International Journal of Geo-Information

MDPI

Article

PGTFT: A Lightweight Graph-Attention Temporal Fusion Transformer for Predicting Pedestrian Congestion in Shadow Areas

Jiyoon Lee [1] and Youngok Kang [1,2,*]

[1] Department of Big Data Analytics, Ewha Womans University, Seoul 03760, Republic of Korea; jiyoon821@ewha.ac.kr
[2] Department of Social Studies (Geography), Ewha Womans University, Seoul 03760, Republic of Korea
* Correspondence: ykang@ewha.ac.kr; Tel.: +82-(2)3277-6646

Abstract

Forecasting pedestrian congestion in urban back streets is challenging due to "shadow areas" where CCTV coverage is absent and trajectory data cannot be directly collected. To address these gaps, we propose the Peak-aware Graph-attention Temporal Fusion Transformer (PGTFT), a lightweight hybrid model that extends the Temporal Fusion Transformer by integrating a non-parametric attention-based Graph Convolutional Network, a peak-aware Gated Residual Network, and a Peak-weighted Quantile Loss. The model leverages both physical connectivity and functional similarity between roads through a fused adjacency matrix, while enhancing sensitivity to high-congestion events. Using real-world trajectory data from 38 CCTVs in Anyang, South Korea, experiments show that PGTFT outperforms LSTM, TFT, and GCN-TFT across different sparsity settings. Under sparse 5 m neighbor conditions, the model achieved the lowest MAE (0.059) and RMSE (0.102), while under denser 30 m settings it maintained superior accuracy with standard quantile loss. Importantly, PGTFT requires only 1.54 million parameters—about half the size of conventional Transformer–GCN hybrids—while delivering equal or better predictive performance. These results demonstrate that PGTFT is both parameter-efficient and robust, offering strong potential for deployment in smart city monitoring, emergency response, and transportation planning, as well as a practical approach to addressing data sparsity in urban sensing systems.

Keywords: pedestrian congestion prediction; shadow areas; Temporal Fusion Transformer (TFT); graph neural network (GNN); non-parametric attention; Peak-Weighted Quantile Loss (PWQL); spatiotemporal forecasting; Peak-aware Graph-attention Temporal Fusion Transformer (PGTFT)

check for updates

Academic Editors: Wolfgang Kainz, Hartwig H. Hochmair, Hao Li and Levente Juhász

Received: 13 August 2025
Revised: 21 September 2025
Accepted: 26 September 2025
Published: 28 September 2025

Citation: Lee, J.; Kang, Y. PGTFT: A Lightweight Graph-Attention Temporal Fusion Transformer for Predicting Pedestrian Congestion in Shadow Areas. *ISPRS Int. J. Geo-Inf.* **2025**, *14*, 381. https://doi.org/10.3390/ijgi14100381

Copyright: © 2025 by the authors. Published by MDPI on behalf of the International Society for Photogrammetry and Remote Sensing. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In densely populated cities, road networks consist not only of major arterial roads but also of narrow and intricately connected back streets within residential areas. As these back streets serve as vital components of everyday urban life, the deployment of CCTV for crime prevention, public safety, and integrated monitoring has increased markedly in recent years. Trajectory data collected from these systems are now widely used for applications such as pedestrian congestion estimation, anomaly detection, and illegal parking analysis [1–6].

For effective monitoring, CCTV cameras should ideally be installed at intervals of about 30 m, corresponding to their typical range. In practice, however, comprehensive

coverage is infeasible due to installation and maintenance costs. As a result, CCTV is often concentrated along major roads, leaving many alleys and narrow streets unmonitored. These "shadow areas" create gaps in observational data and limit the ability to capture congestion patterns.

Addressing this challenge requires predictive models capable of inferring pedestrian congestion in shadow areas based on sparse and partially observed data. Accurate estimation in such contexts would contribute to the early prevention of accidents and other risks, while also supporting broader smart city initiatives. Recent advances in graph-based deep learning and Transformer architectures have demonstrated strong performance for road network modeling by exploiting connectivity, correlation, and spatial dependency [7–14]. Yet most existing approaches assume relatively dense and uniformly distributed data, focusing on arterial roads with near-complete coverage. Research on congestion forecasting under sparse conditions remains limited, and studies specifically targeting shadow areas are particularly scarce [15–17].

To fill this gap, we propose the Peak-aware Graph-attention Temporal Fusion Transformer (PGTFT), a lightweight hybrid model that integrates a non-parametric attention-based Graph Convolutional Network (GCN) with the Temporal Fusion Transformer (TFT). The model leverages both physical and semantic relationships among road segments and incorporates a peak-weighted loss function to improve accuracy during high-risk congestion periods. By focusing on environments where continuous monitoring is infeasible, PGTFT addresses the overlooked but socially significant issue of sparse surveillance coverage in urban back streets, offering a practical step toward safer and more cost-efficient smart city monitoring.

The main contributions of this study are as follows. First, we present a hybrid deep learning model—PGTFT—that integrates a non-parametric attention-based GCN with a modified TFT. Unlike conventional graph–temporal models, our approach introduces a soft-attention mechanism that dynamically assigns edge weights based on input statistics rather than parameterized attention, thereby capturing both structural connectivity and functional similarity between road segments. Second, we propose a hybrid adjacency matrix that combines topological proximity (hop distance) with semantic similarity (cosine similarity of congestion distributions), allowing the model to account for non-local but behaviorally similar road segments. Third, we introduce a peak-aware learning strategy, including a novel Peak-weighted Quantile Loss (PWQL), which enhances sensitivity to high-congestion periods that are critical for real-time monitoring and public safety management. To the best of our knowledge, this is the first study to dynamically weight the loss function based on observed congestion magnitudes rather than static attributes.

The remainder of this paper is organized as follows. Section 2 reviews related work on graph- and Transformer-based models in traffic and spatial analytics. Section 3 details the architecture of PGTFT. Section 4 presents experimental results using a pedestrian congestion dataset collected from urban back streets, including comparisons with baseline models such as TFT [15], GCN-TFT [16], and LSTM [17], as well as an ablation study. Section 5 concludes with implications and directions for future research.

2. Related Works

Recent studies have increasingly explored the use of Transformer and GCN models—two core components of the proposed framework—for traffic flow prediction and sparse-data inference. Attention-based Transformer architectures, which have demonstrated strong performance across multiple domains, have been widely adopted for traffic forecasting [15,18–21]. In parallel, GCN-based approaches that model road networks as

graphs have been applied to traffic prediction, congestion estimation, and sparse-data analysis [11,22–24].

GCNs are widely used for capturing spatial dependencies in structured data such as those of road networks. Yu et al. [22] formulated traffic prediction as a time-series problem over a graph and proposed ST-GCN, a purely convolutional model without recurrent structures, which alternates temporal and spatial graph convolutions to capture spatiotemporal patterns. Zhao et al. [11] combined GCN and GRU to jointly model spatial and temporal features, while Ding et al. [24] introduced IDG-PSAtt, which employs a dynamically learned spatiotemporal graph structure with a probabilistic sparse self-attention mechanism. These approaches highlight the flexibility of GCNs in representing road connectivity and spatiotemporal interactions.

Transformer architectures and their variants are particularly effective for long-term time-series forecasting and have been widely applied to traffic prediction. Wang et al. [20] proposed the Spatiotemporal Fusion Transformer (STFT), which incorporates seasonality encoding, tubelet embedding for efficiency, and a diffusion-based token permutator for regional traffic forecasting. Xu et al. [19] developed Spatial-Temporal Transformer Networks (STTN), alternating between spatial and temporal Transformers to capture directional dependencies and long-range dynamics. Zhang et al. [25] designed a Sparse Attention-based Transformer to improve prediction in congestion-prone segments, while He et al. [18] introduced TEA-GCN, which combines Transformer layers with an adaptive GCN module and local-global temporal attention for dynamic spatiotemporal learning. Among these, the TFT proposed by Lim et al. [15] has become a widely adopted framework for complex time-series forecasting, integrating static covariate encoders, variable selection networks, and multi-head attention. Li et al. [16] further extended TFT by incorporating GCN to model graph structures in spatiotemporal data.

In urban environments, traffic data from back streets are often sparse or incomplete due to limited sensor deployment and high infrastructure costs, which has prompted increasing interest in methods to address these gaps. Tan et al. [26] used matrix completion with day-mode similarity to impute missing data, while Sun et al. [27] improved prediction in low-data regions by transferring models trained on data-rich areas. Kong et al. [28] proposed GraphSparceNet (GSNet), a scalable GNN with linear time and space complexity, designed to balance efficiency with predictive accuracy in large-scale traffic datasets.

Despite these advances, high-precision predictions still require dense, fine-grained data at the node and edge level. For back streets, however, physical limitations and sparse sensor coverage make it difficult to collect such structured data. Consequently, congestion forecasting in shadow areas remains severely constrained. Moreover, models trained on datasets from arterial roads often fail to capture the unique spatial and contextual dynamics of back streets. To address these challenges, it is essential to develop models tailored to sparse conditions and trained with data that reflect the specific characteristics of such environments. This study responds to this need by proposing a sparse-data-based predictive model designed to operate effectively in shadow areas with limited information.

3. Proposed Method

This study proposes the PGTFT, a fusion deep learning model designed to jointly learn the structural connectivity of road networks and the irregular volatility of sparse data (Figure 1). PGTFT builds upon the TFT introduced by Lim et al. [15], extending it to integrate time-series forecasting with road network structures.

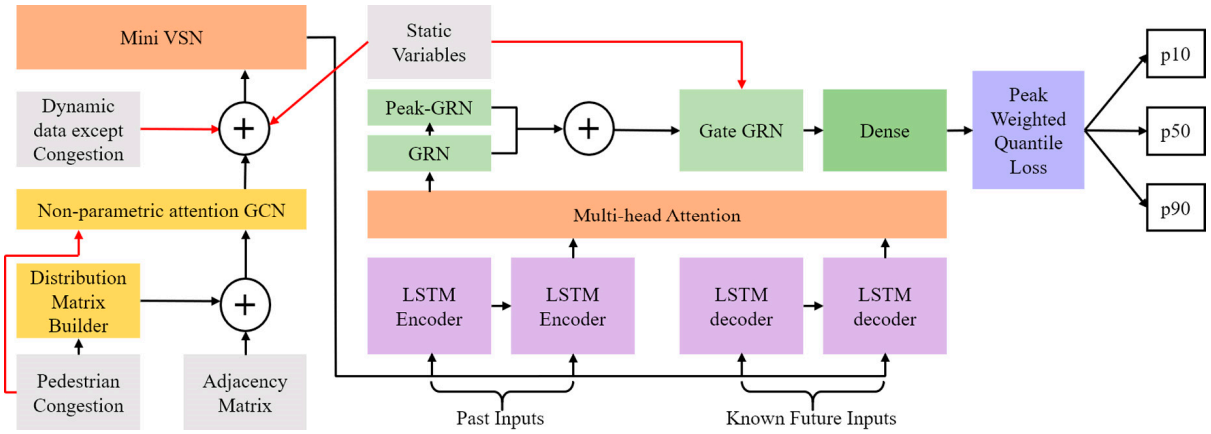


Figure 1. Architecture of the proposed PGTFT model, combining TFT with non-parametric attention GCN, peak-aware GRN, and peak-weighted quantile loss.

TFT consists of key components such as the Variable Selection Network (VSN), GRN, and interpretable multi-head attention, which together provide strong capabilities for modeling both dynamic temporal patterns and static contextual features. While retaining these strengths, PGTFT introduces several modifications and simplifications to better address the unique characteristics of pedestrian and vehicle data collected from sparsely monitored and highly variable back streets.

The major architectural innovations of PGTFT are as follows:

- Non-parametric attention GCN: A non-parametric attention mechanism is incorporated into the GCN to dynamically estimate the relative influence of road segments based on structural connectivity.
- Hybrid adjacency matrix: To capture both topological proximity and functional similarity, PGTFT combines hop-based adjacency with cosine similarity of congestion distributions, enabling more precise modeling of inter-road interactions.
- Simplified variable selection: Recognizing the limited informativeness of static features in back street environments, the original VSN is simplified to avoid unnecessary relearning and reduce computational overhead.
- Lightweight attention mechanism: The static enrichment step in TFT's interpretable attention module is removed, retaining only the attention weight outputs to balance accuracy with computational efficiency.
- Peak-aware GRN: An additional GRN module is used to re-encode high and low attention values from the multi-head attention block, enhancing sensitivity to congestion peaks in time-series data.
- PWQL: Replacing the standard quantile loss, the proposed loss function emphasizes peak values, improving prediction performance for high-congestion segments that are critical for real-time urban monitoring.

3.1. PGTFT

3.1.1. Non-Parametric Attention GCN

In this study, we introduce a lightweight attention mechanism into the GCN by applying GELU activation and softmax normalization to the output of the fused adjacency matrix and input features prior to linear transformation. This design simplifies the attention mechanism of the Graph Attention Network (GAT) [29] while retaining its essential functionality. Equation (1) illustrates how GAT computes the attention coefficient between a target node i and its neighboring nodes N_i. GAT uses masked attention to avoid full pairwise computation, but the cost still increases with the number of neighbors. Let $\vec{h}_i$ and

$\vec{h}_j$ denote the input features of nodes i and j respectively, $\boldsymbol{W}$ a shared linear transformation, and $\vec{a}$ a learnable vector:

$$\begin{aligned} \alpha_{ij} &= \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \\ e_{ij} &= \text{LeakyReLU}(\vec{a}^{\top} [\boldsymbol{W}\vec{h}_i || \boldsymbol{W}\vec{h}_j]) \end{aligned} \tag{1}$$

To better handle real-time, congestion-sensitive data, our model replaces the pairwise operations with matrix multiplication and introduces the GELU activation [30], which probabilistically suppresses less relevant signals while enhancing informative ones. In the subsequent GCN layer, the attention coefficients are used for message passing. Following Kipf and Welling [31], who noted that activation functions in GCNs can be flexibly chosen, we adopt the Swish (SiLU) activation [32] for its self-gating and gradient-friendly properties:

$$\begin{aligned} \boldsymbol{W} &= \text{Softmax}(\text{GELU}(\boldsymbol{A} \cdot \boldsymbol{X})) \\ \boldsymbol{H}_{out} &= \text{SiLU}(\boldsymbol{W} \cdot \boldsymbol{W}_{lin} + \boldsymbol{b}) \end{aligned} \tag{2}$$

where $\boldsymbol{A}$ is the fused adjacency matrix, $\boldsymbol{X}$ is the input feature matrix, and $\boldsymbol{W}$ and $\boldsymbol{b}$ the attention-like weights and bias. GELU is defined as $\text{GELU}(x) = x \cdot \Phi(x)$, where $\Phi(x)$ is the Gaussian cumulative distribution function, and SiLU as $\text{SiLU}(x) = x \cdot \sigma(x)$, where $\sigma(x)$ is the sigmoid function.

This non-parametric attention design offers several advantages. Unlike conventional GCNs, which aggregate neighbor features using fixed adjacency weights, our model dynamically generates softmax-based importance scores directly from the inputs. In contrast to GAT, where attention coefficients are learned as parameters, our approach derives them non-parametrically, reducing computational cost while preserving flexibility. The combination of GELU and SiLU further improves gradient flow, resulting in more stable and effective training.

3.1.2. Hybrid Adjacency Matrix

In a GCN, the input consists of two main components: the node feature matrix $\boldsymbol{X}$ and the adjacency matrix $\boldsymbol{A}$, which defines the connectivity between nodes. In this study, node features are defined as the congestion levels of the six nearest neighboring roads surrounding each target shadow road. The adjacency matrix is initially constructed with weights determined by hop distance (Algorithm 1). The selection of six neighbors reflects a one-hop configuration—three adjacent nodes on each side when a shadow node lies between two intersections. If fewer than six one-hop neighbors exist, we progressively include the nearest nodes from higher-order neighborhoods (up to five hops). Any remaining vacancies are zero-padded to preserve a fixed input size, and padded nodes are excluded from adjacency computations.

The hop-based adjacency matrix $\boldsymbol{A}^{(hop)}$ measures similarity between nodes according to the difference in hop distance from the target node. Specifically, the weight between nodes i and j is defined as:

$$\boldsymbol{A}_b^{(hop)} = \boldsymbol{A}_{ij} = \frac{1}{1 + \left| \text{hop}_i - \text{hop}_j \right|} \tag{3}$$

In graph-based deep learning, adjacency matrices typically assign weights in the range [0, 1] to capture spatial proximity. However, hop distance alone may not fully represent the complexity of real-world interactions. To incorporate functional similarity, we compute a distribution feature vector for each neighboring road segment based on

statistical descriptors of congestion: mean $\mu_{b,n}$, standard deviation $\sigma_{b,n}$, skewness $Skew_{b,n}$, and kurtosis $Kurt_{b,n}$:

$$\begin{gathered} \mathbf{X} \in \mathbb{R}^{B\times T\times N} \\ \mu_{b,n} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{X}_{b,t,n},\ \ \sigma_{b,n} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\mathbf{X}_{b,t,n}-\mu_{b,n}\right)^2+\epsilon} \\ Skew_{b,n} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\mathbf{X}_{b,t,n}-\mu_{b,n}}{\sigma_{b,n}}\right)^3,\ \ Kurt_{b,n} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\mathbf{X}_{b,t,n}-\mu_{b,n}}{\sigma_{b,n}}\right)^4 \\ \mathbf{f}_{\mathrm{b,n}} = [\mu_{b,n},\sigma_{b,n},Skew_{b,n},Kurt_{b,n}] \in \mathbb{R}^4,\ \ \mathbf{F}_b = [\mathbf{f}_{b,1},\cdots,\mathbf{f}_{b,N}]^\top \end{gathered} \tag{4}$$

Algorithm 1 Compute Adjacency Matrix based on Hop Distance Similarity.

Require: *neighbors data*: list of neighbor nodes with id and hop attributes
Require: *top n*: number of neighbors (including padding)
Ensure: *adj*: adjacency matrix of size *top n* × *top n*
Initialize *adj* ← zero matrix of size *top n* × *top n*
for *i* = 1 to *top n* **do**
  if *neighbors data[i].id* = 'PAD' **then**
    continue
  end if
  hop *i* ← *neighbors data[i].hop*
  for *j* = 1 to *top n* **do**
    if *neighbors data[j].id* = 'PAD' **then**
      continue
    end if
    hop *j* ← *neighbors data[j].hop*
    hop difference ← |hop *i* − hop *j*|
    node-to-node similarity ← 1/(1 + *hop difference*)
    adj[i][j] ← *node-to-node similarity*
  end for
  adj[i][i] ← 1.0
end for
return *adj*

Using these descriptors, we construct a distribution-based adjacency matrix $\boldsymbol{A}^{(dist)}$ by computing cosine similarity between normalized feature vectors:

$$\boldsymbol{A}_b^{(dist)} = \cos(\mathbf{F}_b,\mathbf{F}_b) = \hat{\mathbf{F}}_b\cdot\hat{\mathbf{F}}_b^\top \in \mathbb{R}^{N\times N},\ \hat{\mathbf{f}}_{b,n} = \frac{\mathbf{f}_{b,n}}{\|\ \mathbf{f}_{b,n}\ \|_2}$$

The final fused adjacency matrix is defined as a weighted average of the hop-based and distribution-based matrices:

$$\boldsymbol{A}_b = \alpha\cdot\boldsymbol{A}_b^{(hop)} + (1-\alpha)\cdot\boldsymbol{A}_b^{(dist)},\ \alpha\in[0,1] \tag{5}$$

This design allows the model to capture both physical and functional relationships between roads. For example, (1) physically adjacent roads may exert varying influence depending on the similarity of their congestion patterns, and (2) physically distant roads may still strongly influence a target road if their congestion profiles are functionally similar.

3.1.3. MiniVSN

In the standard TFT architecture, the VSN and the Static Covariate Encoder employ a GRN to extract salient features from the input variables. As defined in Equation (6), the GRN alleviates the vanishing gradient problem through a Gated Linear Unit (GLU [33]) and dynamically adjusts the contribution of each variable. Specifically, $\boldsymbol{a}$ denotes the input data, $\boldsymbol{c}$ the contextual vector, $\boldsymbol{W}_{(.)}$ and $\boldsymbol{b}_{(.)}$ the trainable weights and biases, and $\odot$ the element-wise Hadamard product.

$$\begin{aligned} \mathrm{GRN}_{\omega}(\boldsymbol{a}, \boldsymbol{c}) &= \mathrm{LayerNorm}(\boldsymbol{a} + \mathrm{GLU}_{\omega}(\boldsymbol{\eta}_1)), \\ \boldsymbol{\eta}_1 &= \boldsymbol{W}_{1,\omega} \quad \boldsymbol{\eta}_2 + \boldsymbol{b}_{1,\omega}, \\ \boldsymbol{\eta}_2 &= \mathrm{ELU}(\boldsymbol{W}_{2,\omega} \quad \boldsymbol{a} + \boldsymbol{W}_{3,\omega} \quad \boldsymbol{c} + \boldsymbol{b}_{2,\omega}), \\ \mathrm{GLU}_{\omega}(\boldsymbol{\gamma}) &= \sigma(\boldsymbol{W}_{4,\omega} \quad \boldsymbol{\gamma} + \boldsymbol{b}_{4,w}) \odot (\boldsymbol{W}_{5,\omega} \quad \boldsymbol{\gamma} + \boldsymbol{b}_{5,\omega}) \end{aligned} \tag{6}$$

While this architecture is highly effective for large-scale time-series tasks, uniformly applying such complexity to sparse datasets may reduce efficiency and robustness. To address this, we propose a simplified VSN (miniVSN) tailored for sparse pedestrian data.

Given $\mathbf{X} \in \mathbb{R}^{B \times T \times N \times D}$, where B is batch size, T the sequence length, N the number of variables, and D the feature dimension, miniVSN computes variable importance scores and weighted representations through lightweight operations. Specifically, a dot product between each variable representation $\mathbf{h}_{b,t,i}$ and a learnable vector $\boldsymbol{w}$ yields a scalar importance score, which is normalized across variables using softmax. A corresponding weighted representation s_2 is then derived, and both outputs are aggregated via a weighted sum:

$$\begin{aligned} s_1 = \mathbf{h}_{b,t,i}^{\top} \cdot \boldsymbol{w} + b, \;\; s_2 = \mathbf{x}_{b,t,i}^{\top} \cdot \boldsymbol{w}_i + b_i \\ \mathrm{miniVSN}_{b,t} = \sum_{i=1}^{N} \mathrm{softmax}_i(s_1) \cdot s_2 \end{aligned} \tag{7}$$

Through this process, miniVSN assigns higher weights to informative features while suppressing less relevant ones, thereby producing a refined representation with substantially lower computational cost. This design allows PGTFT to selectively emphasize the most critical variables for prediction while avoiding the overhead of full GRN-based selection.

3.1.4. Peak Aware GRN

In the original TFT, contextual information is not only processed in the VSN but also further refined in the GRN together with outputs from the multi-head attention mechanism, thereby enriching temporal features. This design is effective for large and diverse datasets; however, in road networks with sparse coverage and limited variability across segments, static data often provides little additional information. In such settings, emphasizing peak patterns in the time series becomes more valuable than repeatedly propagating static features.

To this end, we introduce a Peak-aware GRN, an additional operation that strengthens peak-related signals by reusing the output of the multi-head attention layer and its gating mechanism. Specifically, the attention output $\boldsymbol{A}_{attn}$ is first processed through a GRN to reinforce baseline temporal context. The same output is then reused as both input and context in a secondary GRN, denoted as GRN_{peak}, to further amplify peak patterns. This process is defined as:

$$\begin{aligned} \text{Peak Aware} &= \mathrm{GRN}_{peak}(\boldsymbol{A}_{attn}, \boldsymbol{A}_{attn}) \\ \boldsymbol{A}_{enhanced} &= \boldsymbol{A}_{attn} + \text{Peak Aware} \end{aligned} \tag{8}$$

By explicitly reinforcing peaks, the model captures sharp variations in congestion more effectively, which is particularly important for real-time monitoring in narrow urban back streets where sudden fluctuations pose safety and management challenges.

3.2. Loss Function: Peak Weighted Quantile Loss

The original TFT adopts the quantile loss function proposed by Zhou [34], shown in Equation (9). Quantile loss evaluates errors at multiple quantile levels (e.g., 10th, 50th, 90th percentiles), making it robust to non-Gaussian distributions and more resilient to over- and under-estimation compared to conventional losses such as Mean Squared Error. Let Ω denote the training dataset with M samples, W the model parameters, and Q the set of target quantiles:

$$\begin{gathered} L(\Omega, W) = \sum_{y_t \in \Omega} \sum_{q \in Q} \sum_{\tau=1}^{\tau_{max}} \frac{\mathrm{QL}(y_t, \hat{y}(q, t-\tau, \tau), q)}{M \tau_{max}} \\ \mathrm{QL}(y, \hat{y}, q) = q(y - \hat{y})_+ + (1-q)(\hat{y} - y)_+, \ \ q \in Q \\ (.)_+ = max(0, .) \end{gathered} \tag{9}$$

Building on this framework, PGTFT introduces the PWQL to improve prediction accuracy in high-congestion scenarios. PWQL amplifies the penalty for larger target values, making the model more sensitive to peak regions. As defined in Equation (10), the quantile loss is scaled by a weighting function $w(y_t)$, where the coefficient $\alpha \in [0, 1]$ controls the degree of emphasis. In this study, $\alpha = 0.7$.

$$\begin{gathered} L(\Omega, W) = \sum_{y_t \in \Omega} \sum_{q \in Q} \sum_{\tau=1}^{\tau_{max}} \frac{\mathrm{PWQL}}{M \tau_{max}} \\ \mathrm{PWQL} = w(y_t) \cdot \mathrm{QL}(y, \hat{y}, q) \\ (.)_+ = max(0, .) \\ w(y_t) = 1 + \alpha \cdot y_t \end{gathered} \tag{10}$$

Unlike conventional weighted quantile loss functions that adjust weights based on sample reliability or predefined importance, PWQL directly uses the target value to determine the weight. To the best of our knowledge, this is the first study to adopt such a strategy, representing a key methodological contribution. Notably, when $q = 0.5$, quantile loss reduces to the Mean Absolute Error (MAE). Consistent with the original TFT, experiments in this study were conducted using $Q = \{0.1, 0.5, 0.9\}$.

4. Performance Evaluation

4.1. Datasets

The dataset used in this study was derived from pedestrian and vehicle trajectories collected at one-second intervals from 38 CCTVs installed in the Indeokwon area of Anyang, Republic of Korea, between 20 and 27 May 2024. Congestion values were computed at one-minute intervals.

Congestion was quantified by estimating the minimum spatial area required for each moving entity based on instantaneous speed, direction, and acceleration. At each time step, the Time to Collision (TTC) was calculated between the current and predicted positions one second ahead. From this, a minimum safety buffer was derived for each entity, representing the region in which it could move without risk of collision. The union of all safety buffers was then computed, and its ratio to the total road area defined the congestion level. In total, the dataset includes 387,427 pedestrian trajectories and 317,990 vehicle trajectories, yielding 62,223,691 spatio-temporal points. Details of the calculation process are available in [1].

As shown in Figure 2, the presence of numerous shadow nodes results in substantial data sparsity across the study area. Figure 3 presents a one-hour aggregated heatmap, illustrating distinct spatial and temporal variations in congestion across back streets. Because ground-truth values are unavailable for shadow nodes, a masking strategy was employed for model evaluation: nodes with observed values were temporarily treated as

unobserved, their congestion inferred from neighboring nodes, and predictions compared against actual observations.

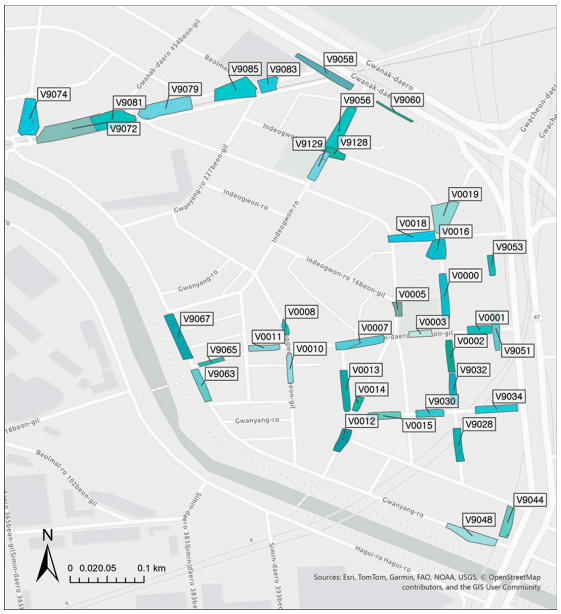


Figure 2. Locations and coverage areas of the 38 CCTVs in the Indeokwon study area; uncovered road segments denote shadow nodes.

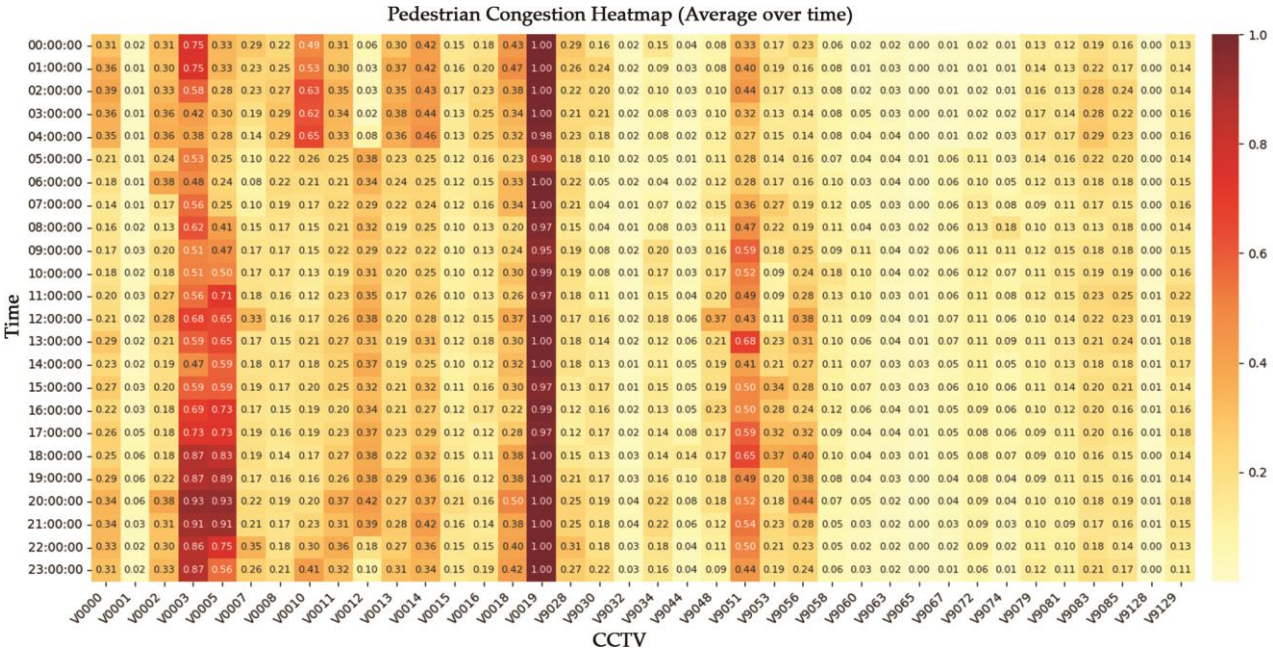


Figure 3. Hourly average pedestrian congestion heatmap for 38 alleyways over a 7-day period. Each cell shows the mean congestion level for the corresponding hour, with CCTV IDs denoting monitored alleyways.

For each target road segment, the six nearest adjacent roads were used as reference inputs. The prediction task was designed to forecast congestion over the next two hours using the preceding six hours of historical data. Dynamic variables included: (1) pedestrian congestion on adjacent roads, (2) a binary peak-hour indicator, (3) the rate of change in congestion relative to the previous minute, and (4) average congestion across 1-hop neighbors. Static variables included: (1) hop distance from the target road, (2) road length, and (3) road width.

Connectivity within the network was represented by hop counts between road segments. Each CCTV corresponded to a specific road segment (Figure 2). For instance, segment V0002 has four 1-hop neighbors (V0000, V0001, V0003, V9032) and seven 2-hop neighbors (V0005, V0007, V0016, V9028, V9030, V9034, V9051). The six adjacent roads were selected from within a 1–5 hop range, with zero-padded virtual roads added when fewer than six neighbors were available; these padded nodes were excluded from adjacency computations using a padding mask. Peak hours were defined as 07:00–10:00 (morning commute), 12:00–14:00 (lunch), and 17:00–20:00 (evening commute).

4.2. Model Training and Hyperparameters Tuning

All experiments were performed in Python 3.10 using the PyTorch 2.2.2 deep learning framework with CUDA 11.8 for GPU acceleration. Training was performed for up to 20 epochs with a batch size of 12 and a learning rate of 0.001. The hidden dimension was set to 128, with 8 self-attention heads and 2 layers across all experiments. Optimization was carried out using AdamW with an L2 regularization (weight decay) coefficient of 0.001 to improve generalization and stabilize learning by preventing excessively large weight updates.

Model performance was evaluated using both average quantile loss and the proposed PWQL. Quantiles were set to p10, p50, and p90, with losses computed individually and then aggregated. For PWQL, the weighting coefficient α was fixed at 0.7, as comparative experiments with alternative values indicated this setting yielded the best balance of accuracy and stability. Additional experiments analyzed loss behavior with respect to the proportion of padding nodes among the six adjacent road inputs.

To evaluate model robustness under varying levels of data sparsity, we generated two modified datasets by applying distance thresholds of 5 m and 30 m when defining neighboring roads. The 5 m threshold represents a stricter criterion in which only very close neighbors are considered. In dense metropolitan networks, this often results in fewer than six valid neighbors, requiring the use of padding nodes (PAD) to maintain a consistent input size. In contrast, the 30 m threshold relaxes this criterion, allowing more distant neighbors to be included. This setting typically provides sufficient valid neighbors, thereby reducing the proportion of padding nodes and yielding more complete inputs. Comparing model performance under these two scenarios—PAD-dominated inputs at 5 m versus densely observed inputs at 30 m—provides clear insight into how different degrees of input sparsity influence predictive accuracy.

Computational complexity was assessed by comparing the total number of trainable parameters across models, peak GPU memory consumption, and inference time across models, providing a measure of efficiency relative to baseline architectures. All experiments were conducted on a system equipped with an NVIDIA GeForce RTX 3080 Ti GPU (12 GB), a 12-core Intel64 Family 6 Model 151 Stepping 2 CPU, 16 GB of system memory, and Windows 10.

4.3. Benchmarks

To verify the effectiveness of the proposed PGTFT, we conducted comparative experiments with existing time-series and graph-based models, as well as ablation studies to assess the contribution of each component. All experiments were performed on the same dataset with identical input settings and evaluation metrics to ensure fair comparison.

4.3.1. Baseline

Three baseline models were selected:

- LSTM: A recurrent neural network commonly used for time-series forecasting. Performance was evaluated using quantile loss.
- TFT: A Transformer-based model designed for complex time-series tasks. Two versions were tested: one with quantile loss and another with PWQL.
- GCN-TFT: An extension of TFT that incorporates a GCN layer to represent road connectivity through adjacency matrices. This model was also evaluated with both quantile loss and PWQL.

4.3.2. Ablation

To evaluate the impact of each core component, ablation experiments were conducted with the following variants:

- PGTFT-GCN (Q loss): Replaces the non-parametric attention with standard GCN-based spatiotemporal learning, trained with quantile loss.
- PGTFT-GCN (Peak Q loss): Same as above but trained with PWQL.
- PGTFT with non-parametric attention (Q loss): Uses the proposed non-parametric attention module but without peak-weighted loss.
- PGTFT with non-parametric attention (Peak Q loss): The full proposed model, integrating non-parametric attention with peak-weighted loss.

4.4. Evaluation Metrics

Model performance was comprehensively evaluated using 5-fold cross-validation, and the best-performing fold for each model was reported. Prediction accuracy was assessed through MAE, Root Mean Squared Error (RMSE), and Mean Quantile Loss, with quantile-specific errors at the 10th, 50th, and 90th percentiles calculated to reflect the distributional nature of time-series forecasting. These accuracy metrics capture both central tendency and variability, enabling a more detailed evaluation of model robustness in sparse-data settings.

To assess computational efficiency and validate the lightweight design of PGTFT, additional complexity-related metrics were reported. Alongside the total number of trainable parameters, which represents structural compactness, we measured peak GPU memory usage and average inference time. These indicators provide practical evidence of model efficiency, complementing traditional accuracy metrics by highlighting trade-offs between predictive performance and resource consumption. FLOPs were not used, as precise estimation is challenging for attention-based architectures; instead, the selected indicators more directly reflect the computational cost of training and real-time deployment.

By jointly considering accuracy, quantile-based uncertainty, and efficiency measures, the evaluation framework enables a balanced comparison of predictive capability and computational burden across all models.

4.5. Experimental Results

The experimental results are presented in four parts to provide a comprehensive evaluation of the proposed PGTFT model. First, we compare predictive accuracy under different neighbor thresholds and examine how loss functions affect performance (Section 4.5.1). Second, we analyze model compactness and computational efficiency using parameters, GPU memory, and inference time (Section 4.5.2). Third, we present visual comparisons between predicted and observed congestion values as well as quantile-based uncertainty analyses (Section 4.5.3). Finally, we investigate the effect of the fused adjacency matrix by contrasting it with the hop-only design and discussing its implications for different urban contexts (Section 4.5.4).

4.5.1. Performance Under Different Neighbor Thresholds

When the neighbor threshold was set to 5 m, the results (Table 1) show that the proposed PGTFT combined with PWQL achieved the highest overall performance. Incorporating peak values into the loss function proved particularly effective in sparse data environments, as evidenced by the lowest MAE (0.059), lowest RMSE (0.102), and the best 90% quantile loss (0.018), outperforming all baseline models. Under the 30 m threshold, where most neighboring nodes contained valid observations, the best performance across all metrics (MAE, RMSE, and quantile losses) was achieved by PGTFT integrating the non-parametric attention GCN with the standard Quantile Loss (Table 2).

Table 1. Test performance of benchmark models and PGTFT under the 5 m neighbor threshold.

Model	GCN Type	Loss Type	MAE	RMSE	Mean Loss	50%	90%
LSTM		Q	0.080	0.135	0.025	0.040	0.023
TFT		Q	0.066	0.124	**0.023**	**0.033**	0.023
		Peak Q	0.094	0.155	0.035	0.055	0.034
GCN-TFT	GCN	Q	0.069	0.129	0.024	0.035	0.023
		Peak Q	0.075	0.132	0.028	0.044	0.024
PGTFT (Ours)	GCN	Q	0.082	0.170	0.026	0.041	0.022
		Peak Q	0.079	0.139	0.031	0.046	0.034
	Non-parametric attention GCN	Q	0.071	0.130	0.024	0.035	0.021
		Peak Q	**0.059**	**0.102**	**0.023**	**0.033**	**0.018**

Table 2. Test performance of benchmark models and PGTFT under the 30 m neighbor threshold.

Model	GCN Type	Loss Type	MAE	RMSE	Mean Loss	50%	90%
LSTM		Q	0.074	0.130	0.024	0.037	0.024
TFT		Q	0.073	0.131	0.025	0.037	0.022
		Peak Q	0.074	0.132	0.032	0.045	0.036
GCN-TFT	GCN	Q	0.081	0.145	0.024	0.041	0.021
		Peak Q	0.073	0.121	0.028	0.043	0.025
PGTFT (Ours)	GCN	Q	0.080	0.154	0.026	0.040	0.024
		Peak Q	0.091	0.165	0.036	0.059	0.032
	Non-parametric attention GCN	Q	**0.062**	**0.115**	**0.019**	**0.031**	**0.016**
		Peak Q	0.071	0.123	0.026	0.041	0.022

For GCN-TFT, applying PWQL improved MAE and RMSE relative to the standard Quantile Loss, but the latter performed better for mean quantile loss and for the 50% and 90% quantiles. A similar trend was observed for PGTFT, where the standard Quantile Loss produced more stable results when sufficient data were available. These findings suggest that peak-sensitive loss functions are beneficial under sparse conditions, but conventional quantile loss remains more effective in well-observed settings. Notably, PGTFT demonstrated reliable performance even without peak weighting, indicating its robustness across different data densities.

Across both experimental settings, PGTFT consistently outperformed baselines. Under the 5 m threshold, results were comparable to TFT in terms of mean and 50% quantile loss, but the non-parametric attention GCN provided additional advantages by assigning softmax-based importance scores within the graph. This improved performance relative to models with standard GCNs, highlighting the role of non-parametric attention in enhancing

accuracy. The Swish activation function, with its self-gating properties, also contributed to superior training stability compared to ReLU.

These results further indicate that, under the 5 m threshold, models trained with PWQL achieved consistently lower errors than those trained with the standard Quantile Loss, confirming the advantage of PWQL for peak detection in sparse settings. In contrast, under the 30 m threshold, the standard Quantile Loss yielded superior performance, suggesting that conventional loss functions remain more effective when sufficient observational data are available.

4.5.2. Model Compactness and Efficiency

Model compactness in this study was evaluated using the total number of trainable parameters, peak GPU memory usage, and average inference time, which together provide practical evidence of computational efficiency beyond predictive accuracy. Tables 3 and 4 present the averaged five-fold results, including standard deviations, under the 5 m and 30 m neighbor threshold settings, respectively.

Table 3. Five-fold average performance and efficiency of models under the 5 m neighbor threshold.

Model	GCN Type	Loss Type	Mean MAE	Mean RMSE	Mean Peak GRU Memory (MB)	Mean Inference Time (ms)	Total Params
LSTM		Q	0.112 ± 0.028	0.193 ± 0.051	23.45 ± 0.64	0.28 ± 0.02	208,131
TFT		Q	0.107 ± 0.030	0.196 ± 0.057	90.24 ± 9.67	8.88 ± 0.25	3,019,773
		Peak Q	0.113 ± 0.025	0.204 ± 0.045	90.24 ± 9.67	8.69 ± 0.07	3,019,773
GCN-TFT	GCN	Q	0.094 ± 0.023	0.168 ± 0.040	118.37 ± 12.64	9.86 ± 0.78	4,030,189
		Peak Q	0.097 ± 0.022	0.172 ± 0.039	118.37 ± 12.64	9.40 ± 0.07	4,030,189
PGTFT (Ours)	GCN	Q	0.100 ± 0.020	0.179 ± 0.033	**65.19 ± 4.98**	**5.48 ± 0.11**	1,540,356
		Peak Q	0.105 ± 0.025	0.186 ± 0.045	65.19 ± 4.98	5.60 ± 0.09	1,540,356
	Non-parametric attention GCN	Q	0.094 ± 0.023	0.167 ± 0.041	65.78 ± 5.10	5.55 ± 0.13	1,540,356
		Peak Q	**0.091 ± 0.031**	**0.162 ± 0.053**	65.78 ± 5.10	5.57 ± 0.15	**1,540,356**

Table 4. Five-fold average performance and efficiency of models under the 30 m neighbor threshold.

Model	GCN Type	Loss Type	Mean MAE	Mean RMSE	Mean Peak GRU Memory (MB)	Mean Inference Time (ms)	Total Params
LSTM		Q	0.104 ± 0.023	0.191 ± 0.043	23.45 ± 0.64	0.29 ± 0.05	208,131
TFT		Q	0.098 ± 0.018	0.183 ± 0.036	90.24 ± 9.67	8.47 ± 0.04	3,019,773
		Peak Q	0.107 ± 0.021	0.203 ± 0.041	90.24 ± 9.67	8.42 ± 0.13	3,019,773
GCN-TFT	GCN	Q	0.099 ± 0.014	0.180 ± 0.032	118.37 ± 12.64	9.06 ± 0.10	4,030,189
		Peak Q	**0.095 ± 0.019**	**0.169 ± 0.034**	118.37 ± 12.64	9.28 ± 0.17	4,030,189
PGTFT (Ours)	GCN	Q	0.100 ± 0.020	0.177 ± 0.038	65.52 ± 5.15	5.41 ± 0.19	1,540,356
		Peak Q	0.114 ± 0.015	0.205 ± 0.025	65.52 ± 5.15	5.23 ± 0.07	1,540,356
	Non-parametric attention GCN	Q	0.116 ± 0.046	0.206 ± 0.088	65.79 ± 5.10	5.45 ± 0.20	1,540,356
		Peak Q	0.104 ± 0.026	**0.177 ± 0.049**	65.79 ± 5.10	5.31 ± 0.10	**1,540,356**

As shown in Table 3, under the 5 m threshold, PGTFT with non-parametric attention and PWQL achieved the best overall predictive performance, with an average MAE of 0.091 and RMSE of 0.162. In terms of efficiency, PGTFT with a GCN-based design demonstrated the most favorable balance, requiring only 65.52 MB of GPU memory and 5.23 ms of inference time. The non-parametric variant showed comparable efficiency, with only marginal differences. Importantly, PGTFT maintained a lightweight structure with approximately 1.54 million parameters—roughly half the size of TFT and GCN-TFT (3–4 million)—while still delivering equal or superior accuracy. Although LSTM required the fewest parameters (about 200,000) and was the most computationally efficient, its predictive accuracy lagged significantly. These findings confirm that PGTFT achieves an effective trade-off between efficiency and accuracy by simplifying variable selection while reinforcing peak-sensitive learning.

In contrast, Table 4 summarizes results under the 30 m threshold. Here, GCN-TFT with PWQL achieved the lowest MAE and RMSE values. However, these improvements came at a substantial computational cost, requiring 118.37 MB of GPU memory, approximately 9 ms of inference time, and over 4 million parameters. Given the marginal performance gains relative to this added complexity, PGTFT remains the more practical and efficient alternative for real-world applications.

4.5.3. Visualization of Prediction Performance

To provide an intuitive understanding of model performance, we visualized the prediction results under the 5 m threshold and examined both median predictions and quantile-based forecasts. These visualizations complement the quantitative metrics by showing how closely the proposed model aligns with observed values and how effectively it captures peak events and predictive uncertainty.

Figure 4 compares the ground truth with PGTFT predictions under the 5 m threshold. The black line represents observed congestion, the red line denotes PGTFT trained with PWQL, and the gray dashed line corresponds to PGTFT with the standard Quantile Loss. Across all cases, PGTFT predictions most closely follow the ground truth. In Figure 4a,b,d, the model accurately captures both the timing and magnitude of peaks, while in Figure 4c, where peak activity is minimal, the predictions are slightly smoothed yet still closer to the ground truth than those of competing models.

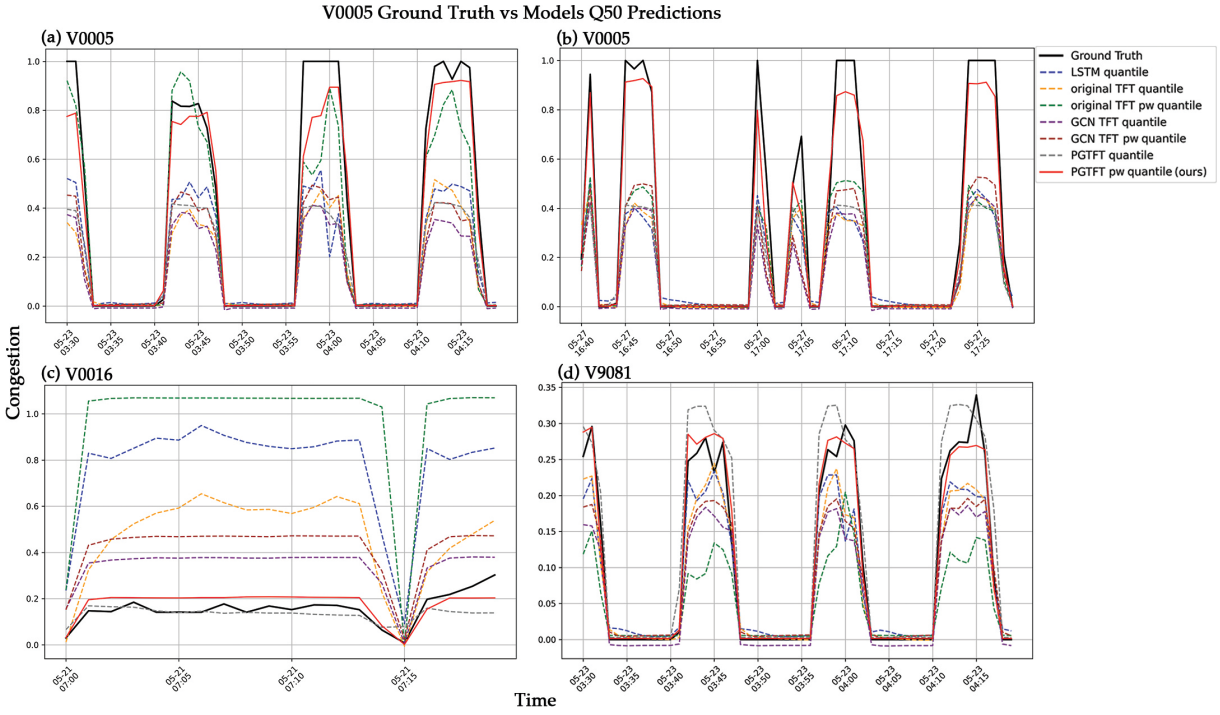


Figure 4. Comparison of ground truth and predicted median congestion (Q50) under the 5 m neighbor threshold. PGTFT most closely follows observed values, capturing peak timing and magnitude more accurately than baselines.

Figure 5 illustrates the quantile-based predictions generated by PGTFT. By producing forecasts at the 10th, 50th, and 90th percentiles, the model provides a distributional view of congestion dynamics. The median prediction (Q50) tracks the overall trend but underrepresents extreme peaks, whereas the Q10–Q90 range reflects predictive uncertainty. Most observed peaks fall within the Q90 band, indicating the model's ability to probabilis-

tically anticipate extreme events. This demonstrates that PGTFT not only captures central tendencies but also provides uncertainty-aware forecasts for high-risk congestion scenarios.

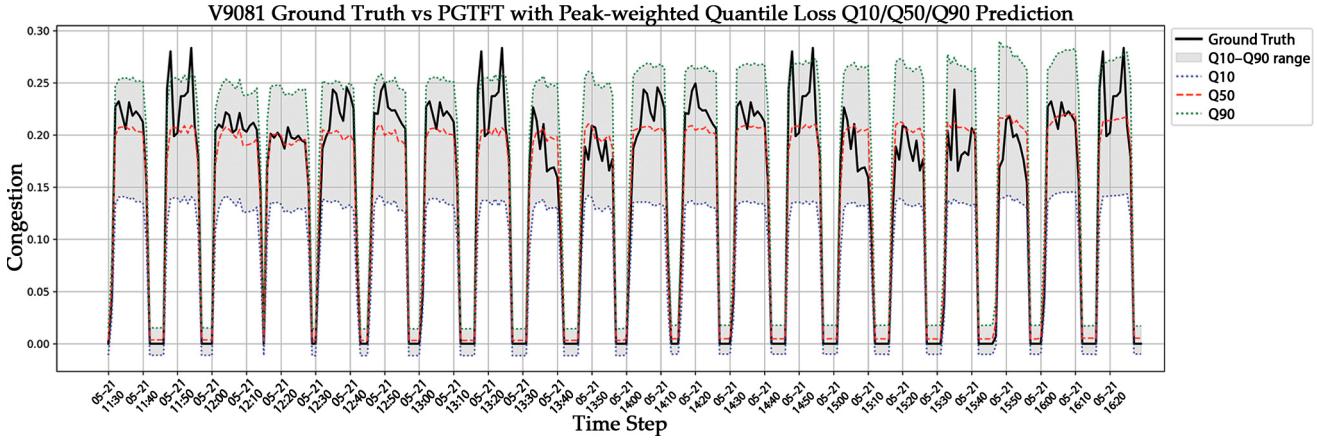


Figure 5. Ground truth and PGTFT predictions with PWQL at Q10, Q50, and Q90. Most peaks fall within the Q90 range, reflecting the model's ability to capture extreme events and predictive uncertainty.

4.5.4. Analysis of the Fused Adjacency Matrix

This experiment was designed to evaluate the practical effect of the hybrid fused adjacency matrix, which combines hop-based and distribution-based similarities, on prediction performance. While the conceptual details of this design were introduced earlier in Section 3.1.2, here we focus on visualizing and interpreting its impact through the experimental results shown in Figure 6.

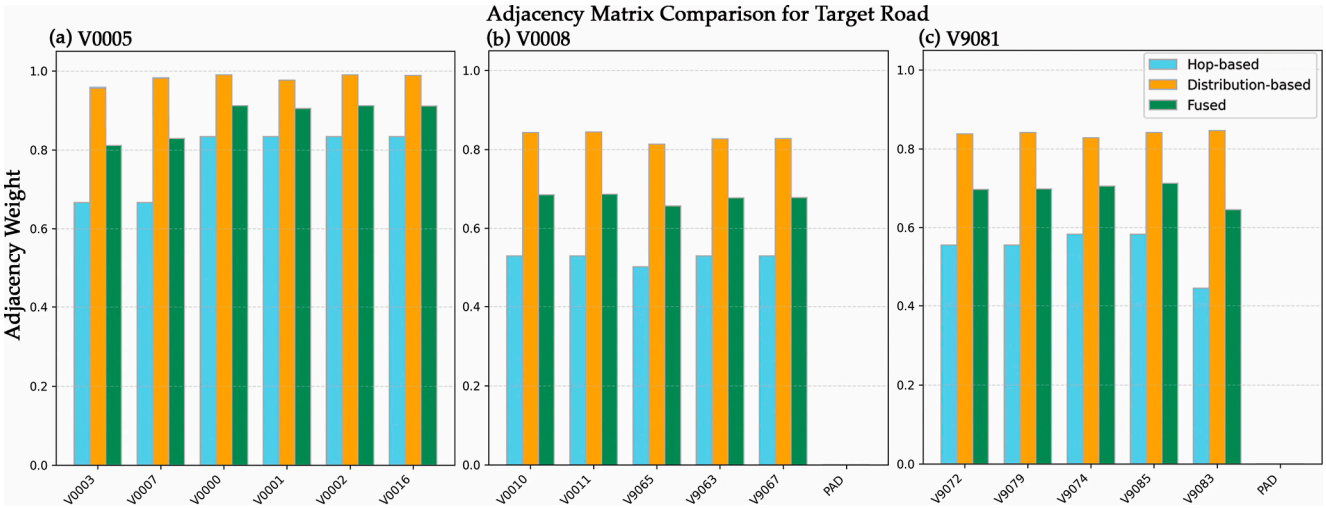


Figure 6. Comparison of hop-based and fused adjacency matrices for a target road segment. The fused design integrates hop distance and distributional similarity, while PAD nodes are excluded from computations.

Figure 6 presents the fused adjacency matrix for a target road segment. Compared to the hop-only matrix, the fused matrix shows visibly richer structure, as even when hop-based weights are low, blending them with cosine similarity in a 1:1 ratio ($\alpha = 0.5$) elevates the overall inter-node similarity values. This integration highlights both local structural consistency and functional similarities across distant but behaviorally similar roads. The choice of $\alpha = 0.5$ was made based on empirical comparisons, which confirmed it provided the most balanced and effective representation for this dataset.

Although the fused matrix exhibits clear numerical differences from the hop-only version, the predictive performance remained largely unchanged. This outcome can be explained by the proportional increase in all neighbor weights, which preserved relative importance among nodes after softmax normalization in the non-parametric attention GCN. The result underscores both the robustness of the proposed architecture and the adequacy of hop-based structural information for the current dataset.

Nonetheless, the fused design has broader implications. In dense urban networks, where neighboring roads are highly connected and congestion patterns vary significantly, relying solely on hop-based proximity may fail to capture functional similarities across distant segments. In such cases, the fused matrix can reduce predictive uncertainty and improve generalization. Conversely, in suburban or highway settings with linear road structures and limited external influences, hop-based adjacency may be more reliable. These findings suggest that while hop-only adjacency was sufficient in the present case, the fused design provides a flexible and adaptive strategy that may yield stronger benefits in more complex or heterogeneous traffic environments.

5. Conclusions

In many urban back streets, pedestrian and traffic conditions can be partially monitored using CCTV-derived trajectory data; however, cost and infrastructure constraints prevent full coverage, creating "shadow areas" where no observations are available. These gaps limit the accuracy and continuity of congestion forecasting across the road network. To address this challenge, this study proposes PGTFT, a lightweight extension of the TFT designed to infer congestion in unmonitored segments by leveraging partially observed data from surrounding roads. The model integrates a non-parametric attention-based GCN, a peak-aware GRN, and a PWQL, enabling it to capture irregular congestion patterns while maintaining computational efficiency.

The contributions of this work are fourfold. First, PGTFT demonstrates robustness under sparse conditions, maintaining stable performance even when neighboring data are limited. Second, the incorporation of peak-aware components improves sensitivity to high-congestion periods. Third, the model achieves accuracy comparable to or exceeding more complex baselines while requiring only 1.54 million parameters, highlighting its efficiency. Fourth, the use of a hybrid adjacency matrix allows the simultaneous representation of both physical connectivity and functional similarity between road segments.

Experimental results across different sparsity levels confirmed that PGTFT consistently outperformed conventional baselines while preserving lightweight complexity. Although the fused adjacency matrix provided only marginal improvements for the dataset used, it is expected to yield greater benefits in settings with more heterogeneous congestion patterns.

This study also has limitations. The dataset was restricted to a single region in Korea, which constrains generalizability, though a neighbor-threshold strategy was applied to diversify connectivity scenarios. In addition, performance varied between the 5 m and 30 m threshold settings, reflecting the influence of data sparsity. Future research should address these limitations by incorporating larger and more diverse datasets and by exploring reinforcement learning strategies for dynamically modeling road connectivity.

The practical implications of this study are significant. By filling observation gaps in areas with limited CCTV coverage, PGTFT offers municipalities a cost-effective tool for enhancing urban safety and monitoring. Its applicability extends beyond congestion forecasting to support intelligent transportation systems, including optimal routing, emergency response planning, and urban design in environments with uneven sensing infrastructure. In addition, the adaptability of the proposed framework suggests broader utility in domains such as vehicular traffic management and urban logistics, where uneven sensing

infrastructure similarly limits data availability. Furthermore, the lightweight design makes PGTFT suitable for real-time applications on resource-constrained platforms, enabling integration with edge devices or mobile sensing units.

In conclusion, PGTFT provides a parameter-efficient and adaptable solution that balances predictive accuracy with computational cost. Building on its demonstrated capability to infer congestion in shadow areas, the model holds strong promise for real-world deployment in smart city applications, while future work should further validate its scalability and extend its functionality across diverse urban contexts.

Author Contributions: For Conceptualization, Jiyoon Lee; Methodology, Software, Validation, Formal Analysis, Jiyoon Lee; Data Curation, Jiyoon Lee, Youngok Kang; Writing—Original Draft Preparation, Jiyoon Lee; Writing—Review and Editing, Youngok Kang; Visualization, Jiyoon Lee; Supervision, Youngok Kang; Project Administration, Youngok Kang; Funding Acquisition, Youngok Kang. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) and funded by the Ministry of Land, Infrastructure, and Transport of the Korean government (Grant No. RS-2022-00143782).

Data Availability Statement: Data can be provided upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lee, J.; Kang, Y. A Dynamic Algorithm for Measuring Pedestrian Congestion and Safety in Urban Alleyways. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 434. [CrossRef]
2. Sultani, W.; Chen, C.; Shah, M. Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6479–6488.
3. Jo, H.W.; Jo, H.Y.; Lee, C.E.; Kim, M.J. Transformer-based Intelligent CCTV System for Real-time Anomaly Detection in Unmanned Stores. *Smart Media J.* **2024**, *13*, 79–91.
4. Zhu, S.; Chen, C.; Sultani, W. Video anomaly detection for smart surveillance. In *Computer Vision: A Reference Guide*; Springer International Publishing: Cham, Switzerland, 2020; pp. 1–8.
5. Ng, C.K.; Cheong, S.N.; Yap, W.J.; Foo, Y.L. Outdoor illegal parking detection system using convolutional neural network on Raspberry Pi. *Int. J. Eng. Technol.* **2018**, *7*, 17–20. [CrossRef]
6. Gao, J.; Zuo, F.; Ozbay, K.; Hammami, O.; Barlas, M.L. A new curb lane monitoring and illegal parking impact estimation approach based on queueing theory and computer vision for cameras with low resolution and low frame rate. *Transp. Res. Part A Policy Pract.* **2022**, *162*, 137–154. [CrossRef]
7. Liu, K.; Gao, S.; Qiu, P.; Liu, X.; Yan, B.; Lu, F. Road2vec: Measuring traffic interactions in urban road system from massive travel routes. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 321. [CrossRef]
8. Cai, P.; Wang, Y.; Lu, G.; Chen, P.; Ding, C.; Sun, J. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transp. Res. Part C Emerg. Technol.* **2016**, *62*, 21–34. [CrossRef]
9. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
10. Jepsen, T.S.; Jensen, C.S.; Nielsen, T.D. Graph convolutional networks for road networks. In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Chicago, IL, USA, 5–8 November 2019; pp. 460–463.
11. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [CrossRef]
12. Yang, H.; Li, Z.; Qi, Y. Predicting traffic propagation flow in urban road network with multi-graph convolutional network. *Complex Intell. Syst.* **2024**, *10*, 23–35. [CrossRef]
13. Ji, X.; Mao, P.; Han, Y. Traffic state prediction for urban networks: A spatial–temporal transformer network model. *J. Transp. Eng. Part A Syst.* **2023**, *149*, 04023105. [CrossRef]
14. Zhang, K.; Ren, H.; Kang, J.; Guo, C.; Chen, W.; Tao, M.; Dai, H.N.; Wan, S.; Bao, H. TST-trans: A transformer network for urban traffic flow prediction. *IEEE Internet Things J.* **2024**, *12*, 8276–8287. [CrossRef]

15. Lim, B.; Arık, S.Ö.; Loeff, N.; Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]
16. Li, X.; Wu, W.; Guo, H.; Qiao, X.; Lu, Y.; Wu, Y.; Qiao, G. An interpretable temperature prediction method for grain in storage based on improved temporal Fusion Transformers. *Comput. Electron. Agric.* **2025**, *236*, 110414. [CrossRef]
17. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
18. He, X.; Zhang, W.; Li, X.; Zhang, X. Tea-gcn: Transformer-enhanced adaptive graph convolutional network for traffic flow forecasting. *Sensors* **2024**, *24*, 7086. [CrossRef] [PubMed]
19. Xu, M.; Dai, W.; Liu, C.; Gao, X.; Lin, W.; Qi, G.J.; Xiong, H. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv* **2020**, arXiv:2001.02908.
20. Wang, Z.; Wang, Y.; Jia, F.; Zhang, F.; Klimenko, N.; Wang, L.; He, Z.; Huang, Z.; Liu, Y. Spatiotemporal Fusion Transformer for large-scale traffic forecasting. *Inf. Fusion* **2024**, *107*, 102293. [CrossRef]
21. Zheng, C.; Fan, X.; Wang, C.; Qi, J. Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1234–1241.
22. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
23. Cui, Z.; Henrickson, K.; Ke, R.; Wang, Y. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 4883–4894. [CrossRef]
24. Ding, Z.; He, Z.; Huang, Z.; Wang, J.; Yin, H. Traffic flow prediction research based on an interactive dynamic spatial–temporal graph convolutional probabilistic sparse attention mechanism (IDG-PSAtt). *Atmosphere* **2024**, *15*, 413. [CrossRef]
25. Zhang, Y.; Zhou, Q.; Wang, J.; Kouvelas, A.; Makridis, M.A. CASAformer: Congestion-aware sparse attention transformer for traffic speed prediction. *Commun. Transp. Res.* **2025**, *5*, 100174. [CrossRef]
26. Tan, H.; Wu, Y.; Cheng, B.; Wang, W.; Ran, B. Robust missing traffic flow imputation considering nonnegativity and road capacity. *Math. Probl. Eng.* **2014**, *2014*, 763469. [CrossRef]
27. Sun, H.; Li, P.; Li, Y. Traffic Flow Prediction in Data-Scarce Regions: A Transfer Learning Approach. *Comput. Mater. Contin.* **2025**, *83*, 4899–4914. [CrossRef]
28. Kong, W.; Wu, K.; Zhang, S.; Liu, Y. GraphSparseNet: A Novel Method for Large Scale Trafffic Flow Prediction. *arXiv* **2025**, arXiv:2502.19823. [CrossRef]
29. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
30. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
31. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
32. Ramachandran, P.; Zoph, B.; Le, Q. Swish: A Self-Gated Activation Function. *arXiv* **2017**, arXiv:1710.05941.
33. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the 34th International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 933–941.
34. Wen, R.; Torkkola, K.; Narayanaswamy, B.; Madeka, D. A multi-horizon quantile recurrent forecaster. *arXiv* **2017**, arXiv:1711.11053.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.