Towards Dynamic Crowd Mobility Learning and Meta Model Updates for A Smart Connected Campus

Xi Yang Suining He Mahan Tabatabaie Bing Wang
Department of Computer Science and Engineering
University of Connecticut
{xi.yang, suining.he, mahan.tabatabaie, bing}@uconn.edu

Abstract

In this paper, we propose MetaMobi, a novel spatio-temporal multi-dots connectivity-aware modeling and ***Meta*** model update approach for crowd ***Mobi***lity learning. MetaMobi analyzes real-world Wi-Fi association data collected from our campus wireless infrastructure, with the goal towards enabling a smart connected campus. Specifically, MetaMobi aims at addressing the following two major challenges with existing crowd mobility sensing system designs: *(a)* how to handle the spatially, temporally, and contextually varying features in large-scale human crowd mobility distributions; and *(b)* how to adapt to the impacts of such crowd mobility patterns as well as the dynamic changes in crowd sensing infrastructures. To handle the first challenge, we design a novel multi-dots connectivity-aware learning approach, which jointly learns the crowd flow time series of multiple buildings with fusion of spatial graph connectivities and temporal attention mechanisms. Furthermore, to overcome the adaptivity issues due to changes in the crowd sensing infrastructures (*e.g.,* installation of new access points), we further design a novel meta model update approach with Bernoulli dropout, which mitigates the overfitting behaviors of the model given few-shot distributions of new crowd mobility datasets. Extensive experimental evaluations based on the real-world campus wireless dataset (including over 76 million Wi-Fi association and disassociation records) demonstrate the accuracy, effectiveness, and adaptivity of MetaMobi in forecasting the campus crowd flows, with 30% higher accuracy compared to the state-of-the-art approaches.

Keywords

Multi-dots connectivity-aware learning, crowd mobility learning, campus Wi-Fi infrastructure, meta learning, gradient dropout, model generalizability

1 Introduction

Crowd mobility learning, *i.e.,* understanding and capturing the human crowd movement, is the key enabler for various mobile, urban, and ubiquitous computing applications. Accurate crowd mobility learning can benefit the city planners and other facility management departments in monitoring urban events [5], analyzing epidemics [40], and other social recommendation based on location-based services [19]. With the above-mentioned social and business values, particularly after the outbreak of COVID-19, the global crowd analytics market was valued at $912.68 million in 2020, and is projected to hit $5.7 billion by 2030, with a compound annual growth rate of more than 20% from 2021 to 2030 [1].

In this study, we focus on how to design, prototype, and implement a crowd mobility learning system for our university campus using the Wi-Fi association data harvested from the campus wireless local-area network (WLAN) infrastructure. Our system leverages the data from the Wi-Fi access points (APs), which have been ubiquitously deployed on our campus to provide the local users (students, faculty, and staff) convenient Internet access services. When users' mobile devices are associated with the campus WLAN infrastructures [38], their locations can be inferred by the locations of the APs that their mobile devices are associated with. The harvested Wi-Fi association data can thus enable a real-time crowd mobility learning framework.

However, despite the prior studies [12, 16, 21], we need to address the following two major technical challenges before a crowd mobility learning system can be practically deployed for a smart connected campus:

Challenge 1 – Internal Dynamic Complexity within Crowd Mobility Distributions. The crowd mobility patterns (*e.g.,* arrivals/departures or in/outflows at various locations as illustrated in Fig. 1) can be inherently dynamic. The crowd distributions vary across different locations (*e.g.,* academic buildings, recreational centers) and time periods (*e.g.,* semesters and academic years) due to the dynamic schedules, neighboring events, affected crowd mobility routines, as well as the latent individual preferences. Existing mobility learning models [16, 21, 42] largely focus on capturing the long-term repetitive patterns, while the transient changes within mobility patterns due to various inherent factors often make it hard for the crowd mobility models to adapt and generalize.

Challenge 2 – External Impacts of Crowd Sensing In-

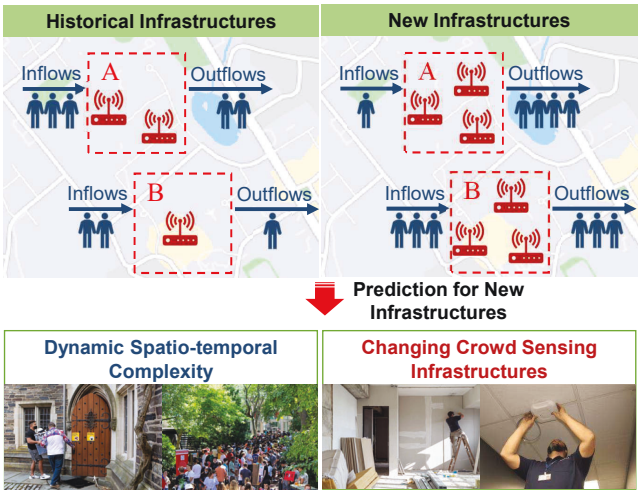


Figure 1: Motivations of our crowd mobility learning designs (crowd flow modeling as an example).

frastructures. We regard the existing campus WLAN as a "crowd sensing" infrastructure with ubiquitously deployed APs as the sensors, similar to cellular-based or camera-based monitoring systems [6, 14]. The harvested Wi-Fi association data contain important crowd mobility patterns for the conventional models to fit upon, learn from, and subsequently predict future patterns, *e.g.,* in/outflows at various locations. However, in practice, this crowd sensing infrastructure may change due to regular maintenance of WLAN, adjusted density and coverage of the APs, or building layout renovation. Such changes may lead to changed spatial, temporal, and contextual patterns of crowd flows captured by the crowd mobility learning system. For instance, as illustrated in Fig. 1, when the campus facility management department changes the building layouts and installs new APs in buildings *A* and *B*, the harvested crowd mobility patterns may change, making the previously-trained crowd mobility models obsolete. Furthermore, existing models [21,34,41], if not adapted properly, may overfit upon the limited accumulation of the Wi-Fi association data that are harvested immediately after the changes of crowd sensing infrastructures, leading to low learnability and generalizability.

To address the above challenges, we have proposed `MetaMobi`, an adaptive multi-dots connectivity-aware modeling and ***Meta*** model update approach for crowd ***Mobi***lity learning. Specifically, in prototyping `MetaMobi`, we make the following three major contributions:

1. **Spatio-Temporal Multi-Dots Connectivity-Aware Crowd Mobility Learning** (Sec. 5.1): We have proposed a novel multi-dots connectivity-aware learning neural network to capture the complex spatio-temporal characteristics of crowd flow patterns at different campus locations that are considered as *dots*, and their important *connectivities* across them. The proposed core design consists of an `EmLearn` module, which learns the spatio-temporal patterns within the crowd mobility embeddings, and a `DiffLearn` module, which differentiates and adapts to the dynamic connectivities across campus locations. Within the `EmLearn` module, we have designed various types of connectivities across multiple dots that represents the geo-locations, including geo-spatial distance as well as their respective building functions on campus, to enhance our spatial embedding learning by a novel multi-graph attention mechanism. Furthermore, we propose an efficient residual structure to enhance the model learnability and efficiency within our temporal embedding learning component. Within the `DiffLearn` module, we have designed a novel attention mechanism to fuse and further differentiate the learned embedded spatial and temporal features, yielding accurate crowd flow prediction.
2. **Generalizable Crowd Mobility Learning Meta Model Updates based on Bernoulli Gradient Dropout** (Sec. 5.2): To overcome the poor learnability and generalizability of existing crowd mobility learning designs [21, 34, 41], we have designed a novel meta-learning framework for model adaptation, and further introduced a *Bernoulli dropout* approach for our meta model update. Specifically, we design a Meta-Training phase upon `MetaMobi` to capture the general crowd mobility patterns from data prior to infrastructure changes. Then we hold the parameters of the `EmLearn` module learned from the Meta-Training phase, and further leverage a short period of the crowd flows of target locations that are changed to further update the initialized `DiffLearn` module. During these processes, we have further designed a Bernoulli gradient dropout to discard the model parameters that are not updated by their gradients, yielding generalizable performance given limited accumulation of the new Wi-Fi association data.
3. **Extensive Data-driven Evaluations and Real-world Experimental Studies** (Sec. 6): We have conducted extensive real-world data studies based on our University of Connecticut (UConn) campus wireless infrastructures (over 76 millions of Wi-Fi associations). In this prototype study, we consider the prediction of crowd flows (*i.e.,* arrivals and departures) at various campus locations (*e.g.,* buildings and APs) to validate our proposed `MetaMobi` designs. Extensive experimental results regarding the spatio-temporal crowd flow prediction and meta model updates have corroborated the *accuracy*, *effectiveness*, and *adaptivity* of `MetaMobi` in forecasting the crowd flows, with an average of more than 30% error reduction compared with the state-of-the-art approaches.

2 Related Work

We briefly review related work in the following two major categories.

• **Deep Mobility Modeling**: With the proliferation of big crowd data, internet-of-things, and deep learning, myriad of deep mobility modeling approaches have been studied for the ubiquitous and mobile computing applications. Lin *et al.* [21] studied a context-aware framework that captures long-range spatial features, and considered various location-based attributes to predict the crowd flow. Jiang *et al.* [15] proposed an online system that extracts the deep trend from

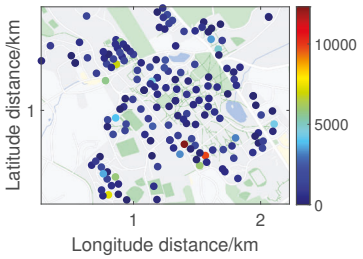
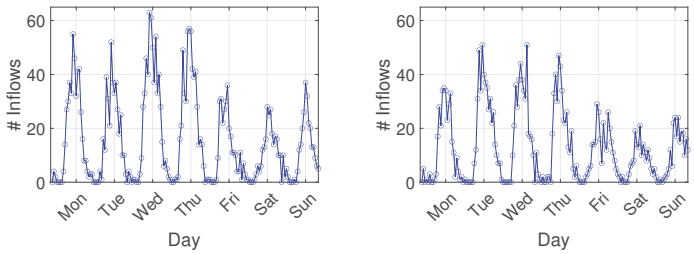


Figure 2: Spatial variations of aggregated crowd flows.



(a) 10-12-2020 to 10-18-2020 (b) 02-15-2021 to 02-21-2021.

Figure 3: Temporal dynamics of the crowd mobility from (a) one week in Fall 2020 and (b) one week in Spring 2021 of the same academic building.

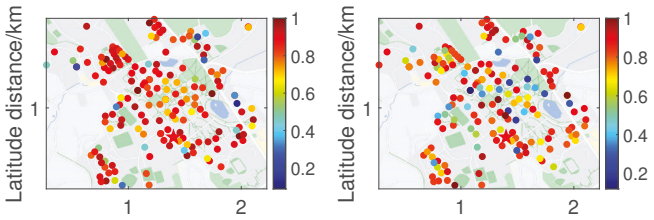


Figure 4: Spatial distributions of the temporal correlations of a building with others during (a) Fall 2020 and (b) Fall 2021. We adopt the cosine similarity here to overcome the scale issue.

a momentary and short-term observations to predict the future mobility. Zhang *et al.* [42] introduced a deep neural network architecture based on residual neural networks [12]. Different from the aforementioned approaches, our work focuses on how to further capture crowd mobility patterns that dynamically vary due to internal spatio-temporal factors as well as external impacts of crowd sensing environments. Our data-driven model designs have been validated using real-world human crowd sensing cases and extensive experimental studies. Our results demonstrate that `MetaMobi` leads to substantial performance improvement over the state-of-the-art approaches.

• **Model Update with Meta Learning**: Meta learning algorithms [3, 9, 23, 28, 29] have recently attracted much attention for enhancing the generalizability and adaptivity of the deep learning models given a set of new tasks or data sources through searching for a good model initialization. To handle few-shot settings (*e.g.,* a limited number of Wi-Fi data records during the period when the infrastructures are changed in our current study) and improve the model adaptivity under "cold-start" scenarios, various model update approaches with meta learning have been proposed and studied recently [24, 26, 30, 33, 39]. Yao *et al.* [39] proposed a meta learning-based approach that trains a model based on the long-term spatio-temporal information from multiple source cities that serves as an initialization for the target cities. Wang *et al.* [33] designed a spatio-temporal meta-model that enhances the generalizability of the model for various spatio-temporal traffic prediction applications. Pan *et al.* [24] proposed a meta learning algorithm that simultaneously learns and predicts the traffic flow for various locations. Rajendran *et al.* [26] introduced a novel data-augmentation technique to reduce the overfitting issues of the learned initialization towards new tasks. Tian *et al.* [30] incorporated a regularization method into their meta learning algorithm to assist the model in reducing the data distribution discrepancy of the training and test datasets.

Inspired by these prior studies, our work focuses on an important crowd sensing and mobility learning problem given the dynamically varying mobility patterns as well as changed sensing infrastructures. We have designed a novel meta model update approach for `MetaMobi` driven by real-world data analytics and insights. Furthermore, to overcome the overfitting given limited initial mobility data, we have further designed a novel Bernoulli dropout for our meta learning approach, and our experimental results have validated the effectiveness and adaptivity of `MetaMobi`.

3 Data-driven Design Motivations

In this section, we first outline the Wi-Fi association dataset harvested from our university campus in Sec. 3.1. Afterwards, we motivate our model designs in three important aspects in Sec. 3.2: (1) spatio-temporally varying features due to human crowd mobility patterns (Sec. 3.2.1), (2) changes in sensing infrastructure (Sec. 3.2.2), and (3) poor learnability and generalizability of traditional approaches (Sec. 3.2.3).

3.1 Real-world Dataset

Collaborating with the campus information technology services, we have collected Wi-Fi association data from our UConn campus network for a total of 10,967 Wi-Fi access points (APs) installed in 231 buildings. Specifically, we have retrieved the Wi-Fi association and disassociation events of all APs on campus on an hourly basis by a server using the standard network protocols [11]. We record important information for each association event, including the user ID (encrypted and randomized for privacy protection), association and dissociation timestamp, the MAC address of the AP where the user connected to the Wi-Fi, and the building name where the AP is installed. Aggregating the number of users whose Wi-Fi association timestamps lie within a time interval in a certain building with an AP, we obtain the number of crowd inflows of the campus locations. Similarly, we generate the number of crowd outflows at the same location by aggregating the number of users whose devices disassociate within the time interval. In summary, we have collected a total of 76,226,504 Wi-Fi association records for the period from 10-10-2020 to 11-22-2021.

In this study, we focus on crowd flow modeling which is a typical case for the crowd mobility studies. Based on the Wi-

Fi associations and disassociations (in-/outflows), we further derive the Wi-Fi-based crowd flow measurements that are used in our prototype `MetaMobi` studies and performance evaluation.

Wi-Fi-based Crowd Flow Measurements: Within a time interval k, we define the human crowd flows, denoted as $\mathbf{F}_k \in \mathbb{R}^{N\times 2}$, as the total number of user arrivals (inflows) and departures (outflows) within the interval k for each of the N *campus locations* (*i.e.,* buildings or APs). Note that here we consider two spatial levels of crowd flows, *i.e.,* building level and AP level. For the buildings with high volumes of crowd flows (say, if hourly inflows or outflows are above 150), we further derive the more fine-grained AP-level crowd flows instead of building-level, which may benefit the campus facility management and emergency department in understanding the crowd mobility. For the rest of the buildings, we aggregate the crowd flows of all APs which suffice for our campus management purposes. Furthermore, we let $\mathbf{F}_k[:,1]$ be the inflows and $\mathbf{F}_k[:,2]$ be the outflows. For the inflows and outflows of each building, we filter out the association/dissociation events generated from user disconnecting from one AP and connecting to another inside the same building to avoid the ping-pong effects [7] and unnecessary computation. To support the connectivity-aware learning, we have further harvested the campus map (GPS coordinates of buildings) and the building functions.

3.2 Design Challenges and Motivations

3.2.1 Dynamic Spatio-Temporal Features

The crowd flow measurement on the campus exhibit strong spatio-temporal dynamics as shown in Figs. 2 and 3. Specifically, we can see from Fig. 2 the total numbers of aggregate inflows within a week during Fall 2020 vary spatially across different buildings on the campus. Fig. 3 further shows the time series of a selected academic building during the same time period in Fig. 2, where we observe that the inflows demonstrate the routine patterns, *i.e.,* high volumes during daytime and low ones at night, while the days approaching or during weekends tend to show lower inflows.

Despite these spatial variations and repetitive patterns (*e.g.,* due to daily commutes) for conventional models to learn and predict, we have also observed that such spatio-temporal patterns may not always hold due to various events during and across different semesters. For instance, Fig. 3 shows the temporal dynamics crowd flow at the same academic building for two weeks in Fall 2020 and Spring 2021, respectively. By comparing Fig. 3a and Fig. 3b, we can observe the changed temporal patterns across the two semesters. Fig. 4 further shows the temporal correlations between the inflow time series of a residential hall (star) and other buildings for 6 weeks in Fall 2020 (Fig. 4a) and 6 weeks in Fall 2021 (Fig. 4b).

Comparing Fig. 4a and Fig. 4b, we see significant differences in the spatio-temporal features. We further calculate the difference in such temporal correlations between Fall 2020 and Fall 2021 semesters for each building and further show the Probability Density Function (PDF) in Fig. 5, from which we note the changes in the crowd flow patterns between the semesters. These may be due to the changing class schedule and the number of registered students, leading to the internal dynamics of the human crowd mobility patterns on campus.

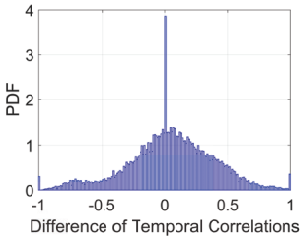


Figure 5: PDF of the differences in the temporal correlations of the two semesters.

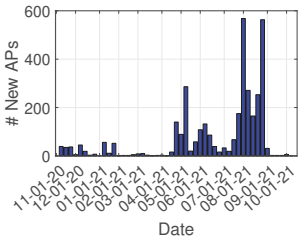


Figure 6: Weekly number of newly installed APs during 2020–2021.

3.2.2 Installation of New APs

In addition to various internal factors, our data analytics show that changes in the external crowd sensing infrastructures can also significantly affect the spatio-temporal features of crowd flows. In particular, when the information technology service (ITS) department changes its AP installation distributions (locations, network coverage, density, and introduction of new APs), the identified crowd mobility patterns in terms of in/outflows may significantly differ from those in the prior periods, leading to performance degradation of conventional crowd mobility learning.

For instance, Fig. 6 plots the number of new APs installed within each week between 11-01-20 and 10-01-21. From Fig. 6, we can observe that a large number of APs were installed during Summer 2021. Clearly, such changed distributions of APs will generate new crowd mobility patterns (in/outflows in our case), and how to make the learning model adapt to such patterns without much prior distribution knowledge (namely the "cold-start" issues) is essential for the effective crowd mobility learning and subsequent campus facility management.

3.2.3 Limitations of Existing Learning Approaches

Given the spatially and temporally dynamic and varying crowd mobility patterns as well as the changing crowd sensing infrastructures, it is challenging to predict the crowd mobility distributions with previously trained models, especially when we only have limited or no knowledge about the new patterns when these phenomena are observed. One may retrain the crowd mobility model based on a small dataset collected in a short period of new mobility patterns, *i.e.,* through a few-shot learning setting, which, however, may lead to model overfitting and poor generalizability.

For instance, we demonstrate in Fig. 7 the loss function in terms of Mean Square Errors (MSE) during the training and validation phases of a conventional crowd mobility model based on deep learning [41]. We train the deep learning model based on only one week of newly generated patterns, and then predict two-week new crowd flows. We see that overfitting happens as the validation loss decreases first but increases after around 40 epochs, despite the drop of the training loss. Without proper model learnability designs and training regularization, it is challenging for the crowd mobility model to adapt to the complex and changed crowd sensing data settings.

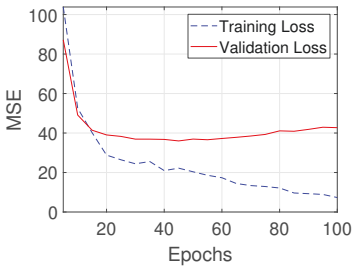


Figure 7: The curve of training and validation loss values of a traditional crowd mobility learning approach.

4 Problem, System, and Model Overview

Motivated by the aforementioned three major challenges, this section details our core model designs of MetaMobi. We first present the problem definition in Sec. 4.1, and then present the system framework in Sec. 4.2, followed by the overview of the model architecture in Sec. 4.3.

4.1 Problem Definition

Given the historical K crowd flows, denoted as $\{\mathbf{F}_k\}$, $k \in \{1, 2, \ldots, K\}$, the campus map and the building functionality, MetaMobi predicts the crowd flows, denoted as $\widehat{\mathbf{F}}$, at the next target time interval.

4.2 System Framework Overview

The information flow of MetaMobi system prototype is shown in Fig. 8. We refer to the Wi-Fi crowd mobility data before the campus WLAN infrastructure changes as the "source" dataset, while the limited amount of data harvested after the changes as the "target" dataset. To adapt to these two datasets, we have designed two different phases for model training and adaptation, *i.e., Meta-Training* and *Pre-Training*.

To achieve adaptive crowd mobility learning (taking the crowd flow as an example), we first train our core model of multi-dots connectivity-aware learning with multiple sets of the source dataset (*e.g.,* based on different semesters in our campus case study) in the Meta-Training phase, so that the model is generalized to various dynamic spatio-temporal features. In the meantime, we incorporate the information of the campus map (*e.g.,* geo-spatial distances) and building functions (*e.g.,* whether a building is mainly related to dining activities) to enhance the learning of spatio-temporal characteristics. Afterwards, the Pre-Training phase leverages the learned parameters, and further trains the model with a small set of the target dataset, making the model adaptive to the new mobility patterns after the changes of crowd sourcing infrastructure. In the Testing phase, MetaMobi leverages the parameters learned in the Pre-Training phase to predict the crowd mobility (*e.g.,* crowd flows) given the dynamic crowd mobility and new WLAN infrastructures.

In real-world deployment, one may prepare the core MetaMobi model based on the datasets harvested before any potential changes in mobility patterns or crowd sensing infrastructures ("Meta-Training"), and adapt the model given a limited period (*e.g.,* a few hours or days) of new mobility data ("Pre-Training") for further prediction of crowd mobility patterns ("Testing").

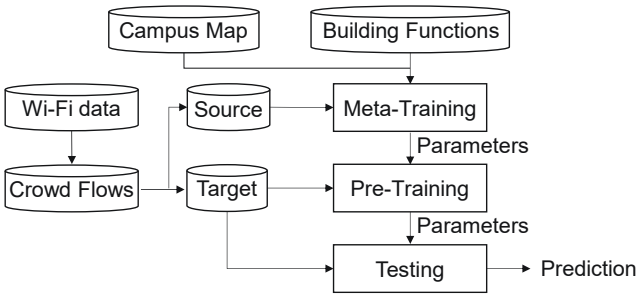


Figure 8: Overview of the system information flow.

4.3 Model Architecture

Given the system framework, Fig. 9 further details the core model architecture of MetaMobi. Our core model of the *multi-dots connectivity-aware learning* consists of two main modules, *i.e.,* the EmLearn module (Sec. 5.1.1), which leverages the *Em*beddings to *Learn* the spatial and temporal connectivities across different location spots through multi-level embedding designs, and the DiffLearn module (Sec. 5.1.2), which *Diff*erentiates, *Learn*s, and adapts to the embedded features from the preceding EmLearn module.

As discussed earlier, our MetaMobi model has two training phases, Meta-Training and Pre-Training, to adapt to the crowd mobility patterns and changed crowd sensing infrastructures. In the Meta-Training phase, we use different sets of source crowd flow data (say, based on different academic semesters in our campus case study) *i.e.,* the periods before the campus WLAN infrastructure is changed, to update the parameters of the EmLearn and DiffLearn modules, and make the model generalized for the target crowd flow data, *i.e.,* the periods when the campus WLAN infrastructure is changed.

Then in the Pre-Training phase, we freeze the parameters of the EmLearn module learned from the Meta-Training phase, and MetaMobi leverages a short period of the crowd flows of target locations to train the DiffLearn module, which is initialized by the parameters learned from the Meta-Training phase. In both training phases, to overcome the *overfitting* with small data, we have designed a *Bernoulli gradient dropout* to randomly drop some parameters that are not updated by their gradients (Sec. 5.2).

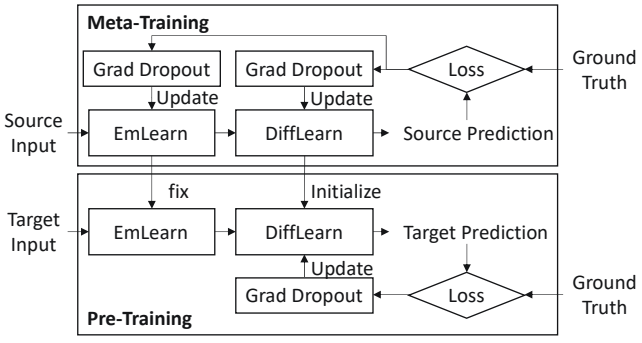


Figure 9: Overview of the core model architecture and training processes of MetaMobi.

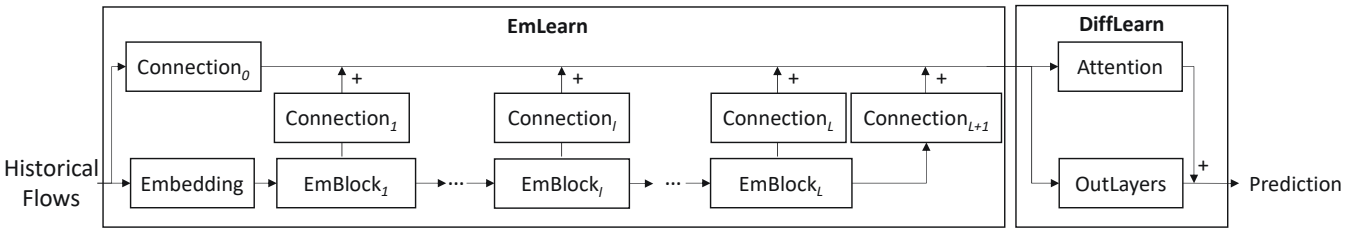


Figure 10: Overview of our multi-dots connectivity-aware learning mechanism.

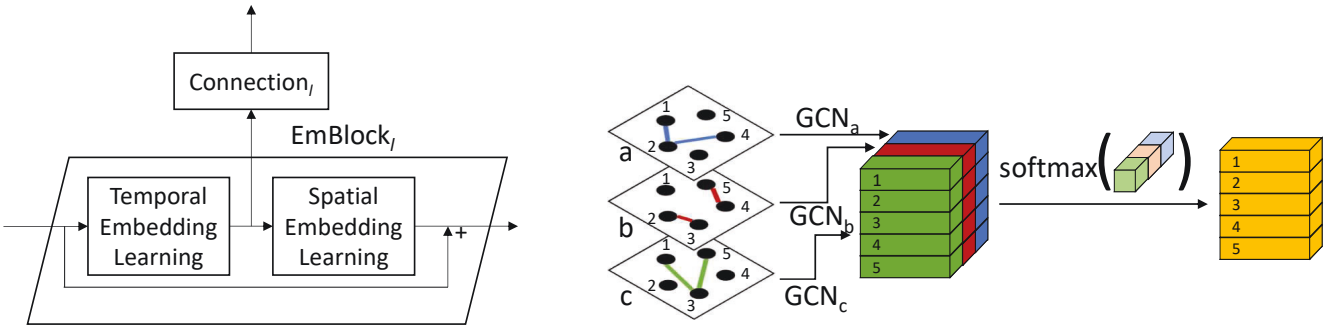


Figure 11: Illustration of (a) `EmBlock` designs; and (b) the Spatial Embedding Learning designs.

5 Core Model Designs

We first present the spatial and temporal crowd learning model architecture in Sec. 5.1. Afterwards, we present the training procedure of our generalizable meta model designs in Sec. 5.2.

5.1 Spatial and Temporal Crowd Mobility Learning Designs

We illustrate our proposed multi-dots connectivity-aware learning mechanism in Fig. 10, which shows the designs of `EmLearn` (Sec. 5.1.1) and `DiffLearn` (Sec. 5.1.2). `MetaMobi` takes in the historical K time intervals of crowd flows, *i.e.,* $\mathbf{F} = \{\mathbf{F}_k\}$, $k \in \{1,2,\ldots,K\}$, as the model inputs, where $\mathbf{F} \in \mathbb{R}^{K\times N\times 2}$. We consider the inputs $\mathbf{F}$ as a heatmap tensor with the height of K, the width of N, and the channel (3^{rd} dimension) of length 2 representing the inflows and outflows.

5.1.1 *Design of `EmLearn` Module*

In the `EmLearn` module, we first pass the inputs through an embedding layer based on a 2D convolutional (`Conv2D`) layer with 1×1 kernel size, to encode the input flows into a high-dimensional embedding, $\mathbf{F}^{\text{Inp}}$, for a comprehensive representation of the input spatio-temporal crowd mobility characteristics, *i.e.,*

$$\mathbf{F}^{\text{Inp}} = \phi(\mathbf{F}), \tag{1}$$

where $\mathbf{F}^{\text{Inp}} \in \mathbb{R}^{K\times N\times C^{\text{Inp}}}$, C^{Inp} represents the channel size of the embedding layer, and ϕ denotes the `Conv2D` operation. Note that in the rest of the paper we consider the 1^{st} dimension of the kernel size for a `Conv2D` layer represents the kernel height, while the 2^{nd} dimension represents the width.

Afterwards, we further feed the preliminary embedded inputs into a series of L *Em*bedding *Block*s, denoted as `EmBlock`s. Each l-th embedding block, denoted as `EmBlock`$_l$, $l \in [1,L]$, is further illustrated in Fig. 11a. Specifically, each `EmBlock` contains two main components to further derive embedded patterns, *i.e.,* the *Temporal Embedding Learning* component and the *Spatial Embedding Learning* component. We further present the designs of these two components as follows.

• **Temporal Embedding Learning.** Temporal Embedding Learning is a `Conv2D` layer with channel size of C^{Tp} and kernel size of $k_l^{\text{Tp}} \times 1$. This component captures the temporal patterns for every k_l^{Tp} time intervals. We note that we do not take into account the zero padding and hence the resulting flow embedding, denoted as $\mathbf{F}_l^{\text{Tp}} \in (K - k_l^{\text{Tp}} \cdot l + 1) \times N \times C^{\text{Tp}}$, has the height of $K - k_l^{\text{Tp}} \cdot l + 1$, and is followed by a Spatial Embedding Learning layer to capture the spatial characteristics of the crowd flows.

• **Spatial Embedding Learning.** The N campus locations form a graph $\mathbf{G}$ with each graph node as a dot representing a campus location. The edges of $\mathbf{G}$ denote the connectivities between the nodes. Through our extensive data analytics, we have observed that various latent correlations between the nodes that might shed different impacts upon the crowd mobility. For example, during the lunch hours the students may depart from the academic buildings for the dining halls that are closest to them.

We denote the connectivities across different nodes of type t as the adjacency matrix, $\mathbf{A}_t \in \mathbb{R}^{N\times N}$, $t \in [1,T]$. In this study, we consider totally $T = 5$ types of nodes (dots), including the *geospatial distance* (unit: m) between locations, and four functionalities of the buildings (*e.g.,* whether this building is mainly related to dining or recreational activities).

For the type of geospatial distance, each element of the adjacency matrix, $\mathbf{A}_t[i,j]$, is given by the inverse of geo-distance between two locations after min-max normalization, *i.e.,*

$$\mathbf{A}_t[i,j] = \mathbf{A}_t[j,i] = \frac{1}{d(i,j)}, \tag{2}$$

where $d(i,j)$ is the normalized geo-distance between locations i and j such that $d(i,j) \in [0,1]$. We set all the diagonal elements $\mathbf{A}_t[i,i] = 0$.

Each building on campus has its own functions, which we categorize into 4 major types in this campus case study, *i.e.,* *Food*, *Residential*, *Athletic*, and *Academic*. The crowd mobility is usually generated by the dynamic functions of the buildings that may attract or steer the crowds. Therefore, for each functionality type, we set $\mathbf{A}_t[i,j] = 1$ if the functionality of buildings i and j are different, yielding a potentially strong mobility trend, and $\mathbf{A}_t[i,j] = 0$ otherwise. We consider an AP has the same functionality of the building where it is located. Similar to the *Distance* type (captured in Eq. (2)), we set all the diagonal elements $\mathbf{A}_t[i,i] = 0$.

To enable awareness of the T types of connectivities, we design T graph convolutional networks (GCNs) [18] for each type, respectively. Specifically, we formulate the t-th GCN, denoted as $\texttt{GCN}_t$, as follows. We normalize the adjacency matrix defined above first by its degree matrix, $\mathbf{D}_t$, and obtain $\widetilde{\mathbf{A}}_t \in \mathbb{R}^{N \times N}$, *i.e.,*

$$\widetilde{\mathbf{A}}_t = \mathbf{D}_t^{-\frac{1}{2}} \cdot (\mathbf{A}_t + \mathbf{I}) \cdot \mathbf{D}_t^{\frac{1}{2}}, \tag{3}$$

where $\mathbf{I}$ is the identity matrix. Then we formulate the graph convolutional operation by

$$\mathbf{F}_{l,t}^{\text{Sp}} = \left[\widetilde{\mathbf{A}}_t \cdot \left(\mathbf{F}_l^{\text{Tp}}\right)^{\intercal} \cdot \mathbf{W}_t\right]^{\intercal}, \tag{4}$$

where the symbol $\intercal$ denotes a transpose operation for the first and the second dimensions and $\mathbf{W}_t \in \mathbb{R}^{C^{\text{Tp}} \times C^{\text{Sp}}}$ is a learnable parameter matrix. The resulting flow embeddings for all T types of correlations are given by $\mathbf{F}_l^{\text{Sp}} = \{\mathbf{F}_{l,1}^{\text{Sp}}, \ldots, \mathbf{F}_{l,T}^{\text{Sp}}\} \in \mathbb{R}^{T \times (K - k_l^{\text{Tp}} \cdot l + 1) \times N \times C^{\text{Sp}}}$. Then we design an attention mechanism to obtain the weighted sum of all T different types of embeddings, *i.e.,*

$$\mathbf{F}_l^{\text{G}} = \texttt{Softmax}(\mathbf{v}_{\text{Att}}) \cdot \mathbf{F}_l^{\text{Sp}}, \tag{5}$$

where $\mathbf{v}_{\text{Att}} \in \mathbb{R}^T$ is a learnable parameter vector, and its output Softmax activation denotes the weights of each type of connectivity or correlation.

We illustrate an example of our Spatial Embedding Learning component in Fig. 11b. Fig. 11b shows that for each individual time interval, we form a graph of 5 nodes, which have three types of correlations between them denoted as {a, b, c} (illustrated by blue, red, and green edges), respectively. Three individual GCNs generate the embeddings of the 5 nodes for each connectivity type (*i.e.,* green, red and blue blocks). Then, an attention layer sums the embeddings over all three types weighted by the Softmax results of a learnable parameter vector, and returns the final $\mathbf{F}_l^{\text{G}}$ (yellow block).

• **Residual Structure.** We further take into account the residual structure for our MetaMobi, since it has been demonstrated to overcome the vanishing gradient problem in neural networks by introduction of identity shortcut connection [12,36,41]. Therefore, we introduce such residual structure of shortcut connections in our model designs, including the EmLearn component and the DiffLearn component.

We illustrate the residual structure in Figs. 10 and 11. For the EmBlock of the EmLearn module, we add to the output of Spatial Embedding Learning layer the input of each EmBlock with the same height of $\mathbf{F}_l^{\text{G}}$. In addition, a *Connection* layer learns the temporal pattern of the output of the Temporal Embedding Learning, $\mathbf{F}_l^{\text{Tp}}$, and adds the learned pattern to the output of previous *Connection* layers. In our design, the l-th *Connection* layer, denoted as $\texttt{Connection}_l$, is a Conv2D layer with kernel size of $(K - k_l^{\text{Tp}} \cdot l + 1) \times 1$, $l \in [0, L]$, and channel size of C^{Con}. We add a $\texttt{Connection}_0$ layer to capture the temporal patterns directly from the historical crowd flow inputs. The output of $\texttt{Connection}_l$ is denoted as $\mathbf{F}_l^{\text{Con}} \in \mathbb{R}^{1 \times N \times C^{\text{Con}}}$. For the last $\texttt{EmBlock}_L$, we use an additional $\texttt{Connection}_{L+1}$ layer to learn the temporal patterns of the output of $\texttt{EmBlock}_L$. The final output of the EmLearn module, $\mathbf{E} \in \mathbb{R}^{1 \times N \times C^{\text{Con}}}$, is given by

$$\mathbf{E} = \texttt{ReLu}\left(\sum_{l=0}^{L+1} \mathbf{F}_l^{\text{Con}}\right). \tag{6}$$

5.1.2 *Design of* DiffLearn *Module*

To further make MetaMobi adapt to the dynamically changing spatio-temporal characteristics learned by the EmLearn module, we have designed a DiffLearn module that differentiates the correlations of the learned spatio-temporal features between campus locations by an attention mechanism.

Specifically, given the outputs of the EmLearn component, $\mathbf{E} \in \mathbb{R}^{1 \times N \times C^{\text{Con}}}$, to compute the attention scores we first generate a query matrix, denoted as $\mathbf{Q} \in \mathbb{R}^{N \times C^{\text{Att}}}$, and a key matrix, denoted as $\mathbf{P} \in \mathbb{R}^{N \times C^{\text{Att}}}$, by

$$\mathbf{Q} = \texttt{Tanh}\left(\mathbf{E} \cdot \mathbf{W}^{\text{Q}}\right), \quad \mathbf{P} = \texttt{Tanh}\left(\mathbf{E} \cdot \mathbf{W}^{\text{P}}\right), \tag{7}$$

where $\mathbf{W}^{\text{Q}}$ and $\mathbf{W}^{\text{P}} \in \mathbb{R}^{C^{\text{Con}} \times C^{\text{Att}}}$ are learnable parameter matrices. Then the attention scores, $\mathbf{a} \in \mathbb{R}^{N \times N}$, between locations are computed by

$$\mathbf{a} = \texttt{Softmax}\left(\frac{\mathbf{Q} \cdot \mathbf{P}^{\intercal}}{\sqrt{C^{\text{Att}}}}\right), \tag{8}$$

where each element, $\mathbf{a}[i,j]$, represents the similarity of the spatio-temporal features between the locations i and j. Then we create another value matrix, $\mathbf{V} \in \mathbb{R}^{N \times C^{\text{Att}}}$, and conduct a weighted summation based on $\mathbf{a}$, *i.e.,*

$$\mathbf{V} = \texttt{Tanh}\left(\mathbf{E} \cdot \mathbf{W}^{\text{V}}\right), \quad \mathbf{H}^{\text{Att}} = \mathbf{a} \cdot \mathbf{V}, \tag{9}$$

where $\mathbf{W}^{\text{V}} \in \mathbb{R}^{C^{\text{Con}} \times C^{\text{Att}}}$ is another learnable parameter matrix. The final attention output $\mathbf{H} \in \mathbb{R}^{N \times 2}$ is

$$\mathbf{H} = \mathbf{H}^{\text{Att}} \cdot \mathbf{W}^{\text{Att}}, \tag{10}$$

where $\mathbf{W}^{\text{Att}} \in \mathbb{R}^{C^{\text{Att}} \times 2}$ is a learnable parameter matrix, in which the last dimension denotes inflow and outflow, respectively.

From the output of EmLearn, $\mathbf{E}$, we adopt two layers of Conv2D, denoted by OutLayers, and create another residual

structure by adding their output to the attention output $\mathbf{H}$ to generate the final prediction $\widehat{\mathbf{F}} \in \mathbb{R}^{N\times 2}$.

5.2 Meta Model Update with Bernoulli Gradient Dropout

5.2.1 Overview and Design Motivations

To make the core model of multi-dots connectivity-aware learning generalizable and adaptive to the dynamic of crowd mobility as well as the WLAN infrastructures across different semesters or AYs, we leverage the meta-learning algorithm [9, 24] to train the model on different sets of source data, *i.e.,* the datasets retrieved before WLAN infrastructure changes, in the Meta-Training phase. In the Pre-Training phase (given the changed crowd sensing infrastructures), we fix the parameters of the `EmLearn` module, θ^{Emb}, and train the `DiffLearn` parameters, θ^{Diff}. To overcome the overfitting problem raised by the few-shot setting of the target datasets, through a novel Bernoulli sampling process, `MetaMobi` drops some of the model parameters' gradients so that they will not be updated, enabling more flexibility and generalizability in capturing the changed mobility patterns.

We show in Alg. 1 the Meta-Training and Pre-Training phases of `MetaMobi`.

5.2.2 Objective Function and Training Designs

We leverage the mean square error (MSE) as the objective function to update the parameters. Specifically, given a dataset u, we formulate the objective function by

$$\mathcal{L}\left(\widehat{\mathbf{F}}_u, \mathbf{F}_u\right) = \frac{1}{2N}\sum\left(\widehat{\mathbf{F}}_u - \mathbf{F}_u\right)^2, \tag{11}$$

where $\mathbf{F}_u$ represents the ground-truth crowd flows in u, while $\widehat{\mathbf{F}}_u$ represents the predicted crowd flows by our multi-dots connectivity-aware learning.

Considering that the source datasets consist of the crowd flow data of U different semesters (in this campus prototype study, we consider partitioning based on semesters or AYs for ease of understanding different crowd mobility behaviors), for each semester, we select a pair of datasets (u, u') for our Meta-Training phase. In the Meta-Training phase, we first initialize the parameters of `EmLearn` and `DiffLearn` modules as in Line 1 of Alg. 1. The Meta-Training phase (Lines 2 – 11) contains an inner loop (Lines 3 – 8) and an outer loop (Lines 9 – 10). Specifically, in the inner loop, `MetaMobi` iterates over all the U semesters. In each iteration, we select one set of crowd flow data, denoted as u, and obtain the prediction $\widehat{\mathbf{F}}_u$. Then we evaluate the error of prediction by Eq. (11), and find $\mathbf{g}^{\text{Emb}}$ and $\mathbf{g}^{\text{Diff}}$ as the gradients of the θ^{Emb} and θ^{Diff} from the loss function, *i.e.,*

$$\mathbf{g}^{\text{Emb}} = \nabla_{\theta^{\text{Emb}}}\mathcal{L}\left(\widehat{\mathbf{F}}_u, \mathbf{F}_u\right), \tag{12}$$

and

$$\mathbf{g}^{\text{Diff}} = \nabla_{\theta^{\text{Diff}}}\mathcal{L}\left(\widehat{\mathbf{F}}_u, \mathbf{F}_u\right), \tag{13}$$

which are illustrated in Line 4.

To regularize the training process, traditional regularization approaches are designed for each single task. However, in a meta-learning setting [9] there are multiple training tasks, and each task corresponds to training on one source dataset (*e.g.,* learning the crowd flows of one semester in this work), where traditional regularization methods may not apply. To address this, we design a Bernoulli gradient dropout algorithm for model regularization. Specifically, our approach drops the gradients of some parameters (Line 5) in a stochastic manner (through Bernoulli sampling [31]) to prevent them from being updated, *i.e.,*

$$\widetilde{\mathbf{g}}^{\text{Emb}} = \mathbf{b} \odot \mathbf{g}^{\text{Emb}}, \tag{14}$$

and

$$\widetilde{\mathbf{g}}^{\text{Diff}} = \mathbf{b} \odot \mathbf{g}^{\text{Diff}}, \tag{15}$$

where $\odot$ denotes the element-wise multiplication operation, and the dropout matrix $\mathbf{b}$ (of the same size of the gradients) follows the Bernoulli distribution with a dropout rate of r. In other words, for each gradient w, we have

$$\mathbf{b}[w] = \begin{cases} 1 & \text{with probability of } (1-r), \\ 0 & \text{with probability of } r. \end{cases} \tag{16}$$

We update θ^{Emb} and θ^{Diff} (Line 6) by the new gradients $\widetilde{\mathbf{g}}^{\text{Emb}}$ and $\widetilde{\mathbf{g}}^{\text{Diff}}$ with a learning rate of γ, *i.e.,*

$$\theta^{\text{Emb}} = \theta^{\text{Emb}} - \gamma \cdot \widetilde{\mathbf{g}}^{\text{Emb}}, \tag{17}$$

Algorithm 1: Meta Model Update of `MetaMobi`.

Input : Total U pairs of (u, u') from the source datasets; a target dataset v.

1 Initialize model parameters θ^{Emb} and θ^{Diff};
`/* Meta-Training. */`
2 **while** *not converged* **do**
`/* Inner loop. */`
3 **for** *each (u, u′) pair* **do**
4 Compute the gradients of the `EmLearn` and `DiffLearn` modules by Eqs. (12) and (13);
5 Randomly drop gradients based on Eqs. (14) and (15);
6 Update θ^{Emb} and θ^{Diff} by Eqs. (19) and (20);
7 Evaluate the objective function for dataset u' by Eq. (11);
8 **end**
`/* Outer loop. */`
9 Compute the gradients of the `EmLearn` and `DiffLearn` modules by Eqs. (19) and (20);
10 Update θ^{Emb} and θ^{Diff} by the `Adam` optimizer;
11 **end**
`/* Pre-Training. */`
12 Fix θ^{Emb} and initialize `DiffLearn` parameters with the current θ^{Diff};
13 **while** *not converged* **do**
14 Compute the gradients of the `DiffLearn` module by Eq. (21) based on target dataset v;
15 Randomly drop gradients based on Eq. (15);
16 Update θ^{Diff} by Eq. (18);
17 **end**

and

$$\theta^{\text{Diff}} = \theta^{\text{Diff}} - \gamma \cdot \widetilde{\mathbf{g}}^{\text{Diff}}. \tag{18}$$

Then, we use the models with the updated parameters to generate prediction of another set of the crowd flow data, denoted as u', from the same semester and evaluate the loss function by Eq. (11) (Line 7).

With the obtained loss functions from all U semesters, the outer loop takes an average of the errors and returns the gradients by the average (Line 9), *i.e.,*

$$\mathbf{g}'^{\text{Emb}} = \nabla_{\theta^{\text{Emb}}} \frac{1}{U} \sum_{u'}^{U} \mathcal{L}\left(\widehat{\mathbf{F}}_{u'}, \mathbf{F}_{u'}\right), \tag{19}$$

and

$$\mathbf{g}'^{\text{Diff}} = \nabla_{\theta^{\text{Diff}}} \frac{1}{U} \sum_{u'}^{U} \mathcal{L}\left(\widehat{\mathbf{F}}_{u'}, \mathbf{F}_{u'}\right). \tag{20}$$

Then we update θ^{Emb} and θ^{Diff} with the gradients $\mathbf{g}'^{\text{Emb}}$ and $\mathbf{g}'^{\text{Diff}}$, respectively, by an `Adam` optimizer [17] (Line 10). At the stage of convergence, θ^{Emb} and θ^{Diff} are updated based on the source dataset so that the model parameters are generalizable for dynamics of crowd flow patterns.

The Pre-Training phase (Lines 12 – 17) enhances the adaptation of the model to the target datasets. Specifically, we fix θ^{Emb} and initialize `DiffLearn`'s parameters with the θ^{Diff} learned from the Meta-Training. Then, we sample a small set of target datasets, v, and compute the gradients of the `DiffLearn` module (Line 14) by

$$\mathbf{g}^{\text{Diff}} = \nabla_{\theta^{\text{Diff}}} \mathcal{L}(\widehat{\mathbf{F}}_v, \mathbf{F}_v). \tag{21}$$

Following the same manner of Eq. (15), we drop the gradients of some parameters of θ^{Diff} (Line 15). Afterwards, we update θ^{Diff} by Eq. (18) (Line 16). After convergence, the core model of multi-dots connectivity-aware learning in `MetaMobi` becomes capable of capturing the new spatio-temporal crowd mobility patterns and adapting to the changed crowd sensing infrastructure.

6 Experimental Results

In this section, we conduct extensive experimental studies to test the effectiveness of `MetaMobi` based on our processed crowd flow data from Wi-Fi association records. We present the experimental settings in Sec. 6.1. Then we discuss the experimental results in Sec. 6.2.

6.1 Experimental Settings

We compare the performance of `MetaMobi` with the following baseline or state-of-the-art approaches.

- `HA`, `RF`, and `DT`: the traditional time-series or machine learning approaches that respectively predicts the crowd flows based on historical average (`HA`) [10], random forest (`RF`) [4], and decision tree (`DT`) [25].
- `LSTM` [13]: which leverages the long short-term memory to predict the crowd flow time series.
- `GCN` [18]: which uses the graph convolution network with geospatial distance as the correlations between nodes for the crowd flow prediction.

Table 1: Overall performance of all studied approaches.

Datasets	**All**		**Existing**		**New**	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE
`HA`	41.311	3.389	41.634	3.335	38.660	3.830
`RF`	39.568	3.602	39.295	3.541	41.810	4.107
`DT`	63.665	4.047	63.326	3.927	66.446	5.032
`LSTM`	43.182	3.452	43.667	3.403	39.198	3.849
`TPA-LSTM`	48.551	3.385	48.357	3.295	50.143	4.119
`CNN`	41.757	3.425	41.096	3.305	47.186	4.411
`DeepST`	44.958	3.637	44.083	3.495	52.144	4.808
`GAT`	87.517	5.700	88.490	5.675	79.529	5.910
`GCN`	88.302	5.401	88.675	5.323	85.237	6.039
`GCNN`	57.176	5.025	57.363	4.991	55.636	5.302
`GCN-LSTM`	43.409	3.480	43.577	3.416	42.030	4.003
`GraphWaveNet`	34.995	3.241	34.578	3.147	38.422	4.008
`MetaMobi`	31.551	2.858	31.181	2.775	34.590	3.538

- `GAT` [32]: which uses the graph node attention to predict the crowd flow.
- `GCNN` [20]: which considers the adjacency matrix as trainable matrices and historical values as the node features to generate the inflows/outflows.
- `GCN-LSTM`: which combines the `GCN` and `LSTM` to predict the crowd mobility.
- `CNN`: which uses 2 layers of 2D convolutional neural network (`CNN`) to predict the crowd mobility.
- `DeepST` [41]: which takes the crowd flow heatmaps as the input and adopts three blocks of spatio-temporal residual neural networks [41] to capture the short-, mid-, and long-term mobility patterns.
- `TPA-LSTM` [27]: which leverages `LSTM` with temporal attention to forecast the multivariate time series.
- `GraphWaveNet` [37]: which captures the spatial features by a self-adaptive mechanism and the temporal features by a stacked dilated 1D convolution component.

Parameter Settings. The parameters of `MetaMobi` are listed below. A total of 2 blocks of `EmBlocks` were used; for each layer, we adopt a kernel size $k_l^{\text{Tp}} = 8$. We set the channel sizes $C^{\text{Inp}} = C^{\text{Tp}} = C^{\text{Con}} = 32$, and $C^{\text{Att}} = 16$. We used a gradient dropout rate $r = 0.1$ and a learning rate $\gamma = 5 \times 10^{-5}$. We used a batch size of 64. `MetaMobi` was trained for 3,000 epochs in both Meta-Training and Pre-Training phases. For all the baselines we use the default parameters and leverage the crowd flows as inputs. For the graph-based baselines including `GCN`, `GCN-LSTM` and `GraphWaveNet`, the geo-spatial distance graph is considered by the adjacency matrix.

Experimental Settings. Recall that we retrieve the crowd flow data based on the Wi-Fi association records during 10-10-2020 to 11-22-2021. We form the *source* datasets by the crowd flows of the 124 buildings for Fall 2020 and the 125 buildings for Spring 2021 semesters. In the Meta-Training phase, we sample 40 days for each semester from the source datasets, with the first 30 days to train the inner loop and the rest 10 days to train the outer loop. Specifically, we sample 10-10-2020 to 11-09-2020 for Fall 2020 and 02-08-2021 to 03-19-2021 for Spring 2021. The selected buildings have few new AP installed and the maximum hourly flows have reached or exceeded 20 during the sampled periods.

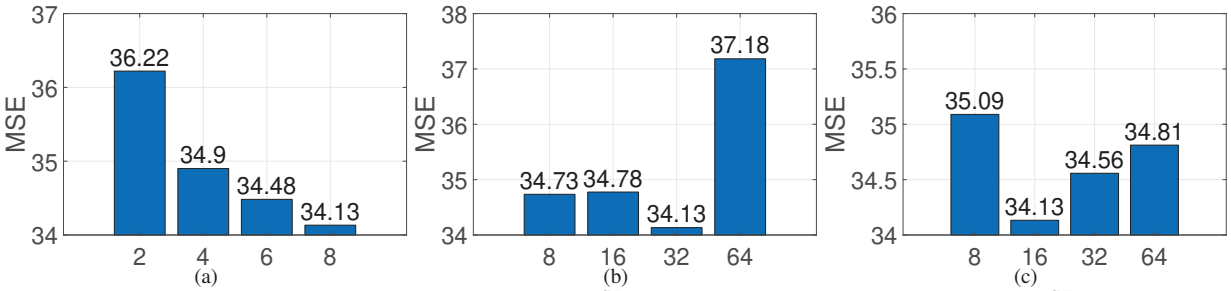
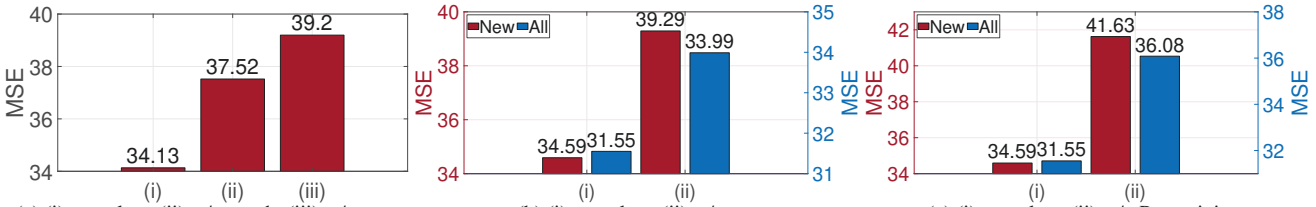


Figure 12: Sensitivity study of MetaMobi on: (a) kernel size, k_l^{ST}; (b) the channel size of the EmBlocks, C^{ST}; and (c) the channel size of the attention layer, C^{Att}.



(a) (i) complete; (ii) w/o graph; (iii) w/o con. (b) (i) complete; (ii) w/o att. (c) (i) complete; (ii) w/o Pre-training.

Figure 13: Results of our ablation studies on MetaMobi.

From our data analysis, we observe the majority of the new APs are installed in the summer 2021. Therefore, we form the *target* datasets by the crowd flows for Fall 2021. We use the first week of Fall 2021, 08-30-2021 to 09-05-2021, from target datasets for Pre-Training, and the following two weeks for testing. We select 221 campus spots including 118 buildings and 103 APs following the same manners as the source datasets, among which we identify 24 campus spots including new APs and buildings with over 90% of APs newly installed.

We evaluate the performance of MetaMobi and baselines on all the 221 campus spots as well as the 24 locations with new APs installed. We use Mean Square Errors (MSEs) and Mean Absolute Errors (MAEs) as the performance evaluation metrics to capture the approaches' robustness to large dynamics as well as the overall predictability. All models were developed in PyTorch, and we train and test these models upon a Linux HPC Server with AMD Ryzen 3960X 3.8GHz, 16GB RAM, and 4 × Nvidia GeForce RTX 3090. The Meta-Training phase of MetaMobi costs around 2.8s per building, and the Pre-Training takes around 0.8s per campus location.

6.2 Experimental Results

Overall Performance: We show the overall performance in Tab. 1. From the table, we see that MetaMobi demonstrates a better performance in the prediction for both campus spots with new APs installed (denoted by "New") and others with existing APs (denoted by "Existing"). We can observe that MetaMobi achieves overall better performance than all the baseline and state-of-the-art approaches.

The HA could not capture the dynamic of spatio-temporal characteristics, while other traditional regression methods like RF (random forest) and DT (decision tree) failed to extract the essential spatio-temporal information for accurate prediction. The recurrent neural network, LSTM, is only able to capture the temporal patterns. On the contrary, the GCN, GAT, and CNN are only able to capture the spatial patterns due to the lack of the temporal designs. Although GCN-LSTM has both spatial and temporal designs, the LSTM could not preserve the spatial information captured by the GCN module. The state-of-the-art approaches such as TPA-LSTM, GCNN, DeepST, and GraphWaveNet did not consider modeling the building functions and differentiating the spatio-temporal correlations across the campus locations, thereof achieving degraded performance in our studies.

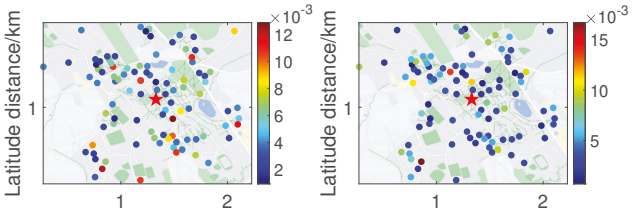
MetaMobi, however, learn the temporal patterns through a series of Conv2D layers and capture contributions of different types of spatial correlations, including distance and building functionalities, to the spatial embedding learning. In addition, it learns the dynamic spatio-temporal features through the attention mechanism, making it generalizable across different datasets with overall better performance.

Model Sensitivity and Ablation Studies: We have conducted our model sensitivity based on the source dataset during the periods between 03-20-2021 and 03-29-2021. We study the effects of *(a)* the kernel size, denoted as k_l^{ST}, *(b)* the channel size of the EmBlocks, denoted as C^{ST}, and *(c)* the channel size of the attention layer, denoted as C^{Att}, on the model performance. Fig. 12a shows that setting the kernel size of 8 has overall the best performance, implying that 8 time intervals provide the most temporal information in our study. Fig. 12b further shows that the model has the best performance given $C^{\text{ST}} = 32$, while the MSE increases when the channel size further increases. This may be attributed to more noisy information retrieved by a large weight. Similar trends can be observed in Fig. 12c for C^{Att}. When C^{Att} is set as 8, it may not suffice to capture the spatio-temporal correlations between campus locations. However, when C^{Att} is larger than 32, more redundant information may be extracted, leading to the accuracy degradation.

We have conducted model ablation studies on various structures of MetaMobi. We show in Fig. 13 the effectiveness of our model designs. Specifically, we evaluate the MSE of MetaMobi based on the 10 days' crowd flow data

of the 125 buildings with few new APs installed during the period between 03-20-2021 and 03-29-2021. We compare *(i)* complete model (denoted by "complete") with: *(ii)* `MetaMobi` without the Multi-Graph Attention Network as the Spatial Embedding Learning in the `STBlock` (denoted by "w/o graph"), and (iii) `MetaMobi` without all the Connection layers within the `EmLearn` except the last one (denoted by "w/o con"). The results are shown in Fig. 13a, from which we can see that the Multi-Graph Attention Network and the Connection layers both have important contributions to the prediction accuracy, demonstrating the effectiveness of the different types of the spatial information extracted by Multi-Graph Attention Network and the residual structure provided by the connection layers.

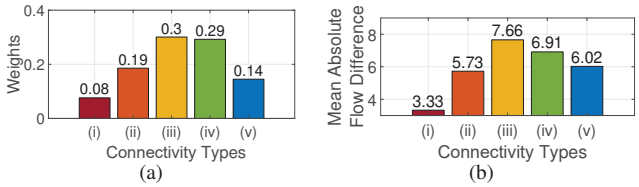
We also evaluate the effectiveness of the `DiffLearn` module on adapting to the target datasets. The evaluation is based on the two weeks' testing set of target dataset from 09-06-2021 to 09-19-2021. We show the results of the complete model compared to the model without the attention layer (denoted by "w/o att") in Fig. 13b. From the figure, we can see the attention mechanism improves the performance on target datasets, demonstrating that the attention mechanism helps the model adapt to the different spatio-temporal mobility patterns from the source datasets. In addition, we further compare the complete model with the one without the pre-training phase, and show the results in Fig. 13c. We can see that pre-training is necessary to update the model with new mobility patterns.



(a) 09-06-2021 11 a.m. (b) 09-06-2021 11 p.m.

Figure 14: Spatial distributions of the attention scores of two selected time slots for an academic building (star).

Visualization: We first illustrate the spatial distributions of the attention scores between an academic building (star) located at the campus center and various campus locations in Fig. 14. We can observe more active correlations at around 11 a.m. than those at 11 p.m., indicating more crowd mobility between that building and the rest of campus captured by `MetaMobi`.



(a) (b)

Figure 15: Illustration of (a) attention weights and (b) mean absolute flow difference on 09-12-2021 of the 5 connectivity types, *i.e.,* (i) geo-spatial distance, (ii) food, (iii) residential, (iv) athletic, and (v) academic.

We further illustrate the attention weights and the mean absolute flow difference between the buildings on 09-12-2021 that are associated with the five different connectivity types, *i.e.,* (i) geo-spatial distance, (ii) food, (iii) residential, (iv) athletic, and (v) academic, in Figs. 15a and 15b. Comparing Fig. 15a with Fig. 15b, we can observe that in general the attention weights calculated by `MetaMobi` capture the mean absolute flow difference, demonstrating the adjacency matrix of each graph indicates the mobility trends between buildings with and without the type of functionalities as discussed in Sec. 5.1.1. For (i), we evaluate the buildings that are close to each other, and find the average absolute differences (in terms of inflows) of hourly flows between all pairs of these neighbors. For (ii) – (v), we find the buildings that belong to the same function type, and find the above mean absolute flow differences. From the figure, we can also observe that overall stronger attractions (and subsequent connectivities) among the buildings with the residential and athletic functions. It is mainly due to the frequent commutes of the students between residential halls usually located at the peripheral of the campus and the athletic facilities around the campus center as shown in Fig. 15b.

7 Discussion

Inclusion of Other Factors: While in this prototype studies, we focus on learning the crowd mobility patterns through the harvested Wi-Fi data, `MetaMobi` is general enough to extend to other signals [14, 35] for more ubiquitous crowd mobility learning. Furthermore, despite our focus on APs that are newly installed (the campus WLAN infrastructure is expanding), our model is adaptive and can accommodate the scenarios when a lot of APs are removed. Other factors such as various campus events held in buildings/classrooms may also significantly influence the crowd flows, which will be investigated in our future studies.

Privacy Consideration: We have discussed with the university internal review board (IRB) and since all sensitive individual identity information has been sanitized, and hence there is no need for IRB approvals for the current studies. Further location privacy preservation for crowd mobility sensing and modeling can be referred to other prior studies [8, 43].

Use Case Generalizability: While this study focuses on campus Wi-Fi association data for our prototype development here, the general insights and models from our work can be easily extended to other more spacious or complex urban environments [22], or use cases such as urban transportation system flow modeling [21] and internet of things (IoTs) traffic prediction [2].

8 Conclusion

We have proposed `MetaMobi`, a novel approach for crowd mobility learning. `MetaMobi` includes a novel multi-dots connectivity-aware learning approach, which jointly learns the crowd flow time series of multiple buildings with fusion of spatial graph connectivities and temporal attention mechanisms. Furthermore, to overcome the adaptivity issues due to changes in crowd sensing infrastructures (*e.g.,* installation of new APs), `MetaMobi` further includes a novel meta model update approach with Bernoulli gradient dropout approach. Extensive experimental studies based on the real-

world Wi-Fi data have validated the effectiveness and accuracy of MetaMobi in predicting crowd flows.

Acknowledgement

We would like to thank the anonymous reviewers and the shepherd Dr. Shijia Pan for their constructive comments. We would like to thank the UConn Information Technology Services (UITS) for their assistance in collecting Wi-Fi association data. This project was supported, in part, by the 2021 NVIDIA Applied Research Accelerator Program Award and National Science Foundation (NSF) under Grant 2118102. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

9 References

[1] Crowd Analytics Market Statistics — Forecast - 2030. https://www.alliedmarketresearch.com/crowd-analytics-market, 2022.

[2] A. R. Abdellah and A. Koucheryavy. Deep learning with long short-term memory for iot traffic prediction. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pages 267–280. Springer, 2020.

[3] A. Antoniou, H. Edwards, and A. Storkey. How to train your MAML. *arXiv preprint arXiv:1810.09502*, 2018.

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] L. Chen, J. Jakubowicz, D. Yang, D. Zhang, and G. Pan. Fine-grained urban event detection and characterization based on tensor cofactorization. *IEEE Transactions on Human-Machine Systems*, 47(3):380–391, 2016.

[6] J. Chu, X. Wang, K. Qian, L. Yao, F. Xiao, J. Li, and Z. Yang. Passenger demand prediction with cellular footprints. *IEEE TMC*, 2020.

[7] A. Draghici, T. Agiali, and C. Chilipirea. Visualization system for human mobility analysis. In *Proc. RoEduNet NER*, pages 152–157. IEEE, 2015.

[8] Z. Fang, B. Fu, Z. Qin, F. Zhang, and D. Zhang. PrivateBus: Privacy Identification and Protection in Large-Scale Bus WiFi Systems. *Proc. ACM IMWUT*, 4(1), Mar. 2020.

[9] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. PMLR, 2017.

[10] J. E. Froehlich, J. Neumann, and N. Oliver. Sensing and predicting the pulse of the city through shared bicycling. In *Proc. IJCAI*, 2009.

[11] C. Gentry and Z. Ramzan. Single-database private information retrieval with constant communication rate. In *Proc. ICALP*, pages 803–815. Springer, 2005.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, pages 770–778, 2016.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] X. Hu, H. Zheng, Y. Chen, and L. Chen. Dense crowd counting based on perspective weight model using a fisheye camera. *Optik*, 126(1):123–130, 2015.

[15] R. Jiang, Z. Cai, Z. Wang, C. Yang, Z. Fan, Q. Chen, X. Song, and R. Shibasaki. Predicting citywide crowd dynamics at big events: A deep learning system. *ACM TIST*, 13(2):1–24, 2022.

[16] R. Jiang, Z. Cai, Z. Wang, C. Yang, Z. Fan, Q. Chen, K. Tsubouchi, X. Song, and R. Shibasaki. DeepCrowd: A deep model for large-scale citywide crowd density and flow prediction. *IEEE TKDE*, 2021.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[19] K. H. Lim, J. Chan, S. Karunasekera, and C. Leckie. Tour recommendation and trip planning using location-based social media: A survey. *Knowledge and Information Systems*, 60(3):1247–1275, 2019.

[20] L. Lin, Z. He, and S. Peeta. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. *Transportation Research Part C: Emerging Technologies*, 97:258–276, 2018.

[21] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin. DeepSTN+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. In *Proc. AAAI*, volume 33, pages 1020–1027, 2019.

[22] M. Liu, L. Li, Q. Li, Y. Bai, and C. Hu. Pedestrian flow prediction in open public places using graph convolutional network. *ISPRS International Journal of Geo-Information*, 10(7):455, 2021.

[23] A. Nichol and J. Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

[24] Z. Pan, W. Zhang, Y. Liang, W. Zhang, Y. Yu, J. Zhang, and Y. Zheng. Spatio-temporal meta learning for urban traffic prediction. *IEEE TKDE*, 2020.

[25] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[26] J. Rajendran, A. Irpan, and E. Jang. Meta-learning requires meta-augmentation. *Proc. NeurIPS*, 33:5705–5715, 2020.

[27] S.-Y. Shih, F.-K. Sun, and H.-y. Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108(8):1421–1441, 2019.

[28] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proc. IEEE CVPR*, pages 1199–1208, 2018.

[29] M. Tabatabaie, S. He, and X. Yang. Driver maneuver identification with multi-representation learning and meta model update designs. *Proc. ACM IMWUT.*, 6(2), Jul 2022.

[30] P. Tian, W. Li, and Y. Gao. Consistent meta-regularization for better meta-knowledge in few-shot learning. *IEEE TNNLS*, 2021.

[31] H.-Y. Tseng, Y.-W. Chen, Y.-H. Tsai, S. Liu, Y.-Y. Lin, and M.-H. Yang. Regularizing meta-learning via gradient dropout. In *Proc. ACM ACCV*, 2020.

[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[33] L. Wang, D. Chai, X. Liu, L. Chen, and K. Chen. Exploring the generalizability of spatio-temporal traffic prediction: Meta-modeling and an analytic framework. *IEEE TKDE*, 2021.

[34] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang. Crowd flow prediction by deep spatio-temporal transfer learning. *arXiv preprint arXiv:1802.00386*, 2018.

[35] X. Wang, Z. Zhou, Y. Zhao, X. Zhang, K. Xing, F. Xiao, Z. Yang, and Y. Liu. Improving urban crowd flow prediction on flexible region partition. *IEEE TMC*, 19(12):2804–2817, 2019.

[36] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proc. ACM SIGKDD*, pages 753–763, 2020.

[37] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

[38] X. Yang, S. He, B. Wang, and M. Tabatabaie. Spatio-Temporal Graph Attention Embedding for Joint Crowd Flow and Transition Predictions: A Wi-Fi-Based Mobility Case Study. *Proc. ACM IMWUT*, 5(4), dec 2022.

[39] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *Proc. WWW*, pages 2181–2191, 2019.

[40] D. Zeng, Z. Cao, and D. B. Neill. Artificial intelligence–enabled public health surveillance—from local detection to global epidemic monitoring and control. In *Artificial Intelligence in Medicine*, pages 437–453. Elsevier, 2021.

[41] J. Zhang, Y. Zheng, and D. Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proc. AAAI*, volume 31, 2017.

[42] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li. Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence*, 259:147–166, 2018.

[43] Y. Zhao, X. Wang, J. Li, D. Zhang, and Z. Yang. CellTrans: Private Car or Public Transportation? Infer Users' Main Transportation Modes at Urban Scale with Cellular Data. *Proc. ACM IMWUT*, 3(3), Sept. 2019.