Enhancing Data-Driven Predictive Modeling of Pedestrian Crowd Flow with Spatial Priors – Case Studies with Post-Event Crowd Data on a University Campus

1st Vivian W.H. Wong*
Department of Urban and Regional Planning, and
M.E. Rinker, Sr. School of Construction Management
College of Design, Construction and Planning
University of Florida
Gainesville, Florida, USA
vivian.wong@ufl.edu

***Abstract*—Accurate crowd flow forecasting is crucial for effective crowd management and public safety in built environments. However, existing studies often overlook the importance of spatial connectivity in predicting crowd movements. This paper addresses this critical gap by incorporating spatial priors into crowd flow predictive modeling. We introduce a framework called the Spatio-Temporal Encoder Network (STEN), designed to simultaneously encode spatial information and temporal crowd data. Our study utilizes a newly collected dataset comprising two scenarios on a university campus, both containing spatial connectivity information. We evaluate the STEN framework's forecasting performance, comparing it with models that rely solely on temporal crowd flow data. Results demonstrate that models whose inputs include both spatial and temporal information consistently yield more accurate prediction outcomes compared to models whose inputs are only temporal crowd flow data. This finding underscores the importance of including spatial priors in enhancing the accuracy and reliability of crowd flow predictive modeling. We envision that our study will highlight the importance of spatial connectivity for future crowd management practices, and that our STEN framework and the accompanying dataset will serve as a valuable resource for researchers developing and evaluating crowd flow prediction models and contribute to identifying potentially congested regions and providing early warnings in our built environments.**

***Index Terms*—pedestrian crowd data, built environment, predictive modeling, graph neural network, recurrent neural network**

I. Introduction

The ability to accurately forecast crowd flow in different regions of a built environment is crucial for effective crowd management, event planning, and public safety. Anticipating future crowd states enables proactive decision-making and resource allocation, potentially preventing overcrowding, congestion, and associated risks. For example, identifying potential congestion hotspots can transform them into valuable places by enabling informed responsive planning and efficient space utilization. However, despite the importance of crowd flow forecasting, research on this problem has been limited by the lack of existing studies that provide both crowd flow data and the spatial connectivity information of the environment. Existing crowd datasets often focus on crowd counting or trajectory prediction, lacking the necessary spatial connectivity information about key places in our built environment.

To illustrate the critical role of spatial connectivity, consider a simplified scenario in a crowded venue with two camera-monitored hallways through which pedestrians are exiting. The spatial relationship between these hallways significantly impacts crowd flow predictions: If the hallways are arranged in parallel, leading directly from the main area to separate exits, a high flow in one hallway would likely correspond to a lower flow in the other, as the total exiting crowd is distributed between them. Conversely, if the hallways are in series, with one leading to the other, we would expect the crowd flow to be roughly equal in both hallways, as the same group of people passes through both in sequence. This simple motivating example underscores how crucial spatial connectivity information is for accurate crowd flow forecasting. Yet, most existing studies do not include such vital data, limiting their predictive power and applicability in real-world scenarios.

While data-driven methods are frequently employed to study pedestrian and crowd behaviors, such data-driven approaches often overlooking the crucial role of spatial connectivity in shaping crowd movements. These studies, while useful for certain aspects of crowd analysis such as pedestrian counting and tracking, do not offer insights on how the built environment could affect crowd movement.

For instance, established pedestrian datasets such as the ETH dataset [1], UCY dataset [2], and the Stanford Drone

*This article builds partially on Chapters 3 and 4 of the author's Ph.D. dissertation at Stanford University, titled "Spatio-Temporal Representation Learning: Applications to Manufacturing Planning and Pedestrian Crowd Analysis." The author would like to thank advisor Dr. Kincho H. Law and dissertation readers Dr. Michael D. Lepech and Dr. Renate Fruchter for their guidance and support. Financial support from the Stanford Blume Fellowship is gratefully acknowledged.

Dataset [3], provide valuable pedestrian tracking information but has no information about the surrounding space.

Similarly, datasets designed for crowd counting, such as the WorldExpo'10 dataset [4], which contains 108 surveillance videos from the Shanghai 2010 WorldExpo, the ShanghaiTech dataset [5], and the UCF-QNRF dataset [6], all featuring extremely dense crowd images. However, these datasets are primarily designed for crowd counting and do not provide the spatial connectivity information necessary for modeling crowd flow between different regions.

Even datasets that offer multiple camera views, such as the CrowdFlow dataset [7] and the CUHK Crowd Dataset [8], primarily target crowd behavior analysis and anomaly detection. These datasets, while providing richer visual information, still fall short in explicitly representing the spatial layout and connectivity of the environment.

This prevalent lack of spatial information in existing crowd studies creates a significant gap in our understanding of crowd dynamics. Crowd movements are not isolated phenomena; they are intricately linked to the spatial configuration of the environment. Factors such as the layout of pathways, the presence of obstacles, and the connectivity between different areas play a crucial role in determining crowd flow patterns.

These constraints call for studies that provide not only varying-density crowd scenarios but also the spatial connectivity essential for a comprehensive understanding and modeling of pedestrian flow between different regions of a built environment. Our study aims to address this gap by incorporating spatial priors into the predictive modeling of crowd flow. Previous research has demonstrated the value of incorporating spatial interactions into predictive models, such as using Long Short-Term Memory (LSTM) networks to simulate pedestrian movements [9] and graph-based methods like Social-STGCNN [10] to capture inter-pedestrian interactions; however, these studies often overlook the spatial connectivity of physical spaces such as doors, stairs, and tunnels.

In this paper, we first briefly introduce the mathematical notations that define the crowd flow forecasting problem, building upon our previous works [11], [12]. Subsequently, we propose a framework named Spatio-Temporal Encoder Network (STEN) for the predictive modeling of crowd flow. STEN is designed to simultaneously encode spatial information and temporal crowd data, thereby capturing the complex interplay between spatial layout and crowd dynamics. Lastly, we present experimental results conducted on a newly collected dataset comprising two scenarios on a university campus. Crucially, both scenarios include detailed spatial connectivity information. We evaluate forecasting performance for up to 4 minutes ahead, and demonstrate that models whose inputs are both spatial and temporal information consistently yield more accurate prediction outcomes compared to models whose inputs are only temporal crowd flow data. This result underscores the importance of including spatial priors in enhancing the accuracy and reliability of crowd flow predictive modeling.

II. Problem Statement

The research problem presented in this section is to forecast pedestrian crowd flow within a complex built environment with multiple Pedestrian Activity Regions (PARs), or locations with potentially high pedestrian activity, such as exits and entrances, and frequent visited services. Specifically, we define this problem as follows: given (1) historical crowd flow data, and (2) spatial prior about the connectivity of pedestrian locations in the built environment, our objective is to predict future crowd flow data at each PAR. In this section, we first elaborate on how a crowd flow state is represented mathematically. Subsequently, we present the formal problem formulation for crowd flow forecasting.

A. Crowd Flow State Representation

Crowd movement inherently manifests in both space and time. The flow of people within each Pedestrian Activity Region (PAR) fluctuates over time, with individuals transitioning in and out of PARs and traversing between adjacent ones spatially. Previous work in [12] presented an integrated data representation, Crowd Mobility Graphs (CMGraphs), to simultaneously represent (1) temporal crowd flow data, and (2) spatial prior about the connectivity of multiple PARs in a built environment.

A single CMGraph represents a snapshot of the crowd mobility state at time step t. A CMGraph can be written as an undirected, unweighted, and dynamic graph $G_t = (V, E, \mathbf{X}_t)$, where t is a discretized time step, V is the set of vertices (i.e. nodes), $E \subseteq V \times V$ is set of edges that represent connectivity between the PARs, and $\mathbf{X}_t$ is the nodal feature matrix. Note that V and E are time-invariant, as we assume the built environment remains unchanged throughout the forecasting period. In other words, it is assumed that neither the PARs nor the routes between them are tentatively added or blocked.

Over a time horizon, a sequence of CMGraphs, one at each time step, is used to represent the crowd flow over time. Each CMGraph also captures spatial information of the PAR connectivity with the graph topology. The following spatio-temporal crowd mobility variables are used to construct the CMGraphs:

1) **Time step, observed and prediction time horizons**: A time step is a discretization of time in the crowd flow forecasting problem, which is a time series problem that involves forecasting future crowd flow information based on past observations — given the crowd flow information during the observed discrete time horizon 1 to T_{obs}, the aim is to predict the crowd flow information from time T_{obs+1} to T_{pred}.
2) **Pedestrian Activity Region (PAR)**: A PAR is represented as a node indexed by v_i, where $i \in \{1, \ldots, N\}$ with N representing the total number of PARs. The node set $V = \{v_1, v_2, \ldots, v_N\}$ is therefore also the set of all PARs.
3) **Spatial connectivity prior**: Spatial connectivity defines the topological linkage between PARs. Two PARs are

considered connected if pedestrians can move directly between them without the need to enter a third region. Spatial connectivity is treated as a prior information, as it is not directly captured by sensors such as surveillance cameras, but is rather manually annotated from provided floor plan information. We represent this connectivity between PARs as time-invariant edges of the CMGraphs, denoted as E, where an edge $e_{jk} \in \{0, 1\}$ connecting node v_j and node v_k is 1 if two egress regions are adjacent to each other and 0 otherwise. The set of edges can be represented as the adjacency matrix $\mathbf{A} \in \{0, 1\}^{N \times N}$.

4) **Temporal crowd flow information**: Crowd flow information at time step t is represented as a feature matrix $\mathbf{X}_t \in \mathbb{R}^{N \times D}$, where N is the number of PAR and D is the number of crowd flow features associated with each individual PAR. In this study, two crowd flow features (i.e. $D = 2$) are used for experiments — pedestrian count over time (in people per second, or pp/s) and timestamp (in seconds).

B. The Crowd Flow Forecasting Problem

Following the above definition of a crowd flow state, the crowd flow forecasting problem is formulated as a sequence generation task that aims to learn a mapping from crowd flow state in the observed time horizon to that in the predicted time horizon.

$$\hat{\mathbf{Y}} = [\hat{y}_t(v)]_{T_{\text{obs}} < t \leq T_{\text{pred}}, v \in V} = f(G_1, G_2, \ldots, G_{T_{\text{obs}}}) \quad (1)$$

where f is the function that maps inputs to the output, the inputs are the time series of observed CMGraph-represented, spatio-temporal crowd flow state $G_1, G_2, \ldots, G_{T_{\text{obs}}}$, and the output is the time series of crowd flow forecasts, $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times (T_{\text{pred}} - T_{\text{obs}})}$.

III. Method: Spatio-Temporal Encoder Network (STEN) Framework

This section proposes the Spatio-Temporal Encoder Network (STEN), a crowd flow forecasting system that iteratively learns and integrates the spatial and temporal features in order to generate prediction outputs. A schematic of the overall system framework is illustrated in Figure 1.

In our earlier section, we described the objective of the forecasting task, which is to learn the function f that maps historical observations to predicted future crowd flow data, as shown in equation 1. The challenge lies in enabling f to capture both spatial and temporal information in the CMGraphs using the dynamic CMGraphs as **inputs** (Fig. 1a). To achieve this, we introduce the concept of embeddings, which are high dimensional vector representations of data features, and are learned by models so that relationships within the data are represented. In our context, **spatial embedding** (Fig. 1b) is learned in order to capture the topological structure or spatial relationships within the CMGraphs. It transforms the information about how different regions or nodes in the graph are connected or related into a high-dimensional vector. Temporal embedding, on the other hand, is concerned with capturing the temporal dynamics or changes of crowd flow features over different time steps. In our approach, the generation of temporal embeddings, derived from sequential spatial embeddings as inputs, effectively results in a **spatio-temporal embedding** (Fig. 1c) that combines both spatial and temporal information. To learn these spatio-temporal embeddings and to combine these representations for generating the final **prediction output** (Fig. 1d), we propose a deep learning framework, spatio-temporal Encoder Network (STEN). STEN consists of two encoder modules for embedding generation:

1) Spatial encoder
2) Temporal encoder

While STEN provides a general framework for crowd flow forecasting, specific models need to be selected for the spatial and temporal encoder modules. The spatial encoder module is a network that generates an embedded graph. Among the many graph neural networks that can achieve this, we experiment with the simple graph convolutional network (GCN) [13] in the scope of this study. Additionally, to remove over-smoothing effect of deep GCN when generating spatial embeddings, we experiment with the addition of dense connections [14].

Similar to the spatial encoder module, the temporal encoder module can consist of any model architecture that can generate an embedding from sequential data input. Examples include multi-layer perceptron (MLP) [15], transformer [16], and recurrent neural network. Experiments in this study is conducted with the temporal encoder module being a the gated recurrent unit (GRU), which is a type of recurrent neural network.

The final step of STEN is a linear layer, where a learnable weight matrix is used to transform the length of the spatio-temporal embedding vector to the length of the desired output, $T_{\text{pred}} - T_{\text{obs}}$.

A. Spatial Encoder: Dense Graph Convolution Network (Dense-GCN)

Dense graph convolutional network (dense-GCN) is employed as the spatial encoder module for STEN. In the following, the motivation and computation methods of GCN and dense connections are introduced.

Due to its simplicity, we first experiment with graph convolutional network (GCN), one of the first GNNs developed and popularly used for several graph learning applications [13]. GCN is chosen for its simplicity and effectiveness in learning node embedding from neighborhoodal information in graphs. For a set of N nodes in a CMGraph $G_t = (V, E, \mathbf{X}_t)$, a GCN layer updates the nodal information using a target node's neighboring nodal information for all nodes. More formally, given a target node v_i, whose node embedding vector is $\mathbf{x}_t(v_i)$ (the i^{th} row of the feature matrix $\mathbf{X}_t$), and its set of neighboring nodes J, a GCN layer updates the target node embedding as follows:

$$\mathbf{x}(v_i)^{(k)} = \frac{\mathbf{W}^{(k)}}{|J|} \sum_{v_j \in J} \mathbf{x}(v_j)^{(k-1)} \quad (2)$$

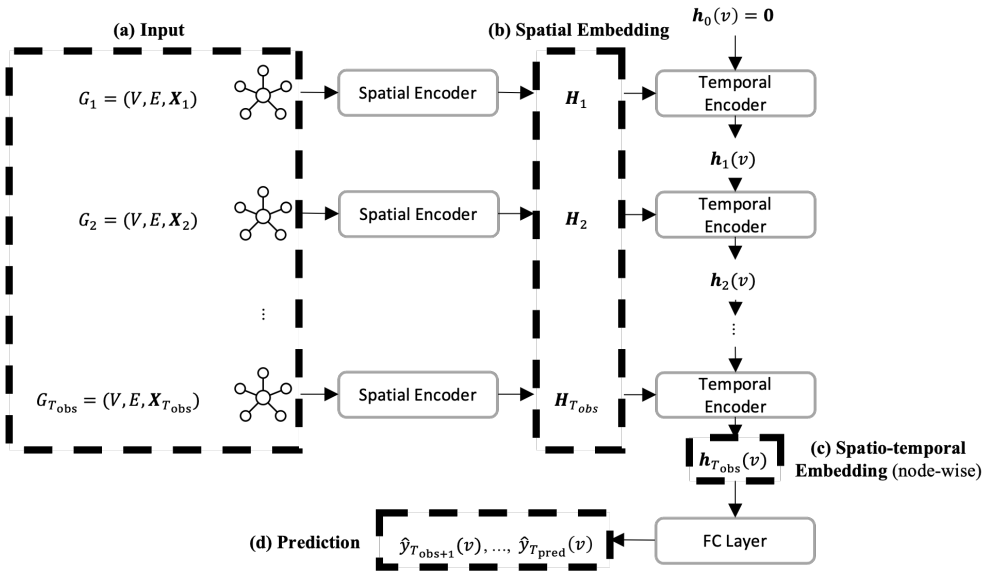

Fig. 1. Illustration of the STEN System. (a) CMGraphs are used as inputs to the system. (b) A spatial embedding of each graph is generated by the spatial encoder module. (c) For each node, a spatio-temporal nodal embedding is generated by the temporal encoder module. (d) A fully connected (FC) layer is used to transform the dimension of the final prediction output, the crowd flow forecasts.

where $\mathbf{W}^{(k)}$ and $\mathbf{x}(v_i)^{(k)}$ are a learnable parameter and the i^{th} node's updated embedding of the k^{th} layer, respectively. In the first layer, $\mathbf{x}(v_i)^{(0)}$ is the initial feature vector of node v_i, for all $v_i \in V$.

Stacking K GCN layers allow us to update node embeddings using information aggregated from nodes in the K-hop neighborhood. After K GCN layers, the embedded graph, G'_t is learned, whose node embedding matrix is $\mathbf{X}'_t$, each row being x'_i, the updated embedding vectors of node i. Each node embedding vector is of an embedding dimension D_{GCN}, a tunable hyperparameter. The dimension of $\mathbf{X}'_t$ is therefore $N \times D_{GCN}$.

Having deep layers of GCN stacked in a model, however, exhibit the issue of over-smoothing, as observed by Li et al. [17]. Dense connections have been shown to be effective in reducing this effect in deep GCNs [14]. The concept of dense connections, first introduced in the convolutional neural network (CNN) model DenseNet [18], involves concatenates an output from earlier layers with an output from later layers. Thus, in this study, we have incorporated the concept of dense connections from CNNs into GCNs by concatenating the GCN-learned spatial embedding, $\mathbf{X}'_t$, with the original input, $\mathbf{X}_t$. The resulting output of this architecture, referred to as Dense-GCN, is denoted as $\mathbf{H}_t \in \mathbb{R}^{N \times (D_{GCN}+D)}$. The schematic of the dense-GCN computation is shown in Figure 2.

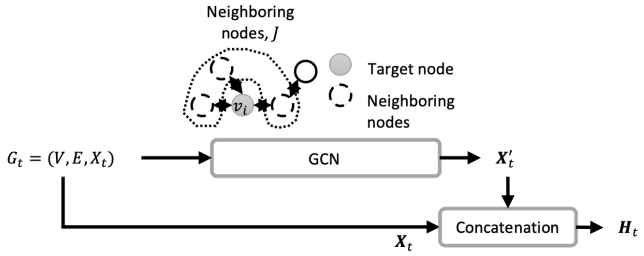

Fig. 2. Spatial Encoder Module: Dense-GCN.

B. Temporal Encoder: Recurrent Neural Network

GRU can be used to encode hidden state representations of time series inputs. Mathematically, each GRU operation in a layer l can be expressed as follows:

$$r_t^{(l)} = \sigma(\mathbf{W}_{ar}^{(l)} a_t^{(l)} + b_{ar}^{(l)} + \mathbf{W}_{hr}^{(l)} h_{t-1}^{(l)} + b_{hr}^{(l)}) \quad (3)$$

$$z_t^{(l)} = \sigma(\mathbf{W}_{az}^{(l)} a_t^{(l)} + b_{az}^{(l)} + \mathbf{W}_{hz}^{(l)} h_{t-1}^{(l)} + b_{hz}^{(l)}) \quad (4)$$

$$n_t^{(l)} = \tanh(\mathbf{W}_{an}^{(l)} a_t^{(l)} + b_{an}^{(l)} + r_t^{(l)} * (\mathbf{W}_{hn}^{(l)} h_{t-1}^{(l)} + b_{hn}^{(l)})) \quad (5)$$

$$h_t^{(l)} = (1 - z_t^{(l)}) * n_t^{(l)} + z_t^{(l)} * h_{t-1}^{(l)} \quad (6)$$

where $\mathbf{W}_{ar}^{(l)}$, $\mathbf{W}_{hr}^{(l)}$, $\mathbf{W}_{az}^{(l)}$, $\mathbf{W}_{hz}^{(l)}$, $\mathbf{W}_{an}^{(l)}$, $\mathbf{W}_{hn}^{(l)}$, $b_{ar}^{(l)}$, $b_{hr}^{(l)}$, $b_{az}^{(l)}$, $b_{hz}^{(l)}$, $b_{an}^{(l)}$, $b_{hn}^{(l)}$ are learnable parameters of the l^{th} layer. $a_t^{(l)}$ is the input to the layer and is equal to the node-wise spatial embedding from the spatial encoder module, $\mathbf{H}_t$, at layer $l = 0$. $h_t^{(l)}$ is the hidden state of the l^{th} layer at time t. $h_{t-1}^{(l)}$ is the hidden state of the layer at time $t-1$ or the initial hidden state at time 0. σ is the sigmoid function. $r_t^{(l)}, z_t^{(l)}, n_t^{(l)}$ are the reset, update, and new gates of the l^{th} layer, respectively. $*$ denotes element-wise multiplication. The final output at time $t = T_{\text{obs}}$ after L GRU layers is then $h_{T_{\text{obs}}}^{(L)}$, a vector with length D_{GRU}, a tunable hyperparameter.

As the temporal encoder performs node-wise operations, for simplicity, we slightly abuse notation and use $h^{(L)}_{T_{\text{obs}}}$ to represent $h^{(L)}_i(T_{\text{obs}})$, which signifies the spatio-temporal embedding for the i^{th} node (i.e., i^{th} PAR). Here, we omit the node-specific subscript i, which previously denoted values corresponding to the i^{th} node. Additionally, it is important to note that in the final step of the proposed STEN system, the vector $h^{(L)}_{T_{\text{obs}}}$ undergoes a transformation to the desired prediction output size via a fully connected (FC) layer.

IV. Experiments

This section present the test set results from two experiments aiming to evaluate the performance of the forecasting models. The first experiment compares the models in across the three different scenarios with a controlled forecasting horizon of 20 steps to provide insight on the overall model performance. The second experiment extends the forecasting horizon in the first experiment to evaluate how the models cope with short-term versus long-term predictions.[1]

A. Data Collection and Preprocessing

To accurately quantify the pedestrian activity within multiple PARs, a challenge arises when these areas are not fully visible through a single camera lens, either due to considerable distances between them or obstructions like walls that impede direct line of sight.To overcome this limitation, we employed an array of cameras for the comprehensive collection of a new dataset: Campus Crowd. Our dataset collection method ensures that the dynamics of crowd movement across dispersed PARs are captured effectively. The mapping of cameras and the ID of the PARs each camera captures is tabulated in Table I.

Adhering to the guidelines set by the Institutional Review Board (IRB), direct access to these video recordings is restricted to safeguard the privacy and personal identifiable information of the individuals in the recordings. Nonetheless, to facilitate research and application development while respecting privacy concerns, we have compiled and released the aggregated crowd counts within each PAR, derived from frame-by-frame crowd count extraction using YOLO-v7tiny, using methods detailed in [11]. To enhance computational efficiency, this analysis was conducted on a reduced frame rate, with the dataset parameters outlined in Table I. Furthermore, to encompass information on how crowds flow over the PARs, the data is released in the form of CMGraphs, whose topology naturally represent the spatial connectivity. The CMGraphs are PyTorch Geometric graph objects [19], which is commonly used in graph-learning research. Our dataset provides a valuable tool for studying crowd dynamics, ensuring both the utility of the data for research purposes and adherence to privacy regulations.

All above-mentioned parameters that describe the cameras, PARs, and video properties of the two scenarios are presented in Table I.

[1]Our publically accessible code of the Campus Crowd dataset and the experimental implementations are available at https://github.com/vivian-wong/Campus-Crowd

1) Scenario 1: Stadium: Raw videos for this scenario were recorded at the end of the 2023 university-wide commencement ceremony at a football stadium. The videos capture dense crowds exiting the stadium through 6 tunnels that were open for entry and exit during the commencement. All tunnels are covered by the recordings, with one camera placed at each tunnel. The cameras used include 5 phone cameras from the iPhone series (one iPhone 7, one iPhone 13, one iPhone 13 Pro Max, and two iPhone 14 Pro Max) and one GoPro Hero 10. To simulate the elevation of regular surveillance cameras, each camera was securely mounted on railings, which are high vantage points near the tunnels. Sample video snapshots, camera placement illustrations, and the PAR definitions are shown in Figure 3.

2) Scenario 2: SEQ: Raw videos for this scenario were recorded following a large class dismissal near a university campus' Science and Engineering Quad (SEQ) on the first day of instruction of a semester. Class attendance is estimated to be at its peak during this time, with around 250 students present. Upon class dismissal, students flow into the engineering quad and exit in 10 different directions. Three cameras (one iPhone 13, one iPhone 13 Pro Max, and one GoPro Hero 10) were mounted at high vantage points to capture activity on the quad. Sample video snapshots, camera placement illustrations, and the PAR definitions are shown in Figure 4.

B. Evaluation Metrics

STEN-based models are evaluated on the test set of the crowd flow dataset. The mean squared error (MSE) between the node feature matrix of the predicted graph sequence and of the true sequence is used as an evaluation metric of prediction accuracy. The MSE loss measures the difference between the PAR-wise predicted crowd flow and the true crowd flow, averaged over the total number of PARs and the total number of time steps, and is therefore

$$\text{MSE} = \frac{1}{NT}\sum_{v\in V}\sum_{t=1}^{T}(y_t(v) - \hat{y}_t(v))^2 \tag{7}$$

where N is the number of nodes/PARs, T denotes the length of the forecasted horizon $T_{\text{pred}} - T_{\text{obs}}$, $y_t(v)$ is the true crowd flow and $\hat{y}_t(v)$ is the predicted crowd flow at the i^{th} node (i.e. PAR) at time t.

The mean absolute error (MAE) is also reported as an evaluation metric, as MSE places more penalization on larger errors with the squared error term, making MSE more susceptible to outliers. MAE measures the average of magnitude difference between the prediction and the true node feature matrices:

$$\text{MAE} = \frac{1}{NT}\sum_{v\in V}\sum_{t=1}^{T}|y_t(v) - \hat{y}_t(v)| \tag{8}$$

C. Implementation Details

The Dense-GCN-GRU model uses a 3-layer ($K = 3$) GCN to learn the spatial representations, and a 2-layer ($L = 2$) GRU to learn the temporal representations. The number of node features is $D = 2$ and are (1) the aggregated crowd

TABLE I

CAMPUS CROWD DATASET DESCRIPTION. FOR PAR IDS, EACH PARENTHESIS INDICATES SURVEILLANCE COVERAGE BY ONE CAMERA. PEDESTRIAN COUNT ARE BASED ON THE ANNOTATION OBTAINED FROM DETECTION AND TRACKING ALGORITHMS.

Parameters	**Stadium**	**SEQ**
Cameras	6	3
PAR ID	(1), (2), (3), (4), (5), (6)	(1,2,3,4), (5,6,7,8,9), (10)
Frame rate of raw recording (fps)	30	23
Processing frame rate (fps)	1	1
Resolution	2K	2K
Recording length (min:sec)	39:58	8:44
Max pedestrian count per camera frame	92	21

Note: fps = frames per second.

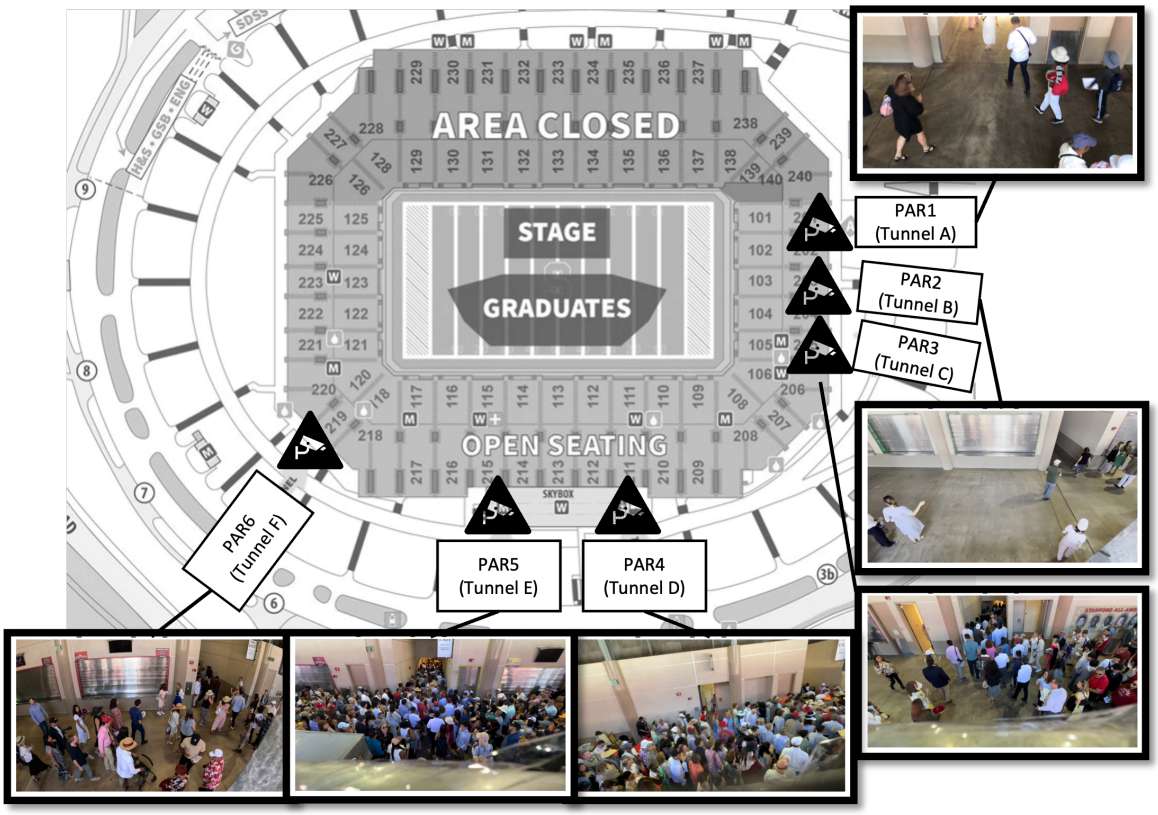

Fig. 3. (a) Sample snapshots and camera placements of Stadium. (b) Edge connectivity of the CMGraphs constructed from the PARs defined.

count, and (2) timestamp for each egress region. The embedding dimension of the GCN encoder is $D_{GCN} = 128$. The embedding dimension of the GRU part is $D_{GRU} = 64$. The graph data is batched into minibatches of size 32 for training. The Adam optimizer with a learning rate of 0.001 is used to train the GCN-GRU model as well as each baseline model for at most 40 epochs. The loss function used is MSE loss, as detailed in Equation 7.

To assess the performance over varying lengths of the forecasting horizon (i.e. short vs. long term forecasting performance), we experiment on $T_{\text{obs}} = T_{\text{pred}} - T_{\text{obs}} = 20, 60, 120, 240$ in the study. In other words, the input and the output sequences are equal in lengths. Since annotations of SEQ and Stadium are 1 FPS, the shortest and longest forecasting horizons assessed are 20 seconds and 4 minutes, respectively.

All training and inference were conducted on the same computer, equipped with an Intel Core i7-7820X processor and a NVIDIA GeForce GTX 1080 Ti GPU.

D. Specific STEN Model and Baseline

For brevity, we refer to the STEN-based model that employs Dense-GCN as the spatial encoder and GRU as the temporal encoder as Dense-GCN-GRU. To assess whether incorporating spatial information with the CMGraphs can indeed enhance forecasting accuracy, we compare the Dense-GCN-GRU model with a baseline model with just GRU. The GRU model treats inputs as purely temporal signals and do not involve the graph's adjacency matrix in its computation,

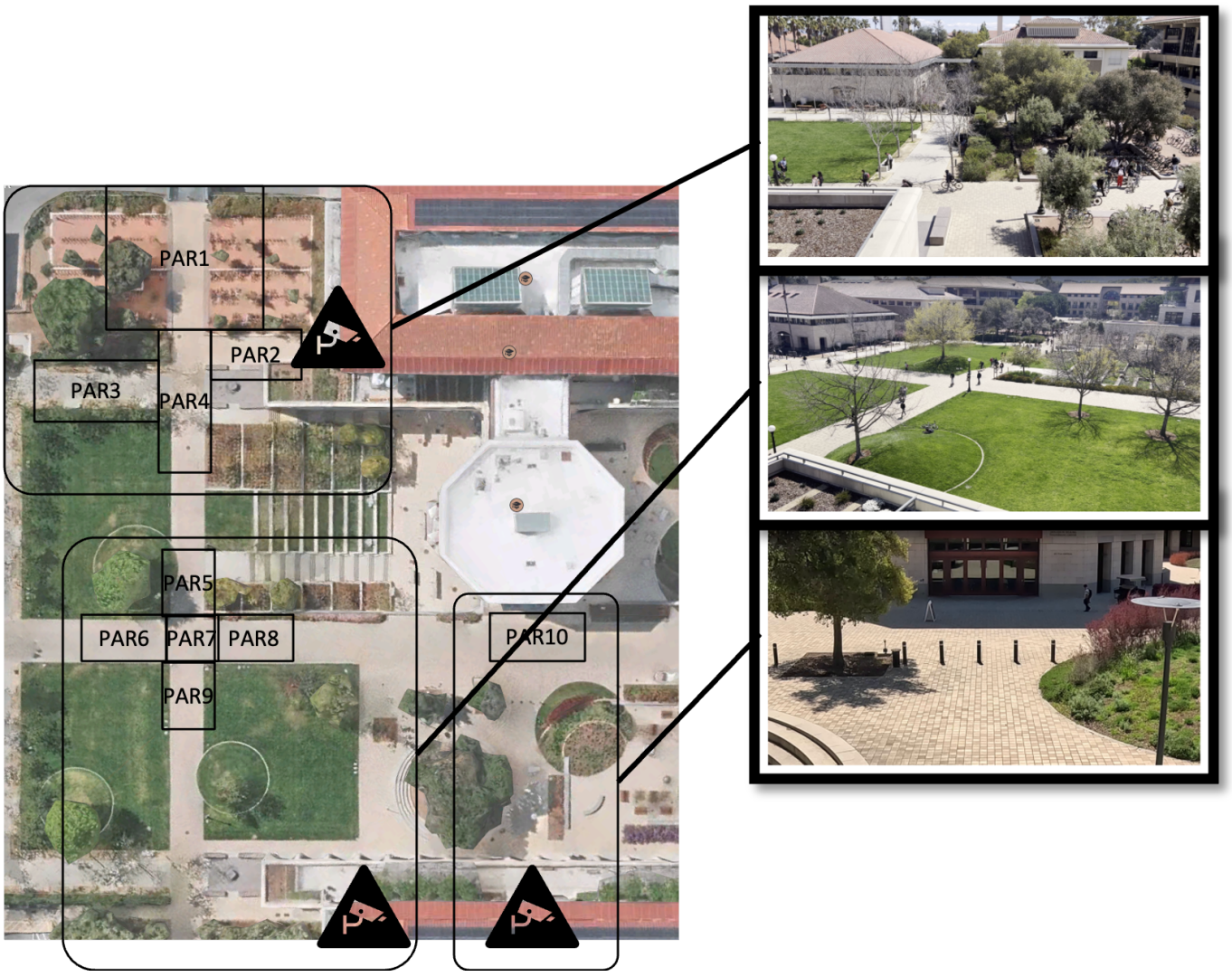

Fig. 4. (a) Sample snapshots and camera placements of SEQ. (b) Edge connectivity of the CMGraphs constructed from the PARs defined.

thereby leaving out the spatial connectivity information given by the floor plan of the surrounding space.

On the other hand, we verify whether the addition of dense connection in the spatial encoder reduces the over-smoothing effect of deep GCNs. To assess the performance of adding the dense connection, we conduct comparison experiments with another baseline method, the GCN-GRU model, which omits the dense connection, and therefore directly uses the un-concatenated spatial embedding $\mathbf{X}'_t$ as the input to GRU (rather than the $\mathbf{H}_t$, the embedding after concatenation, in the Dense-GCN-GRU model).

When choosing a model to deploy into real world application, the computing resources required should also be taken into account. Parameter sizes of each model under the forecasting horizon of 20 is listed in Table II.

All models are trained with the same hardware setup and hyperparameters.

TABLE II
NUMBER OF PARAMETERS FOR EACH MODEL.

Model	Number of Parameters
Dense-GCN-GRU	97,300
GCN-GRU	96,916
GRU	39,316

E. Comparison of Forecasting Performance by Scenarios

1) Short Term Forecasts: The analysis of model performances across various test sets on the short-term forecasting horizon of 20 time steps is plotted in Figure 5, with numeric values are provided in Table III. On both scenarios, Dense-GCN-GRU consistently outperformed the other models, as evidenced by its lower Mean Squared Error (MSE) and Mean Absolute Error (MAE).

GRU, while less accurate than Dense-GCN-GRU, still maintained a moderate level of performance. However, GCN-

TABLE III
MSE AND MAE OF CROWD FLOW FORECASTING MODELS AT A FORECASTING HORIZON OF 20 ACROSS DIFFERENT DATASETS. THE LOWEST MSE AND MAE VALUES FOR EACH DATASET ARE BOLDED.

Model	Scenario	MSE	MAE
Dense-GCN-GRU	SEQ	**0.247**	**0.333**
GCN-GRU	SEQ	0.764	0.627
GRU	SEQ	0.299	0.354
Dense-GCN-GRU	Stadium	**0.034**	**0.130**
GCN-GRU	Stadium	0.076	0.197
GRU	Stadium	0.041	0.142

TABLE IV
MODEL PERFORMANCE COMPARISON AT A FORECASTING HORIZON OF 240 ACROSS DIFFERENT SCENARIOS. MSE AND MAE OF CROWD FLOW FORECASTING MODELS AT A FORECASTING HORIZON OF 20 ACROSS DIFFERENT DATASETS. THE LOWEST MSE AND MAE VALUES FOR EACH DATASET ARE BOLDED.

Model	Scenario	MSE	MAE
Dense-GCN-GRU	SEQ	**0.336**	**0.410**
GCN-GRU	SEQ	0.782	0.645
GRU	SEQ	0.965	0.772
Dense-GCN-GRU	Stadium	**0.036**	**0.140**
GCN-GRU	Stadium	0.048	0.160
GRU	Stadium	0.078	0.201

GRU demonstrated relatively higher error rates, suggesting limitations in its predictive capabilities compared to the other models. One potential explanation for this observation is that the GCN model over-smoothed target node signals, whereas the GRU model does not aggregate neighboring node signals. On the other hand, the Dense-GCN-GRU model preserves the original target node signal through the use of dense connections, resulting in the highest forecasting accuracy amongst the compared models. With the SEQ setting, which has the fewest number of datasets, the over-smoothing effect of GCN-GRU worsens, as exhibited by the unusually high MSE. However, the addition of dense connection is a viable solution, as Dense-GCN-GRU reduces MSE and MAE.

The variation in model performance across datasets suggests that specific model architectures may be more adept at capturing the dynamics of crowd flow in different scenarios. Dense-GCN-GRU's overall efficacy indicates its robustness in handling varied spatial and temporal patterns and dataset sizes for the task of short term crowd flow forecasting.

2) Long Term Forecasts: Similarly, analyses are conducted on the three models and three scenarios with a longer forecasting horizon of 240 time steps. We present the test MSE and MAE of the models in Figure 6 and Table IV.

At the extended forecasting horizon of 240, we observe distinct variations in model performance across different datasets. Similar to the trends noted in shorter-term forecasting, Dense-GCN-GRU consistently exhibits lower MSE and MAE values across all scenarios. This pattern underscores the model's robustness, effectively handling variations in sample sizes, as well as diverse spatial and temporal patterns. Furthermore, GCN-GRU shows improved performance compared to GRU in long-term forecasting tasks, suggesting that the incorporation of dense connections and spatial information through graphs could be advantageous for the task of short and long term crowd flow forecasting.

F. Comparison of Forecasting Performance by Forecasting Horizons

Motivated by the model performance variations in short and long term crowd flow forecasting, we further present a sensitivity analysis of different forecasting horizons of 20, 60, 120, and 240 time steps, averaging the MSE and MAE across the three scenarios.

Figure 7 provides a comprehensive view of how each model copes with short-term versus long-term predictions. Dense-GCN-GRU consistently shows lower MSE and MAE across all forecasting horizons, indicating its robustness and reliability in both short-term and long-term forecasting. The model's prediction accuracy slightly decreases as the horizon lengthens, but its performance remains the highest accuracy.

GCN-GRU, while generally exhibiting higher error rates than Dense-GCN-GRU, demonstrates an improvement in accuracy as the forecasting horizon increases, suggesting its potential suitability for longer-term predictions. However, its performance is still much worse than the Dense-GCN-GRU, which fuses spatial information via dense connection.

GRU shows a decline in performance with increasing forecasting horizon lengths. Its higher MSE and MAE values, especially at longer horizons, highlight potential limitations in capturing complex spatio-temporal dependencies over extended periods.

Overall, the integration of spatial information through graph-based approaches, as seen in Dense-GCN-GRU and GCN-GRU, appears to enhance forecasting accuracy, especially with longer forecasting horizons. Dense-GCN-GRU's consistent performance across various horizons and datasets underscores the effectiveness of combining dense connections with spatial dependencies in graph structures for robust spatio-temporal forecasting.

V. CONCLUSION

In this study, we introduced the STEN framework and the accompanying Campus Crowd dataset, both released as publicly available resources designed to facilitate research on crowd flow forecasting and related spatio-temporal representation learning tasks. Our method addresses the limitations of existing crowd studies by incorporating both crowd flow data and spatial connectivity information from real-world university campus environments. STEN leverages spatial and temporal encoders to combine embedding vectors obtained from both spatial and temporal crowd flow data. Experiments were conducted on two scenarios: a large-scale event at a football stadium and a class dismissal at the Science and Engineering Quad (SEQ). By offering these resources, we aimed to enable researchers to develop and evaluate novel machine learning models for understanding and predicting crowd dynamics in complex built environments.

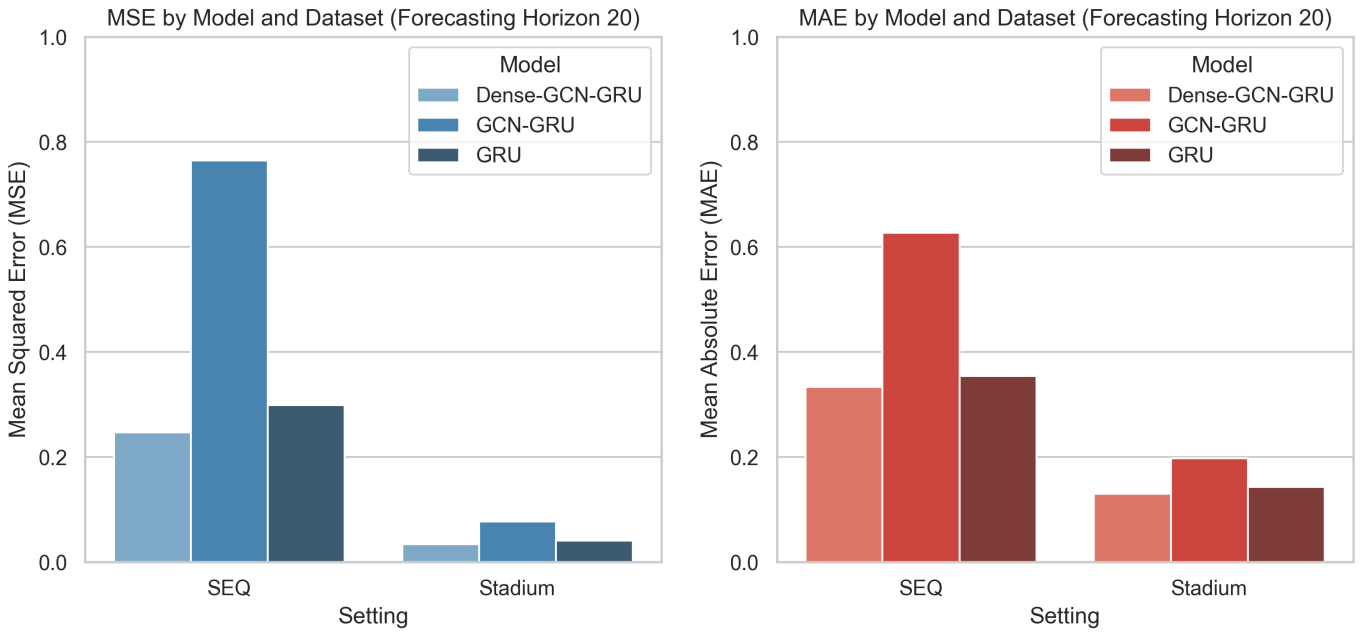

Fig. 5. MSE and MAE of crowd flow forecasting models at a forecasting horizon of 20 across different datasets. Lower MSE and MAE values indicate more accurate forecasts.

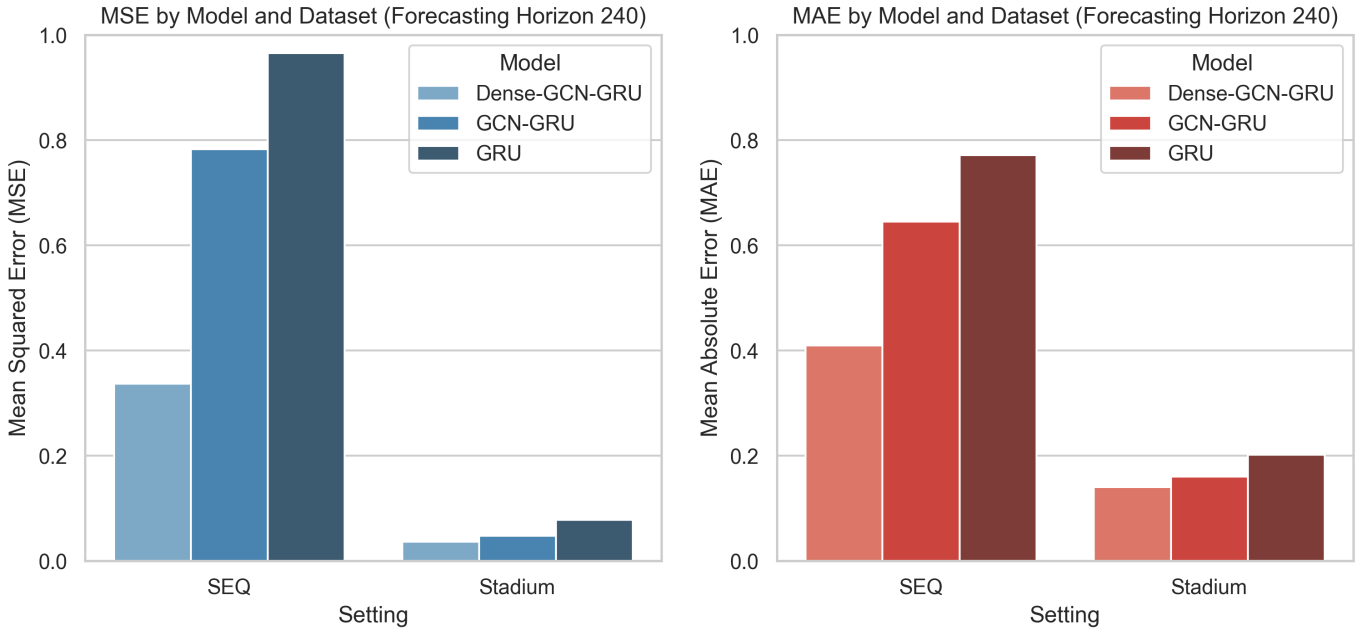

Fig. 6. MSE and MAE of crowd flow forecasting models at a forecasting horizon of 240 across different datasets. Lower MSE and MAE values indicate more accurate forecasts.

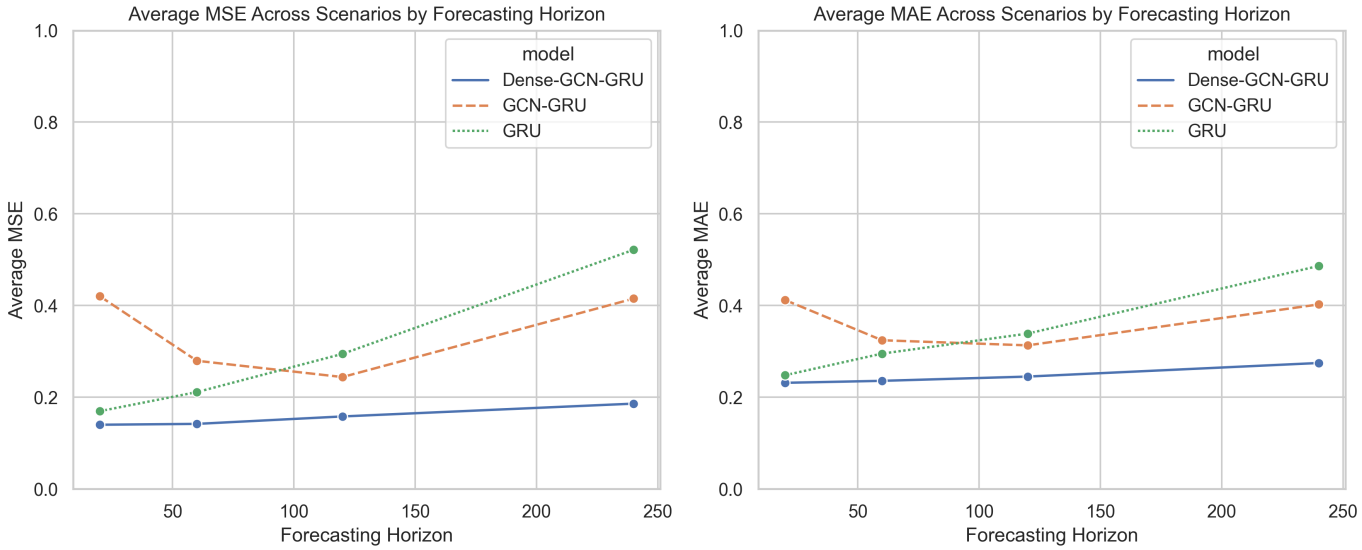

Fig. 7. MSE and MAE of crowd flow forecasting models at forecasting horizons at 20, 60, 120, and 240 time steps. Lower MSE and MAE values indicate more accurate forecasts.

Future work could focus on several aspects to further enhance the value of STEN and the Campus Crowd dataset and expand its applications. On the data side, while the current dataset includes two representative scenarios, future versions could incorporate more diverse settings, such as shopping malls, airports, or city centers, to capture a wider range of crowd behaviors and environmental factors. On the framework side, expanding the current framework with packaged implementations could facilitate a more comprehensive assessment of crowd flow forecasting models. Furthermore, building community tools such as leader board or data challenge could encouraging the research community to contribute to the dataset and to provide feedback that drives innovation in crowd flow forecasting and related fields.

References

[1] S. Pellegrini, A. Ess, K. Schindler, and L. V. Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," *2009 IEEE 12th International Conference on Computer Vision*, 2009.

[2] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by Example," *Computer Graphics Forum*, 2007.

[3] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes," *ECCV*, 2016.

[4] C. Zhang, K. Kang, H. Li, X. Wang, R. Xie, and X. Yang, "Data-driven crowd understanding: A baseline for a large-scale crowd dataset," *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1048–1061, 2016.

[5] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 589–597, IEEE, 2016.

[6] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. A. Al-Maadeed, N. M. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," *ArXiv*, vol. abs/1808.01050, 2018.

[7] S. Ali and M. Shah, "A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6, 2007.

[8] J. Shao, C. C. Loy, and X. Wang, "Scene-independent group profiling in crowd," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2227–2234, 2014.

[9] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] A. Mohamed, K. Qian, M. Elhoseiny, M. Elhoseiny, and C. G. Claudel, "Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction," *arXiv: Computer Vision and Pattern Recognition*, 2020.

[11] V. W. H. Wong and K. H. Law, "Fusion of CCTV video and spatial information for automated crowd congestion monitoring in public urban spaces," *Algorithms*, vol. 16, no. 3, p. 154, 2023.

[12] V. W. H. Wong and K. H. Law, "Modeling crowd data and spatial connectivity as graphs for crowd flow forecasting in public urban space," in *ASCE International Conference on Computing in Civil Engineering*, pp. 202–210, 2023.

[13] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *International Conference on Learning Representations (ICLR)*, 2017.

[14] G. Li, M. Müller, G. Qian, I. C. D. Perez, A. Abualshour, A. K. Thabet, and B. Ghanem, "Deepgcns: Making gcns go as deep as cnns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. Publisher: IEEE.

[15] S. Haykin, *Neural Networks: A Comprehensive Foundation*. USA: Prentice Hall PTR, 1st ed., 1994.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[17] Q. Li, Z. Han, and X.-M. Wu, "Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning," in *AAAI Conference on Artificial Intelligence*, 2018.

[18] G. Huang, Z. Liu, G. Pleiss, L. Van Der Maaten, and K. Weinberger, "Convolutional Networks with Dense Connectivity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[19] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric." https://github.com/pyg-team/pytorch_geometric, 2019.