

# A Cost-Effective Data-driven Approach to Flashover Prediction across Diverse Residential Layouts for Enhanced Firefighters Situational Awareness

Linhao Fan<sup>a</sup>, Hongqiang Fang<sup>b</sup>, Tianshui Liang<sup>c</sup>, Wai Cheong Tam<sup>b\*</sup>, Qixing Zhang<sup>a\*</sup>

<sup>a</sup>State Key Laboratory of Fire Science, University of Science and Technology of China, Hefei, China, [fan7tree@mail.ustc.edu.cn](mailto:fan7tree@mail.ustc.edu.cn), [qixing@ustc.edu.cn](mailto:qixing@ustc.edu.cn)\*

<sup>b</sup>Fire Research Division, National Institute of Standards and Technology, Maryland, USA, [hongqiang.fang@nist.gov](mailto:hongqiang.fang@nist.gov), [waicheong.tam@nist.gov](mailto:waicheong.tam@nist.gov)\*

<sup>c</sup>School of Mechanics and Safety Engineering, Zhengzhou University, Zhengzhou, China, [liangtsh@zzu.edu.cn](mailto:liangtsh@zzu.edu.cn)

\*Corresponding author

## Highlights:

- Developed a generic flashover prediction model adaptable to various layouts.
- Integrated graph attention mechanism to capture sensor connectivity effectively.
- Reduced model development costs for new layouts using transfer learning.

## Abstract:

This paper presents gFlashNet, a generic flashover prediction model, designed to address the limitations of existing models that are restricted to specific residential building layouts. The aim of this research is to improve the scalability and adaptability of flashover prediction models, which is crucial for enhancing fire safety in buildings. By representing the spatial positions of sensors as a graph network, gFlashNet can be applied to various building layouts without requiring any model structure modifications. A graph attention mechanism is integrated to strengthen the model's ability to capture sensor connectivity during fire events. Additionally, transfer learning is employed to reduce development costs by enabling the pre-trained model to be fine-tuned on new layouts using a smaller dataset. The result shows that gFlashNet achieves high prediction accuracy for new layouts with significantly less data, reducing data requirements compared to traditional approaches. This work contributes a novel, cost-effective approach for developing generalizable fire safety models, with significant potential for real-time flashover prediction across diverse residential layouts.

**Keywords:** Flashover Prediction; Spatio-Temporal Graph Neural Network; Model Scalability; Transfer Learning; Smart Firefighting

## 1. Introduction

Situational awareness is crucial during firefighting and fire rescue missions in the context of building fire safety [1]. In residential buildings, fire scenarios are highly variable and complex due

to differences in layout and construction materials. Consequently, extracting information from sensor-collected time series data to enhance situational awareness during firefighting has become a significant research focus in Smart Firefighting [2]. Leveraging the powerful fitting capabilities of deep neural networks, researchers have employed various deep learning models to analyze sensor-collected fire data, aiming to improve various levels of fire situational awareness. These studies include the inversion of important fire scene parameters (e.g., fire source locations and heat release rates) [3–5], the prediction of hazardous fire events (e.g., flashover and backdraft) [6–9], and the determination of the fire development stage [10,11]. The excellent performance of these models has demonstrated significant potential for enhanced situational awareness and has motivated additional research in this field. The deep learning models typically operate with input exclusively from time series data collected by fire detection sensors. Moreover, these models are well-trained during their development phase, ensuring they meet real-time demands in practical applications.

However, a major limitation exists regarding the generalizability of the current works. The deep learning models are typically use time-series data collected from sensors within buildings with a certain arrangement. To ensure consistency in the input format, these studies typically concentrate on a specific building layout, where the sensor nodes located in each room are used as the input data for the model. An alternative approach involves wearable sensors for firefighters, which allows for more flexible data collection. However, Lu et al. [12] using this approach to predict flashover has so far been conducted only within a single room, due to the limited sensing range of individual wearable sensors. Therefore, for multi-room residential buildings, the use of fixed sensors remains the primary solution. But for residential buildings with different layouts, the placement and number of sensors can vary significantly. It means that those models developed for a specific floor plan cannot be directly applied to another. In other words, the model is non-scalable. For example, Fan et al. [13] developed a flashover prediction model using one-dimensional convolution layers, which has a convolution kernel size that matches the number of sensors in a specific residential building. Consequently, the model was only applicable to that particular building layout and its specific sensor node configuration. When the model is applied to a different floor plan, the model's structure must be modified to accommodate the new sensor configuration. This necessitates training a new model from scratch, leading to significant development costs. Tam et al. [14] also noted the limitations of these model's capabilities in terms of generalizability across different floor plans. They attempted to address it by setting virtual sensors to expand the input size of the model to accommodate various floor plan scenarios with a unified input size. However, this approach still involves significant development costs and adds extra computational overhead for simple scenarios.

In summary, these models cannot be directly applied to buildings with different layouts without substantial modifications to the model's structure, requiring a complete retraining process. To address this limitation in model generalizability, this study proposes a novel flashover prediction model, gFlashNet, which leverages graph neural networks and transfer learning [15]. This model is designed to be applicable to a variety of residential building layouts. Specifically, the model is constructed based on the architecture of spatio-temporal graph neural networks. This network enables the model to be directly applicable to different building layouts without any modifications to the model structure during the development phase. Additionally, a graph attention mechanism, which can automatically learn the weights between the sensor nodes, is employed. This helps enhance the model's adaptability to different building layouts. In order to demonstrate the

implementation of this transfer learning strategy, a case study is performed. Firstly, a pre-training model is developed based on a specific house layout. Subsequently, the pre-trained model is fine-tuned with data from another house to obtain corresponding applicable models. A comparison shows that the transfer learning approach effectively reduces the amount of data and training costs required for the flashover prediction model development. It is believed that the outcome of this work can serve as a reference for developing more generic models that can effectively adapt to a variety of different floor plan scenarios. Therefore, this work contributes an innovative and cost-effective method for developing versatile fire safety models, offering substantial potential for real-time flashover prediction across various residential layouts.

The remaining sections are organized as follows. Section 2 introduces the specific problem of flashover prediction and the data set construction strategy. Section 3 describes the model structure and model development strategy. Section 4 details the development process and result analysis of the pre-trained and fine-tuned models.

## 2. Training Dataset Construction

### 2.1 Flashover prediction problem

Flashover is an extremely dangerous event during the fire development process [16], posing a serious threat to the safety of firefighters [17]. This study focuses on the prediction of flashover in residential fires using data collected from fire detection sensors. The time-series temperature data collected by sensors were used as input for the model to ensure its applicability in actual fire scenarios. Considering the workload and physiological stress firefighters face during operations, the model must provide actionable information, minimizing the additional burden of information processing. Therefore, this study considered the movement speed of firefighters in a smoke-filled environment and the highly dynamic nature of fire development, setting the model's output to indicate whether a flashover will occur within the next 30 seconds.

The flashover prediction problem was defined as follows. Given a temperature series  $T \in \mathbb{R}^{(D,N)}$  collected from multiple fire detection sensors within the structure as model input  $X$ , where  $D$  represents the number of sensors and  $N$  represents the length of the time series, the flashover prediction model  $M$  maps the input  $X$  to a binary classification result  $Y \in \mathbb{R}^{(1,2)}$ , indicating the probability of a flashover occurring within the next 30 seconds. The following section describes the dataset construction strategy.

### 2.2 Numerical Setup for Data Generation

A wide range of fire parameters was considered. The fire parameters were: 1) fire location, 2) fire growth rate, 3) nearby-item ignition, and 4) vent conditions. By using a specific probability distribution function and a reasonable range of values for each parameter, various fire scenarios were obtained. This approach ensures the diversity of the fire scenarios. Table 1 summarizes the parameters considered in this study and their reasonable ranges of values. And, window configurations and material details can be found in the Appendix A.

**Table 1.** Summary of parameters considered in the data generation process.

	Fire Conditions	Vent Opening Conditions
--	-----------------	-------------------------

Parameters	Fire location	HRR	Nearby-item ignition criterion	Probability of door opening	Triggering temperature for window opening
Range of Values	Any room except the bathroom	See Table 2	20 kW/m <sup>2</sup> - 50 kW/m <sup>2</sup>	50%	100 °C – 200 °C

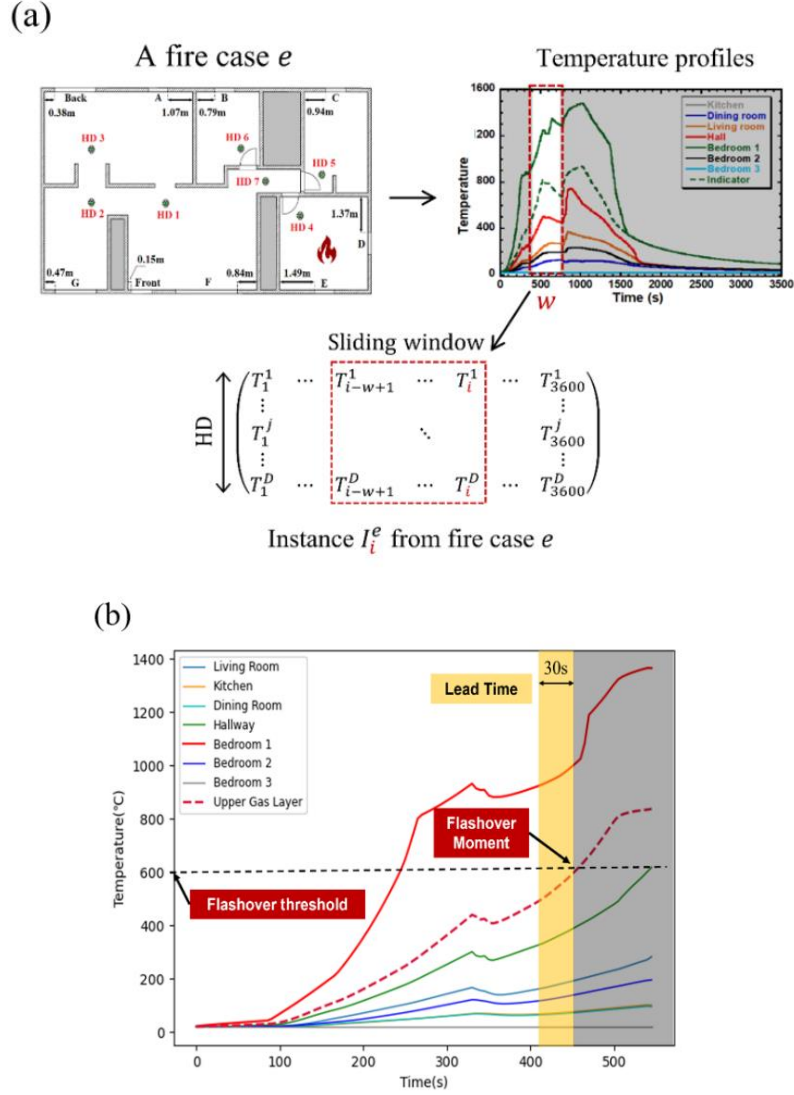
**Table 2.** Heat release rates (HRR) parameters for burning items.

Parameters	$Q_L$ (kW)	$Q_{max}$ (kW)	$t_L$ (s)	$t_{P1}$ (s)	$t_{P2}-t_{P1}$ (s)	$t_E-t_{P2}$ (s)
Range of Values	10 – 30	270 – 3500	50 – 150	50 – 300	0 – 500	100 – 1070

In terms of the fire development within a compartment, the fire was designed to have four-stages containing a smoldering stage, t-square fire growth stage, peak, and decay stage. The HRR curve was designed accordingly. The transition HRR from smoldering to flaming fire ( $Q_L$ ), peak HRR ( $Q_{max}$ ), time to transition ( $t_L$ ) for the heat release rate (HRR) from smoldering to flaming combustion, time to peak HRR ( $t_{P1}$ ), peak time ( $t_{P2}-t_{P1}$ ), and decay time ( $t_E-t_{P2}$ ) are summarized in Table 2. The parameters that describe these four stages and the range of their values were determined based on the reported work from [18]. In this case study, fires with a growth rate ranging from slow to fast were considered. Nearby-item ignition was determined based on the incident heat flux onto the fictitious items (i.e., targets) which were located approximate 0.9 m away from the primary ignition item. The ignition heat flux criteria were determined based on [19] and it was ranging from 20 kW/m<sup>2</sup> to 50 kW/m<sup>2</sup>. There are two types of vent openings: a) doors and b) windows. The doors were assumed to be opened at the beginning of the numerical experiment with a probability of 50 %. The windows were assumed to be fully opened due to a breakage when the top of the window reached a threshold determined by a uniform random generator from 100 °C to 200 °C [20]. In this study, the zonal model, namely CFAST [21], is used as the simulation engine.

### 2.3 Data Processing Strategy

After completing the numerical simulations, a data processing strategy was implemented to convert raw data into a dataset suitable for the model development. The strategy for extracting instances from the raw temperature data for model's input was illustrated in Fig. 1a. For a specific fire scenario,  $e$ , instances were extracted from the raw temperature sequences,  $S$ , using a sliding window,  $w$ . A specific instance can be represented as  $I_i^e = [T^{(1)}, \dots, T^{(D)}]$ , where  $T^{(j)} = [T_{i-w}^{(j)}, \dots, T_{i-1}^{(j)}, T_i^{(j)}]$ . Here,  $w$  is the width of the time window, and the last time point  $i$  is used to label the instance. The width of the sliding window is set to 300 seconds. Given that the temperature data is collected at 5-second intervals during the simulation, the size of each instance  $I_i \in \mathbb{R}^{(D,60)}$ .



**Fig. 1.** Illustrations of (a) instance extraction process and (b) instance labeling strategy.

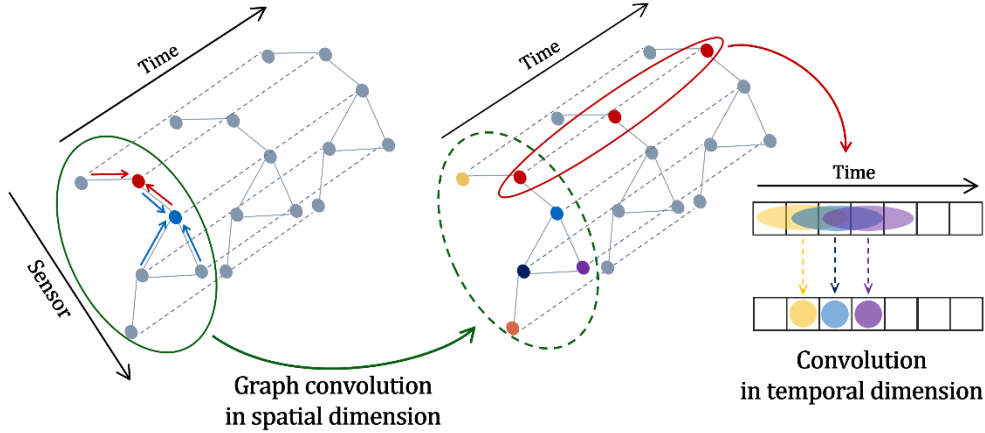
As discussed in Section 2.1, this study considered the flashover prediction problem as a supervised learning problem of multivariate time series classification. Therefore, to conduct supervised model training, it is necessary to select and label the required instances. In this study, the upper layer gas temperature was used as the criterion for determining the onset of flashover. Following the recommendation of Hurley et al. [22], the critical condition for flashover is set to an upper smoke layer temperature of 600 °C. This study adopted the same instance labeling strategy as used in [6,13,14]. First, the moment of flashover (Flashover Moment,  $FO$ ) is determined based on the upper smoke layer temperature reaching 600 °C. If the last time point  $i$  of instance  $I_i$  falls within the yellow area shown in the Fig 1b, i.e.,  $i \in [FO - 30, FO)$ , it indicates that the flashover condition will be met within the next 30 seconds, thus constituting a positive sample. To maintain a balance between positive and negative samples during training, only  $i \in [FO - 60, FO - 30)$  is selected as negative samples.

To effectively train and test the model, the samples need to be further divided into training, validation, and testing sets. First, the fire scenarios are divided into training, validation, and testing scenarios in a 7:2:1 ratio. This prevents data leakage where instances from the same fire scenario appear in both training and test sets. Then, the raw data from the fire scenarios is processed according to the instance extraction and labeling strategy, ultimately resulting in corresponding training, validation, and testing sets. The data is linearly scaled to the range [0, 1] using Min-Max normalization to ensure better convergence during training.

### 3. Development of gFlashNet

#### 3.1 Framework of the ST-GNN

In this study, the framework of Spatio-Temporal Graph Neural Network (ST-GNN) [23] was employed to build the flashover prediction model. The basic framework of ST-GNN was illustrated in the Fig. 2. This framework decouples the extraction of spatial and temporal features of the sensor sequences into independent stages.



**Fig. 2.** The Framework of Spatio-Temporal Graph Neural Network

The graph structure is developed based on the physical locations of the sensor placement, and this structure defines how information is aggregated in the graph neural network. In this graph-based data structure, the model can accommodate any number of sensor nodes by adding or removing nodes, without requiring a strict sequential order among sensors. This flexibility is particularly advantageous when applied to arbitrary building layouts. Following the "Message-Passing" framework [24], a graph neural network can be defined as:

$$\mathbf{m}_u^{(l)} = \text{MSG}^{(l)}(\mathbf{h}_u^{(l-1)}), u \in \{N(v) \cup v\} \quad (1)$$

$$\mathbf{h}_v^{(l)} = \text{AGG}^{(l)}(\{\mathbf{m}_u^{(l)}, u \in N(v)\}, \mathbf{m}_v^{(l)}) \quad (2)$$

where  $N(v)$  is the set of neighboring nodes of node  $v$ . The message construction process for node  $u$  is denoted as  $\mathbf{m}_u$ . Meanwhile,  $\mathbf{h}_v$  represents that node  $v$  aggregates the messages from its neighboring nodes, including itself, to update its state. According to this framework, the node update formula for node  $v$  defined by the graph convolution layer [25] can be expressed as:

$$H_v^l = \sigma \left( \sum_{u \in N(v)} \frac{A_{uv}}{\sqrt{\tilde{D}_{uu}\tilde{D}_{vv}}} (H_u^{l-1}W^l) + \frac{1}{\tilde{D}_v} (H_v^{l-1}W^l) \right) \quad (3)$$

where  $A$  is the adjacency matrix of the given undirected graph  $\mathcal{G}$ . And,  $\tilde{D}$  is a diagonal matrix with  $\tilde{D}_{ii} = \sum_j A_{ij} + 1$ . From Equation 3,  $W^l$  is a learnable weight parameter that defines the message passing (MSG) in Equation 3. The graph convolution defines the aggregation (AGG) in the equation as the weighted average of the representations of the neighboring nodes and the node itself. The weight of neighboring node  $u$  is determined by the edge weight between nodes  $u$  and  $v$ , i.e.,  $A_{uv}$  normalized by the degrees of the two nodes. The weight of the neighboring nodes is determined by the normalization degree of the nodes. The representation of the nodes is normalized by their own degrees. This aggregation method does not include learnable parameters and is entirely determined by the adjacency matrix of the graph structure.

Since the graph convolution aggregates the messages from neighboring nodes, the convolution in the temporal dimension only needs to slide over individual nodes to extract temporal features. As shown in Fig. 2, the size of the convolution kernel  $k$  in the temporal dimension is  $k \in \mathbb{R}^{(1,\ell)}$ .

Because nodes defined based on the graph structure do not have a specific order, the model allows exchangeability of sensors. Additionally, the shared weight parameters for node representations ensure the scalability of the graph neural network to new nodes. The convolution in the temporal dimension observes all nodes sequentially, without being limited by the number of nodes. Therefore, the framework of the spatio-temporal graph neural network is scalable to be used to develop the flashover prediction model with different number of sensors.

### 3.2 Structure of gFlashNet

The structure of generic Flashover Neural network model (gFlashNet) is depicted in Fig 3. The feature extractor in the model has two Spatial-Temporal Blocks (ST-Blocks) [27]. The ST-Block consists of two Temporal-Blocks and one Spatial-Block to carry out the extraction and aggregation of spatio-temporal features from the sequential data. Finally, a fully connected layer is used as the classifier to obtain the final prediction result of the model.

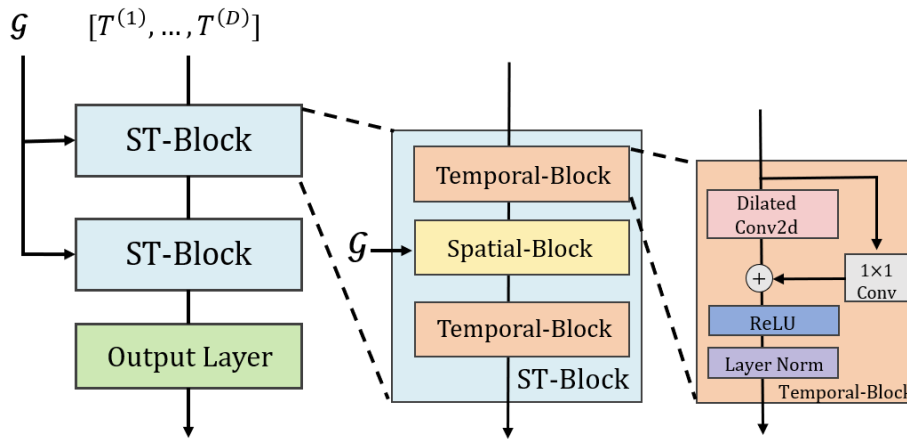


Fig. 3. The model structure of gFlashNet.

For the Temporal-Block, the dilated convolution layer [26] with a kernel size of  $k \in \mathbb{R}^{1 \times l}$  is employed in this study, allowing it to maintain a large receptive field even with fewer convolutional layers. The calculation method for the dilated convolution kernel is:

$$h = \sum_{x_i, w_i, b_i \in (X_{i-d\lfloor l/2 \rfloor, i+d\lfloor l/2 \rfloor}, W, B)} (w_i * x_i + b_i) \quad (4)$$

where  $d$  is the dilation factor, indicating the spacing within the convolution kernel. And,  $W, B$  are the matrices weights of the kernel. Additionally, a  $1 \times 1$  convolution layer is used to incorporate residual connections. ReLU serves as the activation function, and LayerNorm ensures normalization, facilitating efficient convergence of the model.

The Spatial-Block of the model corresponds to a multi-head graph attention layer [28]. In the graph convolution layer defined by Equation 3, the importance of neighboring node  $u$  to target node  $v$  is determined by the weight of their edge  $A_{ij}$  (normalized by node degrees). However, in practice, the input graph structure  $\mathcal{G}$  to the model might be inconsistent with the complex, real-world scenario. It means that the adjacency matrix  $A$  may not accurately reflect the true connectivity strength between two nodes. During a fire, changes in ventilation conditions can alter the connectivity between rooms, meaning that a graph structure from the room layout may not fully represent the spatial relationships between sensors. Therefore, this study uses a graph attention mechanism to automatically adjust the importance of neighboring nodes, which can be expressed as:

$$H_i^k = \sigma \left( \sum_{j \in N(i)} \alpha_{ij} W^{k-1} H_j^{k-1} \right) \quad (5)$$

where the attention score  $\alpha_{ij}$  represents the importance of neighboring node  $j$  to node  $i$ . To calculate the attention score  $\alpha_{ij}$ , the relationship strength  $e_{ij}$  between node  $i$  and node  $j$  is first computed as:

$$e_{ij} = a(WH_i^{k-1}, WH_j^{k-1}) = \sigma(W_2[WH_i^{k-1} \parallel WH_j^{k-1}]) \quad (6)$$

where  $\parallel$  denotes the concatenation operation, and  $W$  is a learnable parameter. To ensure the coefficients of different nodes are comparable, the relationship strength is further normalized using the sum of the relationships over  $j$  and the corresponding attention coefficients,  $\alpha_{ij}$ , obtained through the following expression:

$$\alpha_{ij} = \text{Softmax}_j(\{e_{ij}\}) = \frac{\exp(e_{ij})}{\sum_{j \in N(i)} \exp(e_{ij})} \quad (7)$$

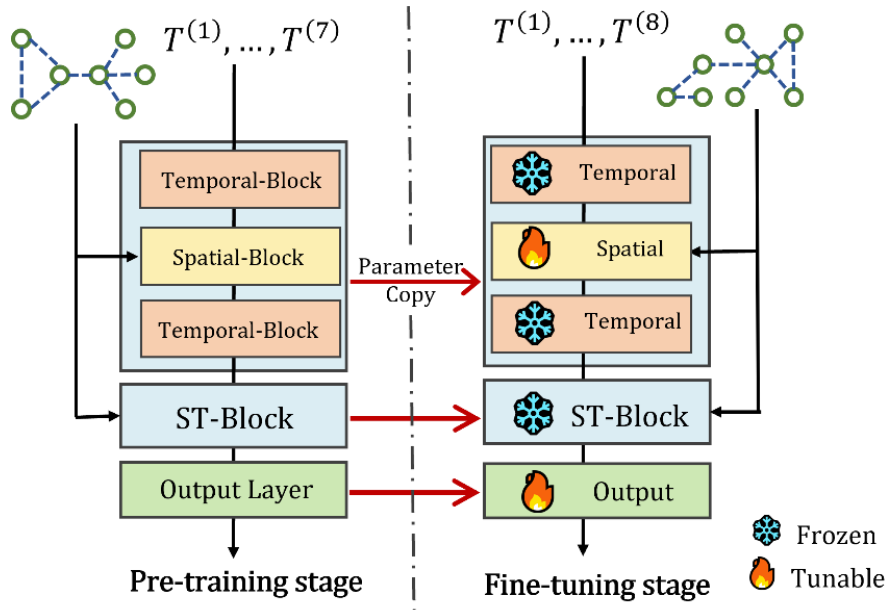
This study employed multi-head attention mechanisms to fully capture the spatial features, where each attention head determines different similarity functions on the nodes. For each attention head, a new node representation can be independently obtained according to equation. The final node representation is the average of the node representations learned by different attention heads:

$$H_i^k = \sigma \left( \frac{1}{T} \sum_{t=1}^T \sum_{j \in N(i)} \alpha_{ij}^t W^t H_j^{k-1} \right) \quad (8)$$

where  $t$  represents the  $t$ -th attention head. The multi-head attention allows the model to learn the spatial relationships and feature representations of the nodes more comprehensively, ensuring the model's scalability and effectiveness in predicting flashover events.

### 3.3 Model Training Strategy

Transfer learning was used and the training strategy is illustrated in Fig 4. Initially, the model was pre-trained on a complete dataset from a specific house layout. During the fine-tuning stage, the parameters of the temporal blocks from the pre-trained model were then frozen. And only the spatial blocks and output layer were modified using data from another house layout. It should be noted that no adjustments to the gFlashNet's model structure were made; only the graph structure corresponding to the house layout needs to be provided. Since the amount of data required for model fine-tuning is much less than that needed for training from scratch, this approach can reduce the cost of developing flashover prediction models for new house layouts.



**Fig. 4.** The illustrations of Model Training Strategy

Considering a variety of room numbers and arrangements in typical residential houses, this study selected nine different residential layouts to evaluate the applicability of the model. Table 3 lists the corresponding building areas, total number of rooms, and the number of each type of room. Home #0 was used to pre-train the model. The other eight residential layouts (Home #1-8), which were used separately to fine-tune the model, covered a wide range of common house types within the United States [29]. Detailed layout diagrams for Home #1-8 are provided in Appendix B to further illustrate the distinct spatial configurations of each layout. Following the dataset construction strategy in Section 2, corresponding datasets were constructed for each residential layout to facilitate subsequent discussions on the model's scalability.

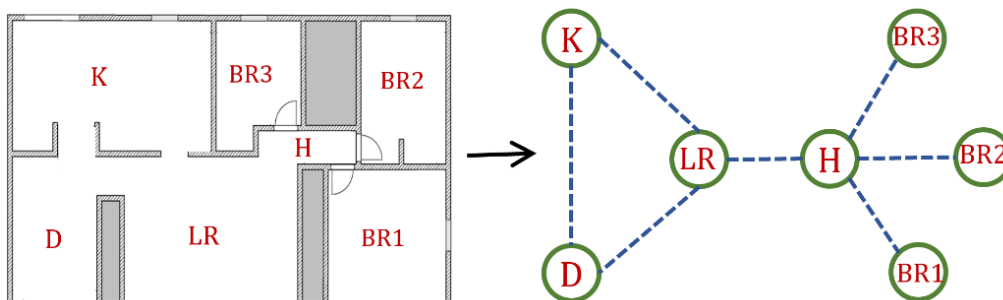
**Table 3.** Summary of building layouts and compartment details (refer to Appendix B for full details).

	Pre-train	Fine-tune (separately)							
Home	#0	#1	#2	#3	#4	#5	#6	#7	#8
Area(m <sup>2</sup> )	107	95	95	95	107	107	130	142	275
Living room	1	1	1	1	1	1	1	1	2
Kitchen	1	1	1	1	1	1	1	1	1
Hallway	1	1	1	1	1	1	2	2	3
Bedroom	3	1	2	2	3	3	2	2	4
Bathroom	0	1	1	1	1	1	1	2	2
Dining room	1	0	0	1	0	1	1	0	1
Total	7	5	6	7	7	8	8	8	13

## 4. Result

### 4.1 Pre-training the Model Based on a House Layout

This study first pre-trained a model using Home #0 to form a baseline model. Fig. 5 depicts the layout of Home #0, which comprises seven compartments: a living room (LR), a kitchen (K), a dining room (DR), a hallway (H), and three bedrooms (BR1, BR2, BR3). It also illustrates the associated graph structure representing this building configuration. The adjacency matrix was constructed according to this graph structure, and the initial adjacency matrix was initialed to represent an unweighted, undirected graph.



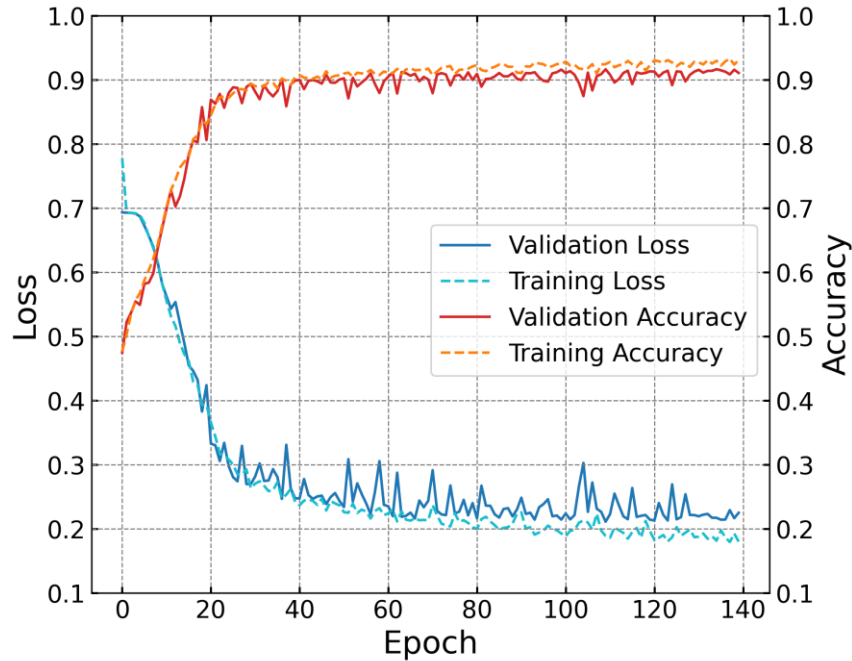
**Fig. 5.** The building layout of Home#0 and the associated graph structure

This study also considered the practical limitations of temperature sensors. In real fire scenarios, sensors fail to function and provide temperature data when exposed to temperatures exceeding their operational limits. To consider data loss due to sensor failure in the data collection through numerical simulation, the following strategy was employed:

$$T_i^{(j)} = \begin{cases} 0, & \text{if } T_i^{(j)} > T_{\text{fail}} \\ T_i^{(j)}, & \text{otherwise} \end{cases} \quad (9)$$

If the sensor temperature,  $T_i$ , exceeds the sensor failure temperature,  $T_{\text{fail}}$ , the sensor temperature is replaced with 0 °C. In this study, the  $T_{\text{fail}}$  was taken to be 150 °C [30]. The temperature sequence and the adjacency matrix representing the spatial relationships of the sensors were the model inputs. Cross-Entropy was adopted as the loss function, and the Adam algorithm was used to

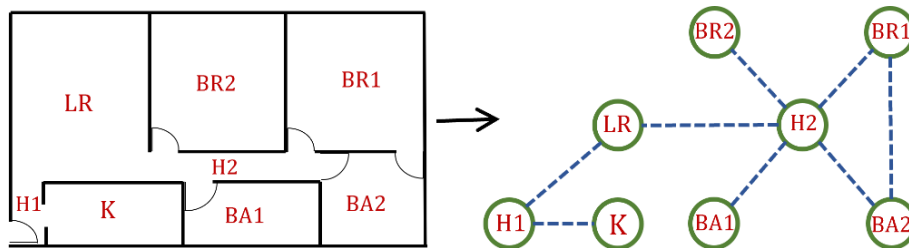
update the model's weight parameters. An early-stop strategy was employed to prevent overfitting. The loss and accuracy curve of the model during the training and validation process are shown in Fig 6. The loss curves demonstrate convergence, with the validation loss stabilizing after approximately 135 epochs, indicating that the model has reached an optimal state. The model achieved an accuracy of 92 % on the test set, indicating that it is a high-performance predictive model.



**Fig. 6.** Loss and accuracy plot for training set and validation set of pre-trained model

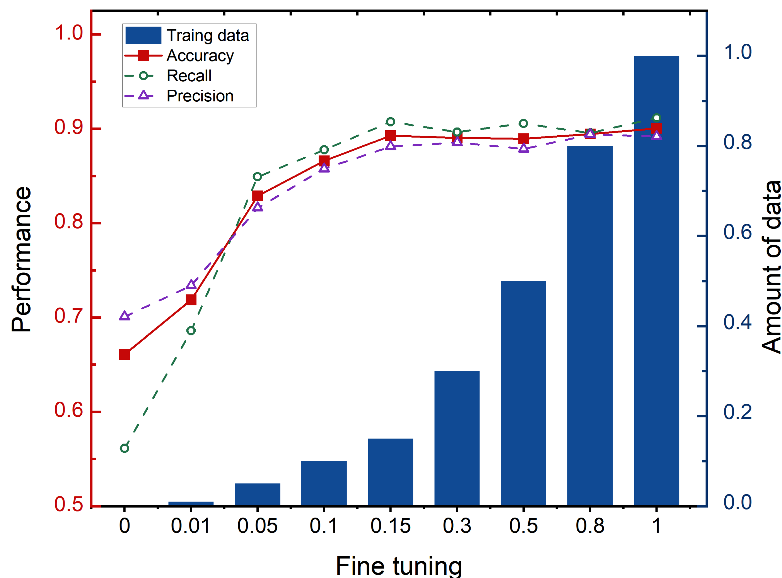
#### 4.2 Fine-Tuning the Model for another House Layout

After obtaining the pre-trained model, the model was fine-tuned using one layout (i.e., Home #7) to demonstrate the model's scalability. Fig. 7 depicts the layout of Home #7, which comprises eight compartments: a living room (LR), a kitchen (K), two hallway (H1, H2), two bedrooms (BR1, BR2), and two bathrooms (BA1, BA2). It also illustrates the associated graph structure representing this building configuration. Obviously, there is a significant difference between this layout and Home #0 as shown in Fig. 5. And the number of sensors is also different.



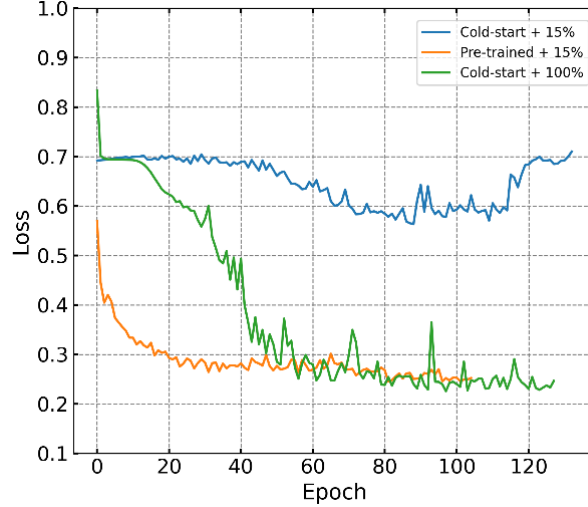
**Fig. 7.** The building layout of Home #7 and the corresponding graph structure.

The study utilized different amounts of data (0 to 1 indicating 0 % to 100 %) from the new layout to evaluate the effectiveness of transfer learning and the results are shown in Fig 8. When no fine-tuning data was used, indicating the pre-trained model was directly tested on the layout of Home #7, the accuracy is about 66 %. As the amount of data used to fine-tune the model increased, the model's performance improved significantly. With 5 % of the fine-tuning data from the new layout, the model's accuracy increased to about 83 %. Upon increasing the fine-tuning data to 15 %, the model achieved nearly 90 % accuracy. Beyond this point, further increases in fine-tuning data did not significantly enhance the model's performance, suggesting that the model stabilizes with about 15 % of the fine-tuning data.



**Fig. 8.** Evaluation of the performance of the fine-tuned model using varying percentages of the training data.

To further analyze the training process during the model development, this study compared the loss curves among three different training strategies: (1) fine-tuning the pre-trained model with 15 % of the training dataset (Pre-trained + 15 %); (2) training the model from scratch using 15 % of the training dataset (Cold-start + 15 %); and (3) training the model from scratch using 100 % of the training dataset (Cold-start + 100 %). The loss curves on the validation set during training for the three strategies are shown in Fig 9. It should be noted that Cold-start denotes that the model was trained from scratch.



**Fig. 9.** The loss on the validation set during training for different strategies

It is observed that the initial loss value for the pre-trained model is lower than that of the cold-start model. Even with only 15% of the training data, the model converged quickly. This is because when fine-tuning the pre-trained model, the model only needs to adjust the spatial blocks based on the new graph structure and does not need to re-learn the temporal features. Additionally, it is noted that training the model from scratch with 100 % of the data resulted in slower convergence and an unstable training process. When training the model from scratch with only 15% of the data, the loss curve indicated overfitting because there was only a small amount of training data, leading to ineffective learning.

The performance of the model, pre-trained with 15 %, indicated that gFlashNet can be directly applied to different house layouts without any structural adjustments. It allows for the use of transfer learning to inherit existing knowledge. By using model parameter weights developed and trained with a large amount of data from one house layout, the pre-trained model's learned general knowledge about flashover prediction. Fine-tuning on different house layouts can yield reliable models, significantly reducing the data amount and cost required to develop corresponding models for other house layouts.

### 4.3 Performance analysis of transfer learning under various house layouts

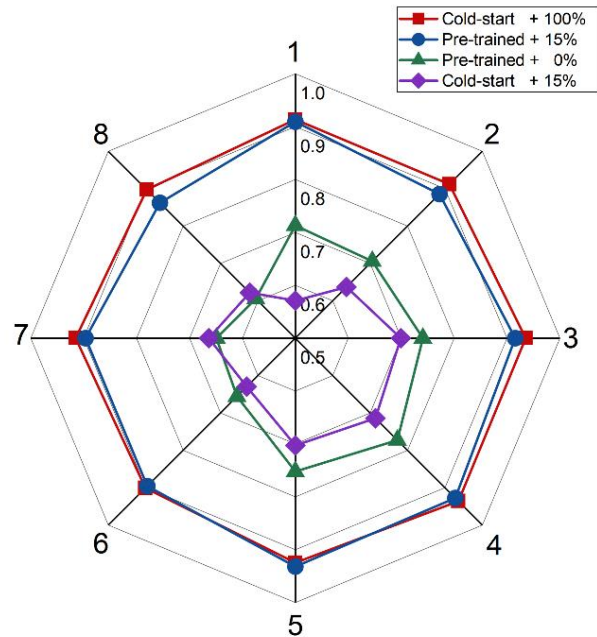
To further demonstrate the effectiveness of the pre-training and transfer learning, this study also tested the performance of the fine-tuned model on the remaining 7 house layouts, as listed in Table 3. Using the model training strategy determined in Section 4.2, the pre-trained model was fine-tuned with 15% of the corresponding fine-tuning data for each house layout. The results on the test sets for each house layout are shown in Table 4. The approximate accuracy of the models fine-tuned on each of the home layouts reached about 90 %. Additionally, the model maintained balanced recall and precision scores.

**Table 4.** The fine-tuned model performance for Home #1-8

Home	#1	#2	#3	#4	#5	#6	#7	#8
Accuracy	90.9%	88.5%	90.8%	91.8%	92.0%	88.6%	89.1%	87.1%

<b>Recall</b>	91.1%	87.8%	91.2%	92.3%	92.6%	89.3%	90.4%	89.2%
<b>Precision</b>	90.8%	89.2%	90.4%	91.5%	91.5%	87.9%	88.1%	85.6%

To further compare the effectiveness of the transfer learning strategies listed in Section 4.2: (1) Cold-start + 100 %, (2) Pre-trained + 15 %, (3) Pre-trained + 0 %, and (4) Cold-start + 15 %, model comparison for all 8 house layouts was carried out. The results were plotted in a radar chart, as shown in Fig 10. As shown in the figure, using a small amount of data to train the model from scratch (Cold-start + 15 %) did not yield a model with good performance. Similar poor model performance was observed from the pre-trained model when no data was used for fine-tuning. (Pre-trained with 0 %). However, by fine-tuning the pre-trained model with a small amount of data (Pre-trained with 15 %), the model's performance significantly improved across all house layouts. Furthermore, the model (Pre-trained with 15 %) can achieve model performance comparable to that of Cold-start with 100 % which demonstrates that transfer learning can greatly reduce the costs of developing flashover prediction models for different house layouts.



**Fig. 10.** The model performance of different development strategies for Home #1-8

## 5. Conclusions

To enhance the scalability of the flashover prediction model, this study developed a more general flashover prediction model, gFlashNet, using the Spatio-Temporal Graph Neural Network (ST-GNN) architecture. Using graph structure and graph attention mechanisms to automatically encode the spatial relationship and connectivity between sensor nodes, the flashover prediction model demonstrated promising performance even when temperature was only available up to 150 °C. Furthermore, this study adopted a transfer learning training strategy, utilizing a pre-trained model developed on one type of house and fine-tuning it with data from other houses. The results indicated that even without any fine-tuning, the pre-trained model can achieve about 70 %

prediction accuracy when it was directly applied to other layouts. By fine-tuning the pre-trained model with only 15 % of the data from the new layout, the performance can be significantly enhanced. Consequently, the amount of required data is significantly reduced, and the pre-trained model converges more quickly during training.

However, it is also important to note that the effectiveness of transfer learning is influenced by the degree of similarity between the pre-trained and target scenarios. In this study, the house layouts share comparable features, such as similar compartment configurations, dimensions of openings, and material properties. These similarities likely contributed to the observed high transferability and the reduced data requirements for fine-tuning. In cases where the pre-trained and target scenarios exhibit significant dissimilarities, such as differences in construction types, ventilation conditions, or compartment layouts, the benefits of transfer learning may be diminished, requiring more substantial fine-tuning efforts or additional data. Overall, this work greatly reduced the cost of developing flashover prediction models for new house layouts while emphasizing the importance of scenario similarity in achieving optimal performance.

## Acknowledgments

The contribution from the State Key Laboratory of fire Science to this work is supported by the Opening Fund of the State Key Laboratory of Fire Science (SKLFS) under Grant No. HZ2024-KF12.

## References

- [1] A.A. Khan, M.A. Khan, K. Leung, X. Huang, M. Luo, A. Usmani, A review of critical fire event library for buildings and safety framework for smart firefighting, *Int. J. Disaster Risk Reduct.* 83 (2022) 103412. <https://doi.org/10.1016/j.ijdrr.2022.103412>.
- [2] X. Huang, W.C. Tam, eds., *Intelligent Building Fire Safety and Smart Firefighting*, Springer Nature Switzerland, Cham, (2024). <https://doi.org/10.1007/978-3-031-48161-1>.
- [3] L. Kou, X. Wang, X. Guo, J. Zhu, H. Zhang, Deep learning based inverse model for building fire source location and intensity estimation, *Fire Saf. J.* 121 (2021) 103310. <https://doi.org/10.1016/j.firesaf.2021.103310>.
- [4] X. Wu, Y. Park, A. Li, X. Huang, F. Xiao, A. Usmani, Smart Detection of Fire Source in Tunnel Based on the Numerical Database and Artificial Intelligence, *Fire Technol.* 57 (2021) 657–682. <https://doi.org/10.1007/s10694-020-00985-z>.
- [5] H. Fang, M. Xu, B. Zhang, S.M. Lo, Enabling fire source localization in building fire emergencies with a machine learning-based inverse modeling approach, *J. Build. Eng.* 78 (2023) 107605. <https://doi.org/10.1016/j.jobe.2023.107605>.
- [6] W.C. Tam, E.Y. Fu, J. Li, R. Peacock, P. Reneke, G. Ngai, H.V. Leong, T. Cleary, M.X. Huang, Real-time flashover prediction model for multi-compartment building structures using attention based recurrent neural networks, *Expert Syst. Appl.* 223 (2023) 119899. <https://doi.org/10.1016/j.eswa.2023.119899>.

- [7] S.A. Yusuf, A.A. Alshdadi, M.O. Alassafi, R. AlGhamdi, A. Samad, Predicting catastrophic temperature changes based on past events via a CNN-LSTM regression mechanism, *Neural Comput. Appl.* 33 (2021) 9775–9790. <https://doi.org/10.1007/s00521-021-06033-3>.
- [8] T. Zhang, F. Ding, Z. Wang, F. Xiao, C.X. Lu, X. Huang, Forecasting backdraft with multimodal method: Fusion of fire image and sensor data, *Eng. Appl. Artif. Intell.* 132 (2024) 107939. <https://doi.org/10.1016/j.engappai.2024.107939>.
- [9] D.J. Garrity, S.A. Yusuf, A predictive decision-aid device to warn firefighters of catastrophic temperature increases using an AI-based time-series algorithm, *Saf. Sci.* 138 (2021) 105237. <https://doi.org/10.1016/j.ssci.2021.105237>.
- [10] H. Fang, S.M. Lo, Y. Zhang, Y. Shen, Development of a machine-learning approach for identifying the stages of fire development in residential room fires, *Fire Saf. J.* 126 (2021) 103469. <https://doi.org/10.1016/j.firesaf.2021.103469>.
- [11] A.A. Ishola, D. Valles, Enhancing Safety and Efficiency in Firefighting Operations via Deep Learning and Temperature Forecasting Modeling in Autonomous Unit, *Sensors* 23 (2023) 4628. <https://doi.org/10.3390/s23104628>.
- [12] Lu, X., He, M., Wang, Z., Hu, H., Ji, J., & Zhu, J. Prediction of flashover time in a compartment fire by CNN-LSTM based deep neural network considering wearable data collection equipment. *J. Build. Eng.* 97 (2024) 110719. <https://doi.org/10.1016/j.jobte.2024.110719>
- [13] L. Fan, W.C. Tam, Q. Tong, E. Fu, L. Tianshui, An Explainable Machine Learning Based Flashover Prediction Model Using Dimension-Wise Class Activation Map, *Fire Saf. J.* 140 (2023). <https://doi.org/10.1016/j.firesaf.2023.103849>.
- [14] W.C. Tam, E.Y. Fu, J. Li, X. Huang, J. Chen, M.X. Huang, A spatial temporal graph neural network model for predicting flashover in arbitrary building floorplans, *Eng. Appl. Artif. Intell.* 115 (2022) 105258. <https://doi.org/10.1016/j.engappai.2022.105258>.
- [15] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A Comprehensive Survey on Transfer Learning, *Proc. IEEE* 109 (2021) 43–76. <https://doi.org/10.1109/JPROC.2020.3004555>.
- [16] R.D. Peacock, P.A. Reneke, R.W. Bukowski, V. Babrauskas, Defining flashover for fire hazard calculations, *Fire Saf. J.* 32 (1999) 331–345. [https://doi.org/10.1016/S0379-7112\(98\)00048-4](https://doi.org/10.1016/S0379-7112(98)00048-4).
- [17] Firefighter fatalities in the United States | NFPA, (n.d.). <https://www.nfpa.org/education-and-research/research/nfpa-research/fire-statistical-reports/fatal-firefighter-injuries> (accessed July 1, 2024).
- [18] P.A. Reneke, M. Bruns, S.W. Gilbert, C.P. MacLaren, R.D. Peacock, T.G. Cleary, D.T. Butry, Towards a Process to Quantify the Hazard of Fire Protection Design Alternatives, NIST (2019). <https://www.nist.gov/publications/towards-process-quantify-hazard-fire-protection-design-alternatives> (accessed July 1, 2024).
- [19] V. Babrauskas, Will the second item ignite?, *Fire Saf. J.* 4 (1981) 281–292. [https://doi.org/10.1016/0379-7112\(81\)90031-X](https://doi.org/10.1016/0379-7112(81)90031-X).
- [20] V. Babrauskas, Glass breakage in fires, *Fire Science and Technology, Inc* (2011) 1-7.
- [21] R.D. Peacock, P.A. Reneke, G.P. Forney, CFAST – Consolidated Model of Fire Growth and Smoke Transport (Version 7) Volume 3: Software Development and Model Evaluation Guide, National Institute of Standards and Technology, 2015. <https://doi.org/10.6028/NIST.TN.1889v3>.
- [22] M.J. Hurley, D.T. Gottuk, J.R.H. Jr, K. Harada, E.D. Kuligowski, M. Puchovsky, J. L. Torero, J.M.W. Jr, C.J. WIECZOREK, *SFPE Handbook of Fire Protection Engineering*, Springer, 2015.

- [23] B. Yu, H. Yin, Z. Zhu, Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting, in: Proc. Twenty-Seventh Int. Jt. Conf. Artif. Intell., 2018: pp. 3634–3640. <https://doi.org/10.24963/ijcai.2018/505>.
- [24] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How Powerful are Graph Neural Networks?, (2019). <https://doi.org/10.48550/arXiv.1810.00826>.
- [25] T.N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, (2017). <https://doi.org/10.48550/arXiv.1609.02907>.
- [26] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, WaveNet: A Generative Model for Raw Audio, (2016). <http://arxiv.org/abs/1609.03499> (accessed July 1, 2024).
- [27] Yu, B., Yin, H., & Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. (2017). arXiv preprint arXiv:1709.04875.
- [28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph Attention Networks, (2018). <https://doi.org/10.48550/arXiv.1710.10903>.
- [29] Persily, A., Musser, A., Leber, D.D., 2006. A Collection of Homes To Represent the US Housing Stock. US NISTIR 7330, National Institute of Standards and Technology, Gaithersburg, MD.
- [30] A. Cowlard, W. Jahn, C. Abecassis-Empis, G. Rein, J.L. Torero, Sensor Assisted Fire Fighting, Fire Technol. 46 (2010) 719–741. <https://doi.org/10.1007/s10694-008-0069-1>.

## Appendix A

In this study, we assumed consistent compartment configurations and material properties across the selected home structures to provide a controlled framework for analyzing flashover prediction. Specifically, the overall ceiling height was set at 2.4 m. And the height and width for doors and doorways are 2.05 m and 0.9 m, respectively. Additionally, while the layouts of the residential buildings vary, the dimensions of the windows remain consistent across the same room types, as outlined in the table R1.

Table R1. Dimensions and Vertical Locations of Windows for Different Room Types

Room Type	Width (m)	Height (m)	Vertical Location (m)
Living Room	1.8	1.4	0.6
Kitchen	1.4	0.85	1.2
Dining Room	1.4	1.4	0.6
Family Room	1.8	1.4	0.6
Bedrooms	1.4	1.4	0.6

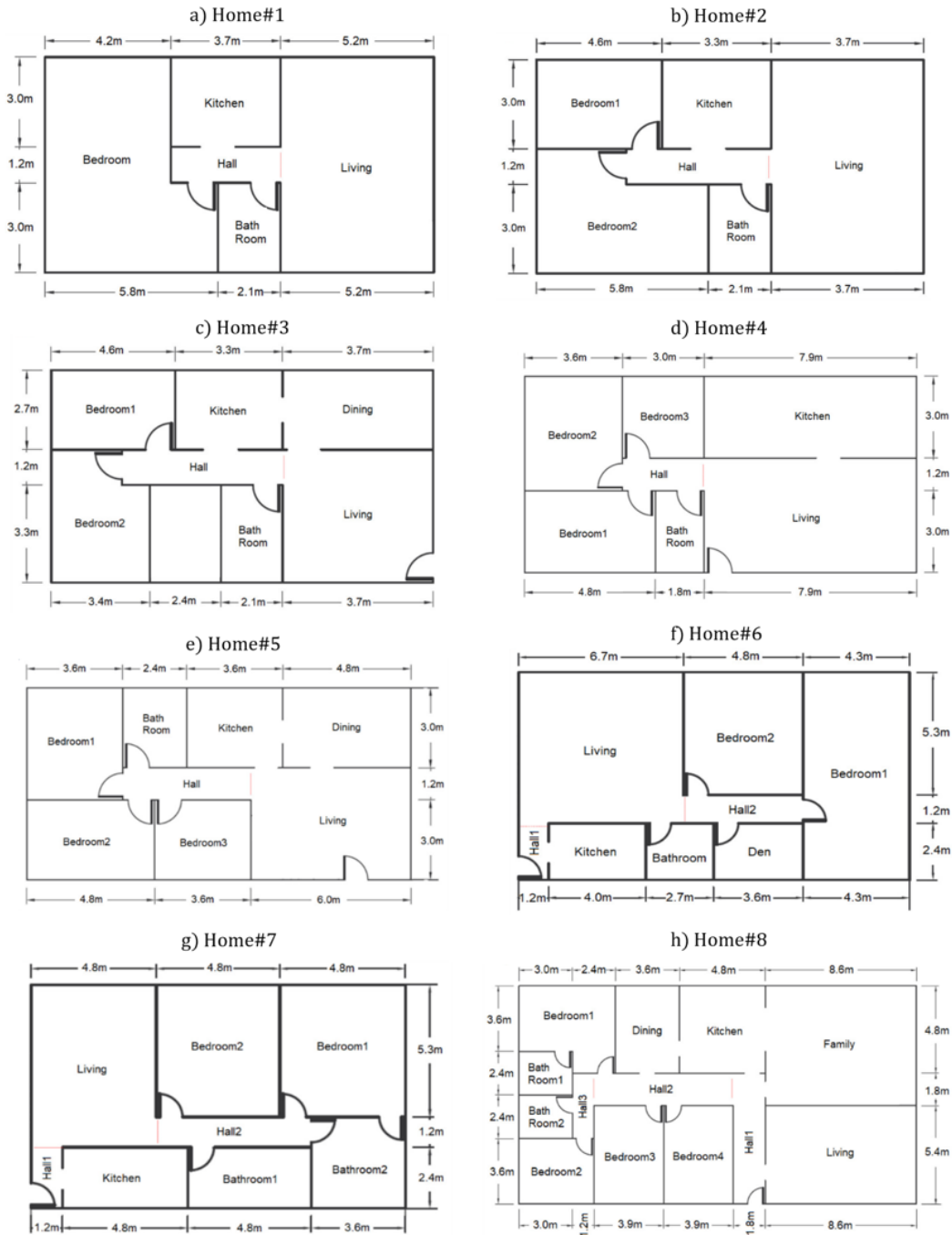
Regarding the housing envelope, the walls and ceilings were modeled with gypsum wallboards, and the floors were constructed with concrete. The glazing material is consistent for all windows, taken as 3 mm single-pane float glass. The thermal properties of these materials, which significantly influence fire dynamics, are listed in Table R2. These configurations were selected to represent typical residential building features and to ensure reproducibility across different scenarios.

Table R2. Summary of approximate thermal properties for building materials.

Materials	Conductivity	Specific Heat	Density	Thickness	Emissivity
Unit	kW/(m · K)	kJ/(kg · K)	kg/m <sup>3</sup>	m	-
Gypsum	0.00016	1	480	0.025	0.9
Concrete	0.0016	0.75	2400	0.15	0.94
Glazing	0.0008	0.8	2500	0.003	0.95

## Appendix B

Fig. B1. presents the detailed layout diagrams for the eight residential layouts (Home #1-8) [29] used in the study.



**Fig. B1.** Layout Diagrams of Residential Buildings (Home #1-8)