WePos: Weak-supervised Indoor Positioning with Unlabeled WiFi for On-demand Delivery

BAOSHEN GUO, Southeast University, China
WEIJIAN ZUO, Southeast University, China
SHUAI WANG*, Southeast University, China
WENJUN LYU, Rutgers University, United States
ZHIQING HONG, Rutgers University, United States
YI DING, University of Minnesota, United States
TIAN HE, Southeast University, China
DESHENG ZHANG, Rutgers University, United States

54

On-demand delivery is an emerging business in recent years where accurate indoor locations of Gig couriers play an important role in the order dispatch and delivery process. To cater to this need, WiFi-based indoor positioning methods have become an alternative method for on-demand delivery thanks to extensive WiFi deployment in the indoor environment. Existing WiFi-based indoor localization and positioning methods are not suitable for large-scale on-demand delivery scenarios due to high costs (e.g., high labor cost to collect fingerprints) and limited coverage due to limited labeled data. In this work, we explore (i) massive crowdsourced WiFi data collecting from wearable or mobile devices of couriers with little extra effort and (ii) natural manual reports data in the delivery process as two opportunities to perform merchant-level indoor positioning in a weak-supervised manner. Specifically, we proposed *WePos*, an end-to-end weak-supervised-based merchant-level positioning framework, which consists of the following three parts: (i) a Bidirectional Encoder Representations from Transformers (BERT) based pre-training module to learn latent embeddings of WiFi access points, (ii) a contrastive label self-generate module to produce pseudos for WiFi scanning lists by matching similarity embedding clustering results and couriers' reporting behaviors. (iii) a deep neural network-based classifier to fine-tune the whole training process and conduct online merchant-level position inference. To evaluate the performance of our system, we conduct extensive experiments in both a large-scale public crowdsourcing dataset with over 50 GB of WiFi signal records and a real-world WiFi crowdsourced dataset collected from Eleme, (i.e., one of the largest on-demand delivery platforms in China) in four multi-floor malls in Shanghai. Experimental results show that *WePos* outperforms state-of-the-art baselines in the merchant-level positioning performance, offer up to 91.4% in positioning accuracy.

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

Additional Key Words and Phrases: Merchant-level Indoor Positioning, WiFi, Weak-supervised Learning

*Corresponding author.

Authors' addresses: Baoshen Guo, guobaoshen@seu.edu.cn, Southeast University, Nanjing, Jiangsu, China; Weijian Zuo, wj_zuo@seu.edu.cn, Southeast University, Nanjing, Jiangsu, China; Shuai Wang, shuaiwang@seu.edu.cn, Southeast University, Nanjing, Jiangsu, China; Wenjun Lyu, wenjun.lyu@rutgers.edu, Rutgers University, NJ, United States; Zhiqing Hong, Rutgers University, NJ, United States, zhiqing.hong@rutgers.edu; Yi Ding, University of Minnesota, Minneapolis, Minnesota, United States, dingx447@umn.edu; Tian He, tianhe@seu.edu.cn, Southeast University, Nanjing, Jiangsu, China; Desheng Zhang, Rutgers University, New Jersey, United States, desheng.zhang@cs.rutgers.edu.

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.
© 2022 Association for Computing Machinery.
2474-9567/2022/6-ART54 $15.00
https://doi.org/10.1145/3534574

ACM Reference Format:
Baoshen Guo, Weijian Zuo, Shuai Wang, Wenjun Lyu, Zhiqing Hong, Yi Ding, Tian He, and Desheng Zhang. 2022. WePos: Weak-supervised Indoor Positioning with Unlabeled WiFi for On-demand Delivery. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2, Article 54 (June 2022), 25 pages. https://doi.org/10.1145/3534574

1 INTRODUCTION

On-demand delivery services (e.g., Instacart [19], Uber Eats [36], DoorDash [11], and Eleme [31]) have become a popular choice for people to order food, medicine, and groceries online, especially after the impact of COVID-19, where Gig couriers [40] usually pick up orders from merchants in the indoor environment and then deliver them to customers in a short time (e.g., 30 mins for food [31] or 1 hour for grocery [9]). The rapid development of on-demand delivery services has driven the growing demand for merchant-level indoor positioning (i.e., detecting whether a courier is in the merchant and which merchant the courier is in) because real-time merchant-level indoor positions of couriers play an important role in the order dispatch process and delivery time estimation in multi-floor buildings of big cities, e.g., New York and Shanghai.

Indoor positioning has been well studied. In infrastructure-based systems, target devices are localized based on additional infrastructures, such as WiFi access points (WiFi APs with the known locations or mapping) [14, 20, 21, 28, 45], Bluetooth beacon [9], and RFID [23, 43]. In infrastructure-free systems, locations are estimated by existing infrastructure (e.g., WiFi, magnetic, etc) without deploying additional hardware [6, 42, 44]. For example, fingerprinting-based solutions collect WiFi fingerprints of specific locations with professional devices or crowdsourcing to build and update the fingerprint database, which needs extensive labor costs. Some studies [1, 15, 25] apply deep learning technologies to learn the mapping between indoor locations and WiFi signal signatures in a supervised or semi-supervised way, which needs labeled WiFi data (or at least a small amount of labeled data) and is hard to deploy in uncontrolled on-demand delivery systems.

In real-world on-demand delivery systems, to achieve merchant-level positioning of couriers given merchant labeled maps, existing state-of-practice positioning solutions consist of (i) anchor-based detection methods [9, 10], which are limited in terms of scalability and cost due to deployment and maintenance; (ii) the manual report-based method [44], which is not generally reliable due to couriers' inconsistent behaviors and complex real-time delivery situations. Compared to these solutions, WiFi-based crowdsourced indoor positioning has become an alternative method for on-demand delivery because WiFi is a more ubiquitous and low-cost infrastructure with no need of deployment or active participation in on-demand delivery.

To build a large-scale and low-cost WiFi-based merchant-level indoor positioning system in the uncontrolled environment (e.g., multi-floor malls) of on-demand delivery systems, we explore the following two opportunities. *(i) Little extra-effort WiFi crowdsensing data during the regular delivery process:* Thanks to the extensive deployment scale of existing WiFi APs in multi-floor malls and massive indoor pickup behaviors of couriers in on-demand delivery services, we collect massive WiFi data from wearable or mobile devices of couriers under their consent with a passive scanning strategy, which is little extra effort because the WiFi collection process is without additional participating activities from couriers, i.e., the couriers perform the regular delivery process. *(ii) Manual reporting order delivery process:* In on-demand delivery, the delivery process of the order is reported by couriers to inform the platform of real-time order status and ensure timely delivery as well as customers' experience. In this work, we leverage unlabeled WiFi data and adopt couriers' reporting behaviors as "semantic" anchors to end-to-end generate in-merchant or out-merchant labels for WiFi scanning lists and establish the merchant-labeled mapping between unlabeled WiFi scanning records and indoor merchants, thus achieving merchant-level positioning in on-demand delivery.

However, it is still challenging to perform merchant-level indoor positioning for on-demand delivery given the above two opportunities because: (i) The nature of uncontrolled commercial indoor environment: considering the large-scale and low-cost need for indoor positioning in on-demand delivery, the locations of WiFi APs are unknown and the pre-collected mapping is not applicable due to labor costs. The missing, noise and the impact of mobile WiFi in the real-world commercial indoor environment make accurate mapping very challenging to obtain. (ii) Uncertain Reporting Behaviors: The manual order progress reporting is unreliable with a significant number of early or late progress reports, especially in the arrival-merchants time due to the overdue penalty. Hence it is inapplicable to utilize manual report arriving and departure times to infer the actual indoor status (as pseudo labels) directly.

To address the challenges, we design and implement *WePos*, an end-to-end weak-supervised representation learning-based merchant-level positioning system. The key insights of *WePos* include: (i) The WiFi scanning data have rich spatial-temporal contextual relations because the scanning data are highly related to the position and continuous indoor movement of couriers duo to the regular order pick up process, which motivates us to extract the contextual features (i.e., the relative position of WiFi APs and position-related features of WiFi lists) with representation learning; (ii) The WiFi data scanned during in-merchant status share high similarity because of the staying behaviors regularity (couriers spend several minutes inside merchants with less movement waiting for food preparation), which motivates us to generate weak-supervised mapping through inaccurate reporting behaviors. Specifically, in *WePos*, (1) we design a BERT-based pre-train model to extract the spatial-temporal contextual relations and generate the latent representation of different WiFi APs and scanning lists leveraging massive unlabeled crowdsourced WiFi data; (2) we propose a SimCSE-based module to generate in/out merchant pseudo label by enhancing the similarity due to staying behaviors regularity; (3) a lightweight deep classifier is designed to achieve online position inference. Our contributions are as follows:

- Firstly, to the best of our knowledge, we conduct the first data-driven study on weak-supervised indoor localization for on-demand delivery with little extra effort crowdsourced and unlabeled WiFi data. The massive crowdsourced WiFi data collected under couriers' consent give us the opportunity to identify latent features of unlabeled WiFi lists and the manual reporting order process provides the weak-supervised information. To protect privacy, all identifiable IDs, such as courier ID, merchant ID, and the SSID and BSSID of WiFi APs are replaced by serial identifiers. In addition, we do not record or track individual WiFi APs but focus on collecting aggregate WiFi scanning lists and signal strength.
- Secondly, we design *WePos*, an end-to-end weak-supervised merchant-level positioning framework, which consists of three parts: (i) pre-training: a transformer-based module to learn latent embeddings of WiFi access points and enhance the impact of possible anchor WiFi APs and reduce the negative impact of mobile WiFi; (ii) pseudo-label-generation: a contrastive label self-generate module to form pseudos for WiFi scanning lists by matching between similarity embedding clustering results and couriers' reporting behaviors; (iii) a deep learning-based indoor positioning module to fine-tune the pre-train model and conduct online merchant-level position inference.
- Thirdly, we evaluate the performance of *WePos* with both a large-scale public crowdsourcing dataset with over 50 GB of WiFi signal records and a real-world WiFi crowdsourced dataset collected from Eleme (i.e., one of the largest on-demand delivery platforms in China) involving four large multi-floor malls. Experimental results show that *WePos* outperforms state-of-the-art baselines in the merchant-level positioning performance and achieve up to 91.4% in positioning accuracy. For the long-term robustness, we conduct experiments with six-month data and the results show that the positioning accuracy of *WePos* maintains 86% after four months, which outperforms other baselines. We also show that *WePos* achieves better performance in arrival time estimation compared with the state-of-the-art, TransLoc.

2 MOTIVATION

2.1 Background for Merchant-level Indoor Positioning in On-demand Delivery

In on-demand delivery services, couriers pick up orders and deliver them with strict time constraints (e.g., 30 minutes). Indicated by an on-demand delivery platform [31], couriers spend around 1/3 of total delivery time in multi-floor indoor environments detected by IODector [50]. Thus indoor positioning of couriers plays an important role in indoor status detection (i.e., arrival and departure indoor merchants), indoor navigation, and order dispatching given the timely delivery nature of on-demand services.

2.1.1 Merchant-level Indoor Positioning in On-demand Delivery. Different from many existing indoor localization scenarios that need to obtain the absolute locations, the indoor positioning of couriers in on-demand delivery allows the merchant-level positioning (i.e., detecting which merchants that couriers are in) because most of the indoor time couriers are staying in merchants waiting for meal preparation and the time spent in indoor route only accounts for a small part. We define the in-merchant duration as the duration between the actual arrival and the departure time of couriers. The out-merchant status means that the courier is not in the merchant (either in other merchants or on the way). Given the real-time WiFi sensing data, the real-time order-waybill data, and the manual reporting process records, the output is whether the courier is in merchants and which merchant the courier is in, and then detect the arrival and departure time through continuous in-merchant detection.

2.1.2 Existing Real-world Indoor Positioning Systems in On-demand Delivery. In real-world on-demand delivery services, existing state-of-the-practice merchant-level indoor positioning methods include GPS-based, beacon anchor-based, and manual-reported based, as shown in Fig. 1. *GPS-based methods* perform indoor positioning poorly because blocked by the buildings, especially in multi-floor buildings. In Beacon-based indoor positioning methods, a case is that Alibaba local services [31] builds a city-wide beacon system (i.e., aBeacon [9]) by deploying the BLE to merchants. *beacon-based positioning methods* have high accuracy but with high maintenance costs. The aBeacon [9] system finally retires because the battery is running and the hardware is broken. The recent manual-report-based method (TransLoc [44]) proposes an arrival time prediction model and a merchant-level localization model under the assumption that couriers' reporting behaviors are consistent for outside merchants and inside merchants. However, couriers' reporting behaviors are not generally reliable and prone to error due to couriers' early reports for avoiding overdue responsibility or forgetting to report.

	Coverage	Accuracy	Maintenance Cost
Manual report	High	Low	Low
GPS based	High	Low	Low
Beacon based	Low	High	High
Wi-Fi based	High	High	Low

Fig. 1. Comparison of State-of-Practice

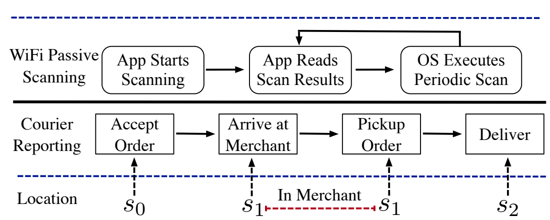

Fig. 2. WiFi scanning and manual reporting process

2.2 Crowdsourced Collection Process of WiFi Scanning Data in On-demand Delivery

Thanks to the widespread deployment of WiFi access points of merchants in indoor environments, especially in multi-floor buildings (e.g., malls), WiFi-based methods have the potential to achieve real-time merchant-level indoor positioning for on-demand delivery.

2.2.1 WiFi Scanning Data. As is shown in Table. 1, each collected WiFi record has five fields, including the courier's ID, timestamp, the service set identifier (SSID), the basic service set identifier (BSSID), and the received signal strength indication (RSSI). To uniquely identify each WiFi AP, we use the "WiFi ID" to represent the tuple of SSID and BSSID (i.e., <SSID, BSSID>) and compute the SHA256 hash value for each "WiFi ID".

Table 1. WiFi scanning records

Field	Value
Courier ID	C001
Scanning Time	1610251200
SSID	'Starbucks'
BSSID	11:22:9f:33:7f:44
RSSI	-60 dB

2.2.2 WiFi Data Collection Process. As shown in Fig. 2, the WiFi data collection process includes two parts: couriers' manual reporting process and WiFi passive scanning process. The couriers' reporting process and WiFi passive scanning process are separated to avoid affecting the delivery tasks of couriers considering strict deadlines of orders (i.e., 30 minutes). The WiFi scanning data are collected by couriers' smartphones using the passive WiFi scanning strategy without additional participating activities from couriers, i.e., the couriers just perform regular delivery and report order process records on the smartphones.

2.2.3 Passive Scanning Strategy. In the WiFi scanning process, to reduce energy consumption, we design a passive WiFi scanning strategy. The app that installed couriers' smartphones reads WiFi scanning results from the OS periodically rather than active scanning. The read action consumes little energy and hence could be at a high frequency (e.g., 1 second). After receiving a read request, the OS returns the WiFi list to the app and uploads it to the server of the on-demand platform.

2.3 Challenges

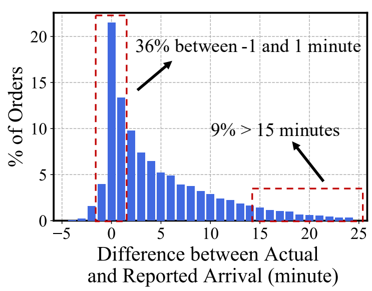

Fig. 3. The difference of Arrive Time

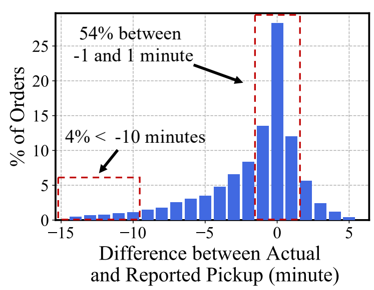

Fig. 4. The difference of Pickup time

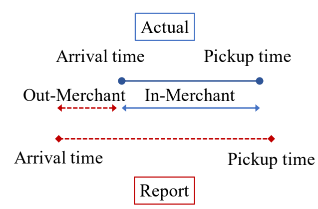

Fig. 5. A case of the difference between real indoor status and reported indoor status of a courier

2.3.1 Inaccurate Manual Reporting Behaviors. Although manual reporting records of the delivery process give us the contextual information to infer couriers' locations and detect the in/out merchants status, it is still inapplicable to build the mapping between collected WiFi data and the merchant-level positions because manual reporting behaviors are sometimes uncertain and unreliable compared with the ground truth. Particularly, from Fig. 3, we find that 64% of the orders have issues of early or late reporting more than 1 minute and about 9% of orders even differ by more than 15 minutes. In Fig. 4, we find that 46% of the delivery process have early or late reporting more than 1 minute and 4% of the orders even differ by more than 10 minutes. Fig. 5 illustrates a case of the difference between real indoor status and reported indoor status. We find that if we utilize reporting behaviors to infer the in-merchant status or infer couriers' locations, 6 minutes will be misjudged as in-store, which has a negative impact on the following order assignment to this courier. When a new order occurs, suppose the platform computes the distance and travel time according to the inaccurate real-time locations, the order may be dispatched to an inappropriate courier (e.g., further or more likely to be overdue) due to the distance and time cost error. The impact of merchant-level positioning is also discussed in Section 6.1.

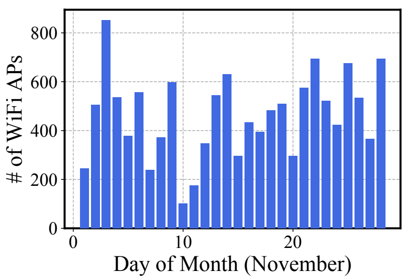

Fig. 6. Number of WiFi APs

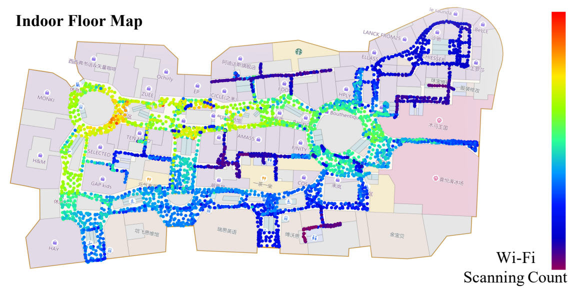

Fig. 7. WiFi scanning count in one floor of a mall

2.3.2 Uncertain and Label-lacked Crowdsourced WiFi Records. Thanks to the crowdsourcing nature of on-demand delivery, we collect WiFi crowdsourcing data with little extra cost by couriers in the delivery process without extra worker participation. However, we find that using these WiFi records is not straightforward because of data uncertainty, unbalanced distribution, and label-lacked. Fig. 6 shows the number of scanning WiFi APs collected during the pickup process of one merchant daily. We find that the number is highly dynamic with the maximum number over 800 and the minimum number less than 200. Fig. 7 shows the WiFi scanning records distributions are unbalanced. Specifically, Unlike existing public crowdsourced WiFi datasets that have coordinates or location labels, crowdsourced WiFi records in on-demand delivery are label-lacked due to collecting in the wild, which introduces a great challenge to map the WiFi signals to real-world locations. Most importantly, indoor maps are not available because delivery services involve thousands of malls in different cities in our platform, which are expensive to obtain all indoor maps. The mobile WiFi of pedestrians in malls also has a negative impact on merchant-level positioning.

2.4 Design Insights of Weak-supervised Merchant-level Positioning

In summary, the challenges of achieving large-scale low-cost WiFi-based merchant-level positioning in on-demand delivery include: (i) the crowdsourced WiFi scanning records are context-lacked (including the position of WiFi APs and the pre-collected mapping) and label-lacked in real-world on-demand delivery systems. (ii) the idea of using manual reporting records to map WiFi lists to merchants is not applicable because manual reporting records contain massive reporting errors due to human behaviors uncertainty. We utilize the following two insights to address the above two challenges.

2.4.1 Spatial-Temporal Contextual Relations of Unlabeled WiFi Scanning Records. In our work, we aim to realize merchant-level positioning considering the required accuracy of on-demand delivery. The WiFi scanning data have rich spatial-temporal contextual relations because the scanning data are highly related to the position and continuous indoor movement of couriers due to the regular order pick-up process. In indoor environments, especially in multi-floor buildings, the distribution of merchants is relatively close. In one delivery process, couriers usually arrive and pick up orders from multiple merchants one by one in the same mall. Couriers' indoor pick-up processes and indoor routes are usually patterned considering her/his experience and familiarity. To extract the contextual features (i.e., the relative position of WiFi APs and position-related features of WiFi lists), we design a BERT-based model (i.e., a pre-training representation learning model) to extract features leveraging the massive unlabeled WiFi crowdsourced data in our scenarios.

2.4.2 *Weak Labels: Staying Behaviors Regularity.*

Compared to previous learning-based indoor localization works with available labels, it is difficult to obtain the actual location of the inside-merchants or to detect in-merchants status based on the existing platform considering the inaccurate indoor GPS signal and the inaccurate manual reports. However, we find that most couriers usually spend several minutes inside merchants waiting for food preparation, which results in WiFi scanning data during one inside-merchant process sharing high similarity. According to this insight, we propose a SimCSE-based module to generate in/out merchant pseudo labels by enhancing the similarity due to staying behaviors regularity to avoid the negative of using inaccurate manual reporting records to map WiFi lists to merchants directly.

3 PROBLEM FORMULATION AND SYSTEM OVERVIEW

In this section, we first detail the problem formulation of merchant-level indoor positioning for on-demand delivery with crowdsourced WiFi data, then we introduce the system overview of the proposed *WePos*.

Table 2. The Main Notations and Definitions

Terms	Definition
ID_p	ID_p is the <SSID, BSSID> tuple of WiFi AP p
v_p, v_l	the RSSI value and level of WiFi AP p
$\boldsymbol{r}_i$	one WiFi scanning Record i
$\boldsymbol{T}_r$	**<ID-RSSI> token** (ID_p, v_l) of scanning Record $\boldsymbol{r}_t$
$\mathcal{R}_t$	Wi-Fi scanning list at timestamp t
c_{id}, o_{id}, m_{id}	courier id, order id, merchant id
x, h	WiFi training instance and its embeddings of $\mathcal{R}_t$
$sim(h_1, h_2)$	the similarity between WiFi embedding $h1$ and $h2$
τ	a temperature hyperparameter

3.1 Problem Formulation

For ease of the following presentation, we describe key definitions used in the proposed method as follows and list the relevant notations in Table 2.

Definition 1: WiFi ID-RSSI Token. According to Table. 1, we first define each WiFi scanning record with a tuple $r(c, t, o, ID, v)$, which means the collected WiFi scanning records by courier c at time t during she/he delivers order o, ID represents the ssid and bssid of WiFi APs, and v is the sensing RSSI value (e.g, -70). To uniquely identify each WiFi AP, we use the "WiFi ID" to represent the SSID+BSSID tuple (i.e., <SSID, BSSID>) and compute the SHA256 hash for each "WiFi ID". Then we define the ID-RSSI token, which is the combination of scanning "WiFi ID" and the scanning RSSI value. To reduce the total token number, we discretize the continuous RSSI values into 10 intervals (range from -100 to 0, e.g., $[-100, -90)$ as level 1) and obtain the final "ID-RSSI" token id. And we utilize T_i to represent the i-th "ID-RSSI" token.

Definition 2: WiFi Scanning List. A WiFi scanning list is defined as a list of WiFi Scanning Records at the same timestamps and denote by $R = \{r^1, ..., r^i, ...\}$, r^i means the scanned WiFi record from WiFi ap^i.

Definition 3: WiFi ID-RSSI Token Vocabulary. Inspired by the emerging BERT-based self-supervised pre-train techniques in natural language processing, we construct the WiFi ID-RSSI Token Vocabulary (similar to the word vocabulary in natural language processing) by collecting "ID-RSSI" tokens that have been scanned more than 5 times in two-month WiFi scanning data in the pre-train process. As is shown on the left of Fig. 8, we construct the WiFi ID-RSSI Token Vocabulary by assigning a unique index (i.e., ID) to each ID-RSSI Token.

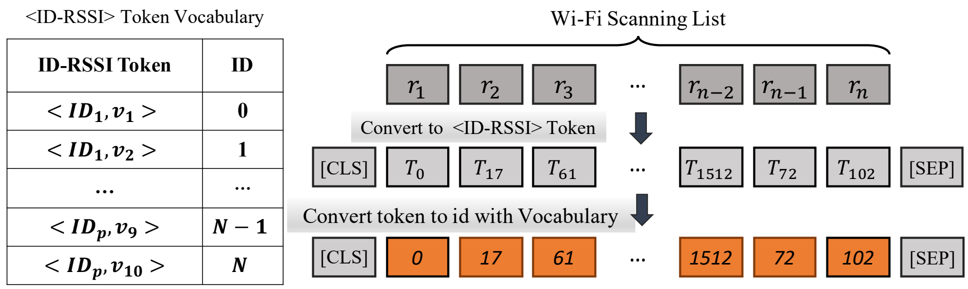

Fig. 8. An example of WiFi ID-RSSI tokenization.

Definition 4: ID-RSSI Tokenization. Fig. 8 gives an example of the ID-RSSI tokenization process. At the scanning time t, we obtain a WiFi scanning list R_t, which includes multiple WiFi records. We first rank the WiFi scanning records in descending order according to their WiFi RSSI strength. Then we construct the ID-RSSI token T_r by combining WiFi ID and the RSSI value. Then we encode each ID-RSSI token T_r in the scanning list with the corresponding index (i.e., ID) in the WiFi ID-RSSI Token Vocabulary. After the ID-RSSI tokenization step, all tokens can be converted into their corresponding IDs.

3.2 System Overview

As is shown in Fig. 9, the proposed model first conducts pre-processing of the raw sensing WiFi records and then conducts the end-to-end indoor localization through *WePos*, which consists of three components: (i) Pre-training module of WiFi Embedding representation, (ii) Label-self-generation: mobility aware WiFi similarity clustering and matching, and (iii) Fine-tuning: Deep in-merchant detection classifier.

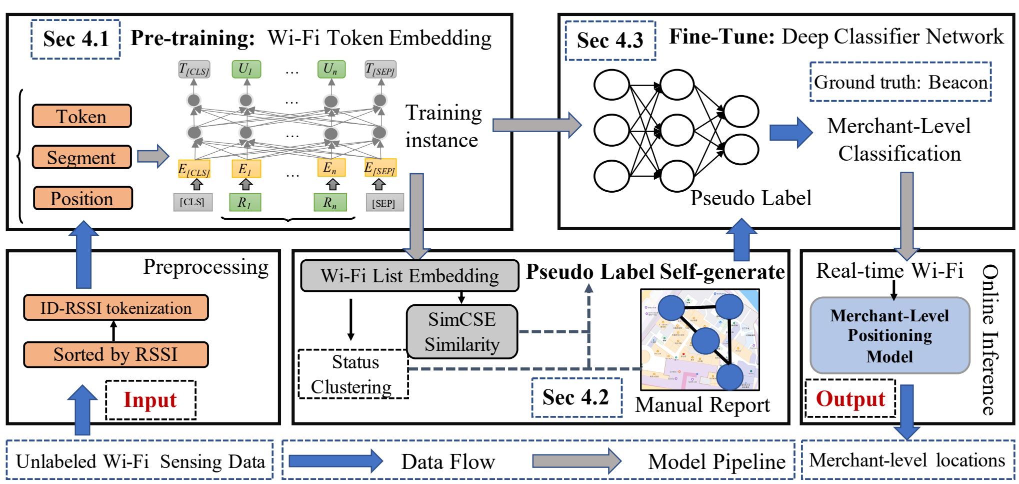

Fig. 9. System Framework of WePos

(1) Pre-processing. In the pre-processing, we first improve the quality of the raw sensing data and process the WiFi data into the format that our model needs. For most raw data collected from the infrastructures, they suffer from several issues that make direct investigation ineffective, e.g., missing values, inconsistent data format and repetitive data fields in separate tables. Further processing is required to better understand the data, such as WiFi sensing data segmentation based on order delivery process, outliers filtering, etc. For each unlabeled WiFi scanning list R, we first sort the WiFi scanning records $r(u, t, o, ID, v)$ by their RSSI value v. Then we combine ID-RSSI of WiFi access points as a token and conduct ID-RSSI tokenization to obtain the model input.

(2) WiFi-BERT: Pre-train module of WiFi Embedding representation. In this module, we extract the latent representation of each WiFi access point in our scenario for further specific tasks (e.g., merchant-level positioning). **Intuitions of WiFi-BERT:** Due to the lack of labeled data, it is not easy to extract spatio-temporal feature representations and location-related information from the raw sensing signal. Bidirectional Encoder Representations from Transformers (BERT) [8] is one effective self-supervised learning model for Natural Language Processing (NLP), which makes use of large amounts of unlabeled data to learn contextual relations in text data and accordingly generate effective embeddings for each word with self-supervised learning. In on-demand delivery, the WiFi scanning process during couriers' delivery tasks collects massive WiFi lists but suffers from labeled scarcity. Inspired by BERT-based application in natural language processing, we utilize BERT-based pre-train model to extract general features or representations from raw unlabeled WiFi data, which is detailed in Section 4.1.

(3) Label-self-generation: mobility aware WiFi labeling through contrastive similarity learning. As is illustrated in Fig. 9, The label-self-generate module aims to generate pseudo labels (e.g., in-merchant or out-merchant status) through WiFi scanning lists similarity and couriers' mobility graph extracted from the order manual reporting process. **Intuitions of using SimCSE for label generation:** WiFi-BERT outputs both the embedding of each WiFi scanning token and the embedding of the WiFi scanning list. While embeddings of WiFi scanning tokens from WiFi-BERT represent the relative locations of WiFi APs better at the token level, the embeddings of WiFi scanning lists generated by WiFi-BERT are not enough for on-demand positioning scenarios. In the merchant-level positioning task, we need not only token-level similarity but also the similarity of the whole WiFi-scanning lists. To cater to this need, we utilize unsupervised SimCSE [12] to enhance sentence-level representation ability because SimCSE can learn effective representation by pulling semantically close neighbors together and pushing apart non-neighbors [13], which is useful to separate positive and negative clusters of couriers' indoor status and is detailed in Section 4.2.

(4) Fine-tuning: Deep in-merchant detection classifier. After pre-training and label-self-generation, we can only obtain pseudo labels of the unlabeled WiFi lists as the training set in the training stage. In the testing and online inference stage, to generate detection results of each new arriving WiFi scanning list, and achieve end-to-end real-time merchant-level positioning, we design a deep neural network (DNN) based in-merchant detection classifier, which takes both latent representations from WiFi-BERT pre-training module and pseudo labels from the self-label-generation model as the input, and output the result of merchant-level indoor positioning of on-demand delivery by mapping the pseudo labels to actual merchant-level locations.

4 WEAK-SUPERVISED MERCHANT-LEVEL INDOOR POSITIONING DESIGN

In this section, we first introduce the pre-train module to obtain the latent embedding of massive unlabeled WiFi scanning records; then we show the details of the label self-generate module to form pseudo labels with the weak-supervised manual reporting process; lastly, we propose a deep in-merchant detection classifier to fine-tune [2] the pre-train model with latent embeddings and in-merchant or out-merchant pseudo labels.

4.1 WiFi-BERT: Pre-train Module of WiFi Embedding Representation

To extract latent representation from massive unlabeled crowdsourced WiFi sensing data, we design WiFi-BERT, a BERT-based model [8, 35] to obtain token-level embeddings of WiFi scanning records r and sentence-level embeddings of WiFi scanning list R.

4.1.1 WiFi-BERT Input Representation. To make the transformer-based WiFi-BERT understands the positional information of given WiFi scanning lists and the relative positional relationships of different WiFi tokens, we conduct additional representation embeddings for the input WiFi scanning list, which consists of

- **Token embeddings:** Token embeddings are the pre-trained embeddings for each WiFi ID-RSSI token. After WiFi ID-RSSI tokenization, each ID-RSSI token is fed into the pre-train model to learn a fix-dimensional embedding vector to represent its semantic position.
- **Position embeddings:** In ID-RSSI tokenization, we rank the WiFi scanning records of the WiFi scanning list in descending order according to their RSSI strength. So the order of each ID-RSSI token in a WiFi list can reflect the distance to the WiFi Ap. Position embeddings are high-dimension encodings of the order and position of each token in a WiFi scanning list. All position embeddings are trainable variables and WiFi-BERT learns the positional embeddings during the pre-training process to better represent the order and position of each token under different contexts.
- **Segment embeddings:** In WiFi-BERT, segment embeddings of each token are used when we feed two WiFi scanning lists into BERT to calculate their similarity and classify them (similar to the text classification and sentence similarity computation in NLP). Segment embeddings of tokens are the identifier of two WiFi scanning lists (e.g., 0 for the first WiFi scanning list and 1 for another WiFi scanning list).

4.1.2 Multi-Head Attention Layers. We adopt the attention-based transformer to take WiFi ID-RSSI token embedding as inputs to infer the relationships of different tokens under different semantic contexts and the dependency of different tokens. We first introduce the single head attention:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) = softmax(\frac{QW_i^Q(KW_i^K)^T}{\sqrt{d_k}})VW_i^V \tag{1}$$

where Q, K, V are the query, key and value; W_i^Q, W_i^K, W_i^V are the corresponding self attention weights; $\sqrt{d_k}$ is the dimension of the key vector k and query vector q.

Then, we introduce the multi-head attention in our WiFi-BERT design. Intuitively, multiple attention heads allow for attending to parts of the WiFi scanning Lists R differently (e.g. longer-term dependencies versus shorter-term dependencies or the importance of different WiFi tokens in a different part of the WiFi scanning List).

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{2}$$

where $head_h$ is the single head attention, W^O is the weight matrix, the $MultiHead(Q, K, V)$ executes $head_h$ in parallel against the same input vectors and then concatenates the output vectors coming from the different heads together into an output vector. The multi-head attention layers enable the model to jointly attend to information from different representation subspaces at different WiFi positions.

4.1.3 Pre-training WiFi-BERT with Masked Language Modeling. In NLP tasks, masked Language Modeling (LM) is a modeling method to predict that word referring to the entire sequence with no specifically selected direction. In WiFi-BERT, our setting is to predict the next WiFi ID-RSSI token when some other ID-RSSI tokens in the same WiFi scanning list are given. For one WiFi scanning list, we first mask 15% of WiFi tokens and the masked token is converted into other tokens in the following setting: (1) 80% of masked words are converted into [MASK] token; (2) 10% of masked words are converted into other words randomly. (3) 10% of masked words are left as they are. The aim of masked language modeling is to predict the masked tokens and fill them correctly. During training, the pre-training model predicts the masked token more and more accurately, which indicates the pre-training model can make a better representation of WiFi ID-RSSI.

After pre-training with the Masked LM Task, WiFi-BERT obtains a good representation of each WiFi ID-RSSI token and the embeddings of WiFi scanning Lists. The output embeddings of WiFi-BERT represent the relative

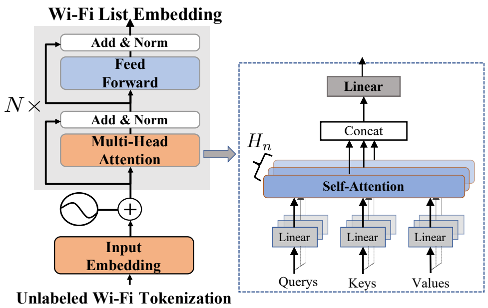

Fig. 10. Details of WiFi-BERT

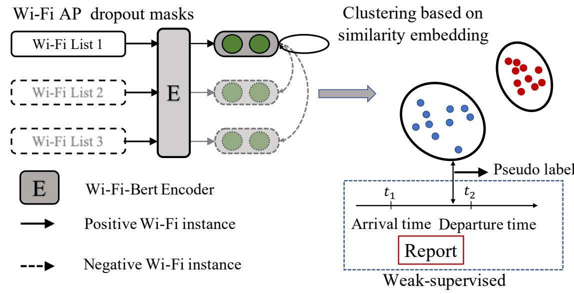

Fig. 11. Contrastive Label Self-generate Design

position of each WiFi ID-RSSI token in signal space. The more similar the two token embeddings are in the high dimension signal space, the closer the two WiFi access points are in the real-world indoor environment.

4.2 Contrastive Label Self-Generate with Similarity Graph Matching

4.2.1 Similarity Aware Contrastive Embedding based on SimCSE. We first construct the positive WiFi training instances for the SimCSE module. In contrastive learning, to learn the effective representation of WiFi training instances by pulling semantically close neighbors together and pushing apart non-neighbors, we first construct the positive WiFi training instances (x_i, x_i^+), which is a critical question in contrastive learning. As is illustrated in Fig. 11, we construct the positive WiFi training instances with only standard dropout used as noise. Specifically, In our contrastive similarity enhancing process, we adopt the positive instances construction methods used in SimCSE [12], which takes a collection of WiFi training instance $\{x_i\}_{i=1}^m$ and use $x_i^+ = x_i$. The intuition of the standard dropout method is that we do not change the original WiFi training instance to achieve data augmentation but simply feed the same input to the encoder twice by applying different dropout masks (z, z'), where z is just the standard dropout mask in Transformers and we do not add any additional dropout. The twice dropout noise can output different embeddings when we input the same WiFi training instances. Because we want similar WiFi scanning records to have more similar sentence embeddings, we set the training objective as follows:

$$\ell_i = -\log \frac{e^{sim(h_i, h_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(h_i, h_j^+)/\tau}}, \tag{3}$$

where N is the number of a mini-batch with N input WiFi training instances, h_i is the embeddings of WiFi scanning list i obtained from the WiFi-BERT encoder, and τ is the temperature hyper-parameter. $sim(h_i, h_i^+$ is the cosine similarity of h_i and h_i^+, which is defined as follows:

$$sim(h_i, h_i^+) = \frac{h_i^T h_i^+}{||h_i|| \cdot ||h_i^+||}, \tag{4}$$

SimCSE for better semantic representation of WiFi scanning lists. As is shown in Fig. 11, after positive WiFi training instances construction, we encode input WiFi scanning lists using our pre-trained WiFi-BERT model (introduced in Sec. 4.1), and then fine-tune all the parameters using the contrastive learning with the objective Equation 3. The input of our fine-tuning framework consists of the positive pairs (the same WiFi scanning instances) and the negative pairs (different WiFi scanning instances). We view this approach as a minimal form of data augmentation: the positive pair takes exactly the same sentence, and their embeddings only differ in dropout masks, while the negative pairs' embeddings differ a lot. After fine-tuning with the contrastive learning objects in an unsupervised manner, our pre-training model WiFi-BERT not only outputs the precise token-level embeddings of each WiFi ID-RSSI token but also has a better representation of WiFi scanning lists, which are used for the following WiFi sensing lists clustering and similarity calculation.

4.2.2 Label-generation through Similarity Clustering and Manual Reporting Matching. After WiFi instances embeddings' similarity calculation, we obtain the latent representation of each WiFi scanning list and form pseudo labels (i.e., in-merchant status and out-merchant status) of each WiFi scanning list in the following two steps.

(1) Separate deep clustering-based embedding similarity. Although we have calculated the embeddings of the WiFi scanning lists through our similarity-based unsupervised fine-tuned WiFi-BERT, we still cannot obtain merchant-level classification labels for each WiFi scanning list. And using couriers' reporting in-merchant duration as the positive instances of actual in-merchant status is not accurate because the manual reporting records are not accurate including reporting arrival time early and forgetting reporting departure time. To generate in-merchant and out-merchant pseudo labels, (i) we first select WiFi scanning lists that are collected before and after the reporting in-merchant time in 5 minutes; (ii) for each pickup process, after data selection, we utilize the KMeans clustering algorithm to cluster WiFi scanning lists with the calculated embeddings h_i of each training instance i as input. After separately clustering, we obtain two clusters in each pickup process but cannot figure out which cluster represents the positive in-merchant status.

(2) Matching between embedding clustering and couriers' reporting behaviors. To further obtain pseudo labels, we utilize the following two steps: (i) the intra-similarity of each cluster is high after separate deep Clustering-based embedding similarity. Then, we collect multiple pickup process that belongs to the same merchant. Each single pickup process has two clusters and we calculate the inter-similarity between different pickup processes to select the positive in-merchant clustering, The intuition is that couriers usually stay in merchants waiting for food preparation, thus the intra-similarity of in-merchant clusters is high. The inter-similarity of multiple in-merchant clusters is also high than out-merchant clusters because WiFi scanning lists of out-merchant clusters are usually collected on the way to pick up foods. (ii) another issue is if a courier totally does not arrive at the shop or just arrives at another merchant during the reporting in-merchant time. For this case, we conduct a matching between embedding clustering and couriers' reporting behaviors. If one in-merchants cluster has low similarity with other clusters during other pickup processes, then we form the pseudo labels of the cluster as out-merchants. The details of the pseudo-label generation process are described in Algorithm 1.

ALGORITHM 1: Mobility aware Contrastive Label Self-generate Algorithm

Input : WiFi-BERT encoder, WiFi scanning lists
Output : Pseudo labels of each WiFi scanning lists

for each indoor pickup process i **do**
    Collect WiFi scanning lists including the reporting in-merchants time;
    Calculating WiFi embeddings through ***SimCSE***;
    Separate clustering**:** obtain embedding clusters of several in-merchant status and out-merchant status;
end
Intra-label merging**:** calculate the intra-similarity of different clusters in different pickup process including same merchants;
Construct reporting graph**:** collect manual reporting records and construct reporting graph with reporting in-merchant during;
Clustering and matching**:** similarity clustering and matching with manual reporting process;
Pseudo labels generation**:** form in-merchant or out-merchant pseudo labels for WiFi scanning Lists.

4.3 Deep Classification Fine-tuning: In-merchant Status Detection

After the pre-training with massive unlabeled WiFi sensing data and the application of the label self-generation module, we form each WiFi scanning instance with the pseudo label (i.e., in-merchant or out-merchant) for the training process. To automatically generate pseudo labels for new WiFi scanning lists and conduct real-time online merchant-level positioning, we propose a task-specific DNN-based classifier with supervised training, which takes the embeddings of WiFi training instances as the training set and the in-merchant and out-merchant pseudo labels as training labels. The DNN classifier has a two-layer fully connected neural network structure, and it takes the latent representations obtained from fine-tuned WiFi-BERT as inputs and the cross-entropy loss function of the classifier can be expressed as follows:

$$\ell_c = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{5}$$

where M is the number of classes, $y_{o,c}$ is binary indicator (0 or 1) if class label c is the correct classification for observation o, and $p_{o,c}$ is the predicted probability that observation o is of class c.

4.4 Iterative Training and Online Merchant-level Positioning

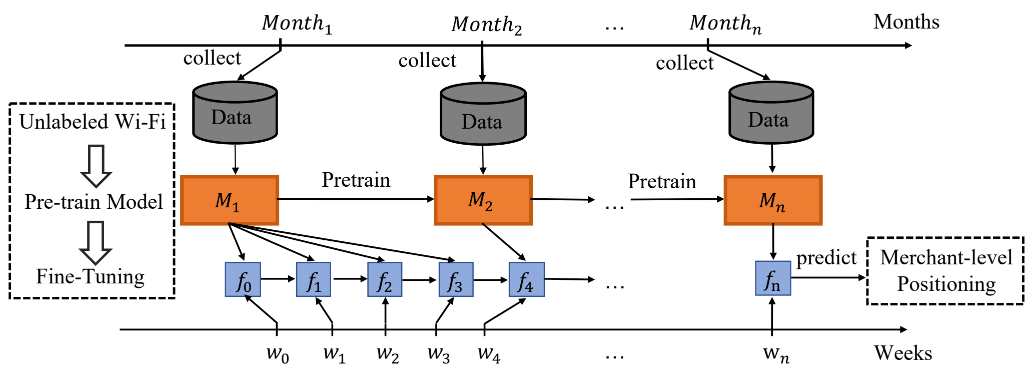

Fig. 12. Online merchant-level positioning and iterative learning

As is illustrated in Fig.12, we use two months of data for pre-training and one week for fine-tuning to ensure the model efficiency and accuracy. For the pre-training, we obtain the past two-month WiFi sensing data from the database at the start of each month to pre-train the model. For the fine-tuning, we set to automatically get data of the past one week from the database every Sunday to train the model and then use the trained model to conduct merchant-level positioning this week in the real-world delivery environment.

5 EVALUATION

In this section, we first give a description of two datasets used for performance evaluation and detail the data management and preprocessing process. Then we describe the evaluation methodology we used (i.e., metrics and baselines). Lastly, we show the evaluation results and ablation studies under different practical factor settings.

5.1 Datasets Description

We evaluate our design based on two crowdsensing data-sets: (i) WiFi crowdsourcing data in real-world On-demand delivery: the real-world dataset collected from an on-demand delivery platform in the commercial business including the WiFi sensing data, Beacon sensing data, and the manual reporting data we used; (ii) Public

WiFi crowdsourcing data: the public indoor positioning dataset consists of WiFi sensing data and corresponding GPS locations. We first introduce the on-demand sensing datasets.

Table 3. An example of on-demand indoor WiFi crowdsourcing dataset.

WiFi Data	Courier ID	SSID	BSSID	Detect time	RSSI
	Cxxx	Starbucks	46:DC:xx	1606698572	-70
Beacon Data	Courier ID	Beacon ID	Merchant ID	Detect time	RSSI
	Cxxx	Bxxx	Mxxx	1606698572	-60
Spatial Info.	Merchant Lat.	Merchant Lng.	Courier ID	Order ID	Merchant ID
	31.230156	121.462316	Cxxx	Oxxx	Mxxx
Temporal Info.	Order create time	Order accept time	Order arrival time	Order pickup time	Order delivered time
	2019/9/1 11:23	2019/9/1 11:26	2019/9/1 11:33	2019/9/1 11:40	2019/9/1 11:51

5.1.1 Details of Crowdsourced Data-sets in On-Demand Delivery. We first detail the crowdsourcing WiFi Dataset, the Beacon sensing data, and the manual reporting data in On-Demand Delivery, which is collected from Eleme [31], one of the largest city-wide delivery platforms in China.

(1) the WiFi sensing dataset of on-demand delivery consists of courier ID, SSID and BSSID of WiFi access points, detect time, and the RSSI value.
(2) the Beacon sensing dataset of on-demand delivery consists of courier ID, beacon ID, merchant ID, detect time, and the RSSI value. The beacon sensing data consists of merchant ID, thus we think a courier is in-merchant status when her/his device receives this merchant's beacon signal.
(3) The manual reporting data consists of temporal information and spatial information. The temporal information consists of order create time, courier accept order time, courier arrival time, courier pick-up order time, and courier delivery time and the spatial information consists of merchant latitude, merchant longitude, courier ID, order ID, and merchant ID. A detailed description of the dataset is shown in Table 3.

5.1.2 Multi-mall Experimental Settings. To evaluate the merchant-level positioning at multiple malls, we collect the crowdsourced WiFi sensing data at four large malls in Shanghai from one of the largest on-demand food delivery platforms in China. We choose these four malls because (i) all of them have massive instant delivery food orders and the courier can scan massive WiFi records at these malls. (ii) all of them have a wide deployment of virtual beacon [10] to detect couriers' physical arrival and departure status and time, which can provide ground truth for merchant-level positioning. Table 4 illustrates the details of multi-mall experimental settings, including the number of floor, number of merchants, number of merchants with virtual beacon deployment, and the duration of data collecting process.

5.2 Evaluation Methodology

5.2.1 Ground Truths and Experiment Settings. We utilize the beacon scanning data as the ground truth. If couriers' devices receive a merchant's beacon and the signal lasts at least 10s (to filter the instances that couriers pass by a merchant but scan its beacon AP), we add an in-merchant status flag. Then, we obtain the in-merchant and out-merchant ground truths by detecting whether merchants' beacons are scanning at the same detect timestamps of each WiFi scanning record.

Table 4. Details of experimental scenarios in four large-malls.

Malls	# of floors	# of participating merchants	# of w/ beacon	Duration
Mall A	3	37	21	6 months
Mall B	3	27	16	2 months
Mall C	2	24	17	2 months
Mall D	2	17	13	2 months

5.2.2 Parameters Settings. We implement our WiFi-BERT and the fine-tuned SimCSE based on the public transformer framework huggingface [16]. Both the proposed model and other baselines are implemented with Pytorch 1.8.0, transformer 4.8.1 in Python 3.8 environment and trained with two GeForce RTX3090, 128G RAM, Intel(R) Xeon(R) Silver 4215R CPU@3.20GHz. The pre-training configurations are as follows: the number of attention heads is 4, the number of hidden layers is 12, hidden layer size is 256, the hidden layers' activation function is gelu, the max position embedding is 512, the learning rate is 6e-5, the number of warm-up steps is 500, and the vocab size is 4938, which represents the number of WiFi access points in a mall during the pre-train data collecting process.

5.2.3 Metrics. For *WePos* and all other baselines, we use three metrics to measure the merchant-level detection of our model, i.e., accuracy, precision, and F1-score.

- The accuracy is defined as $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, here True Positive ($TP$) means the model correctly predicts the positive class (e.g., a WiFi scanning instance is actually in-merchant and our model predicts in-merchant correctly), True Negative (TN) is an outcome where the model correctly predicts the negative class, FP is an outcome where the model incorrectly predicts the positive class, and FN is an outcome where the model incorrectly predicts the negative class.
- The precision of merchant-level in-merchant and out-merchant classification is defined as $Precision = \frac{TP}{TP+FP}$.
- The F1-score of merchant-level in-merchant and out-merchant classification is defined as $F1 = \frac{2*TP}{2*TP+FP+FN}$.

5.2.4 Baseline Approaches.

- **Top-50 WiFi:** This is a rule-based state-of-the-practice method and has been deployed in the real-world on-demand system, which establishes the mapping between WiFi-IDs and merchants with a WiFi ID set that has the top N (e.g., 50) scanned times in a past period T (e.g., 2 months). In the inference phase, if one of the top-50 WiFi is scanned, we treat this WiFi instance as an in-merchant instance.
- **svdLoc:** This state-of-the-practice method first adopts SVD to extract the latent representation of each WiFi access point and WiFi scanning list. After features extraction, we input the features of each WiFi scanning list into a two-layer deep neural network to conduct classification. We use the pseudo-label-self-generation module introduced in Sec. 4.2 to form pseudo labels.
- **GeLoc [3]:** This method obtains the latent representation of each WiFi AP with graph embedding methods. In the online positioning, we utilize two-layer DNN to conduct classification with the same label-self-generation module in Sec. 4.2.
- **DLSTM [4]:** This method utilizes a local feature-based deep long short-term memory (LF-DLSTM) approach to conduct WiFi fingerprinting indoor localization, which first utilizes a local feature extractor to reduce the noise effect and extract robust local features. Then **DLSTM** utilize a DLSTM network to encode temporal dependencies and learn high-level representations for the extracted sequential local features.

5.3 Evaluation Results

5.3.1 Overall Results. By aggregating the evaluation results under the multi-mall setting, we first compare our approach with the baselines mentioned above with WiFi scanning datasets in on-demand delivery.

As shown in Fig. 13, the proposed *WePos* has a better performance in average accuracy and the average merchant-level positioning accuracy of *WePos* is 91.4%, while the average accuracy of DLSTM [4] and GeLoc [3] are 86.79% and 84.33% respectively. The Top-50 WiFi-based in-merchant status detection and the svdLoc have poor performances in the average accuracy than the other two baselines and the proposed *WePos*. Fig. 14 and Fig. 15 also show our *WePos* have a better performance in precision and F1-score metrics with an average precision of 90.12% and an average f1-score of 92.46%, which outperforms other baselines.

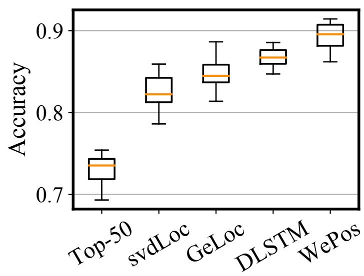

Fig. 13. Positioning Accuracy

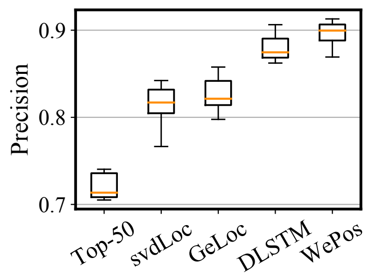

Fig. 14. Positioning Precision

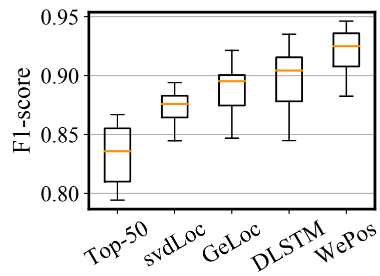

Fig. 15. Positioning F1-score

Performance comparison of merchant-level positioning under multi-mall settings: We also present the merchant-level positioning performance at four large malls in Shanghai to investigate the robustness of the proposed method. Table 5 shows the merchant-level positioning performance comparison of our method and baselines with the precision and F1-score metrics. We find that our *WePos* outperforms other state-of-the-arts baselines in the precision and F1-score of merchant-level positioning at these four malls and the performance order is *WePos* > DLSTM > GeLoc > svdLoc > Top-50. The precision of *WePos* is large than 90% in all malls and is high to 93.69% at the mall C, while the precision of other baselines are less than 90%. The F1-score of *WePos* is also better than that of baselines. The performance comparison of the merchant-level positioning demonstrates the robustness of *WePos* across different malls and environments.

Table 5. Performance comparisons of different methods under Multi-mall settings.

Models	Mall A		Mall B		Mall C		Mall D	
	Precision	F1-Score	Precision	F1-Score	Precision	F1-Score	Precision	F1-Score
Top-50	0.7150	0.8338	0.7790	0.8668	0.7492	0.8566	0.7831	0.8783
svdLoc	0.8146	0.8787	0.7665	0.8272	0.8120	0.8446	0.8024	0.8353
GeLoc	0.8118	0.8951	0.8195	0.9008	0.8577	0.9001	0.8458	0.9081
DLSTM	0.8761	0.8946	0.8678	0.9015	0.9063	0.9159	0.8916	0.9138
WePos	**0.9012**	**0.9246**	**0.8828**	**0.9181**	**0.9369**	**0.9358**	**0.9311**	**0.9233**

5.3.2 Ablation Studies. To verify the importance of our technical components in the proposed *WePos*, we compare and analyze the results between our model and two variations with four-mall data:

- *WePos-s: WePos-s* is the proposed *WePos* without SimCSE based similarity enhancing module.
- *WePos-t: WePos-t* is *WePos* without token-level masked language modeling. *WePos-t* conducts pre-training with a sentence-bert directly and then utilizes the pre-train model to generate pseudo labels and make classification.

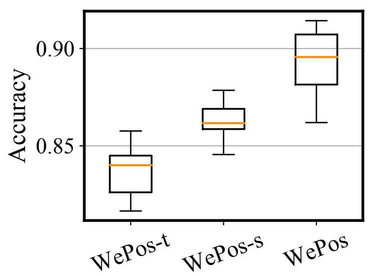

Fig. 16. Positioning Accuracy

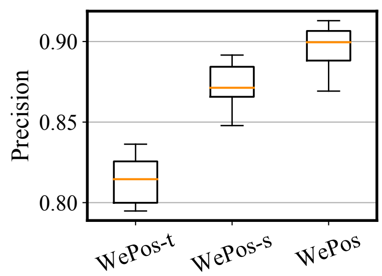

Fig. 17. Positioning Precision

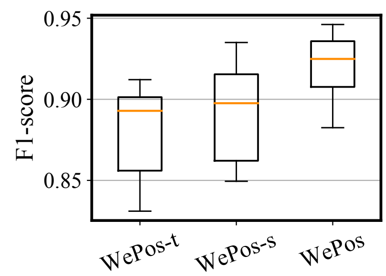

Fig. 18. Positioning F1-score

We first show the comparisons of the merchant-level positioning accuracy in Fig. 16 and find that the variation *WePos-t* has the worst performance, which achieves an average accuracy of 83.86%. *WePos* have the highest average accuracy, offering up to 91.4% and WePos-s has an average accuracy of 88.61%. Fig. 17 and Fig. 18 also show that *WePos* have a better performance in precision and f1-score metrics than other two variations, which proves the effectiveness of the masked language modeling and the SimCSE module in our design. The lesson we learned is that we should consider both token-level pre-training and sentence-level unsupervised fine-tuning when we extract latent representations.

5.3.3 Long-term Robustness Evaluation. We further conduct the long-term evaluation of merchant-level positioning accuracy compared with other baselines with 6-month data collected from Mall A (Jintie City Square mall) to verify the long-term robustness of *WePos*. We utilize the first two-month data to train our method and baselines to extract latent embedding and conduct the performance comparison of the accuracy of merchant-level positioning in the following months. Fig. 19 plots the merchant-level positioning accuracy of our *WePos* and baselines in following 16 weeks. As Fig. 19 shows, *WePos* achieves the best positioning accuracy performance compared with state-of-the-art baselines all the time. In the first week, the accuracy of *WePos* is 92.46%. After 16 weeks, the accuracy of *WePos* still maintains up to 86%, while the performance of other baselines decreases greatly compared with *WePos*, which demonstrates that the proposed weak-supervised learning-based framework is able to maintain the system's performance for a long time.
Impact analysis of fine-tuning module on classification performance: In the pre-training stage, we obtain the past two-month WiFi sensing data from the database at the start of each month to pre-train the model. For the fine-tuning module, we set to automatically get data of the past one week from the database every Sunday to train the model and then use the trained model to conduct merchant-level positioning this week in the real-world delivery environment. Fig. 19 also shows that the performance deterioration of *WePos* is slight in the first four weeks of the fine-tuning process after pre-training. In the 4th week, the accuracy is 90.25% with only a 2.21% decrease, which proves the effectiveness of the fine-tuning strategy of *WePos*.

5.3.4 t-SNE Visualization of WiFi List Embeddings. Fig. 20 visualizes the latent embeddings of unlabeled WiFi scanning lists obtained from SimCSE. Because the aim of similarity aware contrastive embedding based SimCSE is to make the similar WiFi training instances' embeddings closer in the latent space, the embedding from SimCSE

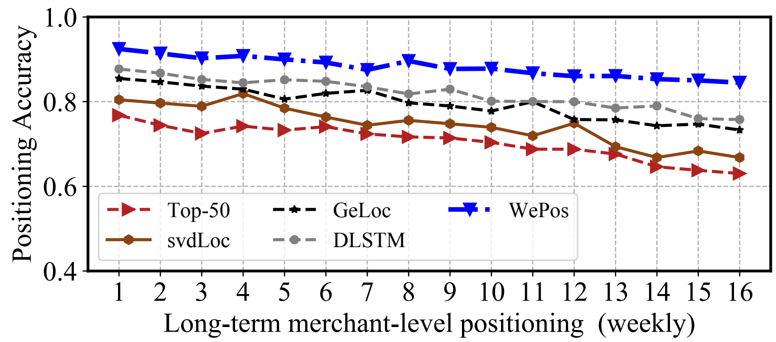

Fig. 19. Performance comparison of Long-term merchant-level positioning

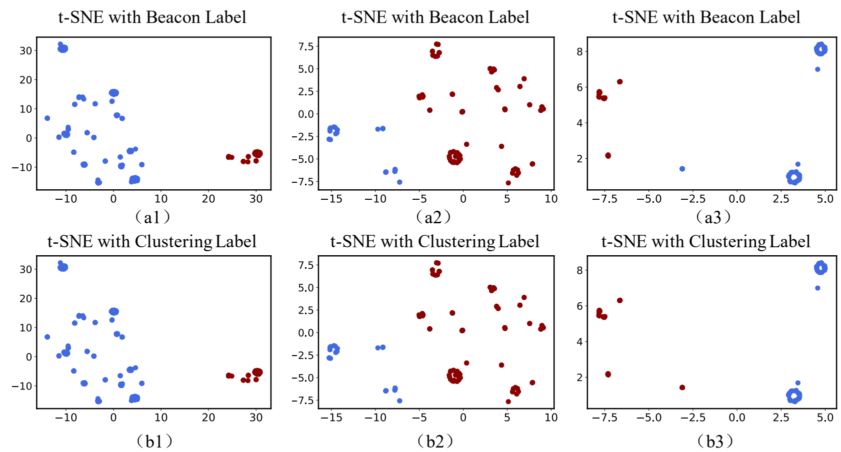

Fig. 20. Visualization of the embeddings of WiFi scanning lists using t-SNE [37]. The sensing data are labeled with beacon and clustering pseudo labels respectively and different colors mark different merchant-level status (i.e., in-merchant and out-merchant status).

needs to better distinguish the WiFi instance between in-merchant status and out-merchant status. Fig. 20 (a1), Fig. 20 (a2), and Fig. 20 (a3) are WiFi embeddings' distribution of three separate order pickup processes, the label is obtained from the beacon and treated as ground truths. Fig. 20 (b1), Fig. 20 (b2), and Fig. 20 (b3) are WiFi embeddings' distribution where the WiFi embeddings are labeled by the proposed deep clustering methods. The same color means that the pseudo labels we formed are correct compared with the ground truths (i.e., beacon labels). We find that through contrastive embedding-based SimCSE, we can achieve accurate clustering to distinguish the difference between in-merchant and out-merchant WiFi instances.

5.4 Application: Performance Comparison of Arrival Time Estimation

5.4.1 Technical Contribution Comparison with TransLoc. Before conducting the performance comparison of arrival time estimation, we first clarify the technical contribution comparison with TransLoc [44] . TransLoc [44] proposes an arrival time prediction model and a merchant-level localization model with human-selected sparse features extracted from manual reporting data. Compared with TransLoc, we not only utilize manual reporting data but also explore extra crowdsourced unlabeled WiFi scanning data as our new opportunity, which are collected during the delivery process with little extra-efforts. Leveraging massive unlabeled WiFi data, the proposed BERT-based model end-to-end generates effective embeddings of WiFi scanning lists, which saves the manual feature selection process. In addition, the crowdsourced unlabeled WiFi data provide more spatio-temporal context of couriers' indoor status than sparse features purely extracted from manual reporting data in TransLoc. We also utilize contrastive learning-based SimCSE to enlarge the distance of embeddings between positive (in-merchant) and negative (out-merchant) to generate the pseudo labels, which extracts precise labels with learned representations from sparse and inconsistent couriers' reporting behaviors (weak labels).

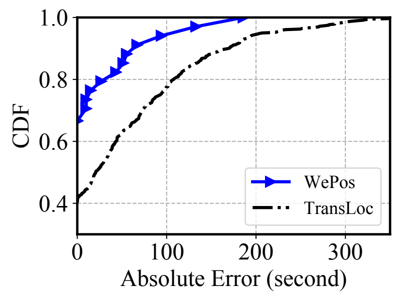

Fig. 21. Arrival Time Estimation

5.4.2 Performance Comparison between TransLoc and WePos. After evaluating the effectiveness of the proposed *WePos* in merchant-level positioning, we further evaluate the arrival time estimation performance compared with TransLoc [44] based on two-month data in a mall area (Mall A) to show the advantages of our model in the downstream applications in on-demand delivery systems. In experiments, we first utilize *WePos* to obtain the merchant-level positioning results (i.e., relative locations) of each WiFi scanning list and then output the arrival time with the sliding window detection. Fig. 21 shows the absolute error distribution of arrival time estimation. We find that our *WePos* has a better performance with a smaller mean error (22.6s) than TransLoc (53.3s). In summary, compared with using human-selected sparse features for positioning, the benefits of *WePos* includes: (i) the WiFi-BERT extracts latent representation from massive unlabeled WiFi scanning data and obtains more contextual relations automatically while using sparse features requires human operations; (ii) the learned embeddings of unlabeled WiFi data after pre-training have the potential to be applied to multiple task-specific downstream problems while using sparse features requires specific feature processing for different tasks.

5.5 Performance Comparison in Public Crowdsourced WiFi Dataset

5.5.1 Details of Public WiFi Crowdsoucing Dataset. The public crowdsourced WiFi sensing dataset [18] is collected from malls in different cities in China with over 50 GB WiFi signal records. The main fields of this public dataset consists of WiFi BSSID of WiFi APs, the RSSI value, the floor number, and the detect timestamps.

Table 6. An example of public crowdsoucing WiFi based indoor Positioning data-set [18].

	WiFi BSSID	Scanning RSSI	Floor	detect timestamp
Public WiFi Sensing Data.	CA:C2:B2:xxxx	-55	2 (range from -1 to 7)	1606698972

5.5.2 Performance Comparisons. To further prove the effectiveness and the generalizability, we conduct performance comparisons between our *WePos* with other baselines in this large crowdsourced public WiFi sensing dataset. Unlike the sensing dataset in on-demand delivery we used above, the public WiFi sensing dataset does not consist of arriving at and staying at merchants' behaviors. To evaluate the positioning performance on the

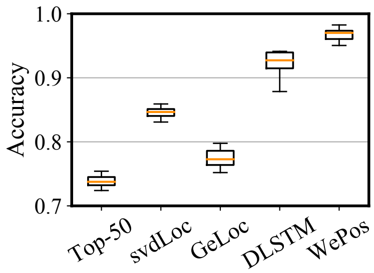

Fig. 22. Positioning Accuracy

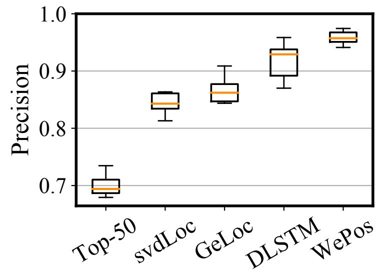

Fig. 23. Positioning Precision

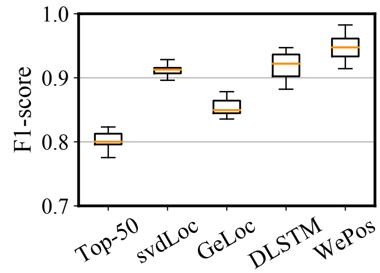

Fig. 24. Positioning F1-score

public WiFi sensing dataset, we first extract latent embeddings based on different baselines and our model, then we utilize floor classification problem (i.e., classify the floor number of each WiFi scanning list) as the fine-tuning task. The experimental results show that *WePos* achieves the best performances floor classification positioning accuracy (96.92%), precision (96.04%), and f1-score (95.19%) compared with other baselines.

6 DISCUSSION

6.1 Insights and Lessons Learned

During working on the paper, we learned the following insights: *Crowdsourcing WiFi collection by couriers are potential ubiquitous ways to construct WiFi fingerprints of merchants and achieve indoor positioning in a low-cost manner* because of two reasons: (i) Under the couriers' consent, we utilize couriers' smartphones to scan WiFi when we detect that couriers are indoor status by IODector [50], which is less-extra-efforts because we do not affect couriers' original delivery tasks. (ii) Crowdsourced WiFi collection collects abundant WiFi scanning records because couriers in our platform pick up and deliver massive orders in one day. In addition, the anchor behavior that couriers stay in merchants waiting for food preparation helps the collected WiFi scanning records share a high similarity, which is essential in the pre-training phase of our model.
Importance of the merchant-level positioning: The impact of the accurate merchant-level classification on order dispatch and delivery process for on-demand delivery are as follows:

- **Order dispatching:** Through merchant-level positioning, we obtain the real-time merchant-level indoor locations, which are utilized by the platform to calculate the dispatching cost (e.g., distance or estimated travel time) between the courier and the merchant to dispatch orders to couriers appropriately.
- **Order delivery time estimation:** Couriers' accurate real-time indoor locations reveal the accurate time of couriers' arrival and departure at indoor shops, offering accurate labels or time-related features for the pick-up time, food preparation time, and delivery time estimation, which helps customers choose shops to order food, merchants prepare meals, and the order dispatching and delivery.

6.2 Limitations and Future Works

(1) Due to lack of fine-grained position labels, we only conduct merchant-level indoor positioning and detect couriers' arrival and departure behaviors. A fine-grained localization would be a nice feature to have but it may not be able to provide higher system gain because it may need onsite configuration or fingerprint of each position. We believe the merchant-level positioning is suitable for the on-demand delivery system and the experimental results in a public data-set with fine-grained position labels prove that our model can achieve more accurate positioning in the wild. (2) In this paper, we study the instant delivery data from Shanghai, which may provide

some obstacles when deploying our model to other cities. However, we believe this obstacle is manageable and our model has the potential to be generalized to other cities. This is because we did not use any city-specific model design and data collected in Shanghai are representative compared to other tier-1 cities in China.

6.3 Ethics, Privacy, and Data Protection Compliance

The data sets used in our research are rendered irreversible anonymous in such a way that the individual is no longer identifiable. Hence the research conducted complies with the general data protection regulation (GDPR)[7]. We did not utilize personal information from the couriers (e.g., age, gender, or personal income) to protect the privacy of the passengers. Specifically, we have taken three steps for privacy protection: (i) *Anonymization:* All data analyzed is anonymized by service providers; (ii) *Minimal Exposure:* We process data that are useful for our order dispatch project, including timestamps, locations, and GPS trajectory records. Then we drop other information for the minimal exposure; (iii) *Aggregation:* Our model analyzes the aggregated results and does not focus on an individual courier or a specific user. Hence, the learned model is less likely to reveal sensitive information about specific individuals. In addition, we do not record or track individual WiFi APs but focus on collecting aggregate WiFi scanning lists and signal strength to protect privacy. We will release one month of the data we collected for the research community to validate our results and conduct further research.

7 RELATED WORKS

7.1 Indoor Localization and Positioning

Studies on indoor localization can be divided into two categories, infrastructure-based [17, 20–22, 45] and infrastructure-free [6]. In infrastructure-based indoor localization systems, indoor locations are estimated based additional infrastructures such as WiFi [14, 20, 21, 28, 45], bluetooth beacon [9], and RFID [23, 43]. For example, abeacon [9] proposes an operational nation-scale indoor sensing system based on Bluetooth Low Energy (BLE) to infer the indoor status of couriers (e.g., arrival and departure at the merchants) by detecting the signals of beacon devices deployed in merchants. LiFS [45] conducts indoor localization system based on off-the-shelf WiFi infrastructure and mobile phones and investigate novel sensors integrated into modern mobile phones and leverage user motions to construct the radio map of a floor plan, which is previously obtained only by site survey. Although recently device-based localization work [22, 38, 41, 46, 47] can achieve high accuracy for indoor positioning, it is hard to build a large-scale low-cost indoor positioning system under the uncontrolled environment, i.e., in the "wild", because of the prohibitive high costs for large-scale deployment in industry. As for the infrastructure-free system, locations are estimated by existing infrastructure (e.g., WiFi, magnetic, etc) without deploying additional hardware [6, 44, 49]. Particularly, indoor localization methods in infrastructure-free systems consist of geometric-based methods and fingerprint-based methods. For instance, MAIL [27] designs a multi-scale attention-guided indoor localization network to identify attention values for different scales in different locations. Chintalapudi et al. [6] propose an EZ Localization algorithm to perform indoor localization using users' locations observations in the setting with WiFi coverage but where do not assume knowledge of the physical layout, including the placement of the APs.

7.2 Deep Learning based Positioning and Sensing Applications

Advances in deep learning in recent years have motivated learning-based solutions for localization [1, 4, 5, 27, 29, 39, 43, 48] and other sensing Applications [26, 33, 34] (e.g., activity detection). Saeed et al. [33] propose a novel self-supervised technique for feature learning from sensory data that does not require access to any form of semantic labels, i.e., activity classes and learn a multi-task temporal convolutional network to recognize transformations applied on an input signal. In [1], a deep learning based indoor navigation framework DLoc is proposed to overcome traditional limitations of RF-based localization approaches (like multi-path, occlusions, etc.). Qian et al.

[29] propose two novel deep learning-based models, which utilize a convolutional mixture density recurrent neural network for indoor next location prediction and utilize a variational autoencoder-based semi-supervised learning model to compute accurate user location. Chidlovskii et al. [5] propose a semi-supervised deep learning method based on the variational autoencoder deep network to further improve the localization accuracy in a complex, multi-building and multi-floor environment with a small set of annotated WiFi observations and a massive set of unlabeled ones. Chen et al. [4] propose a local feature-based deep long short-term memory (LF-DLSTM) approach for WiFi fingerprinting indoor localization, which first utilizes a local feature extractor to reduce the noise effect and extract robust local features. Then, it utilize a DLSTM network to encode temporal dependencies and learn high-level representations for the extracted sequential local features.

7.3 Representation Learning with Unlabeled Data

Deep Representation Learning has been widely studied in recent years to extract latent features of raw data [15, 30], especially real-world massive unlabeled data [26, 32–35]. In the natural language processing (nlp) area, deep representation Learning is widely applied to sentence and word token embeddings under different text contexts [24, 32], especially the pre-train and fine-tuning framework. Specifically, Bidirectional Encoder Representations from Transformers (BERT) [8, 35] are designed to pre-train deep bidirectional representations from the massive unlabeled text by jointly conditioning on both left and right context. After pre-training, the pre-trained BERT model can be fine-tuned with just one additional output layer [2] to create state-of-the-art models for a wide range of tasks, such as question answering and language inference. As for sensing applications, some studies [26, 33, 34] extract a latent representation of sensing data by learning from the massive unlabeled sensing data. Ma et al. [26] propose an end-to-end multi-task deep clustering framework, which consists of a CNN-BiLSTM autoencoder to form a compressed latent feature representation and a deep neural network (DNN) with the latent features and pseudo labels for activity recognition. In our work, we focus on learning effective representations of WiFi scanning records, which follows the general direction of reducing reliance on annotated data, by using massive unlabeled to pre-train model to learn embeddings and then utilize the pre-train model to fine-tune the following specific tasks. However, our model is different from the above works, because our work is in a weak-supervised manner and conducts label-self-generated through contrastive similarity learning considering couriers' uncertain behaviors.

8 CONCLUSION

In this paper, we propose an end-to-end weak-supervised merchant-level positioning framework to achieve real-time indoor positioning for on-demand delivery services with unlabeled WiFi data and uncertain manual reporting. To make full use of massive crowdsourced unlabeled WiFi scanning records and uncertain manual reporting records, we build *WePos*, a weak-supervised representation learning based merchant-level positioning system, which consists of (i) a BERT-based pre-train model to learn the latent representation of WiFi scanning lists; (ii) a contrastive label self-generation module to form pseudo labels for instances; (iii) a deep classifier to achieve end-to-end real-time location inference. Experiments have been conducted on the real-world WiFi scanning data and manual reporting data collected from four multi-floor malls in a large on-demand delivery system. Experimental results show that the proposed approach outperforms state-of-the-art baselines, offering up to 91.2% in merchant-level positioning accuracy. We also show that *WePos* achieves a better performance in long-term robustness evaluation and arrival time estimation evaluation.

ACKNOWLEDGEMENTS

This work was supported in part by Science and Technology Innovation 2030 - Major Project 2021ZD0114202, National Natural Science Foundation of China under Grant No. 61902066, Natural Science Foundation of Jiangsu

Province under Grant No. BK20190336 and Fundamental Research Funds for the Central Universities under Grant No. 2242021R41068.

REFERENCES

[1] Roshan Sai 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 *MobiCom '20: The 26th Annual International Conference on Mobile Computing and Networking, London, United Kingdom, September 21-25, 2020.* ACM, 17:1–17:14.

[2] Md Abul Bashar and Richi Nayak. 2021. Active Learning for Effectively Fine-Tuning Transfer Learning to Downstream Task. *ACM Trans. Intell. Syst. Technol.* 12, 2, Article 24 (Feb. 2021), 24 pages. https://doi.org/10.1145/3446343

[3] Yuanyi Chen, Minyi Guo, Jiaxing Shen, and Jiannong Cao. 2017. GraphLoc: A Graph-Based Method for Indoor Subarea Localization with Zero-Configuration. *Personal Ubiquitous Comput.* 21, 3 (June 2017), 489–505. https://doi.org/10.1007/s00779-017-1011-7

[4] Zhenghua Chen, Han Zou, JianFei Yang, Hao Jiang, and Lihua Xie. 2020. WiFi Fingerprinting Indoor Localization Using Local Feature-Based Deep LSTM. *IEEE Systems Journal* 14, 2 (2020), 3001–3010. https://doi.org/10.1109/JSYST.2019.2918678

[5] Boris Chidlovskii and Leonid Antsfeld. 2019. Semi-supervised Variational Autoencoder for WiFi Indoor Localization. In *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN).* 1–8. https://doi.org/10.1109/IPIN.2019.8911825

[6] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. 2010. Indoor Localization without the Pain. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking* (Chicago, Illinois, USA) *(MobiCom '10).* Association for Computing Machinery, New York, NY, USA, 173–184. https://doi.org/10.1145/1859995.1860016

[7] European Commission. 2019. *EU data protection rules.* https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[9] 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, 859–872. https://www.usenix.org/conference/nsdi21/presentation/ding

[10] 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* (Virtual Event, USA) *(SIGCOMM '21).* Association for Computing Machinery, New York, NY, USA, 705–717. https://doi.org/10.1145/3452296.3472911

[11] DoorDash. 2020. DoorDash. https://www.doordash.com/en-US.

[12] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[13] R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. 1735–1742. https://doi.org/10.1109/CVPR.2006.100

[14] Suining He and S.-H. Gary Chan. 2016. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. *IEEE Communications Surveys Tutorials* 18, 1 (2016), 466–490. https://doi.org/10.1109/COMST.2015.2464084

[15] Tao He, Qun Niu, Suining He, and Ning Liu. 2019. Indoor Localization with Spatial and Temporal Representations of Signal Sequences. In *2019 IEEE Global Communications Conference (GLOBECOM).* 1–7. https://doi.org/10.1109/GLOBECOM38437.2019.9014159

[16] huggingface. 2020. huggingface.co. .

[17] Mohamed Ibrahim, Ali Rostami, Bo Yu, Hansi Liu, Minitha Jawahar, Viet Nguyen, Marco Gruteser, Fan Bai, and Richard Howard. 2020. Wi-Go: Accurate and Scalable Vehicle Positioning Using WiFi Fine Timing Measurement. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services* (Toronto, Ontario, Canada) *(MobiSys '20).* Association for Computing Machinery, New York, NY, USA, 312–324. https://doi.org/10.1145/3386901.3388944

[18] indoor-location-navigation. 2021. indoor-location-navigation. https://www.kaggle.com/c/indoor-location-navigation/data.

[19] Instacart. 2019. *Instacart.* https://www.instacart.com/

[20] 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 *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems* (Roma, Italy) *(SenSys '13).* Association for Computing Machinery, New York, NY, USA, Article 14, 14 pages. https://doi.org/10.1145/2517351.2517352

[21] Bryce Kellogg, Vamsi Talla, Shyamnath Gollakota, and Joshua R. Smith. 2016. Passive Wi-Fi: Bringing Low Power to Wi-Fi Transmissions. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16).* USENIX Association, Santa Clara, CA, 151–164. https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/kellogg

[22] Manikanta Kotaru, Kiran Raj Joshi, Dinesh Bharadia, and Sachin Katti. 2015. SpotFi: Decimeter Level Localization Using WiFi. *Comput. Commun. Rev.* 45, 5 (2015), 269–282. https://doi.org/10.1145/2829988.2787487

[23] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. 2014. Accurate indoor localization with zero start-up cost. *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM* (09 2014). https://doi.org/10.1145/2639108.2639142

[24] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 9119–9130. https://doi.org/10.18653/v1/2020.emnlp-main.733

[25] Danyang Li, Jingao Xu, Zheng Yang, Yumeng Lu, Qian Zhang, and Xinglin Zhang. 2021. Train Once, Locate Anytime for Anyone: Adversarial Learning Based Wireless Localization. In *Proceedings of the IEEE INFOCOM*.

[26] Haojie Ma, Zhijie Zhang, Wenzhong Li, and Sanglu Lu. 2021. Unsupervised Human Activity Representation Learning with Multi-Task Deep Clustering. 5, 1, Article 48 (March 2021), 25 pages. https://doi.org/10.1145/3448074

[27] Qun Niu, Tao He, Ning Liu, Suining He, Xiaonan Luo, and Fan Zhou. 2020. MAIL: Multi-Scale Attention-Guided Indoor Localization Using Geomagnetic Sequences. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 54 (June 2020), 23 pages. https://doi.org/10.1145/3397335

[28] 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 *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2018, Munich, Germany, June 10-15, 2018*, Jörg Ott, Falko Dressler, Stefan Saroiu, and Prabal Dutta (Eds.). ACM, 350–361. https://doi.org/10.1145/3210240.3210314

[29] Weizhu Qian, Fabrice Lauri, and Franck Gechter. 2021. Supervised and semi-supervised deep probabilistic models for indoor positioning problems. *Neurocomputing* 435 (2021), 228–238. https://doi.org/10.1016/j.neucom.2020.12.131

[30] Zhou Qin, Yikun Xian, Fan Zhang, and Desheng Zhang. 2020. MIMU: Mobile WiFi Usage Inference by Mining Diverse User Behaviors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 149 (Dec. 2020), 22 pages. https://doi.org/10.1145/3432226

[31] Rajax. 2018. *Ele.me*. https://www.ele.me/

[32] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[33] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-Task Self-Supervised Learning for Human Activity Detection. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 2, Article 61 (June 2019), 30 pages. https://doi.org/10.1145/3328932

[34] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, Soren Brage, Nick Wareham, and Cecilia Mascolo. 2021. SelfHAR. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (Mar 2021), 1–30. https://doi.org/10.1145/3448112

[35] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. *arXiv preprint arXiv:1908.08962v2* (2019).

[36] Ubereats. 2021. Ubereats. https://www.ubereats.com/hk.

[37] Laurens van der Maaten. 2014. Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research* 15, 93 (2014), 3221–3245. http://jmlr.org/papers/v15/vandermaaten14a.html

[38] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-Level Localization with a Single WiFi Access Point. In *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation* (Santa Clara, CA) *(NSDI'16)*. USENIX Association, USA, 165–178.

[39] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No need to war-drive: unsupervised indoor localization. In *The 10th International Conference on Mobile Systems, Applications, and Services, MobiSys'12, Ambleside, United Kingdom - June 25 - 29, 2012*, Nigel Davies, Srinivasan Seshan, and Lin Zhong (Eds.). ACM, 197–210.

[40] Wikipedia. 2020. Gig worker. Webpage.

[41] Yaxiong Xie, Zhenjiang Li, and Mo Li. 2019. Precise Power Delay Profiling with Commodity Wi-Fi. *IEEE Transactions on Mobile Computing* 18, 6 (2019), 1342–1355. https://doi.org/10.1109/TMC.2018.2860991

[42] 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* (Los Cabos, Mexico) *(MobiCom '19)*. Association for Computing Machinery, New York, NY, USA, Article 8, 16 pages. https://doi.org/10.1145/3300061.3300133

[43] Huatao Xu, Dong Wang, Run Zhao, and Qian Zhang. 2019. AdaRF: Adaptive RFID-based Indoor Localization Using Deep Learning Enhanced Holography. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3 (2019), 113:1–113:22. https://doi.org/10.1145/3351271

[44] 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* (London, United Kingdom) *(MobiCom '20)*. Association for Computing Machinery, New York, NY, USA, Article 41, 14 pages. https://doi.org/10.1145/3372224.3419198

[45] Zheng Yang, Chenshu Wu, and Yunhao Liu. 2012. Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking* (Istanbul, Turkey) *(Mobicom '12)*. Association for Computing Machinery, New York, NY, USA, 269–280. https://doi.org/10.1145/2348543.2348578

[46] Sangki Yun, Yi-Chao Chen, and Lili Qiu. 2015. Turning a Mobile Device into a Mouse in the Air. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services* (Florence, Italy) *(MobiSys '15)*. Association for Computing Machinery, New York, NY, USA, 15–29. https://doi.org/10.1145/2742647.2742662

[47] Feng Zhang, Chen Chen, Beibei Wang, Hung-Quoc Lai, Yi Han, and K. J. Ray Liu. 2018. WiBall: A Time-Reversal Focusing Ball Method for Decimeter-Accuracy Indoor Tracking. *IEEE Internet of Things Journal* 5, 5 (2018), 4031–4041. https://doi.org/10.1109/JIOT.2018.2854825

[48] Hua Zhang, Biao Hu, Shiqi Xu, Bi Chen, Mian Li, and Bo Jiang. 2020. Feature Fusion Using Stacked Denoising Auto-Encoder and GBDT for Wi-Fi Fingerprint-Based Indoor Positioning. *IEEE Access* 8 (2020), 114741–114751. https://doi.org/10.1109/ACCESS.2020.3004039

[49] Han Zheng, Yan Zhang, Lan Zhang, Hao Xia, Shaojie Bai, Guobin Shen, and Xiangyang Li. 2022. GraFin: An Applicable Graph-based Fingerprinting Approach for Robust Indoor Localization.

[50] Pengfei Zhou, Yuanqing Zheng, Zhenjiang Li, Mo Li, and Guobin Shen. 2012. IODetector: A Generic Service for Indoor Outdoor Detection. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems* (Toronto, Ontario, Canada) *(SenSys '12)*. Association for Computing Machinery, New York, NY, USA, 113–126.