# Modeling Crowd Data and Spatial Connectivity as Graphs for Crowd Flow Forecasting in Public Urban Space

Vivian W.H. Wong;[1] Kincho H. Law;[2]

[1]Engineering Informatics Group, Dept. of Civil and Environmental Engineering, Stanford University. Email: vwwong3@stanford.edu
[2]Engineering Informatics Group, Dept. of Civil and Environmental Engineering, Stanford University, Stanford, CA (corresponding author). Email: law@stanford.edu

## ABSTRACT

Predicting crowd flow patterns in a physical space can be useful for infrastructure management and safety planning. A simple representation of individuals in Euclidean space is insufficient for representing people's spatial distribution and movements over time. This paper describes a spatiotemporal graph formulation, namely Crowd Mobility Graphs (CMGraphs), to represent the spatiotemporal data. The CMGraphs model employs dynamic node features that store temporal crowd flow information, while the time-invariant edges represent spatial connectivity of locations of interests in the surrounding space. The spatiotemporal formulation using the CMGraphs allows for crowd flow prediction. Specifically, graph neural network is used to aggregate neighborhood nodal information on CMGraphs to capture spatial connectivity. Subsequently, recurrent neural network is employed to generate future sequences of crowd flow. An experiment is conducted using a publicly available video dataset at a train station to demonstrate the effectiveness of the proposed CMGraph formulation for crowd flow forecasting.

## INTRODUCTION

Understanding the distribution of people in a physical space is an important aspect of safe management and operation of a facility. For instance, uncontrolled crowding at egresses causes slow evacuation during disasters, and overly dense crowds can cause stampedes (Wang et al. 2015). The ability to predict crowd flow and distribution in a dynamically changing environment can be useful for infrastructure management and safety planning.

Predicting the movement of individuals in crowded scenes is a complex task as both spatial and temporal factors can influence their movement in a physical space. While many existing works model individuals as time series signals for movement prediction, for example using recurrent neural networks (Alahi et al. 2016; Mohamed et al. 2020), the spatial information of the surrounding space is rarely taken into account.

Representation of spatial connectivity and spatiotemporal crowd flow in Euclidean space is difficult. This study introduces Crowd Mobility graphs (CMGraphs) that is able to simultaneously represent spatial and temporal information. The spatial connectivities are modeled by the edges of the CMGraph and the temporal information are modeled with the dynamically changing nodal signals. CMGraphs can be exploited to facilitate the study of the crowd flow problem. Specifically, we propose a deep learning Dense-GCN-GRU model that uses graph convolutional network (GCN) and gated recurrent unit (GRU) to conduct crowd flow forecasting. By applying our CMGraph representation and Dense-GCN-GRU model to a publicly available surveillance video dataset at a train station, we demonstrate that the CMGraph formulation can be used for the spatiotemporal crowd forecasting problem. Furthermore, we compare the Dense-GCN-GRU model with two baseline methods to show the advantage of incorporating spatial connectivity and dense connection.

The remainder of the paper is organized as follows: the Literature Review section describes related research to the topic of spatiotemporal modeling for forecasting applications. The Problem Formulation section then describes the forecasting problem and the formulation of CMGraphs. The Spatiotemporal Forecasting of Crowd Flow section outlines the Dense-GCN-GRU methodology for the prediction of crowd flow. The Case Study section presents experimental results using a train station dataset, as an illustration of the potential application of the suggested spatiotemporal modeling approach. Finally, the Conclusion section presents concluding remarks.

## LITERATURE REVIEW

This section briefly reviews prior research concerning human forecasting applications and graph-based spatiotemporal modeling approaches for various domains. Social LSTM (Alahi et al. 2016) uses a Long Short-Term Memory (LSTM) network to simulate pedestrian movements at several time steps and to capture dynamic interactions between pedestrians. Recently, researchers have begun incorporating spatial interactions into graph-based methods. By putting a kernel function on the weighted adjacency matrix, for instance, Social-STGCNN (Mohamed et al. 2020) captures inter-pedestrian interaction. However, spatial connectivity data of the physical space, such as the locations of doors, stairs, and tunnels, are not incorporated into these models.

Research studies in spatiotemporal forecasting problems outside the context of human mobility have considered the connection of the physical space. Panagopoulos et al. (2021) completed COVID-19 pandemic forecasting by constructing a graph connecting Italy, England, Spain, and France based on country-to-country travel, and then deploying a message-passing graph neural network to predict the pandemic's spread. Highway and taxi vehicular traffic problems have been widely investigated by using graphs to represent existing road networks and employing GCN and recurrent neural network for the forecasting task (Bai et al. 2021; Zhao et al. 2019). Graph formulation for these two problems can be derived from existing transportation networksm, such as road connections. Pedestrian crowds, on the other hand, lack such physical and spatial relationships, making the modeling of pedestrian crowd data into a graph structure considerably more difficult. In this study, we propose integrating spatial connectivity information into temporal crowd flow signals by manually identifying egress locations from floor layouts in our case study.

## PROBLEM FORMULATION

This section introduces the formulation of the crowd flow forecasting problem in terms of a sequence generation task and presents the modeling of crowd flow data as a sequence of Crowd Mobility Graphs (CMGraphs).

**Crowd Flow Forecasting.** The prediction of crowd flow is modeled as 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 $T_{pred}$. To study crowd flow in a built environment, we divide the physical space into $N$ egress regions, such as entry and exit tunnels and doors. Using floor plans of a given public space, these egress regions can be manually identified by observing the locations of such egresses. Often, egresses are equipped with sensors or surveillance cameras that can be used to collect and derive crowd flow information, including volume, density, and speed. Crowd flow information can then be used to construct a feature matrix at each observed time step, denoted as $X \in \mathbb{R}^{N\times D}$, where $N$ is the number of egress regions and $D$ is the number of features. Thus, the crowd flow forecasting problem can be formulated as a sequence generation task that aims to learn a function $f$ that maps historical feature matrices $\left(X(1), X(2), \dots, X(T_{obs})\right)$ to future feature matrices $\left(X(T_{obs}+1), X(T_{obs}+2), \dots, X\left(T_{pred}\right)\right)$.

**CMGraph Formulation to Model Spatiotemporal Data.** We represent crowd flow data recorded over a discrete time span $t = 1, \dots, T_{obs}$ as a set of undirected and unweighted dynamic graphs $\{G(t) = (V(t), E)\}$. The set of vertices or nodes, $V(t) = \{v_1, \dots, v_N\}$, corresponds to the $N$ egress regions. The node feature matrix, $X_t^{N\times D}$, stores the crowd flow data at time $t$, where $D$ is the number of node features. $E$ denotes the set of time-invariant edges, where an edge $e_{ij} \in \{0,1\}$ connecting node $v_i$ and node $v_j$ is 1 if two egress regions are adjacent to each other and 0 otherwise. Two egress regions are adjacent if pedestrians can walk from one region to another without entering a third egress region. The graph topology can also be represented as an adjacency matrix $A \in \mathbb{R}^{N\times N}$, where entry $A_{ij} = 1$ if there exists an edge between node $v_i$ and $v_j$ (i.e., $e_{ij} = 1$). Figure 1 shows an example of at the Grand Central Station in New York, where manually identified egress regions (using the train station's floor plan (Suarez 2015)) are used to construct the topology of a CMGraph.

## SPATIOTEMPORAL FORECASTING OF CROWD FLOW

Exploiting both spatial connectivity (with the adjacency matrix, $A$) and recorded temporal crowd flow signal (with the node feature matrix sequence $X(1), X(2), \dots, X(T_{obs})$, we formulate the crowd flow forecasting problem as:

$$\left(\hat{X}(T_{obs}+1), \hat{X}(T_{obs}+2), \dots, \hat{X}\left(T_{pred}\right)\right) = f\left(\left(X(1), X(2), \dots, X(T_{obs})\right); A\right) \tag{1}$$

where $f$ is the function to be learned. $\left(X(1), X(2), \dots, X(T_{obs})\right)$ is the input sequence (Figure 2(a)), and $\left(\hat{X}(T_{obs}+1), \hat{X}(T_{obs}+2), \dots, \hat{X}\left(T_{pred}\right)\right)$ is the generated output sequence, $\hat{X}(\cdot)$ denoting a predicted value (Figure 2(b)).

To learn the function $f$, we present an approach that uses graph convolutional network (GCN) (Kipf and Welling 2017) with dense connection (Huang et al. 2019) and gated recurrent unit (GRU) (Cho et al. 2014). Dense-GCN serves as a spatial encoder that learns an embedding vector encoding spatial features resulting from the topology of the CMGraphs. The embedded graphs are used as the input to the GRU cells, which learn to encode temporal representations from the time series data. We sequentially stack these two encoders to obtain an embedding vector representing both spatial and temporal features. Lastly, a fully connected (FC) layer processes the resulting embedding to generate the output sequence of the crowd flow. The architecture of the model is shown in Figure 2. The following describe the GCN and GRU operations in more details.

**Dense-GCN as Spatial Encoder.** Figure 2(c-1) shows the process of the spatial encoding with Dense-GCN. For a set of $N$ nodes in a CMGraph $G(t) = (V(t), E)$, 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 $x_i$ (the $i^{th}$ row of the feature matrix $X(t)$), and its set of neighboring nodes $J$, a GCN layer updates the target node embedding as follows:

$$x_i^{(k)} = \frac{W^{(k)}}{|J|}\sum_{j\in J} x_j^{(k-1)} \tag{2}$$

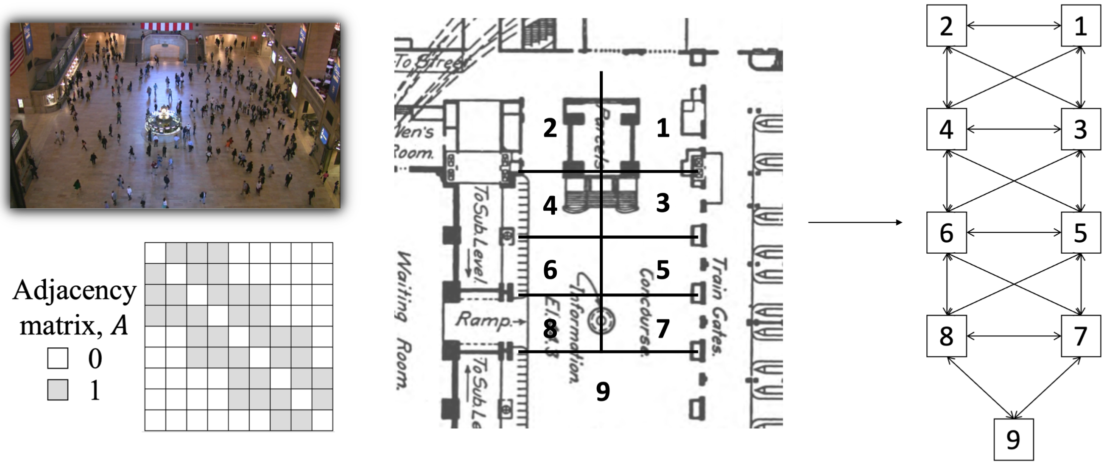

**Figure 1. Egress region division and CMGraph formulation of the Grand Central Station. Grand Central Station floorplan adopted from (Suarez 2015). An entry of the adjacency matrix, $A_{ij}$, is 1 if two egress regions are adjacent.**

where $W^{(k)}$ and $x_i^{(k)}$ are a learnable parameter and the $i^{th}$ node's updated embedding of the $k^{th}$ layer, respectively. In the first layer, $x_i^{(0)}$ is the initial feature vector of node $v_i \; \forall \; v_i \in V(t)$. The dimension of $X'(0)$ is $N \times D$, where $N$ and $D$ are the number of nodes and the number of node features, respectively.

Stacking $K$ GCN layers allow us to update node embeddings using information aggregated from nodes in the $K$-hop neighborhood. After $K$ GCN layers, we have learned the embedded graph, $G'(t)$, whose node embedding matrix is $X'(t)$, each row being the updated embedding vectors $x_i' \; \forall i \in V(t)$. Each node embedding vector is of an embedding dimension $H_{GCN}$, a tunable hyperparameter. The dimension of $X'(t)$ is therefore $N \times H_{GCN}$.

The concept of dense connections, first introduced in the convolutional neural network (CNN) model DenseNet (Huang et al. 2019), involves concatenates an output from earlier layers with an output from later layers. In light of the issue of over-smoothing in deep GCN as observed by Li et al. (2018), skip connections have been shown to be effective in reducing this effect in deep GCNs (Li et al. 2019). Thus, in this study, we have incorporated the concept of dense connections from CNNs into GCNs by concatenating the GCN-learned spatial embedding, $X'(t)$, with the original input, $X(t)$. The resulting output of this architecture, referred to as Dense-GCN, is denoted as $X''(t) \in \mathbb{R}^{N\times(H_{GCN}+D)}$.

**GRU as Temporal Encoder.** GRU can be used to encode hidden state representations of time series inputs, as shown in Figure 2(c-2). Mathematically, each GRU operation in a layer $l$ can be expressed as follows:

$$r_t^{(l)} = \sigma\left(W_{ar}^{(l)} a_t^{(l)} + b_{ar}^{(l)} + W_{hr}^{(l)} h_{t-1}^{(l)} + b_{hr}^{(l)}\right) \tag{3}$$

$$z_t^{(l)} = \sigma\left(W_{az}^{(l)} a_t^{(l)} + b_{az}^{(l)} + W_{hz}^{(l)} h_{t-1}^{(l)} + b_{hz}^{(l)}\right) \tag{4}$$

$$n_t^{(l)} = \tanh\left(W_{an}^{(l)} a_t^{(l)} + b_{an}^{(l)} + r_t^{(l)} * \left(W_{hn}^{(l)} h_{t-1}^{(l)} + b_{hn}^{(l)}\right)\right) \tag{5}$$

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

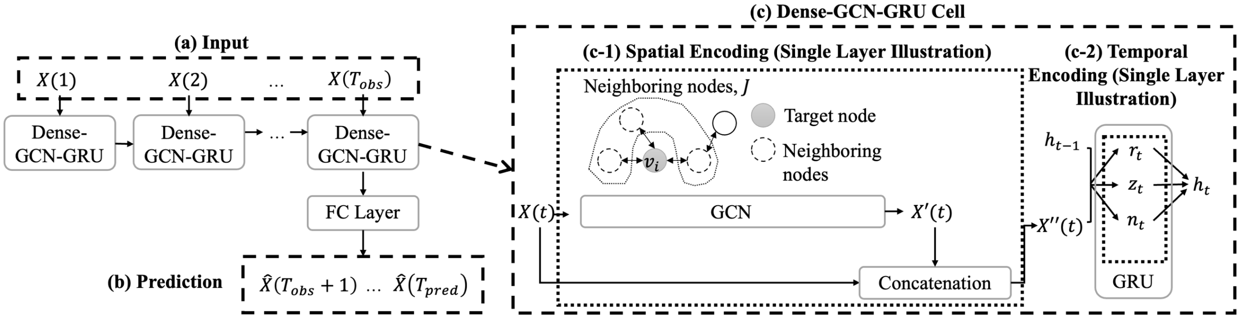

**Figure 2. Architecture of the proposed GCN-GRU model. For simplicity, only a single layer of Dense-GCN ($K = 1$) and GRU ($L = 1$) are drawn.**

where $W_{ar}^{(l)}, W_{hr}^{(l)}, W_{az}^{(l)}, W_{hz}^{(l)}, W_{an}^{(l)}, W_{hn}^{(l)}, b_{ar}^{(l)}, b_{hr}^{(l)}, b_{az}^{(l)}, b_{hz}^{(l)}, b_{an}^{(l)}, b_{hn}^{(l)}$ are learnable parameters of the $l^{th}$ layer. $a_t^{(l)}$ is the input to the layer and is equal to the output from the Dense-GCN part, $X''(t)$, 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. $\sigma$ 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 $t = T_{obs}$ after $L$ GRU layers is then $h_{T_{obs}}^{(L)}$, a vector with length $H_{GRU}$, a tunable hyperparameter.

## CASE STUDY

To illustrate the use of CMGraph to represent spatiotemporal information and Dense-GCN-GRU for crowd flow forecasting, this section describes the experimental results of a case study with the New York Grand Central Station (GCS) dataset, collected by Zhou et al. (2011). The dataset consists of video frames collected with a camera mounted in the train station. Figure 1 shows one of the video frames. Point-wise individual trajectories were manually annotated by Yi et al. (2015). The dataset consists of 17,682 trajectories, with 6,000 video frames at a resolution of 1920×1088, annotated at 1.25 frames per second (FPS). Detailed description of the dataset can be found in the cited references.

**Evaluation Metrics**. 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 predicted node feature matrices $\hat{X}(T_{obs} + 1), \dots, \hat{X}(T_{pred})$ and the true node feature matrices $X(T_{obs} + 1), \dots, X(T_{pred})$ . Denoting the $i^{\text{th}}$ element of a matrix $X(t)$ as $x_{it}$ and $\hat{X}(t)$ as $\hat{x}_{it}$, the MSE is computed as

$$\mathrm{MSE} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T_{pred}} (x_{it} - \hat{x}_{it})^2 \tag{7}$$

The mean absolute error (MAE) is also reported, 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:

$$\mathrm{MAE} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T_{pred}} |x_{it} - \hat{x}_{it}| \tag{8}$$

**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 count, and (2) timestamp for each egress region. The embedding dimension of the GCN encoder is $H_{GCN} = 128$. The embedding dimension of the GRU part is $H_{GRU} = 64$. We use $T_{obs} = 20$ and $T_{pred} = 20$ in the study and split the CMGraph sequences from the GCS dataset into train and test set following a 70/30 ratio. 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).

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. To ensure reproducibility, the code repository is publicly available online at https://github.com/vivian-wong/CMGraph-Crowd-Forecasting.

**Baseline Methods**. We compare the Dense-GCN-GRU model with two baseline models: GRU and GCN-GRU. The GRU model treats inputs as purely temporal signals and do not involve the graph's adjacency matrix in its computation, thereby leaving out the spatial connectivity information given by the floor plan of the surrounding space. On the other hand, the GCN-GRU model omits the dense connection, and therefore directly uses the un-concatenated spatial embedding $X'(t)$ as the input to GRU (rather than the $X''(t)$ in the Dense-GCN-GRU model). All models are trained with the same hardware setup and hyperparameters as Dense-GCN-GRU.

**Results and Analysis**. The experimental results obtained from the different models are presented in Table 1. As shown in the table, the Dense-GCN-GRU model outperforms the GRU model and suggests that considering spatial connectivity enhances the accuracy of crowd forecasting. Notably, we observe that the GCN-GRU model exhibits lower accuracy than both the GRU only model, which disregards spatial information, and the Dense-GCN-GRU. One potential explanation for this observation is that the GCN model oversmoothed 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.

To illustrate the observed results, a sequence of predicted crowd flow is plotted in Figure 3. A notable disparity in forecasting results is seen in the densely populated regions, particularly regions 6, 7 and 9, where the Dense-GCN-GRU better captures the trend of the crowd volume.

**Discussion.** The case study serves as demonstrative analysis for the plausibility of deploying the suggested CMGraph data structure and Dense-GCN-GRU model for predictions of temporal crowd flow, informed by spatial information. In this section, an example of Dense-GCN-GRU predicting temporal crowd flow patterns and qualitative observations is discussed.

Figure 3 demonstrates that the model predicts crowd pattern, as evidenced by the decrease in crowd flow in region 9 and the corresponding increase in regions 7 and 8. In Figure 4, we provide visual evidence from the GCS dataset of an influx of crowd from region 9 dispersing towards regions 7 and 8, in line with the model predictions. In practical applications, predicting where an influx of crowds move to will give facility operators additional time to plan for the dispatchment of additional service support, such as direction guides and signage, and potentially help reduce future congestion. Additionally, the forecasting models developed in this paper could be employed to test and evaluate different space management strategies, such as the placement of

Table 1. MSE and MAE of crowd volume prediction models.

| Predictor | MSE | MAE |
|---|---|---|
| GCN-GRU | 0.243 | 0.337 |
| GRU | 0.113 | 0.240 |
| Dense-GCN-GRU | **0.096** | **0.219** |

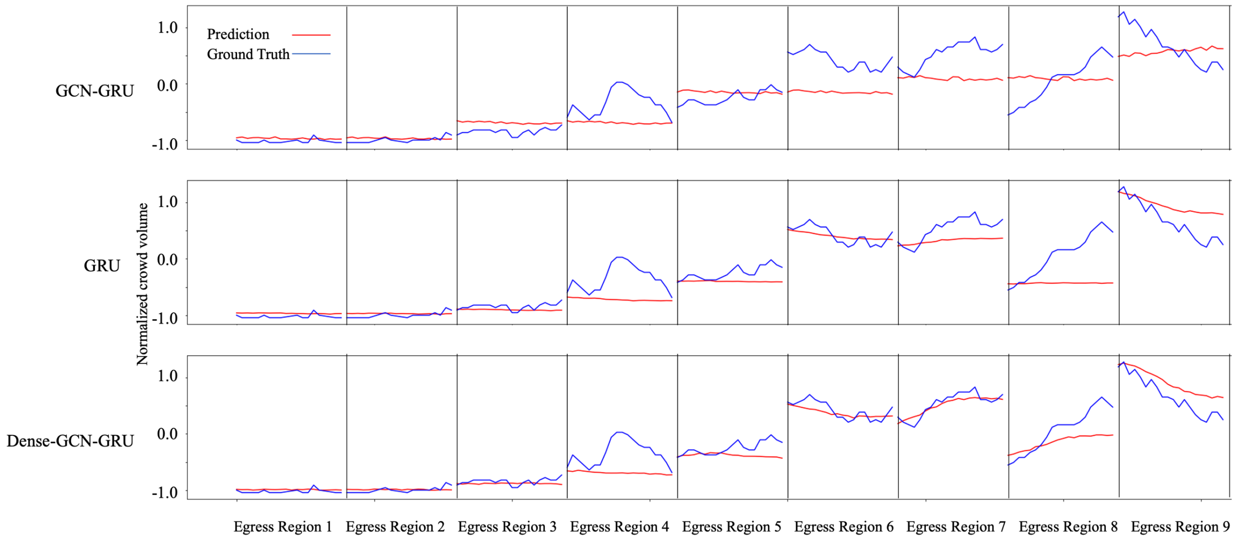

**Figure 3. A sample sequence of the ground truth and predicted crowd flow at each of the 9 egress regions at the GCS. The Y-axis is the crowd volume, normalized to [-1, 1].**

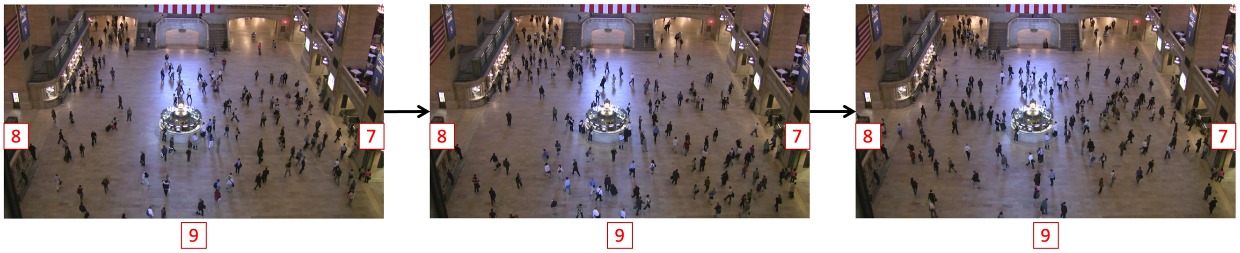

**Figure 4. Frames from the GCS model during the sequence shown in Figure 3, where crowd flows from Egress Region 9 to Egress Region 7 and 8.**

barriers, that could be potentially simulated by modifying the connectivity of the CMGraphs.

## CONCLUSION

In this study, a method to model temporal human crowd flow information with spatial connectivity information (derived from egress locations shown on floor plans) is proposed. A case study on real-world data of the Grand Central Station is performed, where the data of human crowds is modeled into a CMGraph, such that forecasting can be performed with a proposed Dense-GCN-GRU neural network model. The case study results demonstrate that the CMGraph formulation can capture valuable spatiotemporal information, which can be exploited by the forecasting model to predict trends of future crowd flow. Future endeavors will develop an end-to-end framework to automatically extract crowd information from raw videos in crowded scenes, for example, using object detector networks.

## ACKNOWLEDGEMENTS

The research is supported by the Stanford Center at the Incheon Global Campus (SCIGC), which is sponsored in part under the National Program to Subsidize Attracting Foreign Educational Institution and Research Institutes published by the Ministry of Trade, Industry, and Energy of the Republic of Korea and managed by the Incheon Free Economic Zone Authority.

## REFERENCES

Alahi, A., K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. 2016. "Social LSTM: Human Trajectory Prediction in Crowded Spaces." *Proc., 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 961-971.

Bai, J., J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, and H. Li. 2021. "A3T-GCN: Attention Temporal Graph Convolutional Network for Traffic Forecasting." *ISPRS Int. J. Geo-Inf.* 10, 485.

Cho, K., B. van Merriënboer, D. Bahdanau, and Y. Bengio. 2014. "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches." *Proc., SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Association for Computational Linguistics, Doha, Qatar, 103–111.

Huang, G., Z. Liu, G. Pleiss, L. Van Der Maaten, and K. Weinberger. 2019. "Convolutional Networks with Dense Connectivity." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 44 (12): 8704-8716.

Kipf, T., and M. Welling. 2017. "Semi-Supervised Classification with Graph Convolutional Networks." *5th International Conference on Learning Representations*, Toulon, France.

Li, G., M. Müller, G. Qian, I. C. D. Perez, A. Abualshour, A. K. Thabet, and B. Ghanem. 2019. "DeepGCNs: Making GCNs Go as Deep as CNNs." *Proc., 2019 IEEE/CVF International Conference on Computer Vision*, IEEE Computer Society, 9266-9275.

Li, Q., Z. Han, and X.-M. Wu. 2018. "Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning." *Proc., 32nd AAAI Conference on Artificial Intelligence*, Association for the Advancement of Artificial Intelligence, 3538-3545.

Mohamed, A., K. Qian, M. Elhoseiny, M. Elhoseiny, and C. G. Claudel. 2020. "Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction." *Proc., 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 14412-14420.

Panagopoulos, G., G. Nikolentzos, and M. Vazirgiannis. 2021. "Transfer Graph Neural Networks for Pandemic Forecasting." *Proc., 35th AAAI Conference on Artificial Intelligence*, Association for the Advancement of Artificial Intelligence, 35 (6): 4838–4845.

Suarez, S. 2015. "Grand Central Terminal's Original Lighting: Its Significance, Its Relationship With the Current Scheme, and Recommendations for Alternate Considerations." M.S. Thesis, Columbia University, New York, NY, USA.

Wang, J., Y. N. Ding, and D. D. Liu. 2015. "The research on early warning of preventing the stampede on crowded places and evacuated technology." *Proc.,2015 International Forum on Energy, Environment Science and Materials*, Atlantis Press, 1544–1551.

Yi, S., H. Li, and X. Wang. 2015. "Understanding pedestrian behaviors from stationary crowd groups." *Proc., IEEE Conference on Computer Vision and Pattern Recognition*, 488-3496.

Zhao, L., Y. Song, C. Zhang, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, M. Deng, M. Deng, and H. Li. 2019. "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction." *IEEE Transactions on Intelligent Transportation Systems*. 21(9):3848-58.

Zhou, B., X. Wang, and X. Tang. 2011. "Random Field Topic Model for Semantic Region Analysis in Crowded Scenes from Tracklets." *Proc., 2011 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 3441–3448.