching. The merchants were informed about their BLE data and order data will be used for couriers' localization purpose to improve platform delivery performance. The couriers and merchants' IDs are anonymous keys to join different data sets, and any other ID information cannot be tracked or identified in practice. We did not utilize personal information from the couriers, e.g., age, gender, income, to protect the couriers' privacy. So, the IRB is approved. We share the source code [19] and one month of data we used [1, 21] for the community to validate our results and conduct further research.

Limitations. Admittedly, BLE is not perfect for detecting rendezvous due to the following reasons: (1) Some rendezvous events may not be detected due to hardware or software reasons, as studied in [20]. However, we have shown that SmartLOC works robustly given undetected rendezvous events. (2) Distance between merchants and couriers cannot be accurately calculated due to the multi-path effect and blocks in the uncontrolled environment. So we did not use a distance threshold to define a rendezvous. Both official studies [12] and our study (Sec.4.1) suggested that indoor BLE range is limited to less than 20m, which is accurate enough to define "rendezvous" in our scenario. Moreover, we use RSSI statistics as rendezvous' features (Table 2), implicitly considering the distance, as RSSI indicates transmitter-receiver distance. Another limitation is that when a new shop opens in a mall, it will need one or two months to collect the data and train the new graph embedding with the new shop and courier. All the data-based methods (e.g., MDS) share the limitation. Future works can be conducted on

pre-training the model and reduce the data needed to train the model. Note that when a shop closes, we remove the node in the embedding.

Generalization and Implications. Although SmartLOC is designed for on-demand delivery, the idea of “smartphone-based anchors” can be generalized to other location-based services like restaurant recommendation, robots localization, and location-based social networks, as long as a relation between smartphones and locations or context can be mined. For example, our system can be implemented in the ridesharing platform to localize drivers or customers in downtown areas where the GPS signal is weak due to tall buildings. Further, the idea of detecting physical contact [5] with BLE-based smartphone apps has drawn much attention during the COVID-19 pandemic for public health purpose. The lessons learned in SmartLOC (e.g., robustness, energy, privacy), along with our localization learning results, can help the community combat COVID-19 or other epidemics.

8 RELATED WORKS

Indoor Anchors. Studies on indoor localization can be categorized based on the underlying anchors. Wi-Fi is the most widely studied indoor anchors [8, 14, 33, 53], but the Wi-Fi-based methods have the following limitations when applied to on-demand delivery, (1) High energy consumption due to high-frequency scanning [9]; (2) Scanning list unavailable for common iOS APPs [4]; (3) Time-consuming and labor-intensive site survey process. Other anchors like BLE beacons [20, 25] and landmarks [52, 70, 74, 96] have also been proposed, but not sustainable in city-wide on-demand delivery due to high-cost deployment or fingerprinting process and limited coverage in the indoor environment.

Human Input. Recent works use couriers’ report information as human input to localize couriers to the nearest shop, which is cost-free for city-wide application [86]. However, couriers’ reports are inaccurate, which will introduce additional errors to localization results. Moreover, the graph proposed in [86] has no topology information and cannot be used for navigation purposes, leading to poor performance, as shown in our evaluation.

Simultaneous Localization and Mapping (SLAM). SLAM has been studied in robotics for years. In SLAM, a robot builds the environment map and localizes itself simultaneously based on the sensors equipped. However, SLAM suffers from the cumulative error problem [87] and relies on high-sampling-rate sensor data (accelerometer, gyroscope, magnetometers, etc.), which is difficult to obtain due to high energy consumption on smartphones in a commercial setting [45]. Further, a high sampling rate for smartphone sensor data may lead to instability of smartphone APPs, discouraging platforms from using SLAM in practice.

Rendezvous-based Localization. Rendezvous events are considered in some works for indoor localization. Encounter or non-encounter information is used in Social-Loc [43] to improve the Wi-Fi-based localization. M2M [61] uses acoustic signals to detect the encounter events and improves the localization with ZigBee and Wi-Fi. Encounter events used in these methods to build middleware based on other infrastructure for the localization purpose. EBT [69] and Social Spring [55] convert encounters and displacement vectors to a graph. ppNav [88] employs a previous traveler’s trace information along a path to share it with encountered followers.

Deep-Learning-based Sensing. Advances in deep learning in recent years have motivated deep-learning-based solutions for sensing problems like indoor localization [58], sensing fusion [51], and human activity recognition [92]. D-Loc and MapFind [7] solve the mapping and positioning at the same time deep neural network with Wi-Fi signals. RNN is used in [36] to remove the RSSI fluctuation to improve the localization performance. A deep belief network is used in [46] to extract the latent features from unlabeled fingerprints to reduce the time needed for fingerprinting.

9 CONCLUSION

In this work, we perform the first study on couriers' indoor localization with smartphone anchors in on-demand delivery. We adopt merchants' smartphone as wireless anchors under their consent to conduct BLE advertisement for localizing couriers. We build a graph isomorphism network to aggregates the rendezvous event between couriers and merchants to learn the topology and travel time among shops in the malls. We design a mutual localization mechanism to infer merchants' states and couriers' locations iteratively using an attention-based transformer network. We implement the system in a city and evaluate the performance in two malls. Based on the experiment results, our system can improve the localization accuracy by 24%, 43%, 70%, and 76% compared with SmartLOC-, GPS, TransLoc, and WiFi, respectively. We also show that our system can help the delivery platform reduce the overdue rate, lower operational costs, and reduce couriers' detour time with a data-driven case study.

ACKNOWLEDGMENTS

The authors would like to thank anonymous reviewers for their valuable comments and suggestions. Prof. Yunhuai Liu is partially supported by the National Key R&D Program of China 2018YFB2100300, 2018YFB0803400, and National Natural Science Foundation of China (NSFC) 61925202, 61772046.

REFERENCES

- [1] Alibaba Group, Local Service BU. 2021. VALID Data-set Link. <https://tianchi.aliyun.com/dataset/dataDetail?dataId=103969>.
- [2] Amazon. 2020. Amzon Prime Now. [Webpage](#).
- [3] Ganesh Ananthanarayanan and Ion Stoica. 2009. Blue-Fi: enhancing Wi-Fi performance using bluetooth signals. In *ACM MobiSys*. 249–262.
- [4] Apple. 2020. Apple Developer Forums. [Webpage](#).
- [5] Apple. 2020. Apple's Exposure Notification System. [Webpage](#).
- [6] Apple. 2021. Apple Developer, horizontalAccuracy. [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 *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–14.
- [8] Paramvir Bahl and Venkata N Padmanabhan. 2000. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM*, Vol. 2. Ieee.
- [9] Niranjana Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. 2009. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*. 280–293.
- [10] Shelley Bernstein. 2019. *The Realities of Installing iBeacon to Scale*. <https://www.brooklynmuseum.org/community/blogosphere/2015/02/04/the-realities-of-installing-ibeacon-to-scale/>
- [11] Andreas Biri, Neal Jackson, Lothar Thiele, Pat Pannuto, and Prabal Dutta. 2020. SociTrack: Infrastructure-free interaction tracking through mobile sensor networks. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–14.
- [12] Bluetooth SIG. 2021. Understanding Bluetooth Range. [Webpage](#).
- [13] Adam Borkowski, Barbara Siemiatkowska, and Jacek Szklarski. 2010. Towards semantic navigation in mobile robotics. In *Graph transformations and model-driven engineering*. Springer, 719–748.
- [14] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N Padmanabhan. 2010. Indoor localization without the pain. In *MobiCom*. ACM.
- [15] CIO.com. 2019. *6 things marketers need to know about beacons*. <https://www.cio.com/article/3037354/marketing/6-things-marketers-need-to-know-about-beacons.html>
- [16] Deliveroo. 2020. Deliveroo. [Webpage](#).
- [17] Patrick Dickinson, Gregorz Cielniak, Olivier Szymanczyk, and Mike Mannion. 2016. Indoor positioning of shoppers using a network of Bluetooth Low Energy beacons. In *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*. IEEE.
- [18] 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.
- [19] Yi Ding and Dongzhe Jiang. 2021. SmartLOC Source Code. <https://github.com/SmartLoc2021/SmartLOC>.

- [20] 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>
- [21] Yi Ding, Yu Yang, Wenchao Jiang, Yunhuai Liu, Tian He, and Desheng Zhang. 2021. Nationwide deployment and operation of a virtual arrival detection system in the wild. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 705–717.
- [22] DoorDash. 2020. DoorDash. [Webpage](#).
- [23] Eleme. 2020. Eleme. [Webpage](#).
- [24] Xiaomin Fang, Jizhou Huang, Fan Wang, Lingke Zeng, Haijin Liang, and Haifeng Wang. 2020. Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2697–2705.
- [25] Ramsey Faragher and Robert Harle. 2015. Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications* 33, 11 (2015), 2418–2428.
- [26] Fielding. 2020. Persistent Connections. [Webpage](#).
- [27] Gaode. 2020. amap by Gaode. [Webpage](#).
- [28] Mehdi Golestanian and Christian Poellabauer. 2016. Indoor localization using multi-range beaconing: poster. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 397–398.
- [29] Google. 2021. Android Developer, getAccuracy. [Webpage](#).
- [30] Google. 2021. Android Developer, GpsSatellite. [Webpage](#).
- [31] Google. 2021. Wi-Fi scanning overview. [Webpage](#).
- [32] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 273–278.
- [33] Andreas Haeberlen, Eliot Flannery, Andrew M Ladd, Algis Rudys, Dan S Wallach, and Lydia E Kavraki. 2004. Practical robust localization over large-scale 802.11 wireless networks. In *MobiCom*. ACM.
- [34] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.
- [35] Steven M Hernandez and Eyuphan Bulut. [n.d.]. Performing WiFi Sensing with Off-the-shelf Smartphones. ([n. d.]).
- [36] Minh Tu Hoang, Brosnan Yuen, Xiaodai Dong, Tao Lu, Robert Westendorp, and Kishore Reddy. 2019. Recurrent neural networks for accurate RSSI indoor localization. *IEEE Internet of Things Journal* 6, 6 (2019), 10639–10651.
- [37] Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kung Fu, Zheng Wang, Xiaohu Qie, and Jieping Ye. 2020. Heteta: heterogeneous information network embedding for estimating time of arrival. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2444–2454.
- [38] InstaCart. 2020. InstaCart. [Webpage](#).
- [39] Jason T Jacques and Per Ola Kristensson. 2019. Crowdworkeer economics in the Gig Economy. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–10.
- [40] Mr Nicholas Jewell. 2015. *Shopping malls and public space in modern China*. Ashgate Publishing, Ltd.
- [41] 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.
- [42] Christine Julien, Chenguang Liu, Amy L Murphy, and Gian Pietro Picco. 2017. Blend: practical continuous neighbor discovery for bluetooth low energy. In *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 105–116.
- [43] 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 *SenSys*. ACM.
- [44] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [45] Mikkel Baun Kjærgaard, Sourav Bhattacharya, Henrik Blunck, and Petteri Nurmi. 2011. Energy-efficient trajectory tracking for mobile devices. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*. 307–320.
- [46] Duc V Le, Nirvana Meratnia, and Paul JM Havinga. 2018. Unsupervised deep feature learning to reduce the collection of fingerprints for indoor localization using deep belief networks. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 1–7.
- [47] Liqun Li, Guobin Shen, Chunhui Zhao, Thomas Moscibroda, Jyh-Han Lin, and Feng Zhao. 2014. Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service. In *MobiCom*. ACM.
- [48] Lingkun Li, Pengjin Xie, and Jiliang Wang. 2018. RainbowLight: Design and Implementation of a Low Cost Ambient Light Positioning System. In *MobiCom*. ACM.
- [49] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. 2018. Multi-task representation learning for travel time estimation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1695–1704.

- [50] Song Liu and Tian He. 2017. SmartLight: Light-weight 3D Indoor Localization Using a Single LED Lamp. In *SenSys*. ACM.
- [51] Shengzhong Liu, Shuochao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Huajie Shao, and Tarek Abdelzaher. 2020. GlobalFusion: A Global Attentional Deep Learning Framework for Multisensor Information Fusion. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–27.
- [52] 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 *MobiCom*. ACM.
- [53] Alex T Mariakakis, Souvik Sen, Jeongkeun Lee, and Kyu-Han Kim. 2014. Sail: Single access point-based indoor localization. In *MobiSys*. ACM.
- [54] Paul Martin, Bo-Jhang Ho, Nicholas Grupen, Samuel Munoz, and Mani Srivastava. 2014. An ibeacon primer for indoor localization: demo abstract. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. ACM.
- [55] Paul Martin, Yasser Shoukry, Prashanth Swaminathan, Robin Wentao Ouyang, and Mani Srivastava. 2014. Social spring: encounter-based path refinement for indoor tracking systems. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. 156–159.
- [56] Meituan. 2020. Meituan. [Webpage](#).
- [57] Payam Nazemzadeh, Daniele Fontanelli, David Macii, and Luigi Palopoli. 2017. Indoor localization of mobile robots through QR code detection and dead reckoning data fusion. *IEEE/ASME Transactions on Mechatronics* 22, 6 (2017), 2588–2599.
- [58] Ahasanun Nessa, Bhagawat Adhikari, Fatima Hussain, and Xavier Fernando. 2020. A Survey of Machine Learning for Indoor Positioning. *IEEE Access* (2020).
- [59] Timothy J Pierson, Travis Peters, Ronald Peterson, and David Kotz. 2019. Proximity detection with single-antenna IoT devices. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–15.
- [60] Kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. 2018. Widar2. 0: Passive human tracking with a single wi-fi link. In *MobiSys*. ACM.
- [61] Jun-Wei Qiu and Yu-Chee Tseng. 2016. M2M encountering: Collaborative localization via instant inter-particle filter data fusion. *IEEE Sensors Journal* 16, 14 (2016), 5715–5724.
- [62] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. 2012. Zee: Zero-effort crowdsourcing for indoor localization. In *MobiCom*. ACM.
- [63] Niranjini Rajagopal, Sindhura Chayapathy, Bruno Sinopoli, and Anthony Rowe. 2016. Beacon placement for range-based indoor localization. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE.
- [64] 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.
- [65] Piotr Sapiezynski, Arkadiusz Stopczynski, David Kofoed Wind, Jure Leskovec, and Sune Lehmann. 2017. Inferring person-to-person proximity using wifi signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 1–20.
- [66] Hiroyuki Satoh, Makoto Suzuki, Yukou Tahiro, and Hiroyuki Morikawa. 2013. Ambient sound-based proximity detection with smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. 1–2.
- [67] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. 2017. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)* 42, 3 (2017), 1–21.
- [68] Ahmed Shokry, Moustafa Elhamshary, and Moustafa Youssef. 2017. The Tale of Two Localization Technologies: Enabling Accurate Low-Overhead WiFi-based Localization for Low-end Phones. In *SIGSPATIAL*. ACM.
- [69] 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.
- [70] Stephen P Tarzia, Peter A Dinda, Robert P Dick, and Gokhan Memik. 2011. Indoor localization without infrastructure using the acoustic background spectrum. In *MobiSys*. ACM, 155–168.
- [71] Uber. 2020. Uber Eat. [Webpage](#).
- [72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [73] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [74] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No need to war-drive: Unsupervised indoor localization. In *MobiSys*. ACM.
- [75] 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.
- [76] Ju Wang, Hongbo Jiang, Jie Xiong, Kyle Jamieson, Xiaojiang Chen, Dingyi Fang, and Binbin Xie. 2016. LiFS: low human-effort, device-free localization with fine-grained subcarrier information. In *MobiCom*. ACM.

- [77] Keyu Wang, Zheng Yang, Zimu Zhou, Yunhao Liu, and Lionel Ni. 2015. Ambient rendezvous: Energy-efficient neighbor discovery via acoustic sensing. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2704–2712.
- [78] Yu-Lin Wei, Chang-Jung Huang, Hsin-Mu Tsai, and Kate Ching-Ju Lin. 2017. CELL: Indoor Positioning Using Polarized Sweeping Light Beams. In *MobiSys*. ACM.
- [79] Wikipedia. 2020. Time-based One-Time Password. [Webpage](#).
- [80] Wikipedia. 2021. Extract, transform, load. [Webpage](#).
- [81] Wikipedia. 2021. Global Harbor (Shanghai). [Webpage](#).
- [82] Wikipedia. 2021. Standard error. [Webpage](#).
- [83] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).
- [84] Huatao Xu, Dong Wang, Run Zhao, and Qian Zhang. 2019. FaHo: deep learning enhanced holographic localization for RFID tags. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 351–363.
- [85] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).
- [86] Yu Yang, Yi Ding, Dengpan Yuan, Guang Wang, Xiaoyang Xie, Yunhui 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.
- [87] Zheng Yang, Chenshu Wu, and Yunhao Liu. 2012. Locating in fingerprint space: wireless indoor localization with little human intervention. In *MobiCom*. ACM.
- [88] Zuwei Yin, Chenshu Wu, Zheng Yang, and Yunhao Liu. 2017. Peer-to-peer indoor navigation using smartphones. *IEEE Journal on Selected Areas in Communications* 35, 5 (2017), 1141–1153.
- [89] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.
- [90] 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.
- [91] Chi Zhang and Xinyu Zhang. 2017. Pulsar: Towards ubiquitous visible light localization. In *MobiCom*. ACM.
- [92] Fusang Zhang, Kai Niu, Jie Xiong, Beihong Jin, Tao Gu, Yuhang Jiang, and Daqing Zhang. 2019. Towards a diffraction-based sensing approach on human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 1–25.
- [93] Huanle Zhang, Wan Du, Pengfei Zhou, Mo Li, and Prasant Mohapatra. 2016. DopEnc: acoustic-based encounter profiling using smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 294–307.
- [94] Pengfei Zhou, Yuanqing Zheng, Zhenjiang Li, Mo Li, and Guobin Shen. 2012. Iodetector: A generic service for indoor outdoor detection. In *SensSys*. ACM.
- [95] 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.
- [96] Shilin Zhu and Xinyu Zhang. 2017. Enabling High-Precision Visible Light Localization in Today’s Buildings. In *MobiSys*. ACM.