Cross-Zone and Extreme-Aware Mobility Learning of Crowd Interactions with Built Environments

Suining He[1] Bing Wang[1] Kang G. Shin[2] Mahan Tabatabaie[1]
University of Connecticut[1] University of Michigan[2]
{suining.he, bing, mahan.tabatabaie}@uconn.edu[1], kgshin@umich.edu[2]

ABSTRACT

We propose an ***A***daptive ***Cro***wd mobility analytics system based on ***C***ross-zone ***I***nteractive learning (**ACroCI**) to capture, interpret, and forecast how the mobility of human crowds *interacts* with built or man-made environments (e.g., building functions, event occurrences, and changes in the crowd sensing infrastructures). We have conducted a large-scale real-world case study of ACroCI by leveraging the collective and anonymized association data harvested from the campus Wi-Fi access points (APs), to understand how the interactions affect the forecast of crowd flows. We have analyzed the large-scale Wi-Fi association data and derived adaptive learning and data-driven designs on important crowd mobility features like multi-level co-flows and local-global cross-zone interactions. ACroCI accounts for these interactive features and adaptively learns the multi-scale crowd distributions with a novel capsule neural network augmented by interactive attention routing, and accurately predicts the arrivals at, and departures from, each AP. Strengthened by multi-task extreme-aware learning and efficient data imputation, ACroCI further adapts to extreme flows when the crowds interact with events, and initializes its model for altered APs. Our extensive experimental evaluations with $>4.8 \times 10^7$ association data from two large universities have corroborated the accuracy, adaptivity, robustness, and effectiveness of ACroCI in forecasting the crowd interactions with the man-made environments (in terms of crowd flows), achieving a >20% accuracy improvement over the other state-of-the-arts.

1 INTRODUCTION

Crowd mobility analytics (CMA) — i.e., interpreting and understanding the movement of crowds — have become increasingly important for many spacious urban and/or public places, such as college campuses, large airports, and malls, where many people are likely to gather together within and across the human-built or artificial environments like building rooms, corridors, and sidewalks. While the social and behavioral analysis of crowds [11, 24], including the interactions among crowds, has been extensively studied in various ubiquitous and urban computing contexts, how crowds *interact* with the built or man-made environments, such as the building functions, event occurrences inside/across buildings, and infrastructure alterations or changes [2], and how such interaction may benefit mobility modeling, remain a largely unexplored but important subject of CMA.

Human crowds usually move responsively given various built environment settings. For instance, the college students may form certain travel routines on a campus, given their interpretation of relative proximity of buildings (as well as their purposes) such as residential halls (living), recreational centers (recreation), dining halls (dining), and libraries (studies). One may, therefore, expect interactive co-presences of crowds at multiple locations during certain periods of a day. Crowds may also interactively pick alternative entrances, exits, or routes in response to closure of certain corridors or sidewalks. Understanding and incorporating such dynamic interactions within the CMA system design, particularly learning and predicting interactive crowd movements, will benefit various important ubiquitous, mobile, and urban applications, such as event monitoring and facility management.

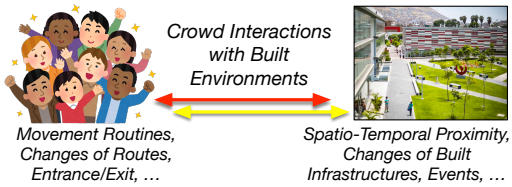

Fig. 1: Illustration of design motivations.

To this end, we propose a novel CMA system with insights of such interactions, using collective and anonymized Wi-Fi association and disassociation data harvested from a campus network infrastructure as a practical case study. Through the system deployment study, we would like to understand how incorporating interactions, as illustrated in Fig. 1(a), between human crowds and the built environments impacts the accuracy and effectiveness of CMA, especially in predicting the mobility of crowd flows. Towards such a ubiquitous CMA system, we must address the following two major technical challenges:

(1) Lack of modeling the local-global and cross-zone interactions with the man-made environments: Crowd mobility is highly complex and often involves *multi-scale* complexities due to hourly, daily, and weekly routines, periodic personal activities and preferences. Furthermore, *co-presences* and *co-flows* of crowds occur not only *locally* at neighboring campus zones but also *globally* across *distant* zones. Conventional learning without considering multi-scale and local-global dependency and interaction modeling cannot capture the underlying sophisticated campus activities and provide satisfactory prediction.

(2) Absence of extreme awareness and adaptivity to interactions with dynamic sensor measurement environments: Besides the inherent spatial and temporal complexities, the distributions of crowds may be affected greatly by transient or abnormal

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3564065

campus events (e.g., a conference gathering or an emergency), leading to a surge or drop of crowd interaction behaviors (say, arrivals or departures). Conventional *extreme-agnostic* Wi-Fi data learning, however, fails to take into account the potential *extreme awareness* within the crowd mobility data. Furthermore, the Wi-Fi infrastructure may change due to network maintenance and reconfiguration, which can impact the Wi-Fi data collection and subsequent model learning. In particular, how to learn and predict crowd mobility with APs that are newly-introduced or relocated without historical data for model training, i.e., handling "*cold-start*" for model initialization, remains largely unexplored.

To address the above challenges, we propose an **A**daptive **Cro**wd mobility analytics system based on **C**ross-zone **I**nteractive learning (**ACroCI**). To better understand how the crowd interactions with the built/artificial environments affect the mobility modeling, we have further designed a crowd flow prediction experiment, i.e., forecasting the flows of crowds at each individual Wi-Fi AP, as a practical and easy-to-evaluate case study. In this case study, at each time interval, for each AP, the *in-flow* is considered to represent the total number of clients entering its close proximity, while the *out-flow* denotes that of those leaving. Specifically, our crowd interaction studies with ACroCI take in the AP-level associations/disassociations and their zone-level representations, statistical time-series features, and external factors. Then, our crowd interaction learning framework, with a novel interactive **A**ttention **R**outing **Cap**sule network (ARCap) as the core, forecasts the AP-level crowd flows.

This paper makes the following contributions:

C1. Spatio-temporal interactive campus crowd mobility analytics (Sec. 2): We have conducted a comprehensive analysis of campus mobile associations and crowd flows using real-world Wi-Fi association datasets, one collected from our university campus and an open-sourced dataset from a university in Northern Europe [18]. We have derived several spatial and temporal representation designs, motivations, and insights regarding the interactions between the crowds and the built environments, including multi-scale temporal and co-flow complexities, local-global cross-zone dependencies, as well as extreme crowd flows, to motivate our adaptive interaction learning model formulation.

C2. Learning interactions between crowds and built environments (Sec. 3): Within ACroCI, the complex spatial correlations across different APs and their neighborhoods are characterized via a novel spatio-temporal mobility heatmap representation, and the crowd flow patterns are further learned by a novel attention routing capsule neural network. The multi-scale co-flows and local-global cross-zone interactions between the crowds and the built environments can be jointly and interactively captured by our novel vectorized capsule structure with attention routing. Then, ACroCI predicts the AP-level crowd flows based on the fusion of multiple learners, yielding high accuracy and robustness.

C3. Awareness and adaptivity augmentation (Sec. 4): We have further designed a novel auxiliary module with novel multi-task extreme-aware learning to capture statistical crowd flow time-series features and forecast the extreme crowd distributions introduced by transient and potentially abnormal university events. We have also designed an efficient data imputation mechanism for our CMA system to handle the cold-start issue of newly-introduced or relocated APs after network reconfiguration.

C4. Extensive experimental evaluations (Sec. 5): With association data from the above-mentioned university campuses, we have conducted extensive experimental evaluations with $>4.8\times10^7$ Wi-Fi association records from over 2.36×10^5 clients in total. These results have corroborated the importance of incorporating interactions between crowds and the built environments, as well as accuracy, robustness, extreme-awareness, and adaptivity of ACroCI in predicting the campus mobility and crowdedness.

2 MOTIVATIONS AND FORMULATIONS

2.1 Crowd Mobility Data Pre-processing

Crowd Mobility Data: ACroCI is based on the following two large-scale campus WLANs.

(a) Campus A: We used our information technology (IT) service of University of Connecticut to collect the Wi-Fi association and disassociation records at our university campus (denoted as Campus A) via the Cisco Prime Infrastructure. The selected APs with the given GPS coordinates (longitudes and latitudes) cover a campus area of 4.4036×10^6 m^2. 1,059 APs and 126,923 clients are studied, obtaining a total of 9.677×10^6 records over a 15-month data collection period (2015/06–2016/09). Each record corresponds to one AP association event, with the MAC addresses of both the associated AP and the user's connected client device, user name (ID), the start time as well as the association duration, and the IP address. We infer the departure or disassociation time by adding the association duration to the start time.

(b) Campus B: The 880 APs selected from another university campus in Northern Europe [18] (denoted as Campus B) cover an area of 1.308×10^6 m^2 and 109,197 clients, producing a total of 3.8593×10^7 Wi-Fi association records during the 16-month long data collection period (2014/01–2015/04).

Crowd Mobility Data Preprocessing: For Campus A, the data collection of individual/crowd locations is enabled by a single-sourced Wi-Fi service provision system managed by our university IT service, and the user anonymity, including privacy of the user ID and device's MAC address, is maintained. Related privacy and ethical considerations have been discussed and vetted by the university's Institutional Review Board or IRB; we were informed that no IRB application was required as only aggregate anonymized data is used in our study. We have anonymized the user names (IDs), client device MAC addresses, and IP addresses (if any) of the association records. We use these anonymized user IDs to map devices to individual users, hence mapping the Wi-Fi association data to the crowd flows, and the IDs are discarded immediately after the mapping. Identities in the dataset from Campus B are also anonymized before its distribution to preserve individual privacy. We infer the AP-level crowd flows (in/out) by:

AP-level Associations/Disassociations: For ease of modeling, we discretize the time domain into intervals (say, each with 60min in our settings). The period k and the index k are used interchangeably, both referring to the k-th time interval. Let M be the number of access points (APs) on a campus. Then, we denote the number of associations and disassociations at AP i ($i \in \{1,\ldots,M\}$) in the k-th time interval as $A_i^{(k)}$ and $D_i^{(k)}$, which form the resultant M-dimensional (M-d) vectors of AP-level associations and disassociations, i.e., $\mathbf{A}^{(k)} = \left[A_1^{(k)},\ldots,A_M^{(k)}\right]$ and $\mathbf{D}^{(k)} = \left[D_1^{(k)},\ldots,D_M^{(k)}\right]$.

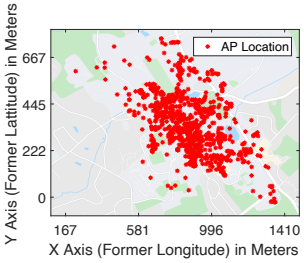

Fig. 2: AP locations studied at Campus A.

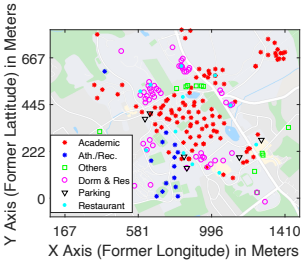

Fig. 3: Points of interests (POIs) at Campus A.

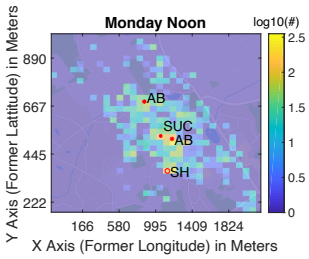

Fig. 4: Heatmap of associations at Campus A (a Monday noon).

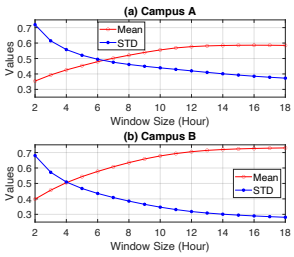

Fig. 5: Multi-level temporal interactions.

2.2 System Design Motivations

We derive the following data-driven motivations regarding interactions of the crowds with the built environments on campus (denoted as **M1**–**M5**).

M1 – Spatial Interactions with Points-of-Interest (POIs).

Taking our university campus (Campus A) as an example, we show in Fig. 2 the AP locations, demonstrating the pervasive coverage of the campus Wi-Fi infrastructures. We also show in Fig. 3 the 219 points of interests (POIs) on Campus A, including academic buildings, athletics or recreation facilities, dormitory and residential area, restaurants and cafes, and parking lots. The geolocations of the POIs are derived from the free editable geographic database OpenStreetMap. We illustrate the spatial heatmap of aggregated AP associations (in $\log_{10}(\cdot)$) during 12:00–13:00 of Monday in Fig. 4. On Monday dense AP associations and "implied" crowdedness can be observed from multiple locations such as academic buildings (denoted as AB) and the student union canteen (denoted as SUC). However, we also observe that on Sunday (the same time period) the magnitude is much lower and student halls (denoted as SH in Fig. 4) as well as academic buildings remain to be hotspots.

One can expect that the campus facility and building distributions have strong impacts on the crowd flows. However, given such a complex and non-uniform distribution of APs, it is very difficult to model their mutual *spatial* interactions for effective crowd interaction learning. To handle the above issue, we leverage the inspiration of image learning, and conduct zone discretization to obtain the aggregate zone-level association data as the important inputs and feature representations for our model training.

Zone Discretization, Zone-level Associations, & Disassociations: We discretize the campus map into G *zones*. Specifically, we divide the map longitudinally and latitudinally into $W \times H$ equal-sized rectangular *grids*. Note that the shape and size of the zones can be customized according to each specific data analysis and prediction task need, and adapted to the map/building accessibility.

We aggregate the number of Wi-Fi associations or crowd arrivals at all APs installed within zone j in the interval k as $Y_j^{(k)}$. Then, we have the zone-level association as $\mathbf{Y}^{(k)} = \left[Y_1^{(k)}, \ldots, Y_G^{(k)}\right]$. Similarly, we find the disassociations or crowd departures as $O_j^{(k)}$, as $\mathbf{O}^{(k)} = \left[O_1^{(k)}, \ldots, O_G^{(k)}\right]$. Then, for each time interval k, we process the given zone-level Wi-Fi associations/disassociations into sequential mobility heatmap frames, denoted as $\mathbf{F}^{(k)}$, as *spatio-temporal mobility representations* for the model input. Note that while the spatio-temporal mobility representations are at the zone level, our prediction output is at each *individual* AP for fine-grained crowd monitoring.

M2 – Multi-scale Temporal Interactions.

We have investigated the temporal intervals to evaluate the multi-scale temporal correlations. Recall that the time domain has been discretized into multiple intervals, each 1 hour long in our setting. Then, by varying the sliding window of T consecutive time intervals, we evaluate the temporal correlations between the associations of APs i and j based on the Pearson correlations, i.e., $\text{corr}\,(i,j) = \frac{\sum_{t=1}^{T}\left(A_i^{(t)} - \bar{A}_i\right)\cdot\left(A_j^{(t)} - \bar{A}_j\right)}{\sqrt{\sum_{t=1}^{T}\left(A_i^{(t)} - \bar{A}_i\right)^2}\sqrt{\sum_{t=1}^{T}\left(A_j^{(t)} - \bar{A}_j\right)^2}}$, where $\bar{A}_i$ represents the mean flow in the window of T intervals at AP i. Similar results can be observed from departures. By varying the sliding window size, we show the means and variations of the temporal correlations of APs at each campus in Fig. 5. Overall, Campus A experiences the lower mean correlations and higher variations than Campus B. From both campus datasets, we can observe that for short-term intervals (say, <3 hours), the correlations tend to be small and varied, which is likely due to the short-term mobility dynamics. On the other hand, for the long-term intervals (say, >9 hours) the correlations tend to be large and consistent, mainly because there exist mobility patterns on various campus function zones. After the sliding window reaches a certain value (say, >12 hours in our cases), the temporal correlations start to converge.

M3 – Co-Flow and Local-Global Interactions.

Complex crowd interactions with different built environments (e.g., buildings) often result from their functions, which can be characterized by their points-of-interest (POIs) in Fig. 3. We further look at the APs in the same POI and obtain the correlations among APs at a pair of different POIs. Specifically, we consider several POIs that are closely related to a student's daily life: *(a)* academic buildings, *(b)* recreational/athletics facilities, *(c)* dormitory and residential zones, and *(d)* restaurants and cafes.

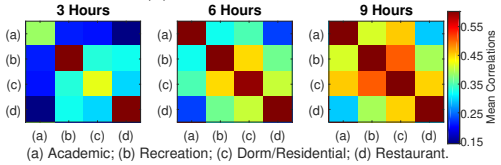

Fig. 6: Temporal interactions at selected POIs of Campus A.

To demonstrate the interactions, Fig. 6 plots the average correlations between each pair of POIs at multiple temporal scales (3, 6, and 9 hours). From the higher correlations (warmer colors) between the two categories of POIs, we can infer more similar trends of client arrivals, i.e., *co-flows*, at the two types of POI zones at the same time. We observe that for short sliding windows, more co-flows are likely to happen in campus zones with recreation/athletics facilities, dormitories, and restaurants, which are closely related to campus life apart from class activities. While for large sliding windows, more co-flows, with much higher correlations, can be observed at dormitories, recreational facilities as well as academic

buildings, demonstrating the long-term routine patterns on campus. We also note from Figs. 3 and 6 that as the recreation and dormitory buildings are largely at the peripheral areas of the campus and with noticeable mutual distances, such dependencies should be carefully modeled via not only *locally* (local nearby zones) but also *globally* adaptive (across distant zones) scopes.

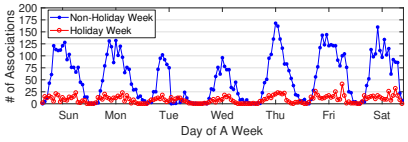

Fig. 7: Dynamics of associations during non-holiday and holiday weeks at an academic building (Campus A).

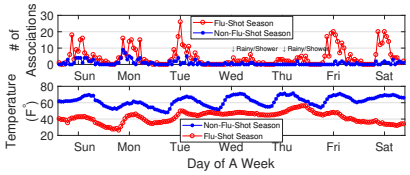

Fig. 8: Dynamics of associations during non-flu-shot and flu shot seasons at the student health center (Campus A).

M4 – Interactions with External Factors.

For Campus A, we further select two locations to further analyze how the crowds may respond to the external factors. Specifically, we select two APs, one inside an academic building and the other inside the student health center (clinic), and illustrate the associations and the crowd flows in Figs. 7 and 8. Fig. 7 shows a clear difference in Wi-Fi associations between holiday and non-holiday weeks on the campus. Another interesting finding comes from Fig. 8, where we compare the crowd flows between two weeks in a flu-shot season (usually October/November in our case) and a non-flu-shot season (in September). The respective temperature trends (average 41°F vs. 62°F, or 5°C vs. 16°C) are also shown in Fig. 8. More clinic visits are likely due to the lowering temperature and flu seasons, implying the needs of forecasting such interactions for public health concerns. We also observe some drops on Wednesday and Thursday due to rain and shower since the student health center is disjoint from other academic buildings and dorms which might deter the crowd mobility.

Motivated by above, for each campus, we collect weather conditions and day of week as the external factors for the crowd flow prediction. Specifically, for Campus A we collect the weather data from the National Oceanic and Atmospheric Administration (NOAA), while for Campus B we find the weather conditions based on the open data portal Weather Archive [1]. As shown in Table 1, we consider weather conditions, meteorological data, as well as event/time. For the categorical factors such as weather conditions and public holidays, we use one-hot encoding [8], i.e., the value is 1 if a condition exists (say, rainy) and 0 otherwise. For the numerical factors such as meteorological data, we conduct the max-min normalization [8]. We concatenate the categorical and numerical factors, and form the external feature vectors $\mathbf{E}$ (the dimension is 11 for Campus A and 15 for Campus B).

M5 – Interactions with Events & Environmental Changes.

We have conducted the following studies upon the crowds' interactions with the event dynamics and environmental alterations (in terms of AP alteration):

Table 1: External factors considered for the two datasets.

Factors	Campus A	Campus B
Weather Conditions	Foggy/Rainy/Misty/ Haze/Snowy (5-D)	Misty/Drizzle/Light Rain/Shower/Snow/ Freezing/Foggy/No Significant Clouds (8-D)
Meteorological Data	Temperature/Humidity/ Wind Speed (3-D)	Temperature/Pressure/Humidity/ Wind Speed (4-D)
Event/Time	Day of Week/Hour of Day/Public Holiday or Not (3-D)	

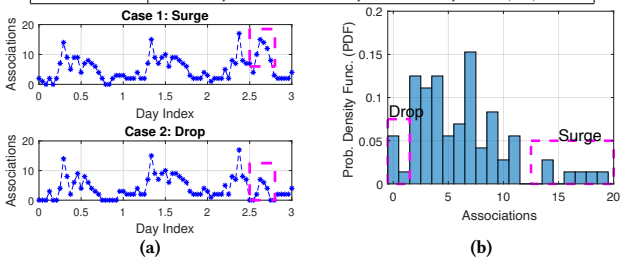

Fig. 9: Time series (a) and the PDF (b) of historical associations for an AP.

(i) Interactions with Events: Transient and abnormal campus events (e.g., a conference or emergency) may lead to significantly higher (surge) or lower (drop) volumes of crowd flows at certain APs than historical average, i.e., the average flows of the same time interval of a day in a week. We show in Fig. 9 regarding *(a)* time series and *(b)* probability distributions of flows in the same historical time interval (14:00–15:00) at an AP in an academic building on Campus A. We can observe that the crowd flow distribution at this AP can have significantly high or low values (highlighted). To model such an *extreme* surge or drop and enhance ACroCI's adaptivity, we take into account multiple *statistical time-series features* (detailed in Sec. 4.1) of the crowd flows at different campus zones as additional inputs for ACroCI. Meanwhile, we use *auxiliary labels* for the extremely high/low AP-level flows for the *joint* model training.

(ii) Interactions reflected by AP Alterations: Due to network maintenance or reconfiguration, some Wi-Fi APs may be introduced, relocated, or removed. A removed AP can be simply masked without further prediction. However, an AP that is either newly-installed or relocated (same MAC addresses but treated as "new" at the model side), may have different spatio-temporal neighborhood features, leading to "cold-start" initialization problem for the crowd interaction learning model due to lack of initial data. For instance, we have observed on Campus A that 11.08% of the APs have been newly introduced within an academic year (fall and spring semesters).

2.3 Problem Definition & Model Overview

Problem Definition. Motivated by the above data analysis, in our CMA prototype study, we consider the crowd flow prediction as a case study to evaluate the usefulness of incorporating interactions within ACroCI.

Specifically, we aim at designing a crowd interaction learning model $\mathcal{P}_{\boldsymbol{\theta}}(\cdot)$ (with model hyperparameters $\boldsymbol{\theta}$) to forecast AP-level crowd flows (arrivals or departures) in a target time interval k, denoted as $\hat{\mathbf{X}}^{(k)}$, which is either associations $\hat{\mathbf{A}}^{(k)}$ or disassociations $\hat{\mathbf{D}}^{(k)}$, given a sequence of w historical representations, $\mathbf{F}^{(\text{hist})} = \left\{\mathbf{F}^{(k-w)}, \mathbf{F}^{(k-w+1)}, \ldots, \mathbf{F}^{(k-1)}\right\}$, as well as the AP-level crowd flows in the previous interval, $\mathbf{X}^{(k-1)}$, external factors, $\mathbf{E}^{(k-1)}$, and other statistical time-series features, $\mathbf{F}^{(\text{stat})}$. It is formally given by

$$\hat{\mathbf{X}}^{(k)} = \mathcal{P}_{\boldsymbol{\theta}}\left(\mathbf{F}^{(\text{hist})}; \mathbf{X}^{(k-1)}; \mathbf{E}^{(k-1)}; \mathbf{F}^{(\text{stat})}\right). \quad (1)$$

Model and System Overview. Based on motivations **M1** – **M5**, we have designed and implemented ACroCI, whose model and

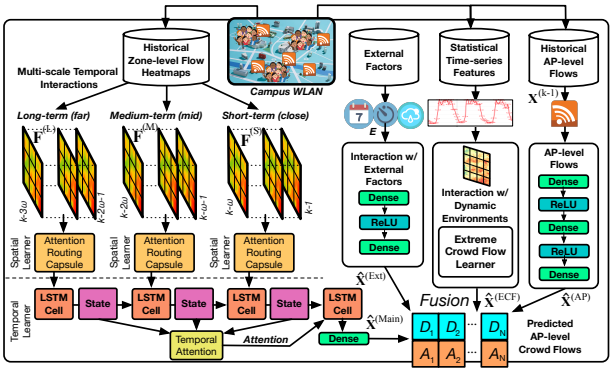

Fig. 10: Model and system framework overview of ACroCI.

system framework is illustrated in Fig. 10. To train the model, we first collect the historical Wi-Fi AP association records, AP locations, campus map, as well as other external factors (including the weather conditions and the university calendar) with the campus IT services. ACroCI pre-processes the above inputs, and the *data-driven studies* derive the model inputs and system parameters for the ACroCI's core model. Specifically, these features are processed and learned by the following learners.

1. Spatial Cross-Zone Interactive Learner (Sec. 3.1): This module captures the zone-to-zone correlations via the heatmap representation of spatial crowd flow distributions. Taking into account the spatial variations, multi-scale co-flows, and local-global cross-zone dependencies (**M1** & **M3**), we pre-process the crowd flows of an interval into a heatmap frame, where each grid element represents aggregated arrivals or departures, and leverage the attention routing capsule for interactive learning.

2. Multi-Scale Attention Temporal Learner (Sec. 3.2): To model the temporal interactions and dependencies of sequential time intervals (**M2**), we further feed the outputs of the spatial learners after processing multiple consecutive heatmap frames, to the temporal learner consisting of long short-term memory (LSTM) encoder and attention decoder.

3. Auxiliary Extreme Crowd Flow Learner (Sec. 4.1): To capture the correlations with extreme flows (**M5**), we extract multiple statistical features of the zone-level crowd flow time series to form another series of heatmap frames for ConvLSTM [8], and simultaneously forecast the crowd flows $\hat{\mathbf{X}}^{(\mathrm{ECF})}$ and the auxiliary labels $\hat{\mathbf{Z}}$. We provide a multi-task extreme-aware learning design beyond the spatial and temporal learners to *jointly* minimize the prediction errors as well as differences in predicted and estimated auxiliary labels.

4. External and AP-level Learners (Sec. 4.2): These two learners are deeply connected neural networks (with Dense layers) that predict $\hat{\mathbf{X}}^{(\mathrm{Ext})}$ and $\hat{\mathbf{X}}^{(\mathrm{AP})}$, given the input vectors of external factors (**M4**) and the crowd flows of the latest time interval prior to the target one, respectively. The outputs $\hat{\mathbf{X}}^{(\mathrm{Main})}$, $\hat{\mathbf{X}}^{(\mathrm{ECF})}$, $\hat{\mathbf{X}}^{(\mathrm{Ext})}$, and $\hat{\mathbf{X}}^{(\mathrm{AP})}$ from all four modules are merged for final prediction.

3 CORE DESIGNS OF ACROCI

3.1 Cross-Zone & Local-Global Interaction Learning

Input Spatio-Temporal Representations. In our prototype studies, we formulate the input spatial distributions of the associations and disassociations in the time interval k into the heatmap frames, i.e.,

$$\mathbf{Y}^{(k)} = \begin{bmatrix} Y_{11}^{(k)} & \cdots & Y_{1W}^{(k)} \\ \vdots & \ddots & \vdots \\ Y_{H1}^{(k)} & \cdots & Y_{HW}^{(k)} \end{bmatrix}, \quad \mathbf{O}^{(k)} = \begin{bmatrix} O_{11}^{(k)} & \cdots & O_{1W}^{(k)} \\ \vdots & \ddots & \vdots \\ O_{H1}^{(k)} & \cdots & O_{HW}^{(k)} \end{bmatrix}, \quad (2)$$

where $Y_{ij}^{(k)}$ ($O_{ij}^{(k)}$) is the number of associations (disassociations) within the i-th row (latitudinal), the j-th column (longitudinal) of the grid map in the k-th interval. The inherent correlations across different campus locations, including their functions and POIs, can be incorporated within the heatmap frames and processed by our subsequent capsule network learning module.

To accommodate and characterize different temporal scales of spatial distributions (Fig. 5 in **M2** and Fig. 6 in **M3**), we formulate spatio-temporal mobility representations $\mathbf{F}$'s in Definition 4 (Sec. 2.3) into the **s**hort, **m**edium, and **l**ong-term input tensors, denoted as $\mathbf{F}^{(\mathrm{S})}$, $\mathbf{F}^{(\mathrm{M})}$, and $\mathbf{F}^{(\mathrm{L})}$, respectively. If we are to predict the associations (or arrivals), each input tensor, $\mathbf{F}^{(x)} \in \left\{\mathbf{F}^{(\mathrm{S})}, \mathbf{F}^{(\mathrm{M})}, \mathbf{F}^{(\mathrm{L})}\right\}$, that is fed to a spatial learner (with the notation $x \in \{\mathrm{S}, \mathrm{M}, \mathrm{L}\}$), comprises ω consecutive heatmap frames from a sliding window or a predetermined number of time intervals, i.e.,

$$\mathbf{F}^{(x)} = \left[\mathbf{Y}^{(k-m\cdot\omega)}, \mathbf{Y}^{(k-m\cdot\omega+1)}, \ldots, \mathbf{Y}^{(k-(m-1)\cdot\omega-1)}\right], \quad (3)$$

where the symbol $x \in \{\mathrm{S}, \mathrm{M}, \mathrm{L}\}$ (either short-, mid- or long-term scale) with, respectively, $m \in \{1, 2, 3\}$. If we are to predict disassociations (or departures), we have $\mathbf{O}$ instead of $\mathbf{Y}$ in Eq. (3). Each heatmap frame in $\mathbf{F}^{(x)}$ is fed as a channel [8] for the input layer (total ω channels per temporal scale). We have empirically studied ω and observed a larger ω reduces errors but with a diminishing return, and hence adopt $\omega = 6$ in our studies.

Designs of Attention Routing Capsule. Given the input heatmap frames of crowd flows, we aim at capturing the complex spatial distributions, correlated co-flows, and local-global cross-zone dependencies (i.e., **M1** & **M3** in Sec. 2.2). However, via our later experimental observations, solely using conventional *scalar*-based networks, like convolutional neural network (CNN), cannot adequately characterize these and encode the transformation of the features. Therefore, we can observe large prediction errors for the complex Wi-Fi data learning and prediction scenarios.

To address this, we introduce the capsule neural network (Cap) [22] within ACroCI, where a structured group of neurons, i.e., *capsule*, forms a *vector* representation of the features. Specifically, each capsule can describe how the input heatmap distribution, as a target of interest, is instantiated as a *vector*, including its spatial position relative to the map, and captures more co-flow and local-global features than the scalar-based CNN.

We note that the conventional capsule network adopts dynamic routing and squash activation function [10, 22]. Such dynamic routing iteratively adjusts the values of the vectors between the capsule layers during training to find the vector agreement and learn the input features. However, the routing cannot further *differentiate* the importance of connections, dependencies, and interactions across spacious campus crowd distributions. To strengthen the model, ACroCI further integrates the *attention* mechanism to parameterize the routing among the capsules [5]. The attention mechanism helps

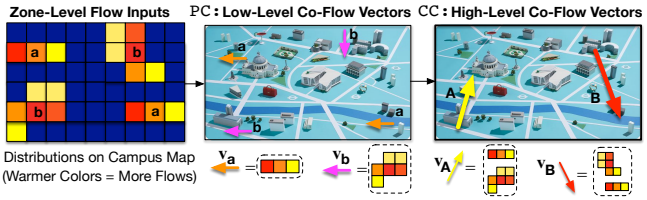

Fig. 11: Illustration of interactive cross-zone learning.

ACroCI learn a compatibility function between low-level and high-level features [17], thus empowering the learning performance and prediction accuracy of the capsule network.

Detailed Layer Designs. In the following, we present the *attention routing capsule* (denoted as ARCap) architecture, which consists of the primary capsule (PC) layer and the convolutional capsule (CC) layer. Before PC and after CC, ACroCI leverages the 2D convolutional layer or Conv2D for input preprocessing and output restructuring. Final output $\mathbf{X}^{(x)}$ is returned after an activation layer.

In an implementation, a capsule layer (either PC or CC) can be reshaped and restructured from the convolutional (say, $\mathsf{Conv}_{K\times K}(\cdot)$ with $K \times K$ kernels) module consisting of $N^{(l)} \times S^{(l)}$ filters [9, 22], where l is the index of the layer, and $N^{(l)}$ and $S^{(l)}$ represent the numbers of capsule channels and dimensions at the l-th layer, respectively. Compared to the convolutional layer, the capsule layer becomes $N^{(l)}$ channels, each of which is a group of neurons returning an output vector. We denote for each capsule layer l the height and width of the input as $H^{(l)}$ and $W^{(l)}$, respectively.

We present the detailed layer designs of ARCap as follows.

(a) *Input*: At the input of the spatial learner, ACroCI first preprocesses the tensor via batch normalization (denoted as $\mathsf{BN}(\cdot)$) and $\mathsf{Conv2D}(\cdot)$, deriving the initial coarse-grained features, i.e.,

$$\mathbf{X}^{(x,0)} = \sigma\left(\mathsf{Conv2D}\left(\mathsf{BN}\left(\mathbf{F}^{(x)}\right)\right)\right), \tag{4}$$

where we adopt ReLU for the activation $\sigma(\cdot)$. For each dataset, the batch normalization is intended to reduce the internal covariate shift within the data. The output $\mathbf{X}^{(x,0)}$ is then fed to the PC Layer at the ARCap structure.

(b) PC *Layer*: Crowd interactions between the nearby zones (say, low-level features of co-flows **a** and **b** in Fig. 11) are captured by the capsule layer characterizing such spatial structural information. Specifically, the PC layer first processes and returns the vector regarding each of the N capsule channels (indexed by n), i.e.,

$$\mathbf{v}_n^{(0)} = \sigma\left(\mathsf{Conv}_{K\times K}\left(\mathbf{X}^{(x,0)}\right)\right), \tag{5}$$

where $x \in \{\mathrm{S}, \mathrm{M}, \mathrm{L}\}$, and we adopt $K = 3$ in the convolution kernel, and ReLU for the activation $\sigma(\cdot)$. The correlations between multiple neighboring zones can be further extracted via the stacked layers. Then, the capsule activation conducts the affine convolutional transformation upon each capsule channel n, i.e.,

$$\mathbf{v}_n = \sigma\left(\mathsf{Conv}_{K\times K}\left(\mathbf{v}_n^{(0)}\right)\right). \tag{6}$$

where we adopt $K = 1$, and tanh for $\sigma(\cdot)$, which converts the output into the range of $[-1, 1]$ to restrict the value scale.

(c) CC *Layer*: CC aims at deriving the crowds' global interactions with the zones (say, high-level co-flow features in groups **A** and **B** from multiple distant zones in Fig. 11) via the capsule attention mechanism. As shown in Fig. 11, the high-level features in groups **A** and **B** in CC (as vectors $\mathbf{v_A}$ and $\mathbf{v_B}$), contain structural information from **a** and **b** (as vectors $\mathbf{v_a}$ and $\mathbf{v_b}$) after PC.

First, *convolutional transformation* converts each capsule channel, and forms new capsule channels with parameters shared locally with multiple channels of the preceding layer's channels, i.e.,

$$\tilde{\mathbf{v}}_n = \mathsf{Conv}_{K\times K}\left(\mathbf{v}_n\right), \tag{7}$$

where we adopt $K = 3$, and the resultant structure enables the attention mechanism interleaving the transformed capsules.

Second, between the transformed capsules and the *CC* layer, we conduct the *attention routing*. The attention weights λ_j's are the softmax [8] outputs of the logarithm probabilities [22] along the capsule channel axis at the previous layer $(l-1)$, i.e.,

$$\lambda_j \triangleq \mathsf{softmax}\left(e_j\right) = \frac{\exp\left(e_j\right)}{\sum_{n=1}^{N^{(l-1)}} \exp\left(e_n\right)}, \tag{8}$$

where the logarithm probabilities e_n [22] can be obtained through the 3D convolution upon the input $\tilde{\mathbf{v}}$, i.e.,

$$\mathbf{e} = [e_1, e_2, \ldots, e_N] = \mathsf{Conv3D}_{1\times 1\times D^{(l)}}\left(\tilde{\mathbf{v}}\right). \tag{9}$$

Unlike conventional dynamic routing [22], attention routing helps ACroCI learn the logarithm probabilities of the agreement coefficients between the ℓ-th and $(\ell-1)$-th layers. Attention routing adjusts the weights λ_j for each spatial location in the convolutional transformed capsules, such that the important locations of interest, as well as the relevant crowd interactions can be further derived.

Third, given Eqs. (7) and (8), each channel output is the weighted average of the logarithm probabilities and the vector output from each preceding capsule channel for the attention routing, i.e.,

$$\mathbf{v} \triangleq \sum_{j=1}^{N^{(l-1)}} \lambda_j \cdot \tilde{\mathbf{v}}_j. \tag{10}$$

The *compatibility* between the preceding (like $\mathbf{v_a}$ and $\mathbf{v_b}$ in Fig. 11) and succeeding features (say, $\mathbf{v_A}$ and $\mathbf{v_B}$ in Fig. 11), i.e., the local-global interactions, is captured by the attention weights, and strengthened during model training. This way, ACroCI captures more co-flow features and cross-zone interactions than other network designs, yielding better accuracy in our experimental studies.

(d) *Output*: The capsule activation at the CC layer is done via another affine transformation, which outputs

$$\mathbf{X}_{\text{out}}^{(x)} = \sigma\left(\mathsf{Conv2D}(\mathbf{v})\right), \quad x \in \{\mathrm{S}, \mathrm{M}, \mathrm{L}\}, \tag{11}$$

where tanh is used for the activation of $\sigma(\cdot)$. Then, we apply and feed $\mathbf{X}^{(x)} = \mathsf{Dense}\left(\mathbf{X}_{\text{out}}^{(x)}\right)$ $(x \in \{\mathrm{S}, \mathrm{M}, \mathrm{L}\})$ to the temporal learner.

3.2 Learning Temporal Interactions

The *Long Short-Term Memory* (LSTM) *encoder* first processes the input time-series and encodes the sequence into a vector representing the context. Let $\circ$ be the Hadamard product, i.e., the element-wise multiplication operation, t be the index of time step within the time-series, (f_t, i_t, o_t) be the output activation vectors from the forget, input/update and output gates [8], and $(\mathbf{c}_t, \mathbf{H}_t)$ be the cell and hidden states at t. ACroCI captures the temporal features via the LSTM structures as follows:

$$f_t = \sigma_g\left(\mathbf{W}_f\mathbf{X}^{(x)} + \mathbf{U}_f\mathbf{H}_{t-1} + \mathbf{b}_f\right), \quad i_t = \sigma_g\left(\mathbf{W}_i\mathbf{X}^{(x)} + \mathbf{U}_i\mathbf{H}_{t-1} + \mathbf{b}_i\right),$$

$$o_t = \sigma_g\left(\mathbf{W}_o\mathbf{X}^{(x)} + \mathbf{U}_o\mathbf{H}_{t-1} + \mathbf{b}_o\right),$$

$$\mathbf{c}_t = f_t \circ \mathbf{c}_{t-1} + i_t \circ \sigma_c\left(\mathbf{W}_c\mathbf{X}^{(x)} + \mathbf{U}_c\mathbf{H}_{t-1} + \mathbf{b}_c\right), \quad \mathbf{H}_t = o_t \circ \sigma_{\mathrm{H}}\left(\mathbf{c}_t\right),$$

where $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_o, \mathbf{W}_c, \mathbf{U}_f, \mathbf{U}_i, \mathbf{U}_o$, and $\mathbf{U}_c$ are the weight matrices for the input and hidden states, and $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_o$, and $\mathbf{b}_c$ are the biases, all of which are hyperparameters to be trained.

To further identify the multi-scale temporal dependency (i.e., **M2** in Sec. 2.2), we integrate within ACroCI the *temporal attention decoder* [3], given the encoded results from the LSTM encoder. By decoding the compressed information from the encoder, the attention decoder finds the locations where the most relevant features are concentrated. The temporal attention mechanism adaptively selects the more correlated hidden states of the encoder in order to match the sample features and produce the final output. Specifically, the output cell state $\mathbf{c}_t$ at the attention decoder is given by the weighted sum of the input mapping (hidden states) from the LSTM encoder of the previous inputs: $\mathbf{c}_t = \sum_{t'=1}^{T} \alpha_{(t',t)} \mathbf{H}_{t'}$. Similar to Eq. (8), the attention weight $\alpha_{(t',t)}$ in the temporal attention is given by a softmax function of

$$\alpha_{(t',t)} \triangleq \texttt{softmax}\left(e_{(t',t)}\right) = \frac{\exp\left(e_{(t',t)}\right)}{\sum_{k=1}^{T} \exp\left(e_{(t',k)}\right)}, \tag{12}$$

where $e_{(t',k)}$ represents a score of how the k-th heatmap frame $\mathbf{F}_k$ in the input sequence matches the hidden state $\mathbf{H}_{t'}$. Specifically, each score is characterized by a feed-forward neural network $\mathbb{F}(\cdot,\cdot)$ with concentration operation [3, 8], i.e.,

$$e_{(t',k)} \triangleq \mathbb{F}\left(\mathbf{H}_{t'}, \mathbf{F}_k\right) = \mathbf{v}_a^{\intercal} \sigma\left(\mathbf{W}_a[\mathbf{H}_{t'-1}; \mathbf{c}_{t'-1}] + \mathbf{U}_a \mathbf{F}_k + \mathbf{b}_a\right), \tag{13}$$

where $\mathbf{v}_a, \mathbf{b}$, $\mathbf{W}_a$, and $\mathbf{U}_a$ are all trainable parameters within the attention decoder, and we adopt tanh as activation function $\sigma(\cdot)$. Let $\mathbf{H}^{(\text{Out})}$ be the output of the last hidden states. The output of ACroCI's temporal learner, denoted as $\hat{\mathbf{X}}^{(\text{Main})} \in \mathbb{R}^M$, is an M-d vector of flows at all M APs, and is generated by the dense neural network following the hidden state output at the LSTM encoder, i.e.,

$$\hat{\mathbf{X}}^{(\text{Main})} = \texttt{Dense}\left(\mathbf{H}^{(\text{Out})}\right), \tag{14}$$

which is the final output of the spatio-temporal integration.

4 ADAPTIVITY MODULE INTEGRATION

4.1 Auxiliary Extreme-aware Learner

To enhance ACroCI's awareness towards the crowd interactions with the transient or emergent campus events (**M5** in Sec. 2.2), we have further designed an auxiliary extreme-aware crowd flow learner based on the statistical time-series features derived for ACroCI.

Processing Time-Series Features. For each campus zone we find the following 9 statistical time-series features of the recent $\omega^{(\text{ECF})}$ historical time intervals: *mean*, *variance*, *autocorrelation*, *entropy*, *trend* coefficients (total 3 features: *trend value* and its *p-value* via linear trend model estimation, and the *variance of the residuals*), *spike*, and *crossing points* (the number of times the mean is crossed by the time series) [7, 13, 15]. Compared with zone-level crowd flows in Eq. (2), these auxiliary features (each forms an $H \times W$ heatmap) provide additional information related to the flow dynamics.

In the meantime, using extensive empirical studies we label the corresponding Wi-Fi associations and disassociations for each AP i at an interval k, $Z_i^{(k)}$, with three conditions when compared with all historical readings at the same time interval (for instance, 8:00–9:00 of all historical Mondays): *(i)* $Z_i^{(k)} = 0$ when the value lies between 5th and 95th percentiles of all historical values (considered *normal*); *(ii)* $Z_i^{(k)} = 1$ when the value rises above 95th percentile (considered *surge*); and *(iii)* $Z_i^{(k)} = -1$ when the value falls below 5th percentile (considered *drop*). Then, for all APs at time interval k, we have $\mathbf{Z}^{(k)} = \left[Z_1^{(k)}, \ldots, Z_M^{(k)}\right]$ as the *auxiliary labels* in our multi-task extreme-aware learning design.

Learning Crowd Interaction with Events. For each campus zone, we find the aforementioned 9 features within a window of $\omega^{(\text{ECF})}$ consecutive time intervals, and the window rolls over the most recent Ω time intervals. Given the Ω most recent $H \times W \times 9$ tensors which form $\mathbf{F}^{(\text{stat})}$, we leverage ConvLSTM [8] to capture spatial and temporal correlations of the features (each heatmap of the 9 features as one channel), followed by a Dense layer (with ReLU for $\sigma(\cdot)$), and forecast the future AP-level crowd flows, denoted as $\hat{\mathbf{X}}^{(\text{ECF})}$. In order to enhance ACroCI's learnability regarding the extreme crowd flows, we design a *multi-task extreme-aware learning* mechanism (Sec. 4.2) which *jointly* captures and outputs both the AP-level crowd flows as well as the auxiliary labels, i.e.,

$$\left[\hat{\mathbf{X}}^{(\text{ECF})}; \hat{\mathbf{Z}}\right] = \sigma\left(\texttt{Dense}\left(\texttt{ConvLSTM}\left(\mathbf{F}^{(\text{stat})}\right)\right)\right). \tag{15}$$

This way, ACroCI correlates the surging, normal, and dropping crowd flows with the statistical features, fits the flows and auxiliary labels simultaneously, and thus enhances the adaptivity towards the transient or abnormal events.

4.2 Output and Model Training

External Learner. ACroCI also takes into account the external features (i.e., **M4** in Sec. 2.2) to assist in the crowd flow inference. Specifically, we normalize each dimension of the external factors into the range of $[0, 1]$, and concatenate them into a vector $\mathbf{E}$, which is fed to a neural network returning $\hat{\mathbf{X}}^{(\text{Ext})} \in \mathbb{R}^M$ as follows:

$$\hat{\mathbf{X}}^{(\text{Ext})} = \texttt{Dense}\left(\sigma\left(\texttt{Dense}\left(\mathbf{E}\right)\right)\right). \tag{16}$$

AP-level Learner. To integrate the most recent AP-level crowd flow features and further adapt to the latest transient crowd flow dynamics, we feed the crowd flows of the last time interval to another neural network, and predict $\hat{\mathbf{X}}^{(\text{AP})} \in \mathbb{R}^M$, i.e.,

$$\hat{\mathbf{X}}^{(\text{AP})} = \texttt{Dense}\left(\sigma\left(\texttt{Dense}\left(\sigma\left(\texttt{Dense}\left(\mathbf{X}^{(k-1)}\right)\right)\right)\right)\right), \tag{17}$$

where $\mathbf{X}^{(k-1)} \in \left\{\mathbf{A}^{(k-1)}, \mathbf{D}^{(k-1)}\right\}$. For Eqs. (16) and (17), we adopt ReLU for activation $\sigma(\cdot)$.

Integration, Output, and Training Objective Function. The final prediction of AP-level crowd flows, $\hat{\mathbf{X}} \in \mathbb{R}^M$, is given by the fusion of the outputs based on spatial-temporal learners ($\hat{\mathbf{X}}^{(\text{Main})}$), extreme crowd flows ($\hat{\mathbf{X}}^{(\text{ECF})}$), external factors ($\hat{\mathbf{X}}^{(\text{Ext})}$), and AP-level crowd flows ($\hat{\mathbf{X}}^{(\text{AP})}$),

$$\begin{aligned}\hat{\mathbf{X}} = \sigma\big(&\mathbf{W}^{(\text{Main})} \circ \hat{\mathbf{X}}^{(\text{Main})} + \mathbf{W}^{(\text{ECF})} \circ \hat{\mathbf{X}}^{(\text{ECF})} \\ &+ \mathbf{W}^{(\text{Ext})} \circ \hat{\mathbf{X}}^{(\text{Ext})} + \mathbf{W}^{(\text{AP})} \circ \hat{\mathbf{X}}^{(\text{AP})}\big),\end{aligned} \tag{18}$$

where $\mathbf{W}^{(\text{Main})}$, $\mathbf{W}^{(\text{ECF})}$, $\mathbf{W}^{(\text{Ext})}$, and $\mathbf{W}^{(\text{AP})}$ are all the learnable parameters which adjust the degrees affected by the different factors, and we also adopt ReLU for the activation function $\sigma(\cdot)$.

As discussed in Sec. 4.1, ACroCI is trained as *multi-task extreme-aware learning*, i.e., to *jointly* minimize the mean squared error between the predicted flows $\hat{\mathbf{X}}$ ($\hat{\mathbf{A}}$ or $\hat{\mathbf{D}}$) and the ground-truth $\mathbf{X}$, as well as the mean squared error between the predicted and ground-truth crowd flow labels, $\hat{\mathbf{Z}}$ and $\mathbf{Z}$, i.e.,

$$\texttt{Loss}(\boldsymbol{\theta}) = \left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_2^2 + \lambda \left\|\mathbf{Z} - \hat{\mathbf{Z}}\right\|_2^2, \tag{19}$$

where $\boldsymbol{\theta}$ represents all the trainable parameters within all the learners of ACroCI and $\lambda > 0$ is a weight parameter. We adopt the Adam optimizer [8] in our training.

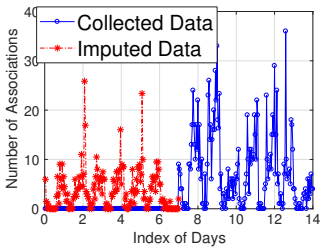

Fig. 12: Collected and imputed data for a newly-installed AP.

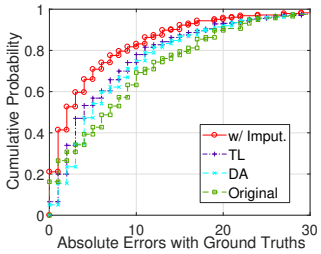

Fig. 13: Cumulative probability of prediction absolute errors.

4.3 Adapting to AP Alteration

As discussed in **M5** of Sec. 2.2, when an AP is newly added or relocated, there is no prior data for model training, leading to the "cold-start" issue. Thus, we design efficient training data imputation for each newly-installed (or relocated) AP using one preceding week's data from its nearby APs in the same zone. Specifically, for each target AP i, we find its geographic distances (in km), $\text{dist}(i, j)$'s ($j \in \{1, \ldots, M'\}$), from all its M' ($M' < M$) nearest APs already deployed in the same zone, and impute AP i's flows using weighted average for ACroCI's model training and initialization, i.e.,

$$\tilde{\mathbf{X}}_i^{(k)} = \sum_{j=1}^{M'} \frac{\beta_j}{\sum_{j=1}^{M'} \beta_j} \mathbf{X}_j^{(k)}, \quad \text{where} \quad \beta_j \triangleq \frac{1}{1 + \text{dist}(i, j)}. \tag{20}$$

We show in Fig. 12 one week's associations (blue) of a newly added AP (due to Campus A's facility reconstruction) after its deployment. We also conduct the data imputation (red) upon the earlier week (no association data) for the model training. We then show in Fig. 13 the improvement via our efficient data imputation, with mean absolute errors (MAEs) reduced from 9.02 (original) to 5.10, showing the applicability of the efficient data imputation. We also compare our approach with transfer learning (TL) [20] and domain adaptation (DA) [19] methods, and our efficient imputation more effectively adapts to such a "cold-start" scenario (~27% error reduction), as TL and DA still largely rely on availability of sufficient training data. We also note that ACroCI can also get timely model updates given newly accumulated association data for those APs after a few hours or days for further accuracy improvement.

5 EXPERIMENTAL EVALUATION

5.1 Experimental Settings

Baseline Methods. Our proposed model is compared with the following conventional and state-of-the-art methods.

- HA and S-HA (**S**easonal HA): HA predicts the crowd flows as **h**istorical **a**verage of those of the same time periods or intervals (e.g., finding the average flow during 8:00–9:00 of all past Mondays to predict the same time slot for a Monday). *(a)* S-HA takes the same historical periods but of the same seasons.
- ARIMA, GP, RNN, LSTM, and DNN: Each of these approaches forecasts with *(3)* **a**uto-**r**egressive **i**ntegrated **m**oving **a**verage (ARIMA), *(4)* **G**aussian **p**rocess, *(5)* **r**ecurrent **n**eural **n**etwork (RNN) [8], *(6)* **l**ong **s**hort-**t**erm **m**emory (LSTM) [8], and *(7)* fully-connected **d**eep **n**eural **n**etwork (DNN).
- CNN and CNN-Att: only capture the spatial distributions of the associations with *(8)* **c**onvolutional **n**eural **n**etwork (CNN) [8] and *(9)* CNN with spatial **att**ention mechanism [4].
- GCN and MTGNN [27]: which leverage *(10)* **g**raph **c**onvolutional **n**eural network [14] and *(11)* **m**ultiple **t**ime-series **g**raph **n**eural **n**etwork, respectively, to predict the crowd flows.
- STCNN, STRNet, and STCaNet: process crowd distribution via *(12)* **s**patio-**t**emporal **c**onvolutional **n**eural **n**etwork [28], *(13)* **s**patio-**t**emporal **r**esidual neural **net**work [28], and *(14)* **s**patio-**t**emporal **ca**psule neural **net**work [9].
- CHAT [12]: is adapted to model the cross-event interactions in predicting the dynamics of the crowd flows.
- CSTN [16]: predicts with the convolution embedded LSTM-based method with a **c**ontextualized **s**patio-**t**emporal **n**etwork.
- T-LSTM: predicts the crowd flows based on **t**emporal pattern attention **l**ong **s**hort-**t**erm **m**emory [25].

Parameter Settings & Evaluation Metrics. Unless otherwise stated, we use the following parameters by default. We set the number of epochs to 1,000, batch size to 256, and learning rate to 0.001. We set 32×32 for heatmap frames on both campuses, 1h for time discretization, and $\omega = 6$ for each scale of temporal correlations. For the spatial learner of ACroCI, we set $\left(W^{(1)}, H^{(1)}, N^{(1)}, S^{(1)}\right) = (14, 14, 8, 8)$, $\left(W^{(2)}, H^{(2)}, N^{(2)}, S^{(2)}\right) = (7, 7, 32, 32)$ for ARCap, the number of filters to 32 for the two Conv2D(·) in Eqs. (4) and (11). For the temporal learner, we set the number of units to 8 for the LSTM modules, and the number of attentions to 16. For the auxiliary extreme crowd flow learner, we have empirically studied and set $\omega^{(\text{ECF})} = \Omega = 5$ and number of filters as 32 in ConvLSTM, and $\lambda = 0.1$ for the multi-task learning objective. For the external learner, we set the output dimension for $\mathbf{E}^0$ to 8. For the AP-level learner, we set the output dimension for $\mathbf{X}^0$ and $\mathbf{X}^1$ to 128 and 32, respectively. For the AP alteration adaptation, we use $M' = 5$ for Eq. (20).

For each campus dataset, we take the first 600 time intervals as the validation dataset and test upon the following 100 intervals to evaluate the model and parameter sensitivity. Apart from that validation dataset, for each campus, we conduct model training and testing on a monthly basis to emulate the real-world deployment. Specifically, we train the model based on data of 25 days (600 intervals in total) before each target test month. For each AP that is newly installed or relocated, we impute one-week data prior to its deployment for model training. We train and test the models through Python/Tensorflow using a GPU server with Intel Core i9 9900K, 32Gb RAM and two NVIDIA RTX 2080Ti 11Gb GPUs. The ACroCI model training time is 1.45s per epoch for Campus A, and 1.51s per epoch for Campus B; its model testing is fast (~0.92ms per time interval) for both datasets.

We evaluate all the schemes using the mean absolute error (MAE) to interpret the overall error *trend*, the root mean square error (RMSE) to demonstrate the *variance* of error distributions, and the poor case rate (PCR) to show the *relative scale* of prediction errors. The PCR is given by the percentage of all predictions which have the excessive over- or under-estimations compared to the ground-truth values, i.e., $\frac{1}{\mathbf{X}_i}|\mathbf{X}_i - \hat{\mathbf{X}}_i| \geq \eta$, where we consider $\eta = 0.6$.

5.2 Evaluation Results

Overall Performance, Rush Hours, and Extreme Events. Table 2(a) lists performance comparison of ACroCI with other algorithms for the two datasets. Overall, ACroCI achieves significantly higher accuracy than others. The conventional time-series-based techniques, (1)–(4), cannot capture the spatial correlations across the campus zones, hence achieving less accurate results. The sequence learning approaches like (5)–(7) cannot handle the spatial and temporal complexities within the crowd flows. Spatial learning approaches like (8)–(11) focus on the zone-to-zone correlations, and hence cannot handle multi-scale temporal complexity.

Table 2: Overall and rush-hour performance for Campuses A and B.

Schemes	(a) Overall Performance						(b) Rush-Hour Performance					
	Campus A			Campus B			Campus A			Campus B		
	MAE	RMSE	PCR	MAE	RMSE	PCR	MAE	RMSE	PCR	MAE	RMSE	PCR
(1)	5.65	9.98	0.313	7.86	12.92	0.346	9.33	13.01	0.447	12.29	18.18	0.508
(2)	4.35	8.19	0.251	7.84	12.30	0.336	8.72	11.92	0.413	11.43	17.17	0.477
(3)	6.15	9.04	0.304	7.57	11.79	0.323	9.24	11.02	0.405	10.08	13.06	0.386
(4)	6.75	10.82	0.351	8.76	12.99	0.363	9.72	13.52	0.465	11.28	14.21	0.425
(5)	3.45	6.76	0.204	6.34	8.98	0.255	6.39	9.59	0.320	10.12	13.68	0.397
(6)	2.97	6.66	0.193	6.26	8.72	0.250	6.85	8.57	0.308	10.18	12.31	0.375
(7)	5.05	7.61	0.253	6.86	9.52	0.273	9.59	10.94	0.411	9.41	11.75	0.353
(8)	2.74	5.75	0.170	6.38	8.37	0.246	3.16	7.29	0.209	8.54	10.03	0.310
(9)	2.89	6.06	0.179	5.27	7.52	0.213	3.42	7.06	0.210	7.23	9.75	0.283
(10)	2.99	4.74	0.154	7.49	10.14	0.294	3.94	8.33	0.245	9.13	12.00	0.352
(11)	3.13	5.82	0.180	6.02	9.11	0.252	3.69	7.74	0.229	8.82	11.40	0.337
(12)	2.53	5.17	0.164	5.68	8.00	0.228	2.89	5.74	0.173	7.61	9.41	0.284
(13)	2.45	4.77	0.144	5.27	7.22	0.208	2.79	5.93	0.174	7.07	9.52	0.277
(14)	2.32	4.57	0.138	5.17	7.16	0.206	2.86	6.87	0.195	7.12	9.46	0.276
(15)	2.35	4.75	0.142	5.12	7.10	0.204	2.97	6.62	0.192	7.26	9.58	0.281
(16)	3.12	6.16	0.186	6.17	9.71	0.265	3.63	7.94	0.231	8.92	11.70	0.344
(17)	2.95	5.73	0.174	6.12	9.12	0.254	3.29	6.73	0.200	7.06	10.52	0.293
ACroCI	**1.53**	**3.18**	**0.094**	**3.69**	**5.63**	**0.125**	**1.92**	**4.39**	**0.126**	**4.47**	**7.27**	**0.196**

ACroCI, however, formulates comprehensive spatio-temporal representations, and hence achieves respectively 51%∼77%, 35%∼70%, and 25%∼51% accuracy improvements over the time-series, sequence learning and spatial learning approaches. Compared to prior efforts on spatio-temporal learning, e.g., (12)–(17), ACroCI captures the spatial heatmap frames via attention routing capsule, which *jointly* leverages vectorization of neural outputs to characterize spatial features, and identifies multi-scale temporal complexities. Furthermore, with the local and global cross-zone dependencies modeled, ACroCI demonstrates superior accuracy than scalar-based networks, leading to 21%∼51% accuracy improvements.

We further pick the rush-hour periods (7:00–9:00, 12:00–14:00, and 16:00–18:00 of the weekdays) to evaluate the prediction robustness of ACroCI and other schemes. Table 2(b) shows higher MAEs, RMSEs, and PCRs during rush hours compared to the overall performance due to more dynamic and larger volumes of crowd flows. We can see that ACroCI still outperforms the other schemes by at least 22.7% for both datasets thanks to the multi-scale designs which adapt to the complex crowd flows.

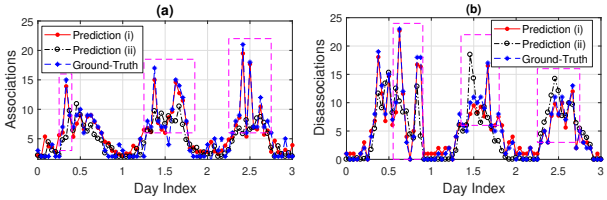

Fig. 14: Ground-truth & predictions for two APs.

We have further studied ACroCI's awareness to extreme crowd flows by showing the results of two APs ((a) & (b)) in an academic building. We compare in Fig. 14 their ground-truth time-series as well as the predicted ones by: *(i)* ACroCI with auxiliary extreme crowd flow learner, and *(ii)* ACroCI without it. We can observe from Fig. 14 that the case *(i)* achieves much better prediction results than case *(ii)* particularly for the extreme crowd flows (highlighted in pink boxes; ∼31.5% MAE reduction). Though extreme crowd flows happen less frequently than normal ones, such adaptivity and extreme-aware improvements will benefit the campus management and response given the foreseen potential extreme events in campus crowd flows.

Model Ablation and Sensitivity Analysis. We have conducted ablation studies on the different components of ACroCI. Fig. 15 presents the performance of several ACroCI variations: *(a)* without entire **t**emporal **l**earner designs (w/o TL), *(b)* without **t**emporal **a**ttention (w/o TA), *(c)* without **ext**ernal factors (w/o ext), *(d)* without **a**ttention **r**outing designs (w/o AR), *(e)* without spatio-temporal representations (w/o **F**), *(f)* without **e**xtreme **c**rowd **f**low learner (w/o ecf), *(g)* without **m**odel **i**nitialization (w/o mi), and *(h)* with all components (w/ all). For w/o AR, we adopt the conventional dynamic routing [22] within the capsule neural network structure of ACroCI. We can see that the MAE benefits more from the representations, spatial attention routing, extreme crowd flow learner, and model initialization (∼27% reduction), while the RMSE benefits more (∼45% reduction) from the temporal learner (including the temporal attention). Our proposed representations and spatial attention in ARCap can help mitigate the overall error trends, and the temporal learner designs assist in adapting to crowd dynamics. Performance improvements from w/o AR also imply ACroCI's preeminence over conventional capsule-based approaches.

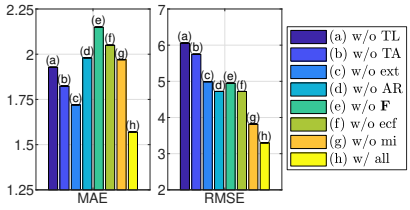

Fig. 15: Ablation studies on ACroCI's various components.

Case Studies & Interaction Visualization. We further focus on Campus A as we are more familiar with the local environments.

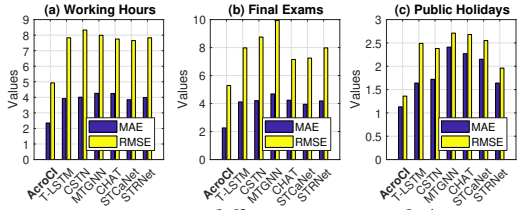

Fig. 16: Various cases at different time periods (Campus A).

Fig. 16 plots the performance of ACroCI and several state-of-the-art schemes, i.e., T-LSTM, CSTN, MTGNN, CHAT, STCaNet, and STRNet for three time periods: *(a)* working hours (8:00–17:00) of the weekdays (holidays excluded), *(b)* final exam weeks (one week in middle December), and *(c)* public holidays (including Thanksgiving Recess and Christmas). Compared to working hours, the mobility patterns during the final exams are more dynamic and hence larger errors can be observed. During the public holidays, all schemes have better performance due to lower crowd flow volumes. In all the above cases, ACroCI outperforms the other schemes by at least 26%, 31%, and 36%, respectively.

6 RELATED WORK

Wireless Mobility Analysis. Wireless mobility analysis has attracted much interest due to its business and social values. Qin et al. [21] mined the user behaviors based on mobile Wi-Fi usage inference. To the best of our knowledge, few of these efforts *predict* AP-level crowd flows using *adaptive* learning upon Wi-Fi association data in a campus network. We fill this gap by presenting an adaptive data learning system for accurate AP-level crowd flow prediction. In addition, we conduct comprehensive studies on how to deal with extreme flows and altered measurement environments, providing new insights for extreme-aware and adaptive wireless mobility analytics.

Various techniques have been proposed for group mobility sensing. Shen et al. [24] studied group detection based on the signal features of Wi-Fi probes; a similar study was conducted by Hong et al. [11]. Vision-based approaches [6, 26] can provide fine-grained crowd sensing, but their pervasiveness and applicability are subject to privacy concerns as well as environmental effects (e.g., none-line-of-sight and poor lighting). Unlike these techniques based on wireless/visual signals, we consider the campus Wi-Fi network setups as a case study and *forecast* crowd flows at each AP, using anonymized and less privacy-intrusive Wi-Fi association and disassociation data.

Crowd Mobility Modeling. Deep learning has recently been adopted to support various urban crowd mobility applications. For mobility learning, Zhang et al. [28] proposed leveraging a residual neural network to predict urban bike and taxi traffic. Huang et al. [12] studied the cross-interaction hierarchical attention networks for urban anomaly prediction. Scellato et al. [23] conducted a pioneering study that leverages time-series analysis models to predict the next location of a user based on her/his historical location visits. `ACroCI`, serving as an interactive wireless mobility data learning and crowd prediction system, differs from others as follows. `ACroCI` focuses on real-world Wi-Fi data learning, and is grounded on comprehensive and extensive data-driven system studies for *smart connected campus* application scenarios.

7 DEPLOYMENT DISCUSSION

Although one user may carry multiple mobile devices like laptops and smartphones, and certain pieces of office equipment like printers may connect to the campus Wi-Fi network, users can be easily identified by their IDs, allowing us to count the number of people (instead of devices) associated with an AP. Through such data pre-processing, we can use $\mathbf{A}^{(k)}$ and $\mathbf{D}^{(k)}$, at the user/client level, as reasonable indicators or ground-truths of user arrivals and departures. In future, we will make further improvements on the ground-truth labeling of arrivals and departures, e.g., calibrating session duration, handling unassociated devices and ping-pong effects, albeit beyond our current scope.

8 CONCLUSION

We have proposed `ACroCI`, a novel crowd mobility analytics system which models the crowds' interactions with the built environments with adaptive Wi-Fi data learning. Integrating with spatial crowd heatmaps, temporal flow dynamics, extreme features, environmental changes, and other external factors, `ACroCI` captures the crowds' interactions with built environments via a novel design of attention routing capsule network. Our extensive experimental studies have corroborated the accuracy, adaptivity, and robustness of `ACroCI`.

We would like to thank the University of Connecticut Information Technology Services (UITS) for their assistance in collecting Wi-Fi association data.

REFERENCES

[1] 2022. Weather Archive. https://rp5.ru/.
[2] Hamed S Alavi, Denis Lalanne, Julien Nembrini, Elizabeth Churchill, David Kirk, and Wendy Moncur. 2016. Future of human-building interaction. In *CHI*. 3408–3414.
[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
[4] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. 2019. Attention augmented convolutional networks. In *ICCV*. 3286–3295.
[5] Jaewoong Choi, Hyun Seo, Suii Im, and Myungjoo Kang. 2019. Attention routing between capsules. In *IEEE ICCV Workshops*.
[6] W. Ge, R. T. Collins, and R. B. Ruback. 2012. Vision-Based Analysis of Small Groups in Pedestrian Crowds. *IEEE TPAMI* 34, 5 (2012), 1003–1016.
[7] M Ghil, P Yiou, Stéphane Hallegatte, BD Malamud, P Naveau, A Soloviev, P Friederichs, V Keilis-Borok, D Kondrashov, V Kossobokov, et al. 2011. Extreme events: dynamics, statistics and prediction. *Nonlinear Processes in Geophysics* 18, 3 (2011), 295–350.
[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
[9] Suining He and Kang G. Shin. 2019. Spatio-Temporal Adaptive Pricing for Balancing Mobility-on-Demand Networks. *ACM TIST* 10, 4 (2019).
[10] Suining He and Kang G Shin. 2019. Spatio-temporal capsule-based reinforcement learning for mobility-on-demand network coordination. In *The World Wide Web Conference*. 2806–2813.
[11] Hande Hong, Girisha Durrel De Silva, and Mun Choon Chan. 2018. CrowdProbe: Non-invasive crowd monitoring with Wi-Fi probe. *ACM IMWUT* (2018).
[12] Chao Huang, Chuxu Zhang, Peng Dai, and Liefeng Bo. 2020. Cross-Interaction Hierarchical Attention Networks for Urban Anomaly Prediction. In *IJCAI*.
[13] Rob J Hyndman, Earo Wang, and Nikolay Laptev. 2015. Large-scale unusual time series detection. In *IEEE ICDM Workshop*. 1616–1619.
[14] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[15] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. 2017. Time-series extreme event forecasting with neural networks at Uber. In *ICML Time Series Workshop*, Vol. 34. 1–5.
[16] Lingbo Liu, Zhilin Qiu, Guanbin Li, Qing Wang, Wanli Ouyang, and Liang Lin. 2019. Contextualized spatial–temporal network for taxi origin-destination demand prediction. *IEEE T-ITS* 20, 10 (2019), 3875–3887.
[17] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).
[18] Ljubica Pajevic, Gunnar Karlsson, and Viktoria Fodor. 2019. CRAWDAD dataset kth/campus (v. 2019-07-01). https://crawdad.org/kth/campus/20190701/wifi-mapping. https://doi.org/10.15783/c7-5r6x-4b46
[19] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. 2010. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Netw.* 22, 2 (2010), 199–210.
[20] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE TKDE* 22, 10 (2009), 1345–1359.
[21] Zhou Qin, Yikun Xian, Fan Zhang, and Desheng Zhang. 2020. MIMU: Mobile WiFi Usage Inference by Mining Diverse User Behaviors. *ACM IMWUT* 4, 4, Article 149 (Dec. 2020), 22 pages.
[22] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *NIPS*. 3856–3866.
[23] Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Andrew T Campbell. 2011. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *Pervasive*. Springer, 152–169.
[24] Jiaxing Shen, Jiannong Cao, and Xuefeng Liu. 2019. BaG: Behavior-Aware Group Detection in Crowded Urban Spaces Using WiFi Probes. In *WWW*. 1669–1678.
[25] Shun-Yao Shih, Fan-Keng Sun, and Hung-Yi Lee. 2019. Temporal pattern attention for multivariate time series forecasting. *Machine Learning* 108, 8 (2019), 1421–1441.
[26] F. Solera, S. Calderara, and R. Cucchiara. 2016. Socially Constrained Structural Learning for Groups Detection in Crowd. *IEEE TPAMI* 38, 5 (2016), 995–1008.
[27] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *ACM SIGKDD*. 753–763.
[28] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *AAAI*. 1655–1661.