SUMO User Conference 2025

Conference paper

https://doi.org/10.52825/scp.v6i.2628



Published: 15 Jul. 2025

# Spatio-Temporal AI Modeling for Urban Traffic Calibration: A SUMO-Based Approach

Pablo Manglano-Redondo[1,*], Alvaro Paricio-Garcia[1], and Miguel A. Lopez-Carmona[1]

[1]Universidad de Alcala, Alcala de Henares, 28805, Madrid, Spain

*Correspondence: Pablo Manglano-Redondo, pablo.manglano@edu.uah.es

**Abstract.** Urban traffic management is a critical challenge in modern cities, necessitating innovative solutions to optimize traffic flow and reduce congestion. This research presents the development of an AI engine leveraging spatio-temporal learning techniques for urban traffic calibration. The proposed methodology leverages digital twin scenarios driven by microscopic simulations, which capture detailed vehicle behaviors—including interactions, lane changes, and driver dynamics to provide granular insights into urban traffic patterns. At the core of the AI engine is the Dynamic Spatio-Temporal Graph Attention Network (DSTGAT), a hybrid model that combines multi-head Graph Attention Networks (GATv2) with Long Short-Term Memory (LSTM) networks. DSTGAT exploits the joint spatio-temporal relationships inherent in traffic data by processing sequential snapshots of urban traffic, where each snapshot is represented as a graph with nodes indicating urban zones and edges carrying continuous flow values. The GATv2 layers, enhanced with residual connections and batch normalization, extract robust spatial embeddings, while the LSTM aggregates these embeddings over time to capture dynamic patterns and predict future traffic flows in real-time. The AI engine incorporates an iterative feedback loop that continuously refines the OD demand using synthetic scenarios, improving estimation accuracy across diverse urban environments. Preliminary results show that the DSTGAT-based framework lowers OD-estimation error on simulated data, suggesting its usefulness as an input to downstream traffic-management strategies.

**Keywords:** Traffic Calibration, Urban Mobility, Transport Policies, DSTGAT, SUMO

## 1. Introduction

Rapid urbanization and the complexity of modern transportation networks have intensified the challenges in traffic management. Modern cities need optimized traffic flow, real-time calibration and the integration of Artificial Intelligence (AI) into traffic systems offers a promising solution.

One of the biggest challenges in deploying deep learning techniques is obtaining proper datasets to train, validate, and test the models. Nevertheless, these models are linked to specific urban environments and the AI designed in one city can hardly be applied to a different one due to the spatial and temporal dependencies. In this

study, we demonstrate how a traffic simulation environment can be used as a digital twin to artificially create datasets, which can then be effectively used to train spatio-temporal models specific to urban traffic patterns. These virtual datasets can bridge the gap between cities with different traffic behaviors, providing a scalable solution for overcoming the challenges imposed by spatial and temporal dependencies. By generating realistic traffic conditions similar to those encountered in diverse environments, the digital twin allows for model transferability and rapid adaptation of AI systems to new cities or changing traffic scenarios, without relying on the availability of large-scale real-world datasets. This approach opens the door for more flexible, data-driven traffic management solutions.

This research introduces an AI engine that uses spatio-temporal learning techniques in microscopic simulation data that capture individual vehicle behaviors, including movements, interactions, and lane changes, for urban traffic calibration. Our approach is built around the Dynamic Spatio-Temporal Graph Attention Network (DSTGAT), which combines multi-head Graph Attention Networks (GATv2) and Long Short-Term Memory (LSTM) networks to merge spatial and temporal information. Each traffic snapshot is modeled as a graph, where nodes represent Traffic Assignment Zones (TAZs) and edges carry continuous flow values. GATv2 layers, enhanced with residual connections and batch normalization, extract robust spatial embeddings that are then aggregated over time by an LSTM to predict future traffic states.

Our AI engine accurately calibrates and predicts traffic flows, a crucial step for effective traffic management. Synthetic scenario testing, varying origin-destination matrices and traffic densities, validates the method's robustness and adaptability across different urban conditions.

Key contributions of this work include:

- A robust pipeline that integrates SUMO with an AI calibration engine, ensuring realistic simulation inputs and dynamic demand adjustments.
- A methodology for pre-training DSTGAT on historical datasets and fine-tuning it online with real-time data, enabling continuous calibration and rapid adaptation to evolving traffic conditions.
- A comprehensive validation framework that yields results closely aligned with real-world traffic behavior.

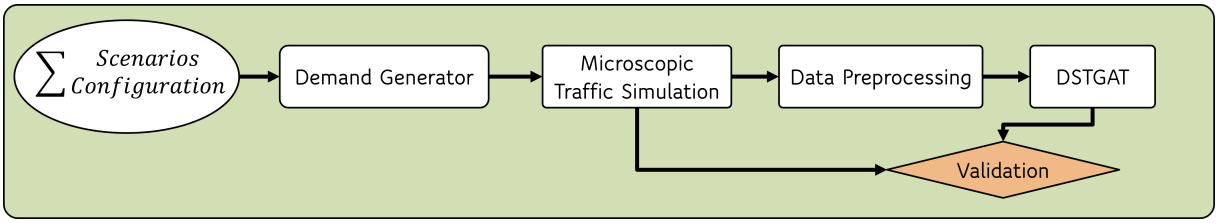


***Figure 1.*** *Overview of traffic calibration workflow.*

Figure 1 illustrates the overall workflow, from the generation of simulation data to the validation and calibration of AI-driven traffic.

The remainder of this paper is structured as follows. Section 2 reviews the relevant literature and theoretical foundations. Section 3 details the design and integration of the AI engine within the simulation framework. Section 4 describes the experimental setup and data analysis, while Section 5 discusses the findings and their implications for urban traffic management. Finally, Section 6 presents the conclusions and outlines the potential future research directions.

## 2. Related Works

The calibration of traffic models in large urban environments is a fundamental research problem, enabling applications ranging from real-time traffic management to long-term transportation planning. Urban traffic flow calibration in large-scale environments is a computationally demanding and data-intensive problem. Over the past decades, various methods have been developed to estimate traffic demand and calibrate simulation models, broadly categorized into analytical methods, simulation-based methods, dimensionality reduction techniques, hybrid approaches leveraging problem structure, machine learning techniques, and more recently deep-learning based methods for spatio-temporal analysis. The effectiveness of these methods depends on factors such as scalability, computational efficiency, and data availability. Dimensionality reduction and hybrid problem-structured algorithms have improved scalability, but challenges remain in achieving efficient and accurate calibration for large-scale transportation networks.

Analytical methods, commonly referred to as demand estimation, rely on mathematical formulations to infer travel demand based on observed traffic counts and historical data. They provide rapid estimation but struggle with complex traffic dynamics. Foundational work in this field includes [1], [2], [3]. These methods are computationally efficient and provide optimal solutions under specific assumptions about network equilibrium and route choice behavior. Other analytical methods include temporal series modeling with ARIMA [4], However, their applicability to large-scale networks is limited due to the need for simplified traffic flow models, which may not capture complex congestion dynamics and non-recurrent traffic conditions accurately.

Simulation-based methods represent a more flexible alternative, where stochastic traffic simulation models are iteratively adjusted until they match observed data. They offer greater flexibility at the cost of high computational costs. These approaches generally employ stochastic optimization (SO) algorithms such as Simultaneous Perturbation Stochastic Approximation (SPSA) [5], [6], [7], [8] and genetic algorithms (GA) [9], [10], [11]. however, they are computationally expensive for large-scale traffic networks. GA and other derivative-free algorithms [12] have been widely used in traffic calibration due to their ability to explore complex search spaces. However, GA is computationally intensive, especially for high-dimensional problems, as it requires extensive function evaluations to achieve convergence.

Given the challenges associated with high-dimensional calibration problems, researchers have explored dimensionality reduction techniques to improve scalability. Methods such as sensitivity analysis [13], [14] and Principal Component Analysis (PCA) [15], [16] help reduce the number of calibration parameters by identifying critical variables that influence traffic dynamics. These methods improve computational efficiency by limiting the search space of the optimization process. However, they rely on assumptions about correlations among variables, which may lead to information loss or suboptimal solutions in complex urban scenarios with highly nonlinear dynamics. Exploiting the underlying structure of traffic calibration problems enhances computational efficiency. These approaches integrate analytical approximations of system behavior into the calibration algorithm. First-order derivative approximations within the calibration process significantly reduce computational requirements[17]. While these approaches improve both efficiency and scalability, their success depends on the availability of accurate analytical formulations of traffic dynamics, which may not always be feasible for highly congested and heterogeneous urban networks.

Machine learning models have been widely used in traffic prediction due to their ability to handle stochastic processes and non-linear traffic patterns. The K-Nearest Neighbors (KNN) method [18] identifies similar historical traffic conditions to predict short-term travel times, offering simplicity but struggling with sparse datasets. Support Vector Machines (SVMs) map non-linear traffic data into high-dimensional spaces for classification, providing robust performance with limited data, but requiring extensive parameter tuning and computational resources.

Probabilistic models such as Bayesian Networks [19] and Hidden Markov Models (HMMs) [20] have been applied to capture uncertainty in OD demand and short-term traffic variations. While Bayesian methods excel in managing missing data, they rely on strong prior knowledge, which may not always be available. Similarly, HMMs effectively model sequential dependencies but face scalability challenges when applied to large transportation networks. The primary limitation of machine learning approaches is their dependency on large labeled datasets, which are often difficult to obtain.

Deep learning has emerged as a powerful alternative, capable of capturing complex spatiotemporal dependencies in traffic data. Long Short-Term Memory (LSTM) networks [21] improve urban travel time prediction by modeling long-range dependencies in sequential data. However, they fail to capture spatial correlations between different traffic zones. To address this, hybrid CNN-LSTM models [22], [23] integrate convolutional neural networks (CNNs) to extract spatial patterns before passing the data to LSTMs for temporal forecasting. This method achieves higher accuracy than standalone CNN or RNN models but remains limited to grid-based spatial data, making it unsuitable for road networks with irregular structures.

To overcome grid limitations, researchers have applied Graph Convolutional Networks (GCNs), which process traffic networks as graph-structured data rather than regular grids. Spectral-based GCNs [24] effectively model global dependencies in transportation networks but require high computational resources. Spatial-based GCNs [25] reduce computational complexity by aggregating local node information, improving scalability. However, GCN-based models still suffer from decreasing accuracy in multi-step forecasting, limiting their reliability for long-term predictions.

Attention mechanisms have been used recently since they enhance prediction models by dynamically selecting relevant information. Self-attention layers [26] have shown success in node classification for graph structures, while multi-level attention networks [27] improve correlation analysis in sensor data. Despite their potential, attention-based models increase computational costs due to the need to train separate models for each time series, making them impractical for real-time applications. Multi-step traffic flow prediction is improved by integrating graph convolutional networks (GCN) for spatial modeling and Attention Encoder Networks (AEN) for temporal correlation considering external factors such as daytime, weekdays, or accidents [28], [29].

Graph Convolutional Networks (GCN) and Gated Recurrent Units (GRU) have been used to used to extract spatio-temporal correlations in urban road networks [30] using traffic checkpoint data for broader coverage and higher reliability. Unlike GPS-based methods, a GCN-GRU model first predicts vehicle trajectories by learning spatial dependencies, then forecasts checkpoint-level traffic flow, achieving higher accuracy than conventional methods.

Dynamic spatio-temporal graph attention networks (DSTGAT) leverage GAT and GRU to capture spatial and temporal dependencies for traffic flow prediction [31]. DSTGAT integrates periodic patterns (adjacent, daily, weekly) and enhances non-adjacent

correlations using Pearson coefficients, while applying as well the attention mechanisms.

## 3. Design of the Dynamic Urban Traffic Calibration Framework

This section presents a dynamic urban traffic calibration framework that integrates three key components: a robust traffic pipeline for preprocessing and simulating traffic data, a demand generation module that creates realistic traffic scenarios, and an AI calibration engine that continuously optimizes control parameters based on live conditions.

### 3.1 Traffic Pipeline

The Traffic Pipeline is the backbone of our simulation framework, orchestrating the entire process from the initialization of the SUMO simulator to the real-time collection and storage of traffic data (see Figure 2). The pipeline begins by launching SUMO with a comprehensive set of customizable options—including configuration for GUI mode, mesoscopic simulation, and selective recording of emissions and vehicle statistics. During this initialization phase, essential configuration files and scripts are verified, the appropriate command-line arguments are constructed, and the traci API is used to start the simulation. Once SUMO is running, the system caches the network's edge information (excluding internal edges) to enable efficient subscription to critical traffic variables, such as vehicle counts, speeds, and emissions.

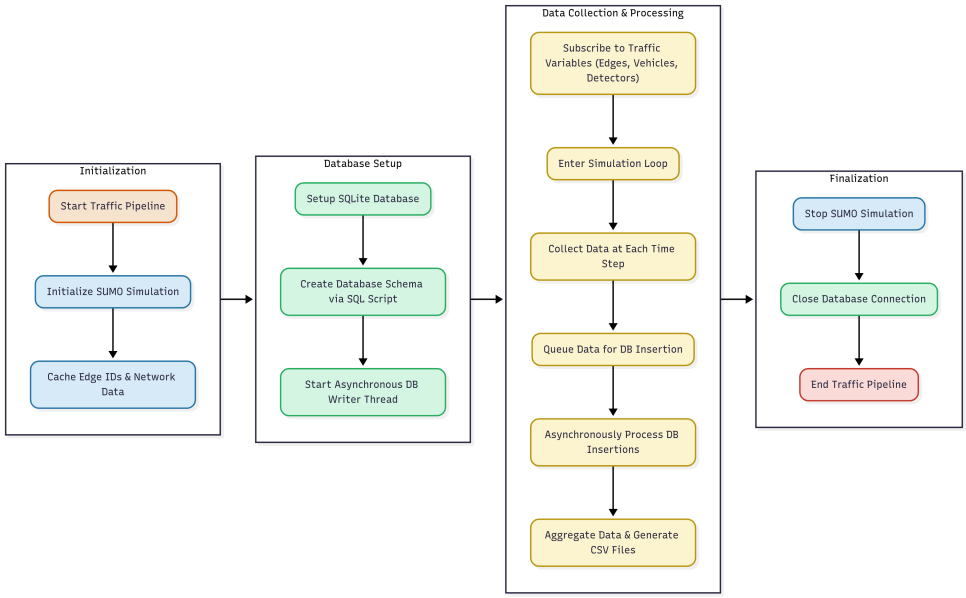


***Figure 2.*** *Traffic pipeline.*

In parallel, the pipeline sets up a SQLite database, which can operate in-memory or on-disk based on user specifications, and executes a predefined SQL script to create the necessary schema for storing various metrics, including edge values, vehicle details, and aggregated flows. Thread locks and an asynchronous writing mechanism ensure thread-safe interactions with the database, allowing the simulation loop to proceed without I/O bottlenecks. Concurrently, the pipeline subscribes to a suite of traffic parameters via traci, with configurable flags controlling the collection of data for edges, vehicles, and detectors. This setup captures a wide range of metrics at each simulation time step, from basic counts and speeds to detailed emission profiles and travel times.

This modular approach enables us to simulate a medium-sized city like Guadalajara with 40,000 vehicles over a 24-hour period and efficiently collect camera snapshots and edge information in just four minutes on a regular desktop PC.

### 3.2 Demand Generation

Traffic simulation input begins with an Origin-Destination (OD) matrix that represents baseline travel demand. This matrix is enriched with additional inputs—such as Traffic Analysis Zone (TAZ) definitions, detector placements, and vehicle heterogeneity—to more accurately reflect real-world conditions.

***Table 1.*** *Fleet Distribution.*

| Fleet Type | Distribution |
|---|---|
| Bus | 0.018 |
| Electric | 0.029 |
| Emergency | 0.008 |
| Motorcycle | 0.112 |
| Passenger | 0.707 |
| Truck | 0.126 |

Based on Spain's official dataset [32], the fleet demand was calibrated to reflect typical urban traffic. Table 1 presents the distribution across vehicle types, ensuring the simulation captures a realistic mix of traffic behaviors.

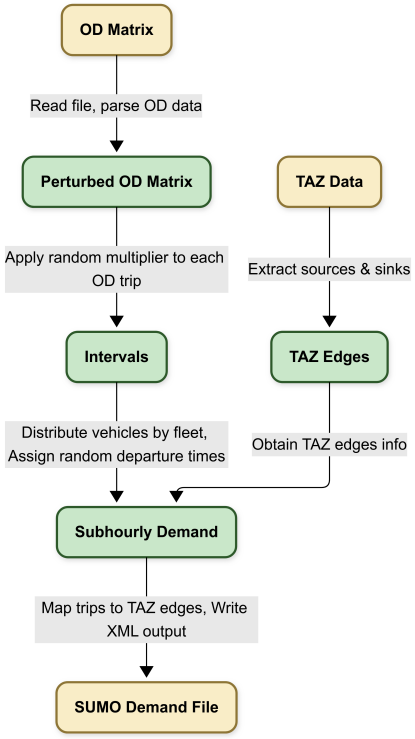


***Figure 3.*** *Demand generation diagram.*

Figure 3 provides an overview of the demand generation process for the traffic simulation. The workflow begins by reading the OD matrix from an input file and parsing its data. This matrix is then perturbed by applying random multipliers to each OD trip, introducing realistic demand variability. In parallel, TAZ data is processed to extract

sources and sinks, leading to the formulation of TAZ edges, which define the spatial structure of the simulation network.

Subsequently, the perturbed OD matrix is segmented into specific time intervals. These intervals are used to distribute vehicles according to the calibrated fleet distribution and assign random departure times. The combined spatial and temporal information results in subhourly demand profiles. Finally, the processed data is mapped onto the TAZ edges and formatted into an XML file, which serves as the input for the SUMO simulation environment.

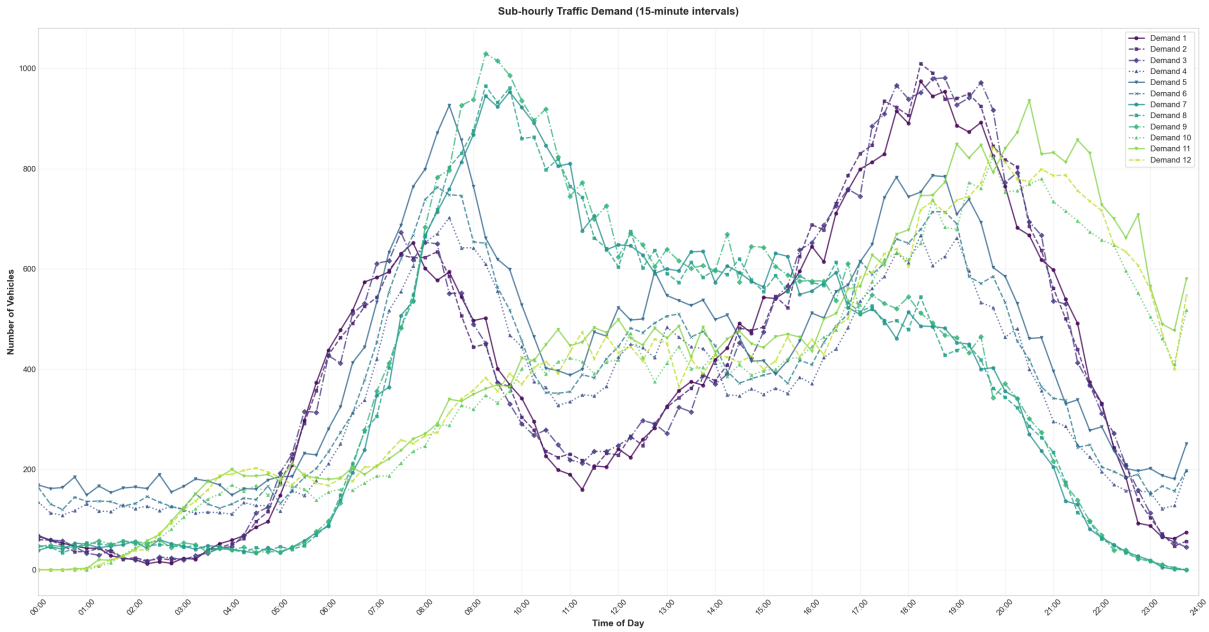


***Figure 4.*** *Historical demand.*

Figure 4 illustrates the sub-hourly traffic demand profiles produced by the workflow described above. Each line represents a distinct demand scenario, reflecting the variability introduced by random perturbations to the baseline OD matrix and diverse demand configurations. These demand profiles serve as the training input for the DSTGAT model, enabling it to learn the spatial and temporal patterns underlying urban travel behavior.

### 3.3 AI-Based Traffic Calibration Engine

At the core of the framework is the DSTGAT model, which fuses spatial and temporal learning to estimate time-varying origin–destination (OD) flows that can later inform traffic-control decisions.

The model processes traffic data represented as a sequence of graph snapshots. Each snapshot encodes the urban network at a specific time step with nodes representing traffic zones and edges carrying traffic flow values. Node features are one-hot encoded, ensuring that each zone is uniquely identified. Spatial dependencies are modeled via multiple layers of a modified graph attention mechanism. In each snapshot, the ResidualGATv2Conv layer applies a GATv2 operation with a residual connection and batch normalization, allowing the model to capture intricate relationships between zones while stabilizing the learning process.

To predict traffic flow between zones, the model combines the embeddings of node pairs connected by an edge. A dedicated multi-layer perceptron (MLP) processes the

concatenated embeddings of each node pair to output a continuous prediction of traffic intensity for that edge. This design allows the DSTGAT to focus specifically on edge-level dynamics, crucial for calibrating traffic control parameters.

The engine utilizes a custom *TrafficWindowDataset* that constructs training snapshots by extracting sliding windows of graph snapshots with a configurable window size and a prediction horizon. Historical data, preprocessed from raw simulation outputs, is used to pretrain the model, capturing long-term traffic patterns. In parallel, a real-time online prediction module ingests current snapshots, and if available, loads updated online weights. Otherwise, it defaults to the pretrained model. Fine-tuning occurs periodically, whereby recent data is used to adapt the model continuously to evolving traffic conditions.

The model is trained using a Mean Squared Error (MSE) loss, enhanced with optional L1 and L2 regularization to prevent overfitting. The training process leverages modern optimization techniques, including adaptive learning rates and mixed-precision training, to ensure efficient convergence. Predictions are validated against live data, with metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) guiding iterative model refinement.

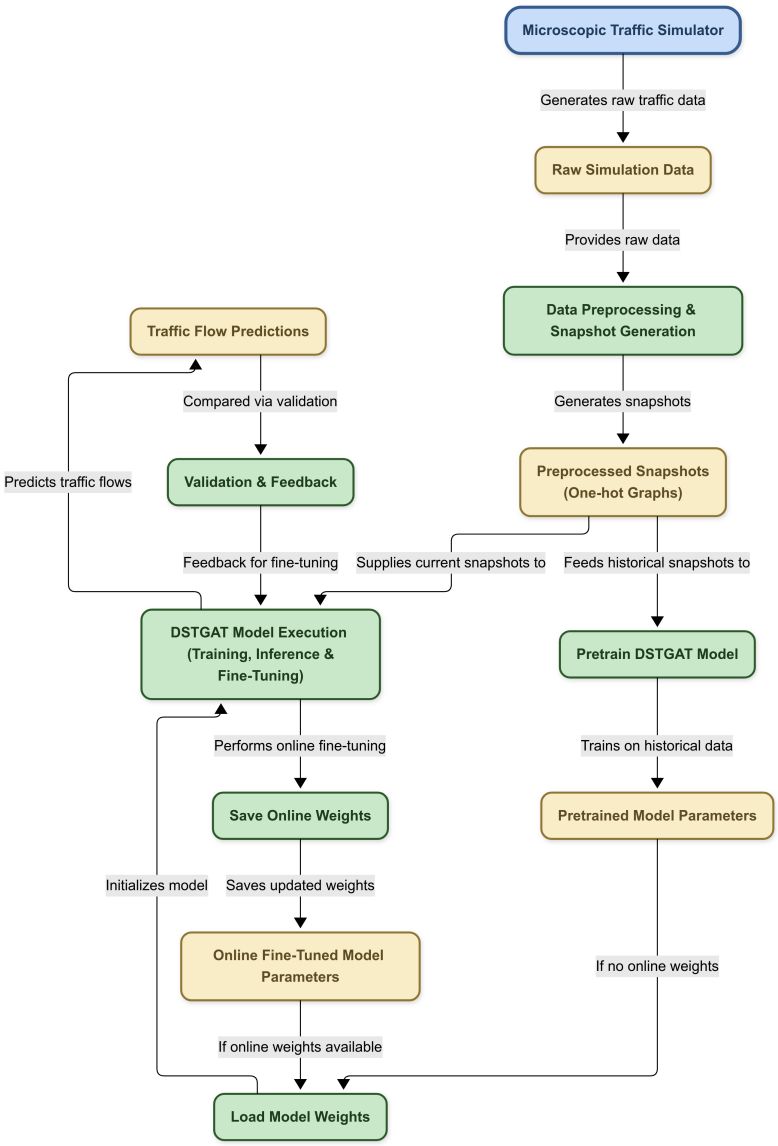


***Figure 5.*** *Workflow diagram.*

Figure 5 outlines the end-to-end workflow of the calibration engine. The process begins with SUMO as a microscopic traffic simulator that generates raw traffic data. This raw data is preprocessed and converted into one-hot encoded graph snapshots, representing the spatial configuration of the urban network at specific time intervals.

These snapshots are employed in two complementary pathways. First, a subset is used for historical pretraining of the DSTGAT model, generating robust parameters that capture long-term traffic patterns. Simultaneously, current snapshots are input for real-time traffic flow prediction. In this case, if updated online weights from previous iterations are available, the model loads these; otherwise, it defaults to the pretrained parameters. The resulting predictions are then validated against live data, forming the basis for iterative fine-tuning.

Validation feedback is integral to this process. The predicted traffic flows are compared to observed values, and discrepancies are used to iteratively fine-tune the model parameters. The updated weights are saved and subsequently integrated into the next prediction cycle, forming a closed-loop system that continually adapts to dynamic urban traffic conditions.

This integrated approach—leveraging spatial attention to capture inter-zone dependencies and LSTM based temporal aggregation enables DSTGAT to learn complex spatio-temporal relationships, providing a robust and adaptive calibration mechanism that enhances urban traffic management.

## 4. Case Study

Guadalajara has been studied in previous research[33], particularly in the context of urban mobility and Low Emission Zones. Given its well-documented road network, diverse traffic flow patterns, and the availability of detailed mobility data, we have selected Guadalajara once again as the testbed for our AI-based traffic calibration framework. This section details the TAZs within the city and their unique characteristics, forming the foundation for our simulation experiments.

### 4.1 Guadalajara Road Network

Guadalajara's urban area is divided into multiple zones, each with distinct mobility patterns and functional roles, as illustrated in Figure 6. This specific grouping is purely illustrative and does not affect the DSTGAT's graph construction or performance, which we categorize as follows:

- **Low Emission Zone (LEZ) (Green Area):** A central district with stricter environmental regulations that limit high-emission vehicles. Traffic dynamics here are influenced by emission control measures and pedestrian-friendly policies.
- **Residential Zones (Blue Areas):** High-density neighborhoods with typical commuting patterns, dominated by local trips, school runs, and connections to commercial or industrial areas.
- **Industrial Zones (Gray Areas):** Characterized by heavier freight traffic and peak-hour congestion tied to industrial shift schedules.
- **Major Entry/Exit Points (Red Nodes):** Highway interchanges and key junctions that regulate traffic flow into and out of the city, serving as boundary conditions in our simulation.

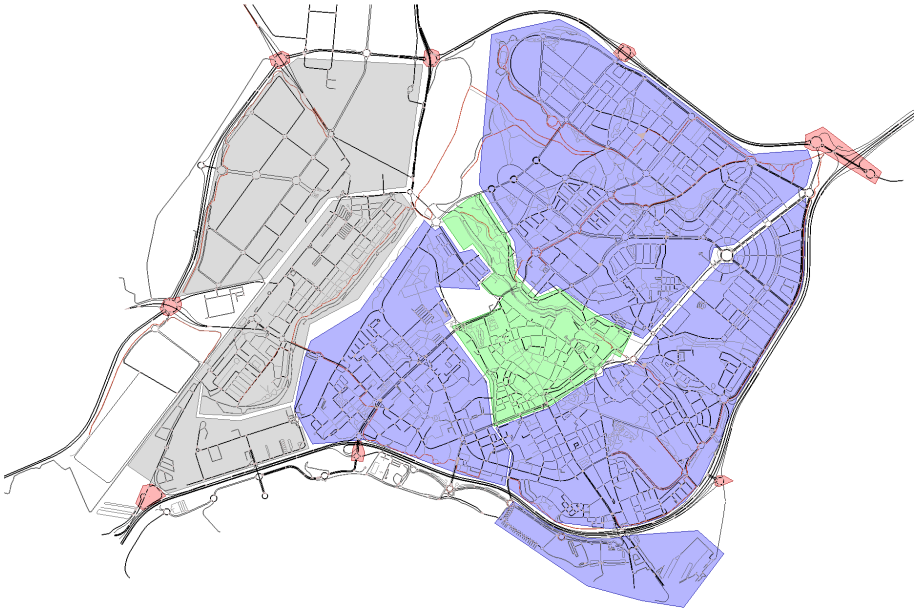

***Figure 6.*** *Guadalajara TAZ configuration.*

Each TAZ serves as a node in our AI-based calibration framework, where traffic flows are continuously adjusted using the DSTGAT model. By including a variety of zone types—ranging from residential to industrial corridors—we capture the diverse traffic behaviors that emerge under different urban functions.

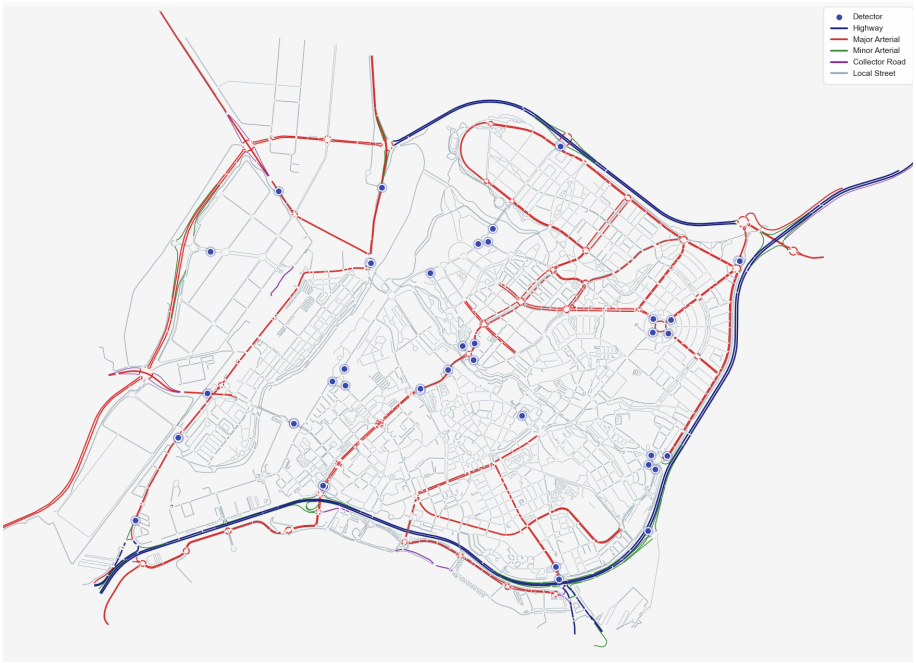


***Figure 7.*** *Network detectors.*

Figure 7 shows the distribution of traffic detectors across highways, major arterials, minor arterials, collector roads, and local streets. These detectors provide real-time flow, speed, and occupancy data, forming the backbone of the model's online calibration. Their strategic placement ensures robust coverage of key network segments and allows the DSTGAT to adapt dynamically to changing traffic conditions.

## 5. Experimental Results

This section evaluates the performance of the proposed AI-based calibration framework on a full-day traffic simulation of Guadalajara. The analysis focuses on comparing predicted and observed traffic flows, quantifying errors at both global and zone-specific levels, and illustrating the model's capacity to capture temporal patterns.

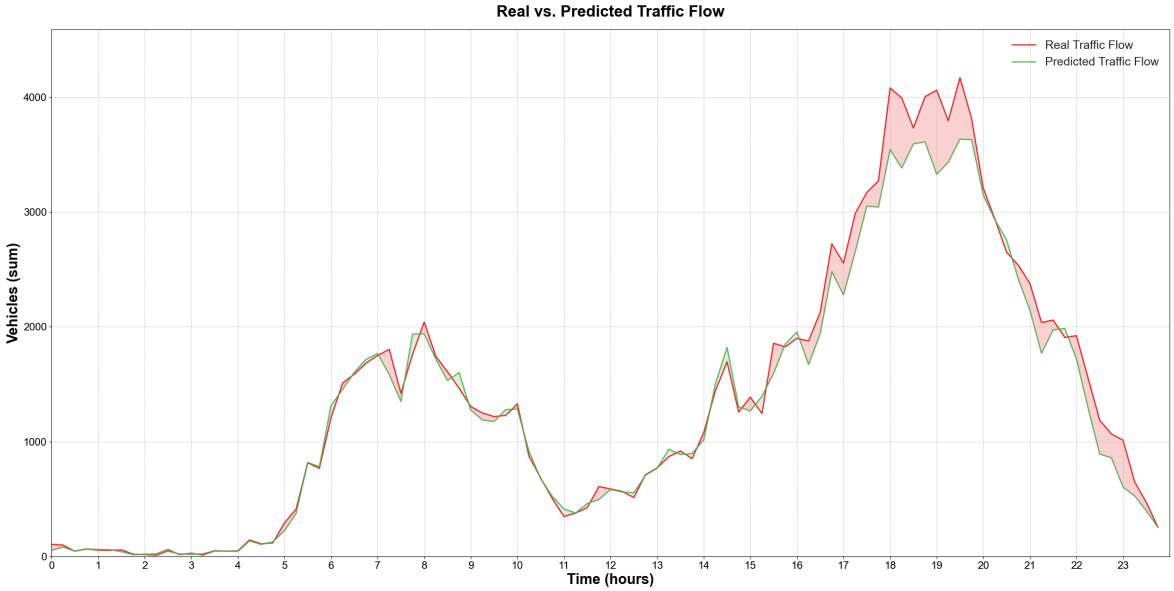


***Figure 8.*** *Predicted traffic flow comparison.*

Figure 8 shows the aggregated comparison between real and predicted traffic flow across the entire urban network. Overall, the model tracks the diurnal traffic pattern closely, accurately capturing both morning and evening peaks. The small discrepancies at certain peak hours highlight the complexity of urban dynamics but remain within acceptable bounds for real-time traffic calibration. The framework achieves a mean absolute error (MAE) of 5.27, showing an error of 5.27 vehicles every 15 minutes, and an $R^2$ of 0.69, indicating strong predictive capability for most time intervals.

Figure 9 zooms in on a single TAZ *aguasvivas* and compares inbound and outbound flows. The model aligns closely with observed values, capturing both the morning and afternoon surges. Minor deviations occur at the highest peak, reflecting localized fluctuations in demand. These results confirm that the DSTGAT architecture, with its combined spatio-temporal attention and online fine-tuning, adapts well to both network-wide and zone-level variations.

Lastly, Figure 10 offers a more granular view, displaying the relative error (%) for each origin–destination TAZ pair. Cooler shades (blue) represent underestimating traffic flows, whereas warmer shades (red) indicate overestimating flows deviations. In general, most cells remain in the low-error range, demonstrating consistent performance across the network. A few TAZ pairs show moderate error spikes, often correlated with complex routes or sudden changes in demand.

In summary, the proposed framework demonstrates robust predictive accuracy for a medium-sized urban environment. By maintaining low errors across diverse TAZ pairs and time periods, the system proves suitable for real-time traffic calibration and management tasks. The detailed error analyses also highlight opportunities for further refinement, such as enhanced detector coverage or tailored modeling of high-variance

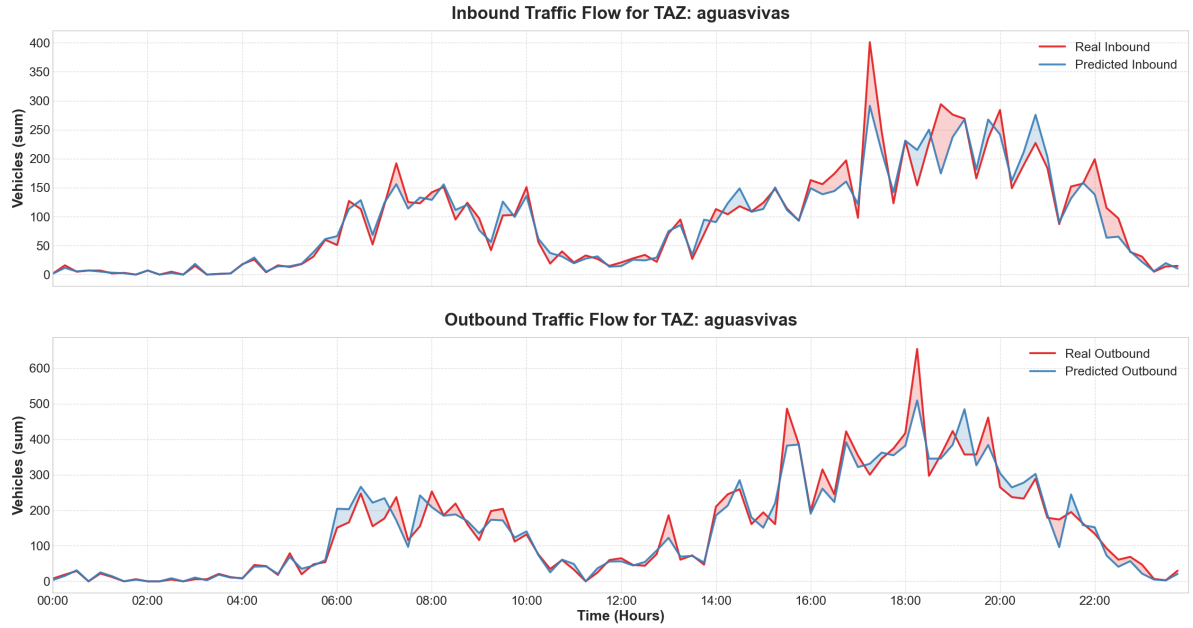


***Figure 9.*** *TAZ traffic flow.*

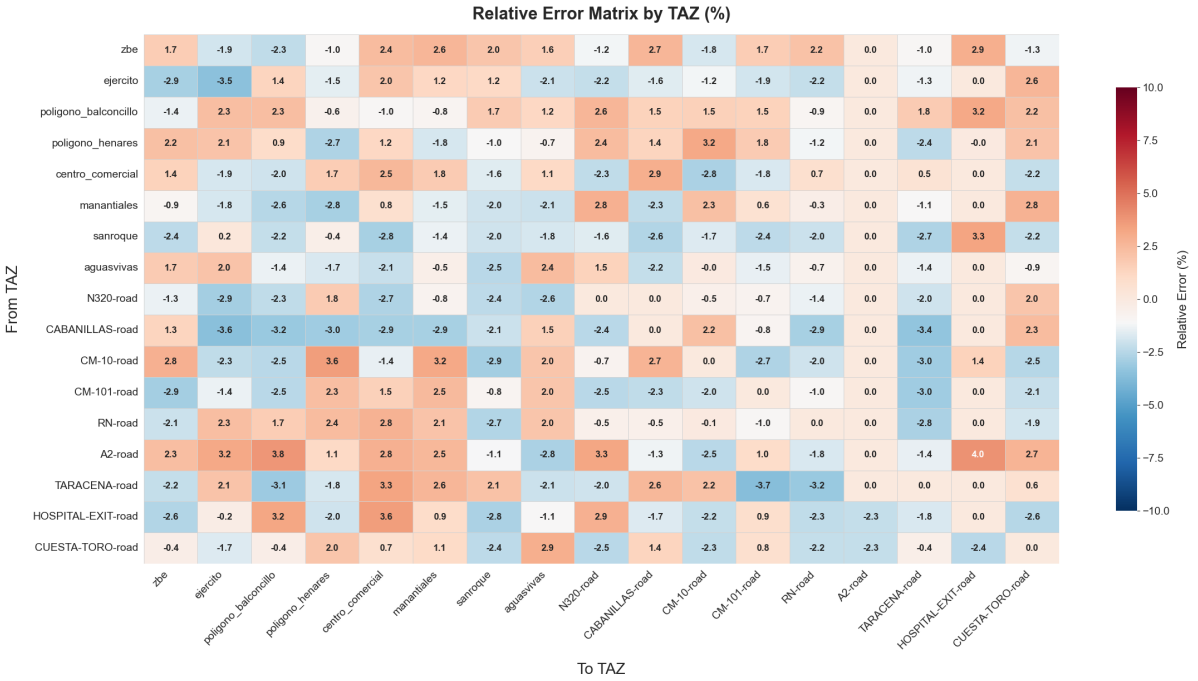


***Figure 10.*** *Relative error matrix.*

routes. Overall, these results underscore the efficacy of integrating advanced AI methods with microscopic simulation for dynamic and data-driven urban traffic calibration.

## 5.1 Data Points and Scalability of the Approach

This section quantifies two key aspects of our method:

- The total number of *training snapshots* (data points) generated from our synthetic scenarios, to ensure sufficiency of data for model learning.

- The *computational complexity* of training and inference, expressed via a density-adjusted graph size and overall processing cost, to assess scalability to larger urban networks.

#### 5.1.1 Training Snapshots and Data Points

To compute the total number of data points used for training, we apply a sliding time-window of length over each full-day simulation of snapshots, with a prediction horizon of snapshots.

$$M = S\left(T - W - H + 1\right) \quad \text{(ec 1.1)}$$

$M$ Total number of data points (snapshots) generated.
$S$ Number of synthetic demand scenarios.
$T$ Number of time steps per day.
$W$ Sliding window size.
$H$ Prediction horizon.
$+1$ Inclusive count to ensure both the very first and very last valid window start are counted.

#### 5.1.2 Computational Complexity and Density Factor

Each traffic snapshot is a graph with $N_{\mathrm{TAZ}}$ nodes.

$$E_{\max} = N_{\mathrm{TAZ}}\left(N_{\mathrm{TAZ}} - 1\right) \quad \text{(ec 1.2)}$$

$E_{\max}$ Maximum possible edges in a fully connected graph.
$N_{\mathrm{TAZ}}$ Total number of Traffic Analysis Zones.

To model realistic network sparsity, we introduce a density factor $\theta$ ($0 < \theta \leq 1$):

$$E = \theta\left[N_{\mathrm{TAZ}}\left(N_{\mathrm{TAZ}} - 1\right)\right] \quad \text{(ec 1.3)}$$

$E$ Actual number of edges (observed zone-to-zone connections).
$\theta$ Density factor: fraction of possible edges present.

Equivalently, the density factor can be computed as:

$$\theta = \frac{E}{E_{\max}} \quad \text{(ec 1.4)}$$

$\theta$ Density factor (0–1).
$E$ Actual edges counted.
$E_{\max}$ Maximum possible edges.

#### 5.1.3 Processing Cost Estimation

The total processing cost scales with the number of scenarios, time steps, and edges:

$$C \propto S \times T \times E = S \times T \times \theta \times N_{\mathrm{TAZ}} \left(N_{\mathrm{TAZ}} - 1\right) \quad \text{(ec 1.5)}$$

$C$ Relative processing cost metric.
$S$ Number of scenarios.
$T$ Time steps per scenario.
$E$ Number of edges per snapshot.
$\theta$ Density factor.
$N_{\mathrm{TAZ}}$ Number of zones.

#### 5.1.4 Example: Guadalajara Scenario

First, we compute the number of training snapshots:

$$M = 12 \times \big(1440 - 10 - 5 + 1\big) = 12 \times 1426 = 17\,112 \quad \text{snapshots}.$$

Next, the graph-edge counts:

$$E_{\max} = 17 \times (17 - 1) = 17 \times 16 = 272,$$
$$E = \theta \times E_{\max} = 0.0846 \times 272 \approx 23 \quad \text{edges}.$$

Finally, the relative processing cost:

$$C \propto S \times T \times E = 12 \times 1440 \times 23 = 397\,440 \quad (\text{edge-evaluations per day}).$$

Generating 17,112 training snapshots from just 12 daily simulations already yields a substantial dataset for DSTGAT training. The graph size, 17 nodes and only 23 edges on average, keeps per-snapshot computations light. Over one full day (1,440 snapshots), the model performs roughly 397,440 edge operations. This demonstrates that for a medium-sized city like Guadalajara, both data volume and computational load are easily handled by standard hardware, validating the practical scalability of our approach.

## 6. Results and Contributions

This research presents a methodology that couples microscopic simulation with spatio temporal deep learning to accelerate the calibration of OD demand for medium-sized cities. Our framework integrates a traffic-simulation pipeline, realistic demand generation, and the DSTGAT model to predict dynamic OD flows in real-time. Experimental results show that our model achieves a MAE of 5.27 vehicles per 15 minutes and an $R^2$ value of 0.69, demonstrating both accuracy and consistency in traffic flow predictions. Considering that the simulation involves 40,000 vehicles, this error represents a very small percentage, underscoring the model's precision.

A key contribution of this study is the capability to simulate medium-sized urban areas such as Guadalajara with 40,000 vehicles over a 24-hour period in just a few minutes. This accelerated simulation process generates high-fidelity synthetic datasets that are essential for training spatio-temporal models tailored to the dynamic nature of urban traffic. In parallel, our advanced AI calibration engine, powered by DSTGAT, effectively fuses spatial and temporal information by combining GATv2 and LSTM layers,

capturing complex inter-zone dependencies and adapting to evolving traffic conditions via online fine-tuning. The framework's modular and scalable design further allows for the flexible incorporation of diverse urban scenarios and additional data sources, ensuring its adaptability to various urban environments. Real-time predictive capabilities, achieved through integrated data processing, asynchronous database management, and API connectivity, ensure continuous monitoring and calibration—a critical requirement for dynamic urban mobility management.

The broader implications of our findings extend beyond improved traffic flow and congestion reduction. By enabling rapid and scalable simulation of large urban environments, our framework lays a robust foundation for future urban planning initiatives and policy-making aimed at enhancing mobility, reducing travel times, and lowering vehicular emissions.

### 6.1 Future Research Lines

In future research, we aim to develop an AI system capable of automatically identifying critical zones within urban networks to optimize the placement of camera detectors. Strategic positioning of these sensors is essential for calibration systems, as it provides the model with comprehensive insights and representative patterns, ultimately enhancing its predictive accuracy.

Furthermore, we will scale the framework to truly metropolitan networks by partitioning the city into hierarchical subgraphs, each running its own DSTGAT instance and exchanging boundary-condition summaries via a lightweight edge-computing layer.

## Data Availability Statement

All available data are included in the paper.

## Author Contributions

Pablo Manglano-Redondo contributed to the conceptualization of the study, developed the methodology, conducted the investigation, validated the results, wrote the original draft, and created the visualizations. Alvaro Paricio-Garcia contributed to the conceptualization of the study, developed the methodology, validated the results, reviewed and edited the writing, provided supervision and managed the project administration. Miguel A. Lopez-Carmona contributed to the conceptualization of the study, validated the results, reviewed and edited the writing, provided supervision, and managed project administration.

## Competing Interests

The authors declare that they have no competing interests.

## Funding

We acknowledge the Catedra MasMovil for Advanced Network Engineering and Digital Services (MANEDS) at Universidad de Alcala (UAH) for the financial support for the research.

## References


[1] E. Cascetta and S. Nguyen, "A unified framework for estimating or updating origin/destination matrices from traffic counts," *Transportation Research Part B: Methodological*, vol. 22, no. 6, pp. 437–455, Dec. 1988, ISSN: 0191-2615. DOI: 10.1016/0191-2615(88)90024-0.

[2] E. Cascetta, D. Inaudi, and G. Marquis, "Dynamic Estimators of Origin-Destination Matrices Using Traffic Counts," *Transportation Science*, vol. 27, no. 4, pp. 363–373, Nov. 1993, ISSN: 0041-1655. DOI: 10.1287/trsc.27.4.363. Accessed: Feb. 26, 2025.

[3] H. Yang, T. Sasaki, Y. Iida, and Y. Asakura, "Estimation of origin-destination matrices from link traffic counts on congested networks," *Transportation Research Part B: Methodological*, vol. 26, no. 6, pp. 417–434, Dec. 1992, ISSN: 0191-2615. DOI: 10.1016/0191-2615(92)90008-K.

[4] B. Williams and L. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003. DOI: 10.1061/(ASCE)0733-947X(2003)129:6(664).

[5] J. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, 1992. DOI: 10.1109/9.119632.

[6] R. Balakrishna, "Off-line calibration of Dynamic Traffic Assignment models /," Jan. 2008.

[7] V. Vaze, C. Antoniou, Y. Wen, and M. Ben-Akiva, "Calibration of dynamic traffic assignment models with point-to-point traffic surveillance," *Transportation Research Record*, vol. 2090, no. 1, pp. 1–9, 2009.

[8] E. Cipriani, M. Florian, M. Mahut, and M. Nigro, "A gradient approximation approach for adjusting temporal origin–destination matrices," *Transportation Research Part C: Emerging Technologies*, Emerging Theories in Traffic and Transportation and Methods for Transportation Planning and Operations, vol. 19, no. 2, pp. 270–282, Apr. 2011, ISSN: 0968-090X. DOI: 10.1016/j.trc.2010.05.013. Accessed: Feb. 26, 2025.

[9] H. Kim, S. Baek, and Y. Lim, "Origin-Destination Matrices Estimated with a Genetic Algorithm from Link Traffic Counts," *Transportation Research Record*, vol. 1771, no. 1, pp. 156–163, Jan. 2001, ISSN: 0361-1981. DOI: 10.3141/1771-20. Accessed: Feb. 26, 2025.

[10] A. Stathopoulos and T. Tsekeris, "Hybrid meta-heuristic algorithm for the simultaneous optimization of the o–d trip matrix estimation," *Computer-Aided Civil and Infrastructure Engineering*, vol. 19, 2004.

[11] L. Kattan and B. Abdulhai, "Noniterative Approach to Dynamic Traffic Origin–Destination Estimation with Parallel Evolutionary Algorithms," *Transportation Research Record*, vol. 1964, no. 1, pp. 201–210, Jan. 2006, ISSN: 0361-1981. DOI: 10.1177/0361198106196400122. Accessed: Feb. 26, 2025.

[12] W. Huyer and A. Neumaier, "SNOBFIT – stable noisy optimization by branch and fit," *ACM Trans. Math. Softw.*, vol. 35, no. 2, Jul. 2008, ISSN: 0098-3500. DOI: 10.1145/1377612.1377613.

[13] Q. Ge and M. Menendez, "An Efficient Sensitivity Analysis Approach for Computationally Expensive Microscopic Traffic Simulation Models," *International Journal of Transportation*, vol. 2, no. 2, pp. 49–64, Aug. 2014, ISSN: 22877940, 22877940. DOI: 10.14257/ijt.2014.2.2.04. Accessed: Feb. 26, 2025.

[14] B. Ciuffo and C. Lima Azevedo, "A sensitivity-analysis-based approach for the calibration of traffic simulation models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1298–1309, 2014. DOI: 10.1109/TITS.2014.2302674.

[15] T. Djukic et al., "Advanced traffic data for dynamic OD demand estimation: The state of the art and benchmark study," in *TRB 94th Annual Meeting Compendium of Papers*, 2015, pp. 1–16.

[16] A. A. Prakash, R. Seshadri, C. Antoniou, F. C. Pereira, and M. E. Ben-Akiva, "Reducing the Dimension of Online Calibration in Dynamic Traffic Assignment Systems," *Transportation Research Record*, vol. 2667, no. 1, pp. 96–107, Jan. 2017, ISSN: 0361-1981. DOI: 10.3141/2667-10. Accessed: Feb. 26, 2025.

[17] G. Flötteröd, M. Bierlaire, and K. Nagel, "Bayesian demand calibration for dynamic traffic simulations," *Transportation Science*, vol. 45, no. 4, pp. 541–561, 2011.

[18] P.-W. Lin and G.-L. Chang, "A generalized model and solution algorithm for estimation of the dynamic freeway origin–destination matrix," *Transportation Research Part B: Methodological*, vol. 41, no. 5, pp. 554–572, 2007.

[19] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006. DOI: 10.1109/TITS.2006.869623.

[20] Y. Qi and S. Ishak, "A Hidden Markov Model for short term prediction of traffic conditions on freeways," *Special Issue on Short-term Traffic Flow Forecasting*, vol. 43, pp. 95–111, Jun. 2014, ISSN: 0968-090X. DOI: 10.1016/j.trc.2014.02.007.

[21] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning," *Transportmetrica A Transport Science*, vol. 15, no. 2, pp. 1688–1711, Jan. 2019, ISSN: 2324-9935. DOI: 10.1080/23249935.2019.1637966.

[22] A. Kumar, D. Garg, and G. Sharma, "Three-Tier Survey of Deep Learning Based Traffic Prediction Schemes," in *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Mar. 2024, pp. 1–6. DOI: 10.1109/ICRITO61523.2024.10522180. Accessed: Feb. 19, 2025.

[23] B. Yu, H. Yin, and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Jul. 2018, pp. 3634–3640. DOI: 10.24963/ijcai.2018/505. arXiv: 1709.04875 [cs]. Accessed: Feb. 26, 2025.

[24] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, *Spectral Networks and Locally Connected Networks on Graphs*, May 2014. DOI: 10.48550/arXiv.1312.6203. arXiv: 1312.6203 [cs]. Accessed: Feb. 26, 2025.

[25] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 922–929, Jul. 2019, ISSN: 2374-3468. DOI: 10.1609/aaai.v33i01.3301922. Accessed: Feb. 26, 2025.

[26] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph Attention Networks*, Feb. 2018. DOI: 10.48550/arXiv.1710.10903. arXiv: 1710.10903 [stat]. Accessed: Feb. 26, 2025.

[27] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction," pp. 3428–3434, 2018. Accessed: Feb. 26, 2025.

[28] J. Ye, S. Xue, and A. Jiang, "Attention-based spatio-temporal graph convolutional network considering external factors for multi-step traffic flow prediction," *Digital Communications and Networks*, vol. 8, no. 3, pp. 343–350, Jun. 2022, ISSN: 2352-8648. DOI: 10.1016/j.dcan.2021.09.007. Accessed: Feb. 19, 2025.

[29] Y. Chen, L. Zheng, and W. Liu, "Spatio-Temporal Attention-based Graph Convolution Networks for Traffic Prediction," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2022, pp. 642–649. DOI: 10.1109/SMC53654.2022.9945522. Accessed: Feb. 19, 2025.

[30] D. Guan, N. Ren, K. Wang, Q. Wang, and H. Zhang, "Checkpoint data-driven GCN-GRU vehicle trajectory and traffic flow prediction," *Scientific Reports*, vol. 14, no. 1, p. 30 409, Dec. 2024, ISSN: 2045-2322. DOI: 10.1038/s41598-024-80563-3. Accessed: Feb. 19, 2025.

[31] Y. Chen, J. Huang, H. Xu, J. Guo, and L. Su, "Road traffic flow prediction based on dynamic spatiotemporal graph attention network," *Scientific Reports*, vol. 13, no. 1, p. 14 729, Sep. 2023, ISSN: 2045-2322. DOI: 10.1038/s41598-023-41932-6. Accessed: Feb. 19, 2025.

[32] *Conjunto de Datos OTLE: Parque nacional de vehículos por comunidad autónoma, provincia, tipo de vehículo y tipo de carburante*. Accessed: Feb. 26, 2025. [Online]. Available: https://apps.fomento.gob.es/BDOTLE/visorBDpop.aspx?i=396.

[33] A. Paricio, M. López-Carmona, and P. Manglano-Redondo, "Optimized Design of Low Emission Zones in SUMO: A Dual Focus on Emissions Reduction and Travel Time Improvement," *SUMO Conference Proceedings*, vol. 5, pp. 247–268, Jul. 2024. DOI: 10.52825/scp.v5i.1143.