Transportation Research Part C 180 (2025) 105360

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Machine learning surrogates for agent-based models in transportation policy analysis

Elena Salomé Natterer [a,*], Saini Rohan Rao [a,b], Alejandro Tejada Lapuerta [b,c], Roman Engelhardt [a], Sebastian Hörl [d], Klaus Bogenberger [a]

[a] Chair of Traffic Engineering and Control, Technical University of Munich, Germany, Munich, Germany
[b] TUM School of Computation, Information & Technology, Technical University of Munich, Garching, Germany
[c] Computational Health Center, Helmholtz Munich, Neuherberg, Germany
[d] Institut de Recherche Technologique SystemX, Palaiseau, France

ARTICLE INFO

Keywords:
Graph neural networks
Transformers
Agent-based simulations
Optimization-based transportation policies
Surrogate models
Street capacity reduction

ABSTRACT

Effective traffic policies are crucial for managing congestion and reducing emissions. Agent-based transportation models (ABMs) offer a detailed analysis of how these policies affect travel behaviour at a granular level. However, computational constraints limit the number of scenarios that can be tested with ABMs and therefore their ability to find optimal policy settings.

In this proof-of-concept study, we propose a machine learning (ML)-based surrogate model to efficiently explore this vast solution space. By combining Graph Neural Networks (GNNs) with the attention mechanism from Transformers, the model predicts the effects of traffic policies on the road network at the link level.

We implement our approach in a large-scale MATSim simulation of Paris, France, covering over 30,000 road segments and 10,000 simulations, applying a policy involving capacity reduction on main roads. The ML surrogate achieves an overall R^2 of 0.91; on primary roads where the policy applies, it reaches an R^2 of 0.98. This study shows that the combination of GNNs and Transformer architectures can effectively serve as a surrogate for complex agent-based transportation models with the potential to enable large-scale policy optimization, helping urban planners explore a broader range of interventions more efficiently.

1. Introduction

Cities around the world face growing challenges in terms of congestion and air pollution, exacerbated by rapid urbanization and population growth. Addressing these issues requires effective policies that reduce car dependency while maintaining flexibility for residents.

Urban planners have a wide range of policy interventions at their disposal, including congestion charging, parking and traffic control measures, the establishment of limited traffic zones or rededicating space allocated for cars. In recent years, additional mobility options such as shared mobility and micromobility have emerged, expanding the set of possible policy measures.

However, designing effective policies is highly complex. Urban planners must not only decide which policies to introduce but also determine the optimal level at which they should be implemented and, for most measures, their optimal spatial extent. The number

* Corresponding author.
E-mail addresses: elena.natterer@tum.de (E.S. Natterer), rohan.rao@tum.de (S.R. Rao), alejandro.tejadalapuerta@helmholtz-munich.de (A. Tejada Lapuerta), roman.engelhardt@tum.de (R. Engelhardt), sebastian.horl@irt-systemx.fr (S. Hörl), klaus.bogenberger@tum.de (K. Bogenberger).

https://doi.org/10.1016/j.trc.2025.105360
Received 15 March 2025; Received in revised form 19 August 2025; Accepted 22 September 2025
Available online 6 October 2025

of possible policy combinations grows exponentially, leading to an immense solution space. This multi-dimensional space is too vast to be explored manually based on experience alone. In practice, stakeholders often limit the solution space to a few scenarios, which in all likelihood will not include the optimal solution, and in the worst case might not even be close to it. Therefore, a more structured approach is needed to narrow down the most promising solutions.

Agent-based simulations are commonly employed to assess how potential policy measures impact the broader transportation system. While these models provide detailed and realistic assessments of the effect of policies in terms of congestion, emissions, and travel times per mode, their computational intensity severely limits the number of scenarios that can be tested in practice.

Agent-based models can be understood as functions that map policy interventions to their corresponding effects - both at the link level, such as changes in traffic volumes on specific street segments, and at the system level, including emissions, mode shifts, and travel times.

Machine learning (ML) has emerged as a powerful tool for approximating complex functions when trained on sufficient data. A key milestone in its application to transportation was the study by Sroczyński and Andrzej (2023), which demonstrated that ML models can replicate traffic patterns with accuracy comparable to microscopic simulations. While not directly related to this work, such advances highlight the potential of ML to approximate intricate relationships in transportation systems.

If machine learning could effectively approximate agent-based models, it could serve as the foundation for fast and scalable surrogates. Even if their predictions were not perfectly aligned with full agent-based simulations, the ability to rapidly filter out ineffective policy setups and thus being able to focus computational resources on the most promising solutions would be a major advantage. Furthermore, such an approach could reveal novel policy combinations that human planners might not have initially considered. Once trained, such a surrogate model can predict the impacts of policies within its training domain within seconds, enabling efficient scenario exploration and large-scale optimization. However, for new policy types or urban contexts, additional training data and retraining are required to ensure reliable predictions. Further, the surrogate model approximates the outputs of agent-based simulation, and its validity ultimately depends on the calibration and realism of the underlying simulation model.

Graph Neural Networks (GNNs) are machine learning models specifically designed to handle graph-structured data. GNNs have gained increasing attention in the transportation domain due to their ability to capture spatial dependencies, making them well-suited for modeling complex network dynamics. As such, intersections or street segments can be represented as nodes, while edges capture relationships such as connectivity or proximity. Their ability to capture spatial dependencies within urban networks makes them an intuitive choice for predicting the impact of policy interventions on link-level.

In this paper, we present a surrogate model for agent-based transport simulations, leveraging a graph neural network in combination with a transformer architecture to approximate the effects of policy interventions. We evaluate its performance using the large-scale MATSim simulation of Paris, France, applying the policy “50% capacity reduction on main roads”. While the developed method is adaptable to different policy interventions and cities, doing so requires generating new simulation data and retraining the surrogate model to reflect the specific characteristics of the new setting.

Our results show that the surrogate model can accurately predict changes in traffic volume per street segment resulting from the policy, demonstrating the feasibility of using a ML model as a surrogate for agent-based models in transport planning. This approach enables efficient solution-space exploration, allowing (1) planners to systematically narrow down policy options and identify the most effective interventions and (2) providing an efficient methodology for objective evaluation in optimization-based approaches.

While machine learning surrogates for transport simulations are gaining traction, most focus on small networks, aggregate outputs, or narrow applications like calibration or pedestrian flows. Few studies address agent-based models at city scale, and to our knowledge, none have used a GNN-Transformer hybrid model to predict link-level traffic effects of policies in large urban networks. This study fills that gap by introducing a scalable ML surrogate model trained on an agent-based simulation of Paris.

This paper is structured as follows: Section 2 reviews the relevant literature, followed by Section 3, which introduces the used methodology. Section 4 provides an in-depth explanation of the surrogate model, while Section 5 outlines the methodology for assessing its accuracy. In Section 7, we introduce the case study based on a MATSim simulation of Paris, France, to validate the proposed approach. Finally, Section 8 presents the results, and Section 9 concludes with key insights, methodological limitations, and directions for future research.

2. Background

2.1. Agent-based models for transport policy evaluation

Agent-based models (ABMs) have become a widely adopted approach for analyzing transportation systems due to their ability to simulate individual decision-making processes and interactions at a fine-grained level (Kagho et al., 2020). Unlike macroscopic and mesoscopic models, which aggregate traveler behavior, ABMs explicitly model each agent (e.g., travelers, vehicles) and their adaptive choices, allowing for a more detailed representation of network dynamics and behavioral responses to policies, as shown e.g. by Horni et al. (2016) and Macal and North (2010).

Agent-based models have been widely applied to assess various transportation policies, including congestion pricing (Ben-Dor et al., 2024), parking fee schemes (Balac et al., 2017), the impact of speed limits on street safety and emissions (Lu et al., 2023), and to evaluate low-traffic zones (Yin et al., 2024; Müller et al., 2023).

ABMs are particularly well-suited for analyzing the impact of emerging mobility solutions and their integration into existing transportation systems. As urban mobility continues to evolve, there is an urgent need for innovative transport solutions that not only improve efficiency and sustainability but also integrate seamlessly with current infrastructure.

The integration of emerging transport modes and mobility schemes into existing transportation systems presents both challenges and opportunities for policymakers and planners. Understanding travel behavior and demand in these evolving systems is essential for designing adaptive infrastructure and policies that can accommodate technological advancements and urban mobility transitions. Modern travelers' decisions are influenced by spatial, societal, and economic factors, adding complexity to the prediction of travel behavior in changing environments. Recent studies emphasize the need to analyze how emerging transport modes - such as ridesharing, electric vehicles, and autonomous vehicles - interact with existing systems. These interactions can be competitive or complementary, depending on regional and national contexts (Bastarianto et al., 2023).

Agent-based models provide a powerful approach to studying these interactions. They can be used to assess the integration of new transport modes with existing systems and to evaluate sustainable mobility transitions, such as shared mobility services (Hörl et al., 2021), electric vehicle adoption (Marmaras et al., 2017), and demand-responsive transit (Auld et al., 2016).

Recent work by Triebke et al. (2023) investigates the use of ABMs, using MATSim as an example, to study traffic dynamics under varying conditions, including scenarios involving capacity reductions. While the study finds that MATSim's mesoscopic traffic flow model has limitations in representing detailed link-level congestion patterns and spillback effects, it demonstrates that the model can produce realistic network-wide travel times when properly calibrated. Accordingly, agent-based models, when carefully calibrated, are well suited for analyzing daily-level impacts of capacity reduction policies on travel demand, routing behavior, and overall network performance, particularly when the primary interest lies in aggregate system effects rather than in detailed microscopic traffic dynamics.

By simulating traveler behavior and mobility trends, ABMs enable researchers and policymakers to explore potential impacts and identify strategies for optimizing urban mobility. Despite these advantages, the increasing complexity of ABMs also poses computational challenges, especially when scaling to large urban regions. As the number of agents and interactions increases, computational costs rise rapidly, making real-time policy testing difficult (Balmer et al., 2008).

2.2. Computational challenges of agent-based models

The disadvantage of agent-based simulations is a high run-time of usually several hours, limiting the amount of scenarios that can be tested. Large-scale ABMs simulate thousands to millions of agents, each with their own decision-making processes. This requires significant memory and processing power, with computational demands growing rapidly as more agents and decision rules are incorporated. Since many ABMs do not parallelize efficiently, execution times often increase non-linearly, making real-time scenario testing infeasible (Auld et al., 2016; Bastarianto et al., 2023).

Improving the scalability and efficiency of agent-based traffic simulations is an open research question. For the MATSim simulator, Laudan et al. (2025) compare local parallelization with distributed computing approaches, but show limitations in acceleration. Furthermore, MATSim is a highly modular framework that is used by numerous domain experts from the transport field that often only have basic programming knowledge. This is a main reason why MATSim is still implemented in Java which is comparably slow with respect to other programming languages, but allows for quick adaptation and extension with new components. An interesting approach, therefore, remains to provide the framework with a high degree of implementation flexibility and find ways to approximate the high-level model outcomes. Smilovitskiy et al. (2025) present a GPU-accelerated transport model for the Isle of Wight using the FLAME-GPU framework. The study demonstrates significant performance gains over traditional CPU-based transport simulations, enabling the simulation of larger vehicle populations and more detailed traffic interactions. However, the computational gains do not fully eliminate scalability issues, particularly for real-time or city-scale applications. Despite leveraging GPU hardware, simulations of large-scale networks still require extensive resources, and the dependency on specialized hardware limits accessibility. Furthermore, GPU-accelerated simulations often focus on the specific task of traffic assignment, but do not cover other aspects such as multimodality or mode choice behavior. An example where such elements are taken into account is presented by Saprykin et al. (2019). However, the rigid programming requirements for the GPU interaction limit modularization and easy adaptation and extension of the software by domain experts with basic programming knowledge as explained above. Manley et al. (2014) introduce a hybrid agent-based modeling framework that balances behavioral realism with computational efficiency by integrating detailed driver behavior with a simplified collective traffic flow model. They demonstrate how this approach enables scalable simulations of urban traffic dynamics while improving computational performance compared to traditional agent-based methods. However, they still face trade-offs between behavioral realism and scalability. The challenge lies in ensuring that large-scale simulations maintain sufficient fidelity in representing individual decision-making and emergent traffic phenomena.

One widely-used alternative to speed up agent-based models for practical applications, is down-sampling the population size. For example, Llorca and Moeckel (2019) analyzed the effects of population scaling in agent-based transport models, investigating how different scale factors impact model runtime, travel times, and link volumes. The authors conclude that scaling down agent populations reduces runtimes but must be done carefully to avoid distortions in travel times. For the Munich metropolitan area they found that a 5% sample produces travel time distributions similar to a full-scale simulation (100% sample) while being 50 times faster. Further, they find that the choice of scale factor depends on the analysis level: aggregate studies may tolerate smaller samples, while detailed corridor or small-area analyses may require full populations.

However, downsampling the population size in agent-based transport models increases the inherent stochasticity due to the random selection of agents, which leads to variability in simulation outcomes. This variability arises because different subsamples may not accurately represent the full population's travel behaviors, affecting demand distribution, route choices, congestion levels, and travel times. Such stochastic effects are particularly pronounced with smaller sampling rates, where statistics can substantially deviate from those of full-scale models. (Ben-Dor et al., 2021)

Concluding, the goal of fully scalable agent-based models remains out of reach, as current research has yet to make them feasible for large-scale applications. In practice, population downsampling is used to reduce computational demands, but even with this approach, testing a single policy can take days. Consequently, the demand for more computationally efficient alternatives of agent-based models has driven interest in machine learning-based surrogates, which could approximate agent-based model outcomes while preserving key behavioral dynamics, enabling faster policy evaluation and optimization.

2.3. Machine learning surrogates for transport simulations

Machine learning surrogates for transport simulations learn the relationships between a subset of simulations' input parameters (e.g., characteristics of the different street segments, due to their nature or due to the introduced policy) and a subset of output variables that are of interest for the surrogate model's use case (e.g., traffic flow metrics on street segment level). The surrogate is trained from data generated by the traffic simulation. Once trained, the surrogate model can rapidly predict the outcomes of various policy scenarios without the need for computationally intensive simulations.

Table 1 provides an overview of machine learning surrogates proposed for different use cases in traffic simulations.

Liu et al. (2020) applied surrogate modeling to calibrate microscopic traffic simulations. The surrogate model is used to improve the computational efficiency of a particle swarm optimization (PSO) algorithm to determine optimal simulation parameters. The goal of the surrogate model is to predict vehicle speeds on three highway links in a case study of Shanghai based on 11 parameters of the applied car-following model. Four machine learning models (Decision Tree, Support Vector Machines, Gaussian Process Regression, and Artificial Neural Networks (ANN)) are tested, with ANNs achieving the best accuracy. The ANN-based surrogate model is then embedded in PSO to optimize parameters, demonstrating superior efficiency and effectiveness. While effective, the authors recognize some limitations: They used a Multilayer Perceptron (MLP) for the artificial neural network but suggest that future studies explore Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) for better performance.

Similarly, Cervellera et al. (2021) present deep learning models as surrogates for microscopic traffic simulations. They focus on finding a feasible amount of training data by leveraging low-discrepancy sequences to cope with the large dimensionality of possible input parameters. In their case study, they calibrate origin-destination (OD) flows for a SUMO network of Genova, Northern Italy. The input to the surrogate model was 20 possible OD-flows, while the model is used to predict traffic counts on 47 edges. The authors show the feasibility of this approach and that deep neural networks outperform Gaussian processes. The authors highlight the need for a deeper theoretical analysis of their approach in the surrogate modeling framework and suggest broader testing across different applications beyond traffic simulation.

Tanaka et al. (2024) propose surrogates for real-time optimization of pedestrian crowd control policies using a neural network-based model trained on microscopic pedestrian simulations. Node-based policy parameters describing the attractiveness of locations in the network are used as input to predict global values for congestion, incentive cost, and travel time in the network when pedestrians move between nodes. As neural networks are differentiable, this property is used to define a gradient-based policy optimization technique. A single origin node is used with up to five termination nodes where pedestrian flows are guided to. Experiments show that the method reduces policy search time by over 14,000 s compared to simulation-based exploration with Bayesian Optimization.

Roman et al. (2025) develop a surrogate for an agent-based transport simulation to provide a tool for urban planners to quickly evaluate scenarios that would need hours to compute if the full-scale simulation were run. Different surrogates - built using XGBoost, Random Forests, Neural Networks, and Mixed-Integer Gaussian Processes - are embedded in a Bayesian optimization loop to support robust scenario selection under uncertainty. In their case study, the goal was to select best possible scenarios for the design of a shared demand-responsive transport (DRT) service. Input to the model were three values representing different scenarios for DRT network, parking spots, and pick-up and drop-off locations. The output of the surrogate model was system-wide performance indicators like carbon emissions, public transport usage, travel delays, and investment costs.

Previously mentioned surrogates focus on approximating aggregated network-wide performance indicators or simulation results at a few spots in the network.

For section-specific, or even lane-specific predictions, graph neural networks have gained increasing attention in the transportation domain due to their ability to capture spatial dependencies. As a class of deep learning models designed for graph-structured data, GNNs represent entities such as intersections or street segments as nodes, while edges encode relationships like connectivity or proximity. Their architecture enables the effective modeling of both local and global spatial dependencies, which has led to successful applications in traffic forecasting. For example, Jiang and Luo (2024) provide a comprehensive review of GNNs in traffic forecasting. With respect to surrogate modeling, Yousefzadeh et al. (2025) propose a surrogate model for microscopic traffic simulators using Graph Attention Network (GAT)-based autoencoders for lane-level predictions. Their model is applied for an urban corridor with nine intersections and trained on 400,000 h of SUMO simulation data and loop detector measurements, enabling accurate, topology-invariant, real-time prediction of lane-wise traffic flows. The architecture focuses on learning compressed latent representations of lane-topology graphs for fast and efficient decoding of traffic states.

Recently, Transformer-based models have emerged as powerful tools for learning from structured spatial data in transportation systems. Originally designed for sequence modeling in natural language processing, Transformers utilize attention mechanisms that allow them to capture long-range dependencies across all input elements, regardless of their position or connectivity. This global receptive field makes Transformers a generalization of Graph Neural Networks, as they are not limited to localized message passing but instead learn dynamic relationships between all nodes. In the context of traffic modeling, this flexibility allows Transformers to effectively represent both local and non-local interactions in complex urban networks.

Table 1
Summary of machine learning surrogates for transport simulations, sorted by year.

Authors	Model	Use Case	ML Method	Data Sources	Evaluation Metrics	Key findings
Liu, Zou, Ni, Gao, Zhang (Liu et al., 2020)	Trans - Modeler (micro-scopic model)	Traffic simulation calibration	Decision trees, Support Vector Machines, Gaussian Process Regression, Artificial Neural Networks (ANN), Particle Swarm Optimization (PSO)	Field data, simulated outputs	Prediction accuracy, computational efficiency	ANNs yielded the best prediction accuracy; ML + PSO methodology improved computational efficiency and effectiveness.
Cervellera, Maccio, Rebora (Cervellera et al., 2021)	SUMO	Origin-destination (OD) demand calibration, traffic light optimization, and strategic mobility planning.	Deep Neural Networks (DNNs), trained using low-discrepancy sampling (Sobol' sequences)	Data generated using SUMO, with OD flow values as inputs and simulated traffic flows as outputs. Medium-sized urban network.	1) Estimation accuracy of traffic flow at measurement points (MAE), 2) Distribution preservation comparing predicted and true flow distributions (MMD), 3) Surrogate model effectiveness in OD demand calibration.	1) Deep learning-based surrogates outperform Gaussian Processes in high-dimensional traffic simulations, 2) Low-discrepancy sampling (Sobol' sequences) improves training efficiency, accuracy, and reduces variance, 3) The surrogate better preserves traffic flow distributions, making it suitable for demand calibration, 4) Combining deep learning with low-discrepancy sampling enhances efficiency without sacrificing accuracy.
Tanaka, Amano, Uchiyama, Hiromori, Nakamura (Tanaka et al., 2024)	Pedestrian ABM	Optimization of crowd control policies (e.g., phased exits, incentives)	Neural network surrogate + gradient-based optimization	Scenargie simulation (agent-based), detour-centric road networks, real-world pedestrian flow data (Koshien Stadium, Japan)	Policy effectiveness score (travel time, congestion, cost), execution time	Surrogate enables fast gradient-based policy optimization; reduces exploration time by over 14,000s; validated on synthetic and real-world pedestrian scenarios.
Narayanan, Makarov, Antoniou (Narayanan et al., 2024)	4-step model	Replicating traffic flows from simulations	GNNs	Synthetic transport simulation data, networks consisting of 15-80 nodes	Accuracy (F1 Score, MAE and R^2) and computational efficiency	Proof-of-concept that a GNN can replicate traffic flows learned from the 4-step model.
Roman, Maheshwari, Do, Adey, Fourie, Ye, Bansal (Roman et al., 2025)	MATSim	Adaptive scenario planning under uncertainty	XGBoost, Random Forest, Neural Nets	600 MATSim simulations (Singapore)	Accuracy, runtime	Surrogates predict outcomes under uncertainty; used in scenario optimization loop.
Yousefzadeh, Sengupta, Karnati, Rangarajan, Ranka (Yousefzadeh et al., 2025)	SUMO	Lane-wise traffic flow at intersections	GAT-based Graph Auto-Encoders	400k h of SUMO simulation + real loop data	MAE, RMSE, inference speed	GAT-based digital twins produce accurate lane-wise flow estimates; topology-invariant; real-time capable.
This study	Generic ABM	Predicting the effects of transportation policies by approximating the outcomes of agent-based simulations	Hybrid Transformer-GNN model	10,000 MATSim simulations of Paris, comprising over 30,000 nodes each	Accuracy (MSE, MAE, R^2, Pearson and Spearman Correlation), Impact of Simulation Stochasticity on the Error	Proof-of-concept that ML models can learn the effect of transport policies from agent-based models on link-level.

Several recent studies highlight the potential of Transformers in traffic applications. For instance, Reza et al. (2022) propose a multi-head attention-based Transformer model tailored for traffic flow forecasting in urban areas, and through rigorous comparison with recurrent neural networks, demonstrate the Transformer's advantage in capturing long-range spatial and temporal dependencies essential for accurate predictions in transportation systems. Xing et al. (2023) introduce STTF, a spatio-temporal Transformer architecture designed specifically for effective congestion prediction in urban road networks, utilizing a newly developed

congestion index and incorporating road network structural information to enhance prediction accuracy and interpretability. Zhang et al. (2025) advance transformer-based traffic flow prediction by embedding implicit spatial relationships and employing an enhanced self-attention mechanism, resulting in improved modeling of complex spatial-temporal dependencies and superior predictive accuracy across multiple real-world datasets.

Despite their potential, hybrid GNN-Transformer architectures have not yet been explored in the context of surrogate modeling for transport simulations. These models combine graph convolution layers to capture local spatial structure with Transformer attention mechanisms to learn global, long-range dependencies across the network. This hybrid design leverages the strengths of both paradigms: the inductive bias of GNNs for spatially structured data and the flexible interaction modeling of Transformers.

One of the few related studies is provided by Narayanan et al. (2024); They define a GNN-based surrogate to replicate outputs of a four-step transport model and introduce an augmented data generation process for training on synthetic networks containing 15–80 nodes. Their experiments assess GNN performance on both classification (discrete flow categories) and regression (continuous flow values) tasks at the link level. While the results are promising, the study is limited to small, synthetic networks and an exclusive focus on the four-step model.

2.4. Contribution of this study

This study presents a scalable graph-based surrogate modeling framework that replicates the outputs of complex agent-based traffic simulations with high fidelity. Specifically, we show that graph neural networks in combination with Transformers can effectively learn the relationship between transportation policies and their impact on traffic volumes at the street-segment (link) level.

While recent studies have developed machine learning-based surrogate models for agent-based transport simulations, including those leveraging GNNs, our work is distinct in its emphasis on fine-grained, link-level predictions across large-scale urban networks. In our case study, we focus on a single policy type - road capacity reduction - implemented in various combinations across urban districts. We evaluate the surrogate model's ability to accurately predict the simulation outcomes for novel, previously unseen policy combinations. This is a realistic setting for transport planning, where a fixed intervention must be evaluated under multiple spatial rollouts.

The presented framework is flexible and can be adapted to other transportation policies and cities, as long as new simulation data is generated and the model is retrained accordingly. The contributions of this study are threefold:

1. **Scalable Surrogate Modeling Validated on a Large-Scale Network.** We propose a scalable surrogate framework based on a dual graph representation of the street network and a hybrid GNN-Transformer architecture, enabling accurate prediction of policy effects at the street-segment level. Unlike prior studies limited to small areas or aggregate metrics, our approach supports detailed, spatially granular forecasts across complex, large-scale urban networks. We validate the method on a high-resolution MATSim simulation of Paris, comprising over 30,000 street segments and 10,000 spatially heterogeneous road capacity reduction scenarios. The model generalizes effectively to unseen policy configurations, demonstrating its practical utility for strategic evaluation and efficient exploration of complex policy spaces.
2. **Systematic Benchmarking of Surrogate Architectures.** We design and execute a two-step benchmarking protocol to assess model performance. In the first stage, we evaluate a range of model architectures - both GNN-based and non-GNN baselines - using compact versions with comparable parameter counts to ensure a fair comparison. As the Transformer outperforms the other models, we further extend this stage by evaluating hybrid architectures that combine GNNs with Transformers. Based on this evaluation, we identify the best-performing architecture. In the second stage, we scale up the selected architecture to assess how increased model capacity influences predictive performance. This systematic comparison provides insights into the trade-offs between model complexity and accuracy.
3. **Quantifying and Accounting for Simulation Stochasticity.** Agent-based models inherently involve stochasticity, particularly when population downsampling is employed to reduce computational demands. We examine how this variability influences surrogate model accuracy, offering insights into the practical limits of predictive performance and informing the interpretation of surrogate outputs trained on inherently noisy simulation data. To our knowledge, this represents a novel exploration of aleatoric uncertainty within transportation surrogate modeling.

3. Methodological approach

Fig. 1 gives an overview of the general methodology: the framework integrates an ABM with a GNN-Transformer model to predict changes in traffic volume at the link level under different policy scenarios. The process begins with an ABM that simulates travel behavior based on three key components: the street network, which represents the street infrastructure; travel modes, which define different transportation options; and a synthetic population, representing individual travelers and their mobility choices.

Two types of scenarios are considered: a *base case* (defined in Section 3.1), which captures existing traffic conditions such as baseline traffic volume $\bar{v}_e$ and positional attributes without any intervention, and *policy scenarios*, where the introduced policy leads to adapted traffic volumes.

To efficiently process this information, the street network is transformed into a *dual graph*, where street segments are represented as nodes, and their connectivity is defined by edges, explained further in Section 4.3. A ML-based surrogate model is then applied to capture relationships between the network and the applied policy, as described in Section 4. Finally, the trained surrogate predicts the *change in traffic volume* at the link level under active policy conditions.

The relevant variables for the analysis are listed in Table 2.

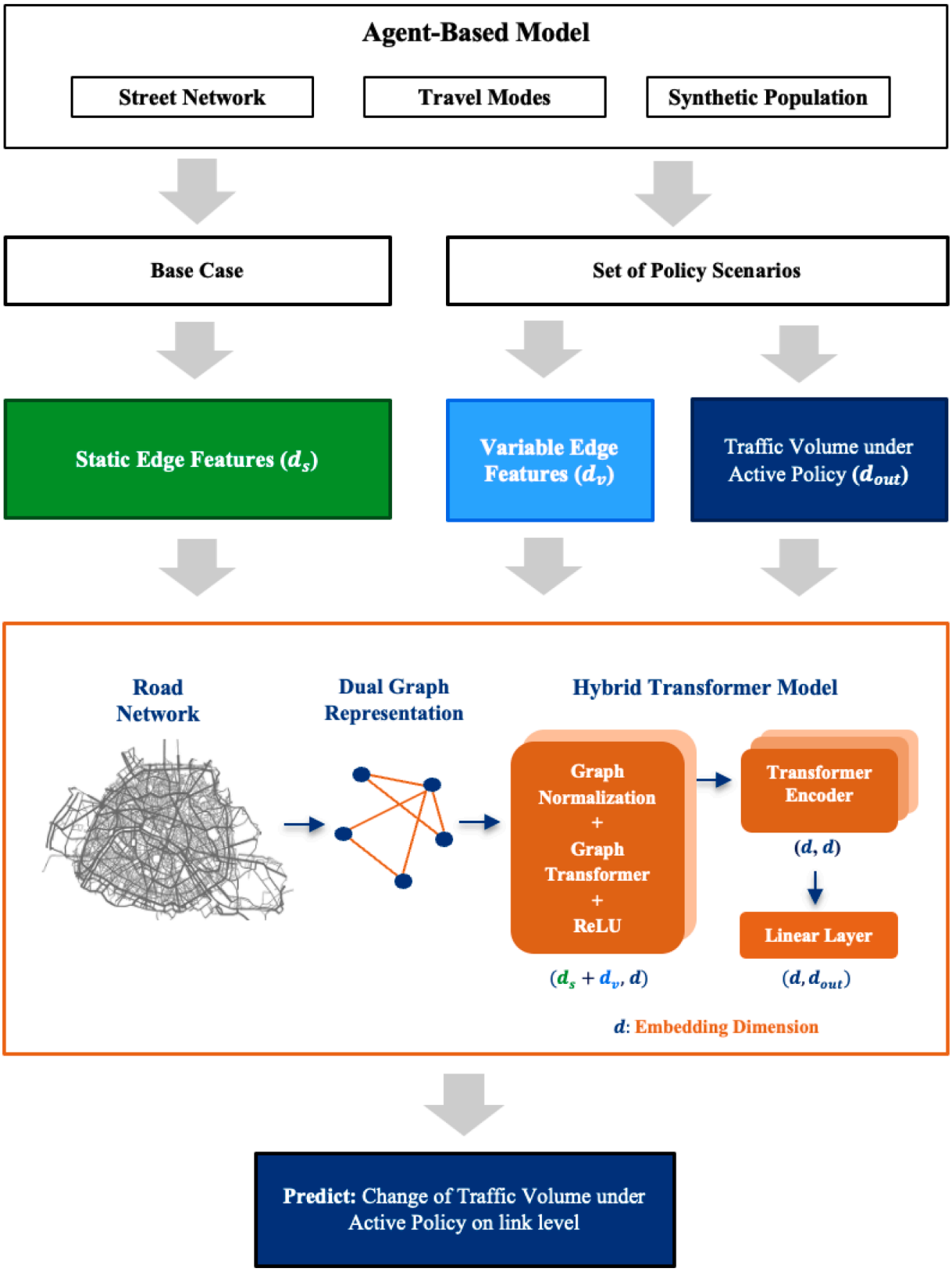


Fig. 1. High-level overview of the methodology: An agent-based model generates training data for a ML-based surrogate model, which predicts link-level traffic volume changes under policy scenarios.

Table 2
Symbols used in this analysis.

Symbol	Description
E	Set of edges in the street network, single edges denoted $e \in E$
R	Set of random seeds for which we perform simulation runs, random seeds denoted $\{r_1, r_2, \dots\}$
P	Set of policies that may be implemented on edge level, policies denoted $\{p_1, p_2, \dots\}$
v_e^r	Traffic volume on edge e in simulation run r, without policy intervention
$\overline{v}_e$	Average traffic volume on edge e, without policy intervention
$y_{e,p}^r$	Simulated *change* in traffic volume on edge $e \in E$ in simulation run r due to policy p.
$y_{e,p}$	Single realization of $y_{e,p}$: Simulated change in traffic volume on edge $e \in E$ due to policy p for one simulation run.
$\overline{y}_{e,p}$	Average simulated change of traffic volumes on edge $e \in E$ due to policy p: $\overline{y}_{e,p} = \frac{1}{R} \sum_R y_{e,p}^r$
$\overline{y}_p$	Mean of the simulated change of traffic volumes over all edges $e \in E$ due to policy p: $\overline{y}_p = \frac{1}{E} \sum_E \overline{y}_{e,p} = \frac{1}{E} \frac{1}{R} \sum_E \sum_R y_{e,p}^r$
$\hat{y}_{e,p}$	Predicted (ML surrogate) change in traffic volume on edge $e \in E$ due to policy p.

3.1. Base case

The *base case* represents a simulation scenario of the agent-based model without any policy intervention, serving as the reference point for predicting changes in traffic volume. It is generated by running multiple simulations with different random seeds and averaging the results at street level.

Consider the simulation, without any policy intervention, repeated with $|R|$ different random seeds. Each simulation repetition produces a traffic volume v_e^r, where $r = 1, 2, \dots, |R|$. For a single edge $e \in E$, the base traffic volume for edge e is then defined by $\overline{v}_e$:

$$\overline{v}_e = \frac{1}{|R|} \sum_{r=1}^{|R|} v_e^r, \tag{1}$$

With this, we can directly define the variance of the base case traffic volume for edge $e \in E$ as:

$$\sigma_{e,b}^2 = \frac{1}{|R|} \sum_{r=1}^{|R|} (v_e^r - \overline{v}_e), \tag{2}$$

which quantifies the dispersion of traffic volume across independent simulation runs with different random seeds.

3.2. Problem statement

We aim to develop a machine learning-based surrogate model that estimates traffic volumes in response to policy interventions, using data from agent-based simulations. Specifically, the model learns how each intervention affects traffic on individual roads within the network.

Mathematically speaking, our objective is to learn a function F that maps a set of input simulation parameters P_{sim} to the corresponding simulation output S:

$$F(P_{sim}) \to S. \tag{3}$$

This defines a regression problem where the goal is to approximate F such that it captures the relationship between the input parameters and the resulting traffic volume changes. Let F_p denote F under policy intervention p, mapping input features to traffic volume changes under p. The goal is to learn the function:

$$F_p : \mathbb{R}^{|E| \times n} \to \mathbb{R}^{|E|}, \tag{4}$$

which maps a set of n street characteristics per edge to the predicted change in traffic volume $\hat{y}_{e,p}$ caused by a given policy intervention p. The impact of a policy intervention on a street is measured by the change in its traffic volume relative to the base case: For each edge $e \in E$, we predict the traffic volume change under intervention p compared to the base case.

To train the model, we use the Mean Squared Error (MSE) loss function, which quantifies the difference between predicted and actual values. Let x_e denote street characteristics for street e:

$$\mathcal{L}(F_p) = \frac{1}{|E|} \sum_{e \in E} (F_p(x_e) - \overline{y}_{e,p})^2 \tag{5}$$

$$= \frac{1}{|E|} \sum_{e \in E} \left(\hat{y}_{e,p} - \frac{1}{|R|} \sum_{r \in R} y_{e,p}^r \right)^2. \tag{6}$$

In an ideal setting, minimizing this loss would require $|R|$ simulation runs per policy intervention to compute the expected change in traffic volume. This distinction is crucial: the observed traffic volume change in a single run, $y_{e,p}^r$, differs from the expected change averaged over multiple runs with different random seeds, $\overline{y}_{e,p}$. Ideally, the surrogate model would predict $\overline{y}_{e,p}$ to account for randomness in the simulation.

However, due to computational constraints, each policy intervention is simulated only once ($|R| = 1$), so the predicted value $\hat{y}_{e,p}$ approximates $y_{e,p}$ rather than the expected mean. Consequently, we train our model using this single realization, minimizing:

$$\mathcal{L}(F) = \frac{1}{|E|} \sum_{e \in E} \left(\hat{y}_{e,p} - y^r_{e,p} \right)^2. \tag{7}$$

where $y^r_{e,p}$ is the observed traffic volume change for a simulation run with random seed r. As a result, the surrogate model approximates a single realization rather than the expected value, inherently incorporating the stochasticity present in the data. Since we always work with a single realization, we denote $y^r_{e,p}$ as $y_{e,p}$ hereafter.

The objective is to find the optimal function F^* that minimizes this loss:

$$F^* = \arg\min_F \mathcal{L}(F). \tag{8}$$

To develop an effective surrogate model, we first define the input and output parameters of the machine learning-based surrogate (see Section 4). The evaluation metrics are described in Section 5.1. Given the high computational cost of full-scale simulations, we train the model on downsampled population data, which increases the variance of traffic volumes across simulation runs with different random seeds. Since the model predicts a single realization of the stochastic simulation rather than the expected value $\bar{y}_{e,p}$, it inherently reflects the randomness in the data and cannot reduce stochasticity beyond what is already present. We further analyze this in Section 6, highlighting the role of base case variance in error estimation.

4. The machine learning surrogate model

As a theoretical basis for our surrogate model, we employ a hybrid architecture combining graph neural networks and Transformers (Vaswani et al., 2017; Shi et al., 2021). These models are well-suited for graph-structured data and can capture both local and global spatial dependencies. While GNNs aggregate information from neighboring nodes to model local interactions, Transformers generalize this process by allowing flexible, learned attention over all nodes, enabling global context modeling. Street networks naturally form graphs, where edges correspond to streets and nodes to intersections. In our case, the predictive target is at the street segment level, making the graph representation and the hybrid architecture a natural choice. Section 4.1 elaborates on the rationale for this approach.

Because most GNNs are optimized for node-based predictions, we transform the street network graphs into dual graphs, where nodes correspond to street segments and edges capture the connections between segments. This transformation, detailed in Section 4.2, allows the model to operate at the level of street segments rather than intersections, making it well-suited for learning spatial dependencies and traffic dynamics. Section 4.3 discusses the model's input-output structure, while Section 4.4 presents the network architecture and outlines the key input features.

4.1. The importance of spatial inductive Bias

A central reason for employing graph neural networks in our surrogate modeling approach is their inherent spatial inductive bias - the ability to naturally learn from graph-structured data. This property is especially critical in large-scale traffic modeling, where spatial dependencies between street segments directly influence traffic dynamics adapted by policies.

In our setting, the street network forms a graph in which interventions on one segment - such as a capacity reduction - can lead to downstream effects like rerouting or congestion in neighboring areas. GNNs leverage message passing to model these interactions explicitly, allowing information to flow across connected nodes. Importantly, this mechanism captures multi-hop dependencies, enabling the network to learn how local changes propagate through the topology of the entire system.

In contrast, non-graph models - such as fully connected neural networks or ensemble methods like XGBoost - lack an inherent understanding of spatial structure. To encode spatial relationships, one would have to manually engineer features that describe the graph (e.g., neighborhood connectivity or topological distances). However, this approach becomes impractical at scale - particularly for complex, long-range interactions.

The attention mechanism, the cornerstone of the transformer architecture, is a permutation equivariant operation, meaning it processes its input as an unordered set of tokens lacking inherent spatial or sequential context. While Large Language Models (LLMs) overcome this by injecting sequential positional embeddings to denote word order, the challenge in our case is to capture the spatial structure of the city, which we model as a graph. To address this challenge, our proposed architecture first employs a graph neural network to encode the local spatial structure. The GNN layers enable information flow between neighboring nodes, producing contextualized embeddings for each street that are inherently aware of their position within the network. Subsequently, these spatially-aware node representations serve as the input to a transformer model, which can then effectively model the crucial long-range interactions across the entire city graph. This hybrid approach consistently outperforms vanilla transformers.

We validate the advantage of spatial inductive bias empirically in Section 8.2, where the hybrid architectures outperform models that do not incorporate graph-structured information. This demonstrates that architectures exploiting spatial inductive bias - by directly encoding local dependencies - yield superior predictive performance in network-scale traffic prediction tasks.

4.2. Dual graph transformation

The first layers of the proposed surrogate are graph-based (GNNs). Most GNN architectures are designed for node-based predictions, whereas in our setting, predictions are required at the level of street segments (i.e., edges in the primal graph). To address this, we

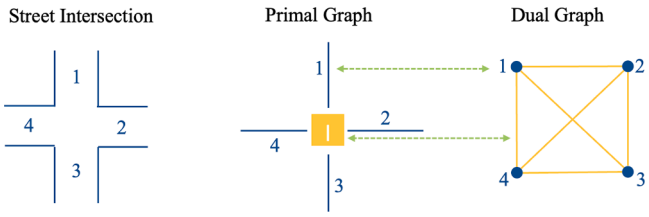


Fig. 2. Illustration of the dual graph transformation: street segments become nodes and intersections create edges between them.

transform the street network into its *dual graph* representation. In the dual graph, each node corresponds to a street segment, and edges represent connectivity between segments at intersections.

Fig. 2 illustrates this transformation. On the left, we show a standard street intersection with four street segments. In the middle, the primal graph represents the intersection as a node connected to four street segments (edges). On the right, the dual graph inverts this relationship: each street segment becomes a node, and two nodes are connected if their corresponding segments meet at the same intersection in the original (primal) graph.

This transformation allows us to apply GNNs directly to street segments, leveraging segment-level features and learning spatial dependencies between connected segments. Importantly, the approach generalizes to complex intersection geometries (e.g., multi-leg intersections, roundabouts), as dual graph edges are created for all pairs of segments that share an intersection in the primal graph. This ensures that the full connectivity and topology of the original street network are preserved in the dual graph representation.

In our setting, the street network is represented as a directed graph, where each edge corresponds to a street segment in a specific direction. Consequently, in the dual graph, each node represents a directed street segment. For streets that are bidirectional, this results in two separate nodes in the dual graph - one for each direction of travel. This ensures that directional effects (e.g., asymmetric traffic volumes or capacities) are preserved and can be explicitly modeled in the GNN.

This representation enables embedding edge-level features (e.g., traffic volume, capacity) as node attributes and provides a consistent structure for learning over the network.

4.3. Input and output of the model

Our objective is to predict policy effects at the level of street segments. To align the data structure with the learning task, we transform the street network into its *dual graph* representation, as explained above.

The model input is the dual graph for a given policy scenario, denoted P_{sim}. Each node in this graph corresponds to a street segment $e \in E$, and is associated with a set of features grouped into two categories:

1. *Static features* provide fixed attributes such as base traffic volume $\bar{v}_e$, capacity and speed limit in the base case, and street segment length. The static features also contain positional features.
2. *Variable features* represent attributes modified by implemented policies, such as reductions in capacity or speed.

We denote the number of static and variable features as d_s and d_v.

The output S captures the impact of the implemented policy at the street level, such as changes in traffic volume relative to the base case, represented by $\hat{y}_{e,p}$ for each street e.

4.4. Architecture

The proposed architecture is based on Transformer convolutions and Transformer encoders, selected for their strong ability to capture long-range spatial dependencies - a crucial feature for modeling urban traffic dynamics in complex street networks. An overview of the architecture is shown in Fig. 3.

The architecture consists of two stages. In the first stage, two layers of TransformerConv (Shi et al., 2021), a type of Graph Transformer, are employed to capture the local graph structure of the road network. Attention is restricted to neighboring nodes, so each road segment considers only its immediate neighbors. The TransformerConv layers implement multi-head attention to dynamically weight the relevance of each neighbor's features, enabling the model to learn which local connections are most important for traffic prediction.

To support the architecture, we incorporate Graph Normalization (GN): A graph-specific normalization technique that improves convergence and reduces covariate shift across layers. GN is particularly effective in deep GNNs, where standard normalization techniques (e.g., batch norm) often underperform due to graph sparsity and variable node degrees. μ_G and σ_G^2 are the mean and variance computed over all nodes in graph G, ϵ is a small constant for numerical stability, and γ and β are learnable parameters.

$$\hat{x}_i = \frac{x_i - \mu_G}{\sqrt{\sigma_G^2 + \epsilon}} \cdot \gamma + \beta \tag{9}$$

Both graph layers use ReLU activations, with their weights initialized using Kaiming or Xavier schemes to ensure stable training dynamics.

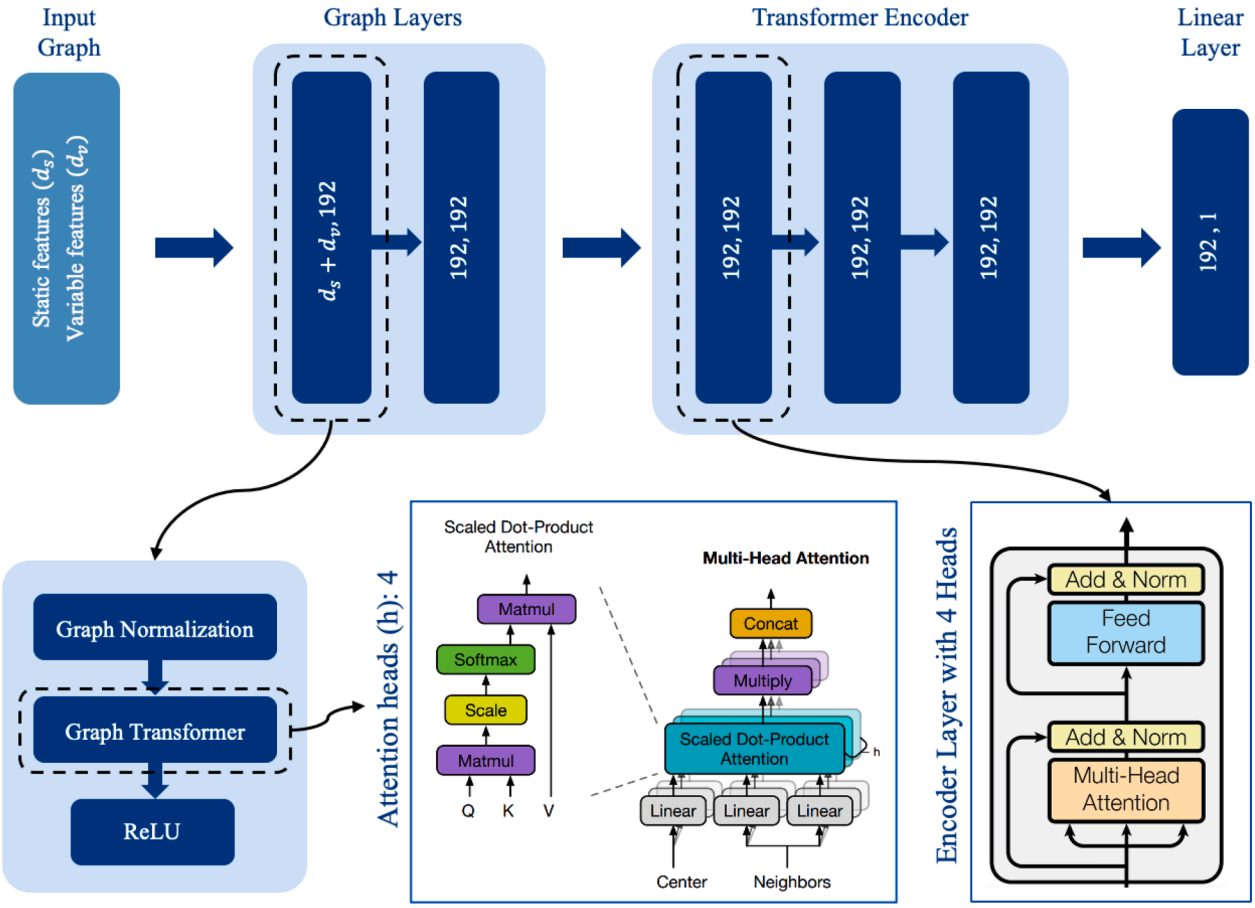


Fig. 3. Architecture of the surrogate model. The attention mechanism for the Graph Transformer follows Shi et al. (2021), while the Transformer Encoder adopts the formulation from Vaswani et al. (2017).

In the second stage, a three-layer Transformer Encoder (Vaswani et al., 2017) is applied with unrestricted self-attention. Each road segment can attend to all other segments in the network, allowing the model to capture global dependencies. Unlike the first stage, which focuses on local relationships, this stage models complex interactions across the entire road network and refines the representations learned in the first stage with global context.

Both stages of the model employ the *attention mechanism*, which allows each node (road segment) to aggregate information from other nodes based on learned relevance. Let n denote the number of nodes in the graph. Let m_k denote the dimensionality of the queries and keys, and m_v the dimensionality of the values. Let $Q \in \mathbb{R}^{n \times m_k}$, $K \in \mathbb{R}^{n \times m_k}$, and $V \in \mathbb{R}^{n \times m_v}$ denote the matrices of queries, keys, and values, respectively. The single-head attention output is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{m_k}}\right)V,$$

where $QK^T \in \mathbb{R}^{n \times n}$ contains the similarity scores between all query-key pairs, division by $\sqrt{m_k}$ stabilizes gradients, and the softmax normalizes the scores row-wise to produce attention weights. Multiplication with V propagates a weighted combination of the values to each node.

In both stages, the model uses four attention heads to capture multiple perspectives on node relationships, with an embedding dimension of 192 and a feedforward dimension of 768 in the Transformer Encoder.

The empirical benefits of this design are detailed in Section 8.2, where different architectures are compared.

To support ongoing research and practical applications, we provide an open-source library[1] for building and benchmarking surrogate models for agent-based simulations. The library offers a modular framework for data preprocessing, graph construction, and model evaluation. It includes flexible implementations of the hybrid architecture as well as nine additional regression algorithms, enabling easy comparison across a wide range of modeling approaches.

[1] The code for the ML surrogate can be found on https://github.com/enatterer/ml_surrogates_for_agent_based_transport_models

5. Methodology for evaluating surrogate model accuracy

While the model is trained by optimizing the MSE between simulated and predicted daily traffic volume changes, additional metrics are used to provide a more comprehensive assessment of its accuracy and robustness.

This section outlines our approach to evaluating the performance of the surrogate model on the link level. First, we introduce the evaluation metrics beyond MSE that are used for a thorough performance analysis. We also explain that model performance is assessed on the entire test set, with accuracy analyzed separately for different street types to identify variations (Section 5.1). Finally, since agent-based simulations inherently introduce stochasticity, understanding its impact on surrogate model accuracy is crucial for a meaningful evaluation of the results. To better interpret the results, we examine the extent to which model error is influenced by simulation stochasticity (Section 6).

5.1. Evaluation metrics

The test set consists of multiple scenarios, where each scenario corresponds to applying a policy p to a specific combination of districts. To evaluate model performance across different street types, we define aggregated evaluation metrics computed over the entire test set for each street type. Let S be the set of scenarios in the test set, and let T be the set of street types. Define E_t as the set of edges in the test set that belong to street type t. The total number of such edges is given by $|E_t| = |S| \times |T|$.

Aggregated base case quantities

The *mean base case traffic volume* for street type t is given by:

$$\bar{v}_t = \frac{1}{|E_t|} \sum_{e \in E_t} \bar{v}_e, \tag{10}$$

where $\bar{v}_e$ represents the base case traffic volume for edge e (Section 3.1).

The *base case variance of traffic volumes*, or simply *base case variance*, for street type t is defined as:

$$\sigma^2_{t,b} = \frac{1}{|E_t|} \sum_{e \in E_t} \sigma^2_{e,b}, \tag{11}$$

where $\sigma^2_{e,b}$ is the variance of traffic volumes across multiple simulation runs for edge e, as defined in Eq. 2.

Aggregated evaluation metrics

To evaluate model performance at the road-type level, we define the following quantities: The *average change in simulated traffic volume* for street type t under policy p is:

$$\bar{y}_{t,p} = \frac{1}{|E_t|} \sum_{e \in E_t} y_{e,p}, \tag{12}$$

where $y_{e,p}$ is the simulated traffic volume change for edge e under policy p. The *variance*, or *total sum of squares* ($SS_{\text{tot},t,p}$), for street type t under policy p is:

$$SS_{\text{tot},t,p} = \sum_{e \in E_t} (y_{e,p} - \bar{y}_{t,p})^2. \tag{13}$$

We use the following metrics for assessing model performance:

1. *Mean Squared Error (MSE)*
   The MSE serves as the primary loss function during training. Although it is a standard regression metric, its squared term disproportionately penalizes large errors, making it more sensitive to outliers. The MSE for street type t under policy p is defined as:
   $$MSE_{t,p} = \frac{1}{|E_t|} \sum_{e \in E_t} (\hat{y}_{e,p} - y_{e,p})^2. \tag{14}$$
2. *Mean Absolute Error (MAE)*
   To complement MSE, we report the Mean Absolute Error, which is more interpretable in practical applications as it represents the average absolute error in the same units as the input (veh/day). Additionally, MAE is more robust to outliers than MSE, as it treats all errors equally rather than amplifying large deviations. This makes it particularly useful in cases where occasional extreme values might otherwise dominate the error metric. The MAE for street type t under policy p is computed as:
   $$MAE_{t,p} = \frac{1}{|E_t|} \sum_{e \in E_t} |\hat{y}_{e,p} - y_{e,p}|. \tag{15}$$
3. *Coefficient of Determination (R^2)*
   Another standard regression metric is the coefficient of determination (R^2), which measures the proportion of variance in the observed values $y_{e,p}$ explained by the model's predictions $\hat{y}_{e,p}$. It provides a standardized measure of predictive accuracy by

comparing the model's residual error against the total variation in the data. The *sum of squared residuals* ($SS_{\text{res},t,p}$) per road type t and policy p, which measures the error between predicted and observed values:

$$SS_{\text{res},t,p} = \sum_{e \in E_t} (y_{e,p} - \hat{y}_{e,p})^2 \tag{16}$$

Finally, the coefficient of determination is given by:

$$R^2_{t,p} = 1 - \frac{SS_{\text{res},t,p}}{SS_{t,p}}. \tag{17}$$

An R^2 value of 1 indicates a perfect fit, while an R^2 of 0 suggests that the model performs no better than simply predicting the mean $\bar{y}_{t,p}$. Negative values imply that the model introduces more error than a naive mean-based predictor. Unlike MSE and MAE, R^2 offers a relative measure of fit, which can be useful for comparing the prediction of the model on different street types with different variance. However, it can be sensitive to outliers and may be less reliable for highly skewed data distributions.

4. *Correlation Metrics: Pearson and Spearman*
To further assess the model's performance, we compute the Pearson and Spearman correlation coefficients. Pearson correlation measures the strength of the linear relationship between actual and predicted values, indicating how well the predictions align in a linear sense. Spearman correlation, in contrast, evaluates the rank-based relationship, making it more robust to non-linear patterns. These correlation metrics complement MSE, MAE, and R^2, offering additional insights into the quality of the predictions.

6. The Impact of simulation stochasticity on surrogate model accuracy

Agent-based models inhibit randomness. When running the same simulation under identical conditions, but with different random seeds, the outputs vary due to the probabilistic nature of agent-based interactions. Further, machine learning surrogates for ABMs rely on training data from downsampled populations, as using the full population is impractical due to the extensive computational resources required, increasing stochastic variability in the simulation outputs. Stochastic variability acts as noise and affects the accuracy of surrogate models: In ML terms, this is called *aleatoric uncertainty*. Unlike *epistemic uncertainty*, which stems from incomplete knowledge and can be reduced with more data or improved modeling, aleatoric uncertainty reflects variability that cannot be eliminated with the given data, only quantified. A clear understanding of aleatoric uncertainty is essential for properly interpreting the results and recognizing the limitations of the model.

6.1. Definitions

Simulation output

The change in traffic volume on each edge e in the network is stochastic due to randomness in the simulation. We model the true simulated change in traffic volume as:

$$y_e = \Delta b_e + \epsilon_e, \tag{18}$$

where:

- $\Delta b_e = b_e^{\text{policy}} - b_e^{\text{base}}$ represents the *deterministic change* in traffic volume caused by the policy. This is a fixed value representing the average impact of the policy on edge e.
- ϵ_e is the *random simulation noise*, representing the variability in the simulation output.

Machine learning model prediction

The machine learning model attempts to predict y_e, producing:

$$\hat{y}_e = y_e + \hat{\epsilon}_e. \tag{19}$$

We assume the prediction error is structured as:

$$\hat{\epsilon}_e = \epsilon_e + \epsilon'_e, \tag{20}$$

where:

- ϵ_e is the original simulation noise.
- ϵ'_e is the *additional error* introduced by the model. Note that this error is *deterministic*, as any output of a machine learning model.

Mean squared error (MSE)

The Mean Squared Error (MSE) is defined as:

$$\text{MSE}(y, \hat{y}) = \frac{1}{|E|} \sum_{e \in E} (y_e - \hat{y}_e)^2. \tag{21}$$

Substituting $\hat{y}_e = y_e + \hat{\epsilon}_e$, and $\hat{\epsilon}_e = \epsilon_e + \epsilon'_e$, the squared error simplifies to $(\epsilon'_e - \epsilon_e)^2$ and thus

$$\text{MSE}(y, \hat{y}) = \frac{1}{|E|} \sum_{e \in E} (\epsilon'_e - \epsilon_e)^2. \tag{22}$$

6.2. Introducing the random variable

When evaluating our machine learning surrogate, our goal is to predict, for each edge $e \in E$, the simulated change in traffic volume, y_e, using the model prediction $\hat{y}_e$. We measure this using the squared error between the simulated values y and the predicted values $\hat{y}$ for all edges $e \in E$. Because y_e includes a stochastic component ϵ_e, it varies between simulation repetitions. Hence, also the MSE$(y, \hat{y})$ varies, and can never be zero.

To rigorously account for this variability, we turn to probability theory, which provides a natural framework for modeling random experiments. In this framework, we treat y_e and $\hat{y}_e$ as random variables to capture the inherent stochasticity of the simulation outcomes. Let:

1. Y_e be the random variable representing the simulated change in traffic volume for edge $e \in E$
2. $\hat{Y}_e$ be the random variable representing the predicted change in traffic volume for edge $e \in E$

Then, let $\mathbf{Y}$ be the random vector $\mathbf{Y} = (Y_{e_1}, Y_{e_2}, \ldots Y_{e_{|E|}})$, consisting of the random variables of all edges, and similarly $\hat{\mathbf{Y}} = (\hat{Y}_{e_1}, \hat{Y}_{e_2}, \ldots \hat{Y}_{e_{|E|}})$. Then, the quantity

$$\mathbb{E}[\mathrm{MSE}(Y, \hat{Y})] = \frac{1}{|E|} \sum_{e \in E} \mathbb{E}[\mathrm{MSE}(Y_e, \hat{Y}_e)] \tag{23}$$

represents the *expected mean squared error (MSE)* between the simulated outcomes Y and the surrogate model predictions $\hat{Y}$, averaged over all edges $e \in E$. Because the simulated outcome Y_e is subject to stochastic variation, it can differ between simulation runs even for the same input conditions. As a result, the squared error between Y_e and the prediction $\hat{Y}_e$ also varies. The expected MSE accounts for this by averaging the squared error over all possible realizations of Y_e.

In short, $\mathbb{E}[\mathrm{MSE}(Y, \hat{Y})]$ answers the question:

> *"On average, across all edges and across all possible outcomes of the stochastic simulation, how well does our surrogate model predict the change in traffic volume?"*

Next, note that Y_e, just like y_e, consists of a deterministic and a random part:

$$Y_e = \Delta b_e + \epsilon_e. \tag{24}$$

The deterministic change Δb_e is constant, and ϵ_e is the stochastic noise, which captures the random variability around the deterministic change. Without loss of generality, we assume ϵ_e follows a normal distribution:

$$\epsilon_e \sim \mathcal{N}(0, \sigma_e^2). \tag{25}$$

Similarly, the machine learning model's prediction is:

$$\hat{Y}_e = Y_e + \hat{\epsilon}_e \tag{26}$$

where again $\hat{\epsilon}_e = \epsilon_e + \epsilon'_e$. ϵ'_e is a constant for any $e \in E$ and thus has variance 0. Thus,

$$\begin{aligned}
\mathbb{E}[\mathrm{MSE}(Y_e, \hat{Y}_e)] &= \mathbb{E}[(Y_e - \hat{Y}_e)^2] && (27)\\
&= \mathbb{E}[(\epsilon_e - \epsilon'_e)^2] && (28)\\
&= \mathbb{E}[\epsilon_e^2] + \mathbb{E}[(\epsilon'_e)^2] - 2\mathbb{E}[\epsilon_e \epsilon'] && (29)\\
&= \mathbb{E}[\epsilon_e^2] + \epsilon_e'^2 - 2\epsilon' \mathbb{E}[\epsilon_e] && (30)\\
&= \sigma_e^2 + \epsilon_e'^2 && (31)
\end{aligned}$$

where Eq. 29 follows from the linearity of the expectation, Eq. 30 follows from that ϵ' is a constant and Eq. 31 follows from the assumption, that the error ϵ_e has mean zero, see Eq. 25.

6.3. The Bias-variance decomposition

Using the decomposition derived above and substituting into Eq. 23, we obtain:

$$\mathbb{E}[\mathrm{MSE}(Y, \hat{Y})] = \frac{1}{|E|} \sum_{e \in E} \sigma_e^2 + \epsilon_e'^2. \tag{32}$$

where σ_e^2 represents the variance of the stochastic simulation outcome Y_e for edge e, and ϵ'_e denotes the bias between the deterministic prediction of the surrogate model $\hat{Y}_e$ and the expected value of the simulation $\mathbb{E}[Y_e]$.

This equation represents the *lowest possible expected error* that the surrogate model can achieve. Even if the model perfectly predicts the deterministic part of the simulation - that is, if the model's prediction $\hat{y}_e$ exactly matches the expected value of the simulation output, $\mathbb{E}[Y_e]$, for every edge $e \in E$ - the inherent randomness in the simulation still prevents the error from being zero.

In other words, no matter how well the surrogate model captures the expected values $\mathbb{E}[Y_e]$, the variance in the simulation outcomes imposes a limit on how closely the model can match the actual values of Y_e for each edge. Thus, the simulation noise sets a fundamental upper bound on the model's prediction accuracy.

7. Case study

To evaluate the effectiveness of our surrogate model, we test it on the large-scale MATSim model of Paris, France. The model is based on a synthetic population of the Île-de-France region surrounding Paris (Hörl and Balac, 2021) which describes households, persons with their sociodemographic attributes and their daily activity sequences including information on when and where (by coordinate) the activities (home, work, shopping, ...) are performed. The process is open-source and entirely based on open data, which makes it fully replicable. In our case, 1% of households from the population of 12 million people are selected for performance reasons. The downsampled synthetic population is then simulated using MATSim for a baseline case of the entire region around Paris, and then cut so that only the agents interacting with the city area and its surrounding high-capacity ring street (*Boulevard Périphérique*) are retained. The MATSim model simulates a typical workday, and we aggregate the traffic volumes at the link level over the entire day. As a result, traffic volume is expressed in units of vehicles per day (veh/day).

Fig. 4 provides an overview of the ML surrogate model designed to predict traffic volume changes resulting from capacity reduction policies. The process begins with link-level inputs: the *policy scenarios*, simulated in MATSim, corresponding to combinations of districts where capacity reductions have been applied (see Section 7.1); and the *base case*, representing the network without interventions and providing static features such as baseline volume, capacity, speed limits, link length and positions, capturing the midpoint coordinates of each link (see Section 3.1). For each simulated policy scenario, we compute the difference in traffic volume at the link level between the simulated policy scenario and the base case, establishing the target variable representing traffic volume changes due to the policy intervention. The ML surrogate model is trained on these inputs, with the dataset split into training, validation, and test sets. It optimizes performance using the MSE loss function.

We first explain the simulation setup, then the data preparation, and finally, the training process of the surrogate model.

7.1. Simulation setup

We begin by generating 10,000 scenarios, each of which has the policy "50% Capacity Reduction" (p) being applied to different combinations of districts. The tested policy p enforces a 50% capacity reduction on all primary, secondary, and tertiary roads within the selected districts in a scenario. The street network consists of approximately 35,000 street segments, with primary, secondary, and tertiary roads (as classified by OpenStreetMap) constituting about 15,000 of these.

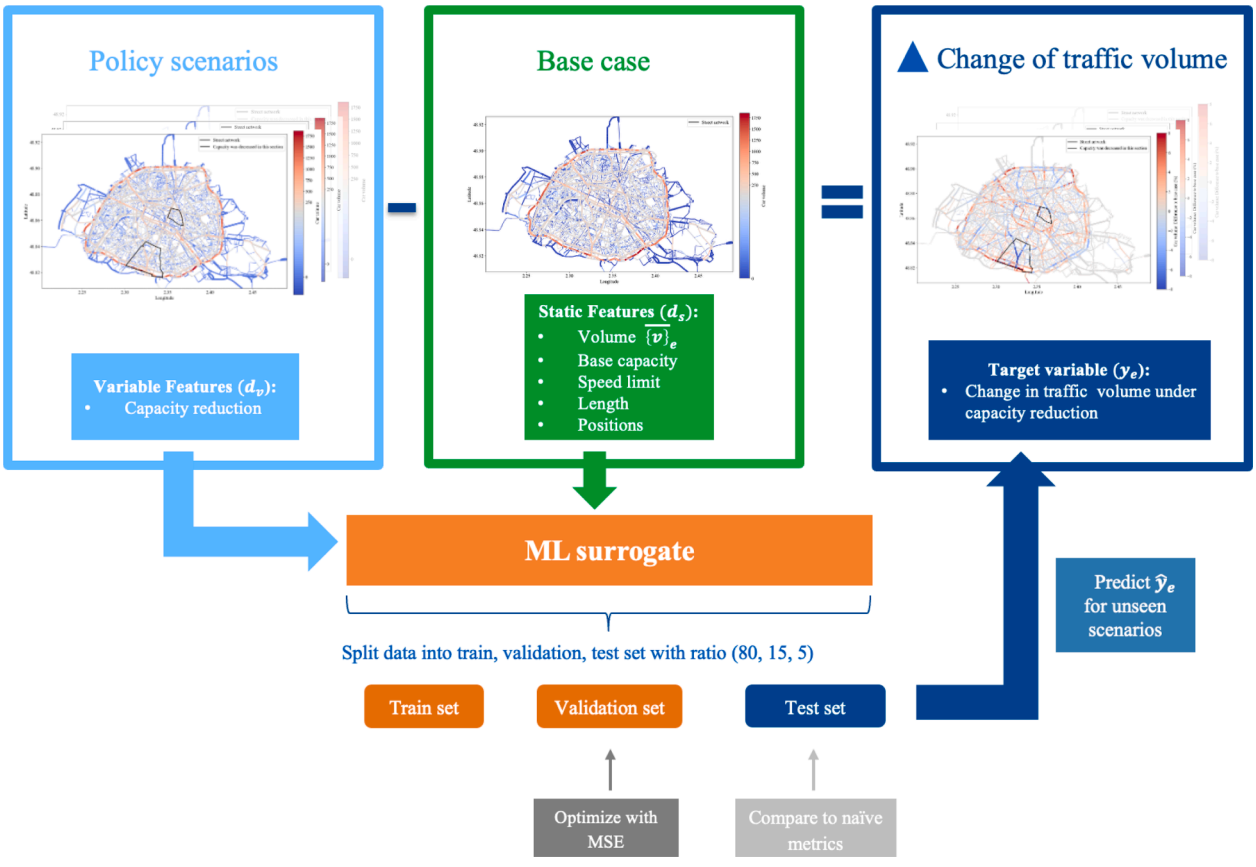


Fig. 4. Workflow of the case study.

Parallelly, we compute the MATSim base case (see Section 3.1) by averaging the traffic volume per street segment across 50 independent runs with different random seeds. The base case serves as a reference point for quantifying the changes in traffic volume, due to the introduced policy.

Paris is divided into 20 districts ("Arrondissements"). To generate district combinations, we follow a two-step process. First, we determine the size of each combination, i.e., the number of districts included. If chosen randomly, there are

$$\sum_{i=0}^{20} \binom{20}{i} = 1,048,576$$

possible combinations. The selection of district combinations follows a binomial distribution, $X \sim \text{Binomial}(20, 0.5)$, with a mean of 10 and a standard deviation of approximately 2.24.

To ensure the model accurately captures the localized effects of policy interventions, we focus on a smaller set of districts. Inspired by real-world transformations in Paris, where efforts have prioritized active mobility by expanding bike lanes and reallocating street space (Natterer et al., 2025), capacity reduction policies have been primarily implemented in Arrondissements 1 - 4, representing the "core" of the city. Therefore, we aim to generate district combinations with around four districts. To construct a diverse dataset, we generate 10,000 combinations using two different sampling strategies:

1. Randomly sample 5,000 combinations with a set length following a normal distribution with mean 4 and standard deviation 1.
2. Randomly sample 5,000 combinations with a set length following an exponential distribution with mean 4.

We run simulations for the selected combinations, resulting in 8,308 successfully completed runs. Each simulation took approximately one hour to complete on a high-performance cluster using twelve cores. The completed simulations are then used for training and testing the surrogate model.

7.2. Data preparation

Each of the resulting simulations are first made compatible with the surrogate's graph layer by turning it into the P_{sim} format (as described in Section 4.3). The corresponding dual graph has $31,635$ street segments (nodes), and $59,851$ edges (representing intersections). As illustrated in the top half of Fig. 3, to analyze policy effects, we subtract the base case traffic volume from the intervention case (resulting from policy simulations), generating $y^r_{e,p}$, which acts as the ground truth for the ML surrogate. At the network level, $y^r_{e,p}$ can be visualized as a difference plot, highlighting changes in traffic volume. Red areas indicate an increase in traffic volume compared to the base case, while blue areas represent a decrease, both driven by the policy intervention. Results presented in this format are shown in Figs. 9, 10, and 11.

As illustrated in Figs. 3 and 4, each street segment is characterized by two types of features: static and variable. The selected features include:

1. *Static features (d_s)*:
   - Daily traffic volume base case ($\overline{v}_e$)
   - Capacity base case
   - Speed limit
   - Length
   - Positional coordinates
2. *Variable features (d_v)*:
   - Capacity reduction

Therefore, in our setup, $d_s = 6$ (accounting for the x and y positional coordinates), and $d_v = 1$, resulting in an input feature dimension of 7. Initially, we considered a broader set of features, including categorical attributes such as street types and allowed transport modes. However, ablation tests revealed that the model did not effectively learn from these additional features, leading us to focus on the selected subset.

As a preprocessing step, all the features, being continuous, are normalized using a standard scaler, ensuring they have zero mean and unit variance. This helps maintain numerical consistency across features and improves model convergence. Finally, the processed scenarios data is split into training, testing, and validation sets with a ratio of (80, 15, 5). 6,646 of the overall 8,308 simulation scenarios are used for training, 1,246 for validation, and 416 for testing.

7.3. Training process of the surrogate model

The ML surrogate model is trained to predict changes in traffic volume on each street segment resulting from a policy intervention, relative to the base case. Its generalization ability is evaluated on unseen district combinations. The model is optimized using Mean Squared Error (MSE), ensuring accurate predictions while capturing network-wide traffic dynamics. Additionally, we log Spearman and Pearson correlation metrics to assess the consistency of the predictions.

Implemented in PyTorch Geometric, the ML surrogate was trained for up to 500 epochs (batch size: 8), with early stopping triggered if the validation loss did not improve for 25 consecutive epochs (5% of the total training). Training was conducted on a single NVIDIA H100 94GB GPU and required approximately 43 h to converge.

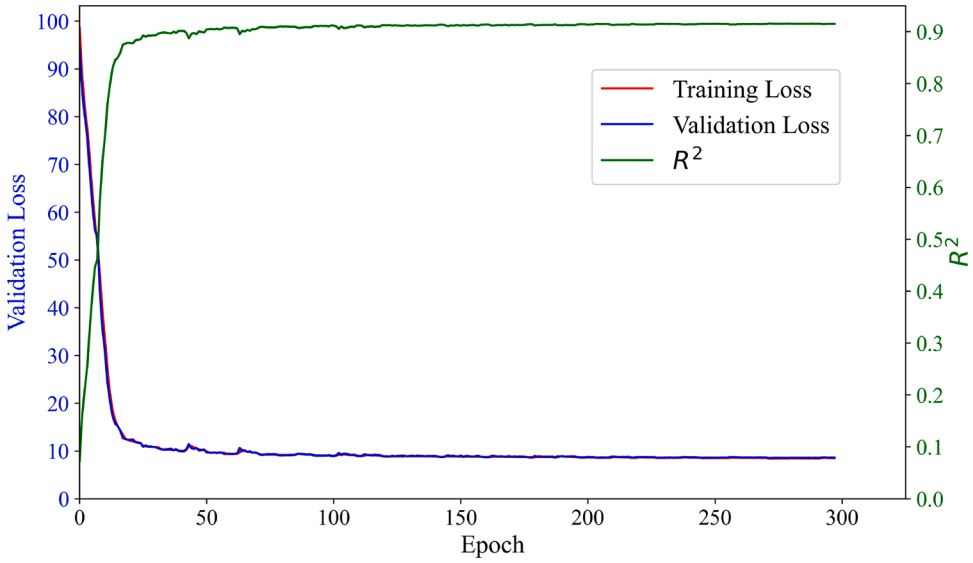


Fig. 5. Validation loss, training loss and R^2.

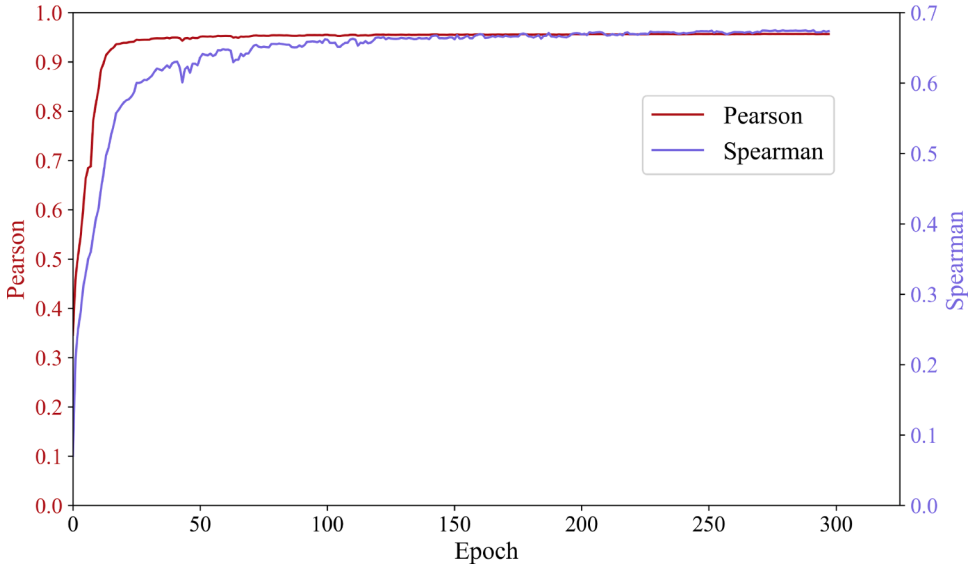


Fig. 6. Spearman and pearson correlation coefficient over the training process.

To stabilize learning, a linear warm-up phase was applied for the first 5% of the total epochs (25 epochs), gradually increasing the learning rate before reaching the peak value of 5×10^{-4}. The training process then followed a cosine decay learning rate schedule, progressively reducing the learning rate to 5×10^{-6} by the last epoch to enhance convergence. Gradient clipping is applied to prevent exploding gradients. The optimizer used was AdamW, with a weight decay of 1×10^{-4}, acting as a regularizer to mitigate overfitting.

During the training process, the model's performance was evaluated using the validation loss, R^2, and correlation metrics (Pearson and Spearman correlation coefficients). The learning curves are illustrated in Figs. 5 and 6. The first figure shows that both training and validation loss (red and blue, respectively) decrease over time, while R^2 (green) increases, indicating improved predictive accuracy. The model learns rapidly at first and continues to converge gradually, with early stopping triggered after 298 epochs - shortly after R^2 surpassed 0.92. The slight difference from the final test set value of 0.91 arises from the distinction between validation and test performance. The second figure highlights the steady increase of both Pearson (brown) and Spearman (purple) correlation coefficients throughout training, demonstrating that the model progressively learns meaningful relationships in the data.

7.4. Computational efficiency comparison

We emphasize that the relevant comparison for practical application is between the simulation time of MATSim and the prediction time of the trained ML model. While training the model requires computational resources, the training phase is a one-time cost per dataset. In contrast, the prediction time determines how quickly new policies can be evaluated once the model is trained, which is the core benefit of using a surrogate approach.

All MATSim simulations were executed on the the *SuperMUC-NG* system at the Leibniz Supercomputing Centre (LRZ) using the Slurm workload manager. For production runs, each job requested 16 compute nodes, with 6 tasks per node and 8 CPUs per task. The compute nodes (e.g., `i01r06c06s12`) are equipped with *2×24-core Intel CPUs*, hyperthreaded to *96 logical processors*, and *80 GB of RAM*. Simulations were run using *OpenJDK 17 (version 17.0.11_9)*, and the main class used for execution was `RunSimulations1pctDistNotConnected`, available in the public GitHub repository[2].

From a dedicated timing test using a single node, the average runtime for one simulation was *38 min and 22 s*, with a standard deviation of *45 s*. Across *100 independent simulation runs*, the 95% confidence interval for the mean runtime ranged from *38 min 12 s* to *38 min 31 s*.

In contrast, once trained, the ML surrogate model can make predictions for an entire city-level policy scenario in *462.47 ms*. All inference were performed on an NVIDIA RTX A5000 24GB GPU using PyTorch Geometric. Batch prediction is fully supported, which means that multiple policy scenarios can be evaluated simultaneously. However, since the batch is treated as a single large disconnected graph, this does not lead to speedups. For example, running inference on a batch sized 8 (as used during training) takes around *3.72 s* (*∼465.12 ms* per graph), while a batch size of 32 takes *15.13 s* (*∼472.74 ms* per graph).

While inference time increases with model size, it remains well within practical limits for interactive policy evaluation. Importantly, no inference-time optimizations such as quantization or pruning have been applied yet, meaning there is still potential to reduce latency further. As detailed in Section 8.2, deeper models generally yield higher accuracy. However, given the diminishing returns in predictive performance for very large models, smaller architectures may provide a better balance between speed and accuracy depending on specific application needs.

These results highlight the substantial computational advantage of the surrogate model: reducing scenario evaluation time from approximately 2,300 s (MATSim) to under 0.5 s (ML surrogate) - a speed-up factor of around 5,000.

8. Results

8.1. Overall performance

The trained surrogate model can predict the effects of policy interventions in randomly selected district combinations within milliseconds. It achieves an overall accuracy of $R^2 = 0.91$ and a MSE of 8.75 on the test set. The base case variance is 8.57. As we will see in Section 8.5, this can help with interpreting the error; in particular, we present an assumption under which the base case variance can represent the part of the error that comes from simulation noise. The model achieves a MAE of 1.71, and strong predictive correlation coefficients, with a Pearson correlation coefficient of 0.96 and a Spearman correlation coefficient of 0.67.

We begin by demonstrating that the chosen architecture achieves superior performance compared to all benchmarked alternatives (Section 8.2). Then we evaluate the model's performance across different metrics and street types (Section 8.3), with a particular focus on the differences in predictions for streets with and without capacity reductions (Section 8.4). Further, we investigate the relationship between simulation stochasticity and model error in Section 8.5, as motivated in Section 6. Section 8.6 analyzes performance consistency across multiple evaluation metrics. Then, we analyze results at the network level (Section 8.7): We first present results for three randomly selected test scenarios to provide a general overview, then we overlay results from all test scenarios to assess the overall error distribution across the network. Finally, in Section 8.8, we present the error distributions and observe that the errors follow expected patterns.

8.2. Benchmarking the architecture

The benchmarking follows a two-stage process. In the first stage, we compare different model architectures using compact models with similar parameter counts and the same number of layers (five), using four attention heads in the attention-based models. We benchmark across three broad categories: graph-based baseline models (GNNs), non-graph baselines (fully connected neural networks and standard Transformer), and hybrid architectures that integrate graph structure into the Transformer. Once the best-performing architecture is identified, we proceed to the second stage, scaling up that architecture to explore the effects of increased model capacity on performance.

We benchmark several GNN variants: Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017), Graph Attention Networks (GAT) (Veličković et al., 2017), Transformer Convolutions (TransConv, a type of Graph Transformer) (Shi et al., 2021), and PointNet Convolutions (PNC) (Charles et al., 2017). These models differ in how they aggregate information across the graph, enabling various forms of spatial reasoning.

As non-GNN baselines, we include two architectures that do not explicitly model spatial relationships: A fully connected neural network (FC), which processes all input features independently without leveraging any spatial structure; and a Transformer encoder

[2] https://github.com/enatterer/matsim-ile-de-france

Table 3
Comparison of model architectures in terms of complexity, inference efficiency and accuracy. The first four rows represent graph-based baseline models (GNNs), including GCN, GAT, TransConv and PNC. The next two rows correspond to non-graph baselines: a fully connected network (FC) and a standard Transformer encoder (TE). The final two rows explore hybrid architectures that integrate graph structure into the Transformer: one using positional features, and the other applying Graph Transformer layers before the Transformer blocks. These comparisons highlight how incorporating spatial structure impacts predictive performance and efficiency.

Model	Complexity			Pred. time	Performance Metrics			
Type	#Params	#Epochs	Train t	Inference t	R^2	MSE	Pearson	Sp.
Graph Baselines								
GCN	1.31 M	750	24.4 h	19.05 ms	0.74	26.22	0.86	0.47
GAT	1.32 M	750	32.4 h	22.45 ms	0.79	21.79	0.89	0.53
TransConv	1.32 M	600	22.7 h	17.18 ms	0.80	20.57	0.89	0.56
PNC	1.65 M	365	15.1 h	30.45 ms	0.80	20.71	0.89	0.56
Non-Graph B.								
FC	1.42 M	739	10.1 h	0.41 ms	0.28	72.76	0.54	0.43
TE	1.32 M	600	93.1 h	589.47 ms	0.89	11.31	0.94	0.55
Hybrid B.								
TE + Pos.	1.32 M	357	55.3 h	581.59 ms	**0.91**	**8.78**	**0.96**	**0.66**
TransConv + TE	1.49 M	500	72.5 h	461.57 ms	**0.91**	9.04	0.95	**0.66**

(Vaswani et al., 2017), which relies solely on self-attention mechanisms and does not incorporate graph connectivity unless explicitly encoded. We also explored the feasibility of other machine learning methods, such as random forests and gradient boosting (e.g., XGBoost). However, these approaches proved infeasible for our problem scale: The input space is extremely high-dimensional (158,175 features), and the model must predict a large output vector (31,635 nodes). Existing implementations struggle with this combination; vector output support is limited, and training one model per output node is computationally impractical. As a result, we focused our benchmarking on neural network architectures, which are better suited to this scale and structure.

8.2.1. *First stage: benchmarking compact models*

Table 3 summarizes the evaluated architectures in terms of model complexity, predictive accuracy, and computational efficiency. The *Complexity* section reports the number of parameters, total training time, and number of training epochs, reflecting the resources required for each model. While we include training and inference times for completeness, we emphasize that these metrics are secondary for our surrogate modeling context. As detailed in Section 7.4, the true efficiency gain lies in comparing full MATSim simulations with the surrogate inference time, rendering training time of the ML surrogate largely irrelevant. Therefore, the key efficiency metric in practical deployment is inference time - the time required to evaluate a new spatial policy rollout. This metric is averaged across the whole test set and is reported in the *Pred. time* column. Finally, the *Performance Metrics* (R^2, MSE, Pearson, and Spearman correlation) provide a comprehensive view of predictive quality.

From the graph-based models, *TransConv* achieves the best results, with an R^2 of 0.80 and the lowest MSE (20.57) at 1.32 million parameters. *PointNetConv* performs similarly but requires a larger model size and exhibits higher inference time, despite faster training. This makes *TransConv* the most effective choice in terms of both speed and accuracy. The strong performance of Transformer Convolution is driven by its advanced attention mechanism that extends beyond the simpler dot-product or concatenation-based attention used in GAT. Unlike GAT, TransConv learns more expressive query, key, and value representations via multi-layer perceptrons (MLPs), enabling adaptive weighting of neighboring nodes based on complex feature interactions. This facilitates capturing long-range dependencies within the graph - critical in traffic networks where policy impacts propagate beyond immediate neighbors. This capability parallels the success of Transformer-based large language models, which rely on similar attention mechanisms to model long-range context effectively.

Among both graph-based and non-graph-based models, the Transformer encoder significantly outperforms the others, achieving an MSE of 11.31 compared to over 20 for all graph-based models. However, a standard Transformer is inherently permutation-invariant and treats inputs as an unordered set. Without spatial bias, it lacks awareness of the underlying graph structure - such as the spatial layout of the road network - which is crucial for effective traffic modeling.

This observation motivated an investigation into how the expressive power of attention can be preserved while more effectively incorporating spatial structure, leading to the design of *hybrid* architectures. A common strategy is to restrict attention to neighboring nodes, as in the Graph Transformer (TransConv); however, we extend this line of inquiry by exploring additional mechanisms. Specifically, we evaluate two alternative approaches for embedding spatial information into the Transformer architecture, alongside the Graph Transformer baseline:

1. Transformer + Positions: A standard Transformer that incorporates positional coordinates to encode each node's location in the network. In our setting, positions are defined on the dual graph, where nodes represent street segments and their coordinates correspond to the segment midpoints.

2. Graph Transformer Layers First, Then Transformer: A hybrid architecture in which we first apply Graph Transformer layers (TransConv), which outperform the other GNN baselines. These layers integrate local spatial structure, and their output is then passed into a standard Transformer layer to capture longer-range dependencies via unrestricted self-attention.

Note that PointNet Convolutions are unique among the tested graph-based models in that they require explicit positional input and are designed to process unordered sets of spatial points. To assess the impact of positional features more broadly, we also tested the other GNN architectures with and without positional inputs. Since this had negligible effect on performance, we report only the results without positional inputs - except for PNC, where they are essential.

The results clearly demonstrate that the hybrid models consistently outperform the baseline Transformer. Both strategies - incorporating positions and adding graph layers before the Transformer encoder - result in significant performance improvements. Specifically, the hybrid models achieve lower MSE (8.78 and 9.04 vs. 11.31) and a higher Spearman correlation coefficient (0.66 vs. 0.55). These findings underscore the importance of integrating spatial structure into the model, whether through explicit positional information or by leveraging the graph topology, confirming the value of combining self-attention with spatial context.

In contrast, the fully connected network, despite having the fastest inference time (under 1 ms), exhibits substantially lower predictive accuracy. This performance gap underscores the importance of spatial inductive bias in learning effective surrogates for complex, spatially structured systems like traffic networks.

8.2.2. Second stage: benchmarking scaled models

Given that hybrid architectures combining Transformers and graph structures achieved the best performance in the compact model evaluation, an extended architecture search was conducted to identify the model with the highest accuracy at scale.

Models were tested under a comparable core configuration - four attention heads in both Graph Transformer (TransConv) and Transformer Encoder (TE) - while varying depth and overall parameter count. To support stable training as model capacity increased, we tuned learning rates, maximum epochs, and early stopping criteria accordingly. Consequently, training times and epoch counts are not directly comparable to those in Table 3.

The results of this fine-tuning process are presented in Table 4, which reports each model's number of layers (#L), parameter count (#Params), inference time, and key performance metrics including R^2, MSE, and Pearson and Spearman correlation coefficients. From the table, the following conclusions can be drawn:

1. Overall, the accuracy of all hybrid models is quite consistent. Nearly all models achieve an R^2 of 0.91, with MSE values ranging from 8.75 to 10.86. The Pearson correlation coefficient is generally 0.95 or 0.96, and the Spearman correlation coefficient is mostly between 0.66 and 0.67. The only exception is "TransConv (1) + TE (4)", which suggests that a single layer of TransConv is insufficient; at least two layers are needed, as demonstrated by the other models in the "Graph + Transformer" section.
2. Regarding scaling, for the Transformer + Positions architecture, five layers of Transformer Encoder are sufficient - adding more layers (ten) does not improve accuracy metrics.
3. The highest performance, albeit by a narrow margin, was obtained with the architecture combining Transformer, positions, and Graph Layers first. This model was subsequently used to generate the reported results.

The inference time remains well within practical limits for policy evaluation, with the most accurate model requiring only 462 ms per scenario - about 5,000 times faster than a full MATSim simulation. While inference speed could be further improved using standard techniques such as pruning or quantization, this study prioritizes predictive accuracy over runtime optimization.

Table 4
Performance, Complexity, and Inference Time for Extended Transformer Architectures. This table compares models that combine Transformer Encoder (TE) layers with spatial information. The section 'Transformer + Positions' adds positional features, 'Graph + Transformer' incorporates Graph Transformer (TransConv) layers, and 'Graph + Positions + TE' combines both Graph Transformers and positional information. The number in parentheses indicates the number of layers per model component.

Model	Complexity		Pred. t.	Performance Metrics			
Structure	#L.	#Params	Inference	R^2	MSE	Pear.	Sp.
Transformer + Positions							
TE + Pos. (5)	5	1.32 M	581.59 ms	0.91	8.78	0.96	0.66
TE + Pos. (10)	10	4.45 M	1.54 s	0.91	8.8	0.96	0.66
Graph Transformer + TE							
TransConv (1) + TE (4)	5	1.24 M	595.09 ms	0.89	10.86	0.95	0.60
TransConv (2) + TE (3)	5	1.49 M	461.57 ms	0.91	9.04	0.95	0.66
TransConv (3) + TE (7)	10	3.42 M	1.09 s	0.91	8.91	0.96	0.67
TransConv (2) + TE (10)	12	4.6 M	1.55 s	0.91	8.98	0.95	0.66
Graph Tr. + Positions + TE							
TransConv (2) + TE (3) + Pos.	5	1.49 M	462.47 ms	**0.91**	**8.75**	**0.96**	**0.67**

As discussed in Section 7.4, the key efficiency benchmark is the comparison between full simulation runtimes and surrogate inference time. Since even the largest model is several orders of magnitude faster, inference time is not a limiting factor.

We therefore identify *TransConv (2) + TE (3) + Pos.* as the best-performing surrogate, striking a strong balance between accuracy and practical deployability in policy planning workflows. In conclusion, we find that this compact model with 1.5 million parameters, consisting of two layers of Graph Transformer (with incorporated positional features), followed by a three-layer Transformer Encoder, effectively combines the power of Transformers with spatial structure.

8.3. Overall performance across street types

To evaluate the model's performance across the test set, we analyze key aggregated performance metrics for different street types.

Firstly, given the central role of variance in our evaluation, we clarify the various variance notations employed throughout this paper, summarized in Table 5. In particular, $\sigma^2_{e,b}$ denotes the variance of simulation outcomes for a single edge e in the base case scenario. The mean of these variances across all edges in a subset E_t - typically representing a specific road type t - is expressed as $\sigma^2_{t,b}$. Lastly, the variance for street type t under policy p, or *total sum of squares* ($SS_{\text{tot},t,p}$), is defined by Eq. 13.

This notation clearly distinguishes between the variability due to simulation stochasticity within individual edges and the variability observed among different edges belonging to the same road type.

Table 6 presents the results per street type. It consists of the columns street type, the number of roads that it consists of (Count), the average traffic volume counted throughout a day in the base case $\bar{v}_t$ (as discussed in Section 3.1), the average variance in the base case $\sigma^2_{t,b}$, the MSE per street type (Eq. 7), the MAE per street type (Eq. 15) and R^2 (Eq. 17), and the variance across street types $SS_{\text{tot},t,p}$ for better interpreting R^2. The table is sorted by its traffic volume in the base case $\bar{v}_t$. Since only a single policy p was tested, the subscript p is omitted in this section for clarity.

The street network consists of approximately 35,000 street segments, with primary, secondary, and tertiary roads constituting about 13,000 of these. Residential roads make up the largest category (nearly 12,000 roads), whereas living streets are the least common, with only 732 street segments. The policy of "50% Capacity Reduction" has been implemented in all primary, secondary, and tertiary roads if in the relevant scenario. The average traffic volume varies significantly depending on the street type. Trunk roads exhibit the highest base volume, with an average of 484.60 vehicles per day, while living streets have the lowest, averaging just 4.81 vehicles per day. We make the following key observations:

1. *Higher traffic volume in the base case leads to greater model accuracy*: Across all street types, higher base case traffic volumes are strongly associated with higher absolute errors in terms of MSE and MAE.

2. $\sigma^2_{t,b}$ and $SS_{\text{tot},t}$ are highly correlated, with a Pearson correlation coefficient of 0.84. They do, however, measure different aspects of variability:
   - $\sigma^2_{t,b}$ quantifies variance within simulation runs across multiple random seeds for the base case.
   - $SS_{\text{tot},t}$ captures variance across all roads of a given type, reflecting differences in traffic volume changes between roads.
3. $\sigma^2_{t,b}$ *and* $SS_{tot,t}$ *generally correlate with higher base traffic volume and accuracy, except for trunk roads*: For most street types, greater traffic volume corresponds to higher variance across multiple simulation runs and across all streets of a given type, leading to improved predictive accuracy. However, trunk roads deviate from this trend.
4. *Trunk roads form an exception to the variance-accuracy relationship*: Despite having the highest base traffic volume - approximately four times that of primary roads - trunk roads exhibit lower variance ($\sigma^2_{t,b}$, and $SS_{\text{tot},t}$) than primary roads.

Table 5
Summary of variance notations used in the analysis.

Notation	Definition	Explanation
$\sigma^2_{e,b}$	$\frac{1}{\lvert R\rvert}\sum_{r=1}^{\lvert R\rvert}(v_e^r-\bar{v}_e)^2$	Base case variance on a single edge e across multiple simulation runs R.
$\sigma^2_{t,b}$	$\frac{1}{\lvert E_t\rvert}\sum_{e\in E_t}\sigma^2_{e,b}$	Average base case variance across all edges e in subset E_t (e.g., all edges of road type t).
$SS_{\text{tot},t,p}$	$\frac{1}{\lvert E_t\rvert}\sum_{e\in E_t}(y_{e,p}-\bar{y}_{t,p})^2$	Variance among all edges in road type E_t.

Table 6
Model performance metrics by street type (aggregated over the test set, as defined in Section 5.1).

Street Type t	Count	$\bar{v}_t$	$\sigma^2_{t,b}$	MSE_t	MAE_t	R^2_t	$SS_{\text{tot},t}$
All Roads	31,635	50.91	8.57	8.75	1.71	0.91	101.44
Trunk	933	**484.60**	18.32	20.79	3.09	0.90	210.82
Primary	5,295	114.90	19.22	19.52	2.95	**0.95**	410.29
Secondary	4,328	51.16	12.72	12.96	2.41	0.84	81.01
Tertiary	3,792	35.35	10.75	10.22	2.19	0.77	44.34
Residential	11,796	13.27	4.33	4.53	1.27	0.73	16.41
Living Streets	732	4.81	**1.23**	**1.44**	**0.66**	0.68	**4.39**

- $\sigma^2_{t,b}$ being lower for trunk than for primary roads means that trunk roads remain stable across different simulation runs with random seeds. For example, in Paris, the motorway ring street around the city, the Boulevard PÃ©riphÃ©rique, consists of trunk street segments. Since this street is less affected by route-choice variability compared to other roads, such as primary roads, traffic volume remains relatively consistent, with minimal stochastic fluctuations during simulation runs.
- The relatively low $SS_{\text{tot},t}$ for trunk roads suggests that trunk roads indicates that traffic patterns among trunk roads are more uniform compared to primary roads. This suggests that individual trunk roads exhibit similar traffic behavior, whereas primary roads show greater variability.
- Additionally, the low $SS_{\text{tot},t}$ contributes to the relatively low R^2. Since R^2 measures the proportion of variance explained by the model, lower variance inherently results in lower R^2 values - even for roads with high traffic volumes. The phenomen of lower variance leading to lower R^2 values, and a seeming contradiction between absolute error metrics (MSE/MAE) and R^2 is further explored in the next section (Section 8.4), where we examine the distinction between roads with and without capacity reduction.

5. *Poor performance on low-volume roads*: In terms of R^2, the model performs worst on roads with comparatively smaller traffic volumes, such as residential ($R^2 = 0.73$) and living streets ($R^2 = 0.68$), indicating that these street types exhibit the least predictable traffic patterns.

8.4. Impact of capacity reduction on model performance

Table 7 presents the model's performance across different street types, distinguishing between roads with and without capacity reduction. Since reductions apply only to primary, secondary, and tertiary roads, the analysis is limited to these categories.

The number of observations (*# Obs.*) represents the total street segments evaluated across all test scenarios. For instance, the 541,671 observations for primary roads with capacity reduction indicate the total number of primary street segments affected at least once across scenarios. Overall, the dataset includes approximately 1.3 million street segments with capacity reduction and 4.3 million without.

This table differs from Table 6 in two key ways. First, it explicitly separates performance metrics for roads with and without capacity reduction, enabling a direct comparison. Second, the *# Obs.* column replaces the *Count* column from Table 6, as a given street segment may be affected by capacity reduction in some scenarios but not in others. For these reasons, the results are reported independently.

A consistent pattern emerges across all street types: streets with capacity reduction exhibit higher R^2, Pearson, and Spearman correlation coefficient values, but also show higher MSE and MAE. This apparent contradiction between error metrics warrants further investigation. Looking closer, we observe also patterns in the characteristics of the two types:

1. Streets with capacity reduction exhibit significantly higher variance, with primary streets, for example, showing a variance of 1,228.67 compared to just 93.76 for those without. Across all categories, variance for streets with capacity reduction is approximately eleven times greater than for streets without. This increased variance naturally inflates R^2 values, as variance in the target variable directly affects its computation:

$$R^2_t = 1 - \frac{MSE_t}{SS_{\text{tot},t}}$$

The higher variance is expected, as streets with capacity reduction experience more extreme fluctuations in traffic across scenarios. Some street segments are *directly affected* in one scenario (where their capacity is reduced) but only *indirectly influenced* in another (where no reduction occurs). This alternation between scenarios leads to a wider spread in observed traffic volume changes, explaining the high variance. Consequently, while R^2 may indicate strong model performance, it can be misleading when variance is high, as absolute errors remain significant despite the improved relative fit.

Table 7
Model performance metrics by street type, with and without Capacity Reduction.

Street Type t	# Obs.	$\sigma^2_{t,b}$	MSE_t	MAE_t	R^2_t	$SS_{\text{tot},t}$	Pearson	Sp.
P/S/T, Cap. Red.	1,344,535	18.12	20.81	3.31	**0.97**	637.48	**0.98**	**0.92**
P/S/T, No Cap. Red.	4,236,105	13.66	**12.86**	**2.32**	0.77	58.30	0.88	0.66
Primary, Cap. Red.	541,671	23.55	28.14	3.85	**0.98**	1228.67	**0.99**	**0.96**
Primary, No Cap. Red.	1,661,049	17.81	**16.7**	**2.65**	0.81	93.76	0.90	0.71
Secondary, Cap. Red.	405,170	16.44	18.51	3.21	**0.91**	206.07	**0.96**	**0.90**
Secondary, No Cap. Red.	1,395,278	11.61	**11.35**	**2.18**	0.73	44.07	0.85	0.63
Tertiary, Cap. Red.	397,694	12.39	13.17	2.68	**0.88**	107.46	**0.94**	**0.86**
Tertiary, No Cap. Red.	1,179,778	10.20	**9.23**	**2.03**	0.61	24.36	0.78	0.60

• "P/S/T" stands for "Primary, Secondary, and Tertiary" roads.
• "Cap. Red." refers to roads affected by capacity reduction policies, while "No Cap. Red." refers to roads unaffected by these policies across the test set.
• "Pearson" denotes the Pearson correlation coefficient, and "Sp." refers to the Spearman correlation coefficient.

2. Roads with capacity reduction systematically exhibit higher base case variance ($\sigma^2_{t,b}$) compared to those without. Investigating this, we find a difference arises from a sampling bias, as detailed in Appendix A. Specifically, our selection method tends to choose districts with higher base case traffic volumes, which in turn have higher variance. Thus, the observed base case variance differences stem from a sampling artifact rather than a causal effect of capacity reduction, influenced by district selection frequencies and their base volumes.

The first observation explains the high R^2 values observed for roads with capacity reduction. Together with the Pearson and Spearman correlation coefficients, these results confirm that the model performs better on roads with capacity reduction than on those without.

8.5. Disentangling model error from simulation noise

In Section 6, we formally derived how the expected mean squared error (MSE) between surrogate model predictions and stochastic simulation outputs decomposes into two additive components: the irreducible variance from the simulation (σ^2_e) and the squared model error $\left(\epsilon'_e\right)^2$). This follows from treating the simulation output $Y_e = \Delta b_e + \epsilon_e$ and model output $\hat{Y}_e = Y_e + \epsilon'_e$ as random variables. The final decomposition in Eq. 32 states:

$$\mathbb{E}[\mathrm{MSE}(Y, \hat{Y})] = \frac{1}{|E|} \sum_{e \in E} \sigma^2_{e,p} + {\epsilon'_e}^2,$$

implying that even a perfect model ($\epsilon'_e = 0$) cannot reduce MSE below σ^2_e, the aleatoric uncertainty inherent to stochastic simulations. The aleatoric component sets a natural performance limit independent of model quality.

In practice, the true expected output $\bar{y}_{e,p} = \mathbb{E}[Y_{e,p}]$ is unknown, as it requires multiple independent simulation runs per policy. Thus, observed MSE includes both model error and simulation noise. We express the empirical MSE as:

$$\mathrm{MSE}_{e,p} = \underbrace{\sigma^2_{e,p}}_{\text{Aleatoric Variance}} + \underbrace{\mathrm{Bias}^2_{e,p}}_{\text{Squared Bias}} + \underbrace{\mathrm{Var}_{e,p}}_{\text{Model Variance}}, \tag{33}$$

where:

1. $\sigma^2_{e,p}$ is the variance of simulation outcomes for edge e under policy p,
2. $\mathrm{Bias}_{e,p}$ quantifies the difference between the model's prediction and $\bar{y}_{e,p}$, and
3. $\mathrm{Var}_{e,p}$ is the surrogate model's prediction variance due to data or model stochasticity (e.g., in ensembling or training randomness).

Directly computing $\sigma^2_{e,p}$ is computationally expensive, as it requires running each policy scenario multiple times. We therefore introduce the following assumption:

Assumption 1. If $\sigma^2_{e,p} = \sigma^2_{e,b}$, where $\sigma^2_{e,b}$ is the simulation variance in the base case (without any policy intervention), then the base case variance can serve as a proxy for aleatoric variance under policy p.

This assumption is motivated by the observation that the core stochastic processes of the simulation do not change significantly under moderate policy interventions, especially when agent behavior and routing mechanisms remain structurally similar. If Assumption 1 holds, we could estimate the aleatoric variance from the base case to estimate the irreducible error component in the MSE.

This leads to an important distinction: in many machine learning applications, aleatoric uncertainty is hard to quantify due to unknown input noise. In contrast, ABMs allow precise empirical estimation of simulation-induced noise, making this a rare case where one could approximate the irreducible component of model error a priori.

The remainder of this section proceeds under this assumption.

Table 8 examines the contribution of base case variance to MSE across street types. The 95% confidence intervals reported in Table 8 were computed via bootstrapping with 1,024 resamples. In each resample, street segments were drawn with replacement within each street type, and base case variance, total MSE, and their ratio were recalculated. This yields robust estimates of uncertainty without relying on parametric assumptions.

A high share indicates that a substantial portion of the prediction error arises from inherent stochasticity in the simulation, and is thus fundamentally irreducible. Notably, a 100% share implies that, provided our assumption holds, the MSE is entirely explained by the base case variance, meaning the surrogate model is operating near its theoretical limit and further improvements would require reducing input uncertainty. A share above 100% indicates that the base case variance $\sigma^2_{e,b}$ overestimates the true variance $\sigma^2_{e,p}$, as it serves only as a proxy. Further, the model's error is close to the inherent simulation noise, approaching close to ideal performance.

The proportion of MSE explained by base case variance is highest for tertiary roads without capacity reduction at *110.59% [111.15%, 111.94%]*, and lowest for living streets at *85.22% [84.16%, 86.23%]*. This suggests that for tertiary roads without capacity reduction, all the model's error is attributable to inherent traffic variability, whereas for living streets, a portion of the error stems from other factors, such as complex interactions or model limitations. The observation that primary roads without capacity reduction, secondary roads without capacity reduction, and tertiary roads with or without capacity reduction all exceed 100% indicates that, in these cases, the base case variance $\sigma^2_{e,b}$ overestimates the true variance $\sigma^2_{e,p}$.

Across all road types without capacity reductions, both the base case variance and MSE are consistently lower compared to roads with capacity reductions or the combined dataset. However, a larger proportion of the prediction error is explained by the base case

Table 8
Base Case Variance and its Contribution to MSE. The 95% confidence intervals for all reported values are computed via bootstrapping with 1,024 resamples; for each resample, street segments are drawn with replacement within each street type, and base case variance, total MSE, and their ratio are recomputed. This provides a robust, non-parametric estimate of uncertainty.

Street Type t	$\sigma^2_{t,b}$ [95%]	MSE_t [95%]	Share of $\sigma^2_{t,b}$ in MSE_t [95%]
All Roads	8.57 [8.56, 8.58]	8.75 [8.73, 8.77]	97.95% [97.76%, 98.13%]
Trunk	18.32 [18.27, 18.37]	20.79 [20.67, 20.92]	88.12% [87.59%, 88.66%]
Primary	19.22 [19.19, 19.24]	19.52 [19.43, 19.61]	98.47% [98.04%, 98.89%]
Primary, Cap. Red.	23.55 [22.81, 22.91]	28.14 [27.85, 28.48]	83.66% [80.26%, 82.08%]
Primary, No Cap. Red.	17.81 [18.00, 18.06]	16.7 [16.64, 16.76]	106.63% [107.62%, 108.27%]
Secondary	12.72 [12.70, 12.74]	12.96 [12.91, 13.01]	98.14% [97.81%, 98.46%]
Secondary, Cap. Red.	16.44 [15.97, 16.04]	18.51 [18.39, 18.64]	88.82% [85.85%, 87.03%]
Secondary, No Cap. Red.	11.61 [11.74, 11.79]	11.35 [11.30, 11.39]	102.34% [103.30%, 104.05%]
Tertiary	10.75 [10.73, 10.76]	10.22 [10.19, 10.25]	105.17% [104.88%, 105.49%]
Tertiary, Cap. Red.	12.39 [12.08, 12.14]	13.17 [13.10, 13.25]	94.05% [91.38%, 92.44%]
Tertiary, No Cap. Red.	10.20 [10.27, 10.31]	9.23 [9.19, 9.26]	110.59% [111.15%, 111.94%]
Residential	4.33 [4.33, 4.34]	4.53 [4.52, 4.54]	95.67% [95.46%, 95.89%]
Living Streets	1.23 [1.22, 1.24]	1.44 [1.42, 1.46]	85.22% [84.16%, 86.23%]

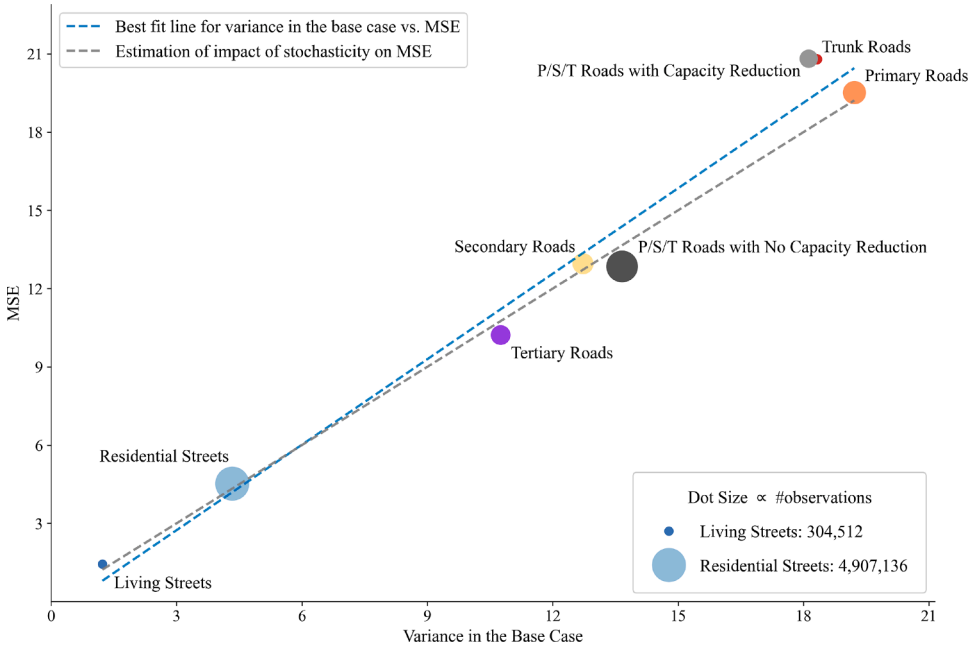


Fig. 7. Relationship between base case variance and MSE across street types. The strong correlation suggests that base case variance is a reliable predictor of model error across different policy scenarios.

variance, indicating that the model's errors in these cases primarily arise from inherent stochasticity. This suggests relatively better predictive performance on roads without capacity reductions compared to those with them.

Across all street types, the 95% confidence intervals for both the base case variance and the MSE are generally narrow, often within a range of 0.02 to 0.10. While the resulting confidence intervals for the share of variance in MSE are somewhat wider - ranging from around 0.3 to 2.1 percentage points - they still indicate a high degree of precision. This suggests that the estimated contributions of variance to MSE are statistically robust and not driven by random variability in the data.

Fig. 7 illustrates the relationship between model error and simulation stochasticity across different street types. Simulation stochasticity, measured as $\sigma^2_{t,b}$, is shown on the x-axis, while model error, represented by MSE, is displayed on the y-axis. Each dot corresponds to a street type, positioned according to its variance and MSE, with dot size reflecting the number of observations. The blue dashed line represents the best-fitting regression line, and the grey dashed line denotes the theoretical lower bound on the MSE, as established in Section 6.

The strong correlation between MSE and base case variance confirms that simulation stochasticity is a major driver of model error. The street types align well with the expected relationship between the MSE and the simulation stochasticity.

We conclude that in our use case, the base case variance can serve as a reliable estimator for prediction uncertainty. This assumption holds also because policy interventions affect only a limited subset of roads in each scenario. While up to 13,415 roads (namely

primary, secondary, and tertiary roads) of the 31,635 roads are eligible for capacity reduction, in practice, only about 3,000 roads (roughly 10% of the total network) are modified per scenario, assuming an even distribution across four districts on average. While some deviations are expected, our findings suggest that base case variance provides a reliable and easily obtainable approximation of aleatoric variance arising from simulation stochasticity in surrogate models for agent-based models.

In summary, aleatoric variance establishes a fundamental upper bound on model performance. When a large portion of the mean squared error (MSE) is attributable to the base case variance, it suggests that further model improvements are unlikely to significantly reduce the total error. Conversely, if this share is low, there remains potential to reduce bias or variance, indicating possible underfitting or overfitting in the model. This variance decomposition offers a practical framework for guiding model evaluation and interpreting the accuracy of surrogate models when dealing with inherently stochastic simulation outputs.

8.6. Evaluation across multiple metrics

Fig. 8 presents a radar plot that visualizes model performance across R^2, Pearson and Spearman correlation coefficients, sorted by their R^2 value. For clarity, the radar plot groups streets with capacity reduction and those without together. Each street type exhibits a distinctive polygonal trace, forming a triangle-like shape with relatively consistent scores across metrics. The radial grid lines at 0.2 intervals highlight substantial performance differences, with deviations of up to 0.3 - 0.4 between the best- and worst-performing categories.

The radar plot shows that the trends observed in Section 8.3 and Section 8.4, corresponding to Table 6 and Table 7, remain consistent across evaluation metrics:

1. *General Hierarchy of Street Types*:
   The ranking of street types by accuracy is consistent across all error metrics. Primary, secondary, and tertiary roads with capacity reductions consistently achieve the highest performance, followed closely by primary and trunk roads. Next are secondary and tertiary roads, then primary, secondary, tertiary roads without capacity reductions, with residential and living streets showing the lowest performance across all metrics.
2. *Impact of Capacity Reductions:* The trend observed in Table 7 is also reflected across additional evaluation metrics: primary, secondary, and tertiary roads with capacity reductions consistently outperform their counterparts without capacity reductions across all metrics.

Thus, the consistent triangle shapes across metrics suggest that the model's performance is stable across different evaluation criteria. This indicates that variations in accuracy stem from inherent differences between street types rather than inconsistencies in the evaluation metrics.

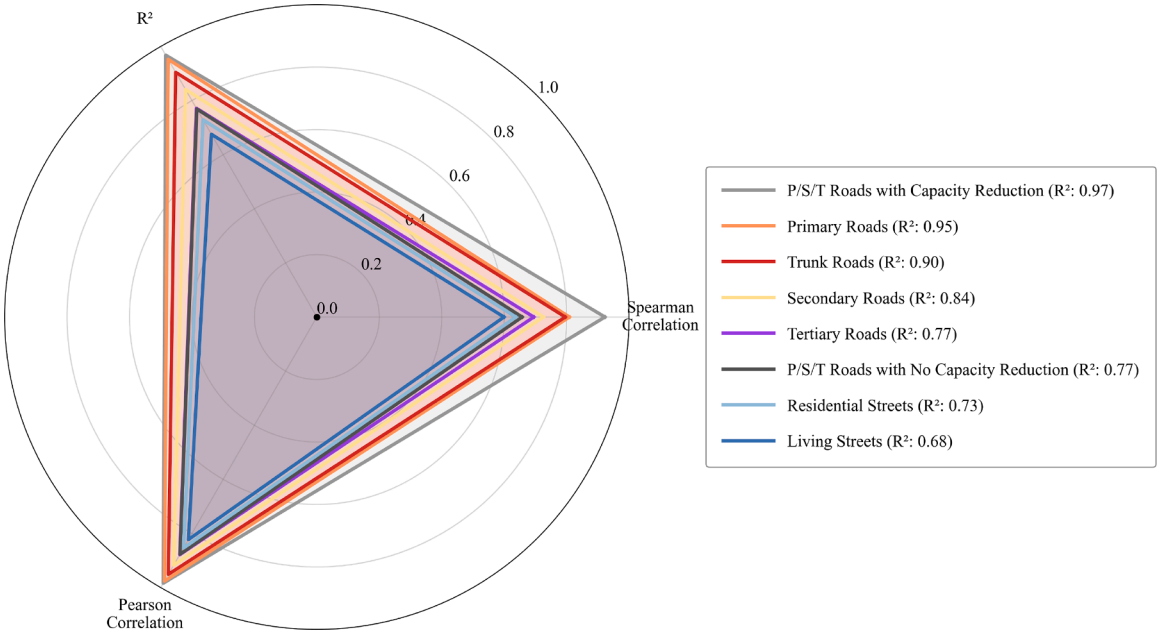


Fig. 8. Radar plot comparing model performance metrics across different street types, sorted by R^2 value.

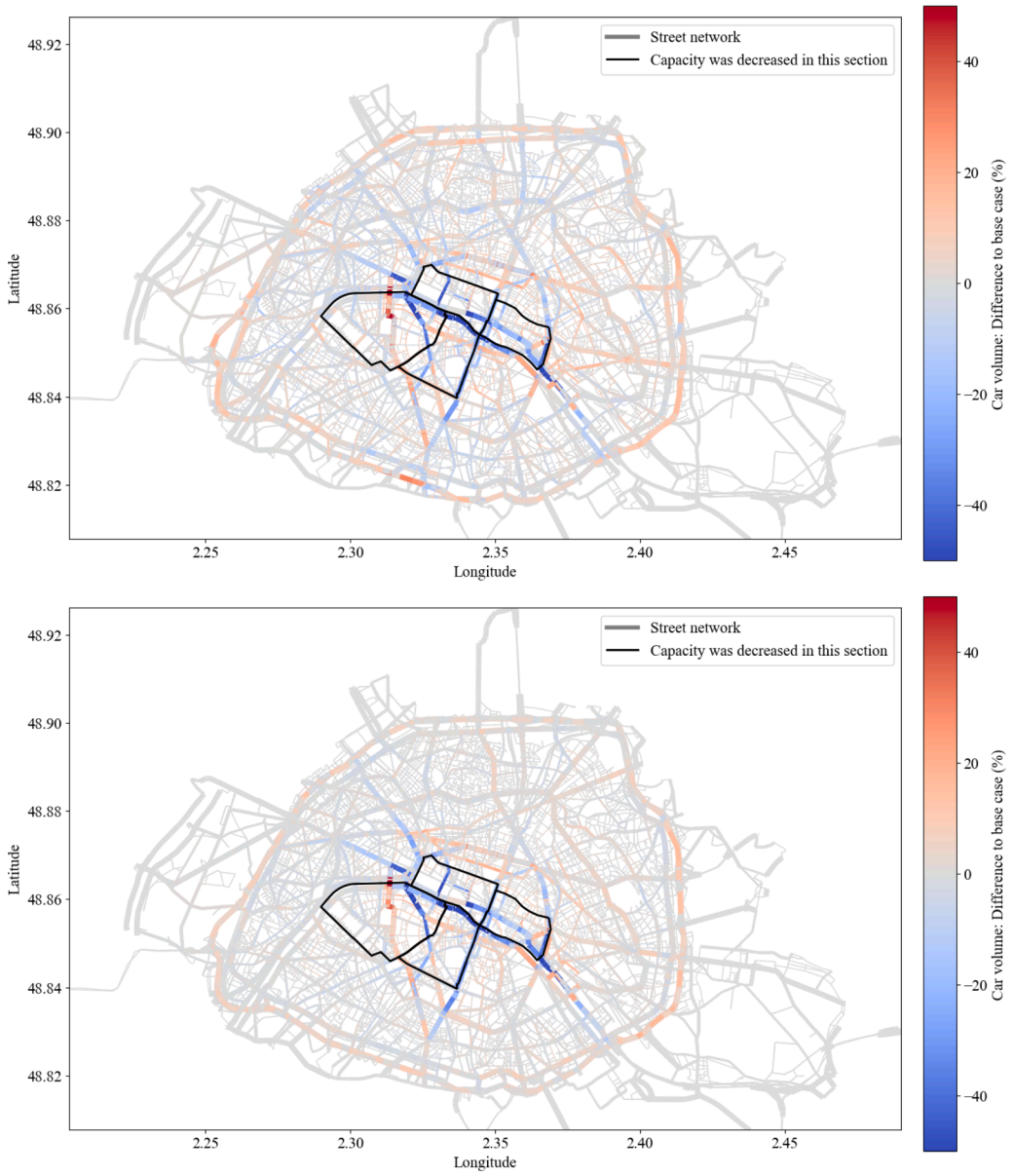


Fig. 9. Test case 1: Simulated (top) vs. predicted (bottom) change in daily traffic volume.

8.7. Network-level evaluation: visual analysis and interpretation

To extend our analysis from link-level to network-level performance, we examine both specific test cases and aggregate patterns.

8.7.1. Individual test case analysis

We first present three randomly selected test cases from our test set in Figs. 9, 10, and 11. These figures compare the outputs of MATSim (top) with our surrogate model predictions (bottom), showing changes in traffic volume relative to the base case (see Section 3.1). The color scale ranges from blue (50% decrease) to red (50% increase) in traffic volume. The visualization approach is inspired by Bogenberger and Weikl (2012).

Test case 1 models a realistic scenario with capacity reductions focused in the central Parisian districts, reflecting the city's actual aggressive policies in these areas. Test case 2 considers capacity reductions in three distinct, separate zones within the city. Test case 3 represents a more extensive scenario, with capacity reductions spread across multiple districts, indicating strong, city-wide interventions.

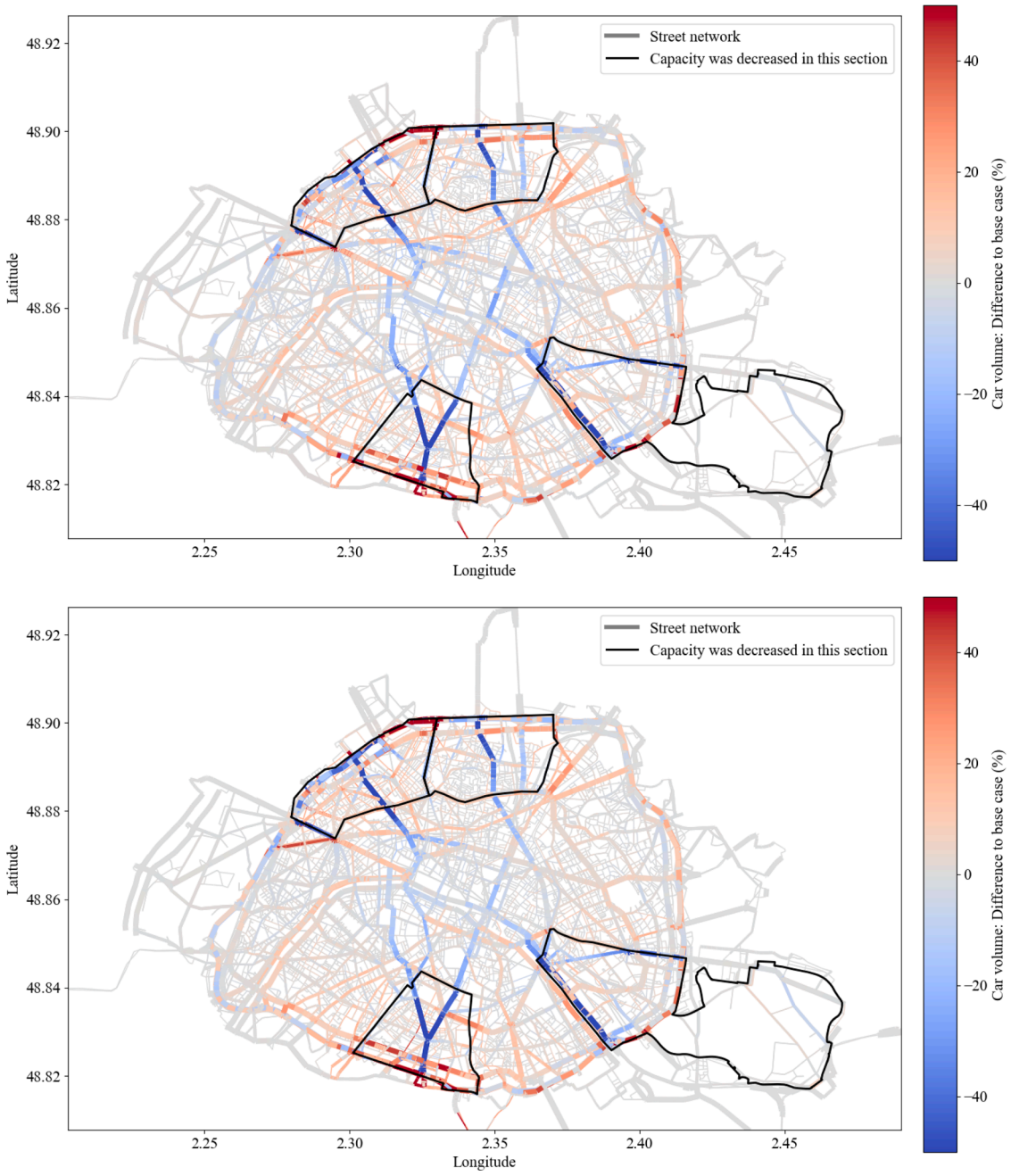


Fig. 10. Test case 2: Simulated (top) vs. predicted (bottom) change in daily traffic volume.

In terms of performance, test case 1 yields an MSE of 1.53 and an R^2 of 0.71. By comparison, test cases 2 and 3 show improved results, with lower MSEs (1.24 and 1.27) and much higher R^2 values (both 0.96), suggesting that the model predicts more accurately when capacity reductions are distributed either across multiple separate areas or widely across the city.

The model demonstrates strong performance in predicting traffic reductions due to capacity reduction policies, visible as blue segments in the plots. Further, the model sometimes does not accurately represent evasive behavior: when comparing the top of the figure (MATSim output) with the bottom of the figure (surrogate model output), red street segments appear more or less pronounced in the surrogate model's output than in the MATSim results. This suggests that the model may sometimes over- or underestimate evasive behavior. Addressing this issue would likely require additional training data to improve the model's ability to differentiate between genuine evasive maneuvers and regular traffic patterns.

These findings align with our quantitative analysis showing generally higher accuracy for roads with capacity reduction and higher traffic volumes, across all evaluation metrics (Section 8.6).

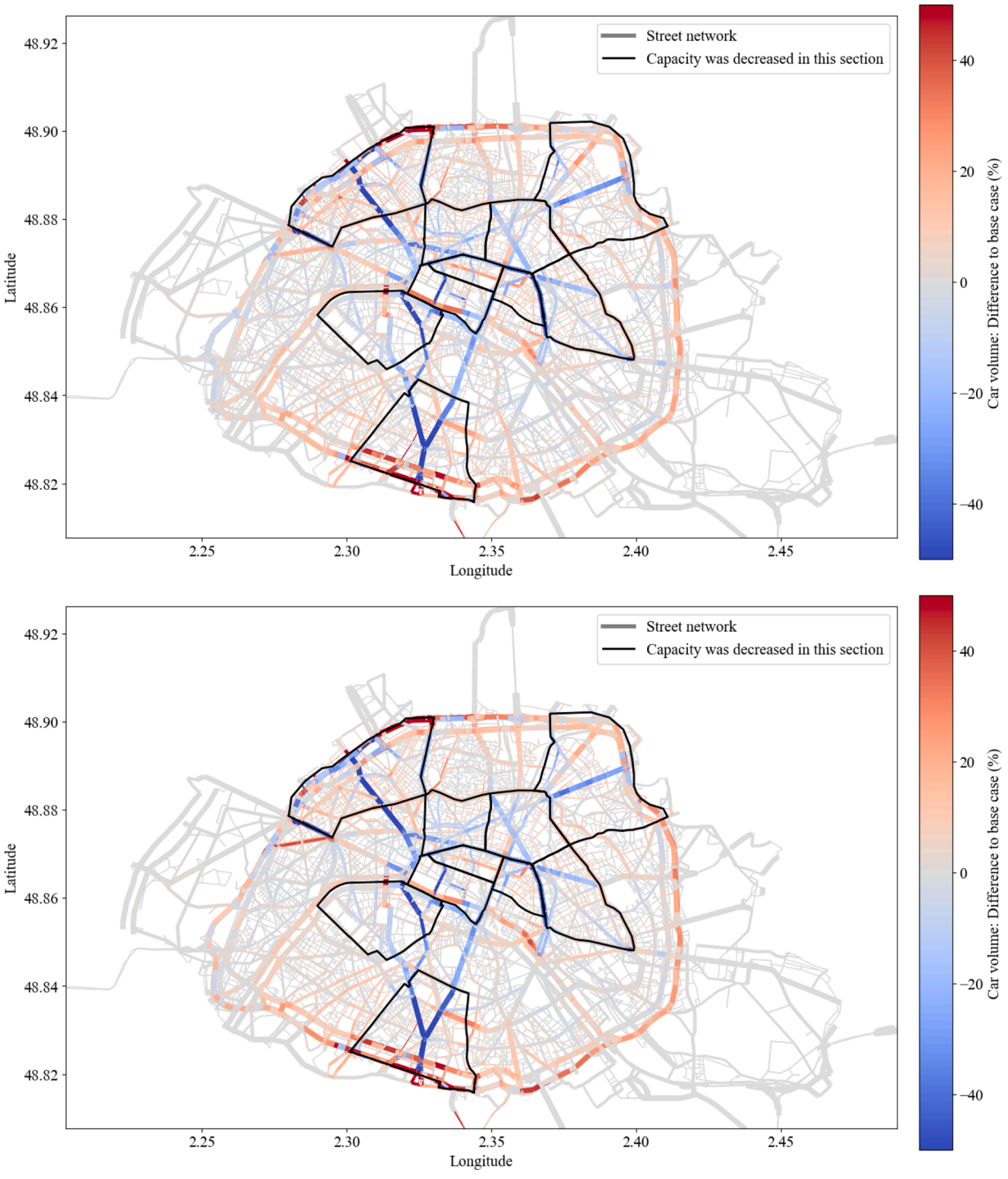


Fig. 11. Test case 3: Simulated (top) vs. predicted (bottom) change in daily traffic volume.

8.7.2. Aggregate pattern analysis

To assess the spatial distribution of prediction errors across Paris, we computed and visualized two complementary maps: one showing absolute differences and the other showing relative differences between predicted and actual traffic volumes (Figs. 12 and 13).

For computing this, we first computed difference plots for each test scenario by calculating the discrepancy between MATSim-simulated and model-predicted traffic volume changes at the edge level. As illustrated in Figs. 9, 10, and 11, we then aggregated these differences across all test cases to compute the average prediction error δ_e for each edge $e \in E$, providing comprehensive assessment

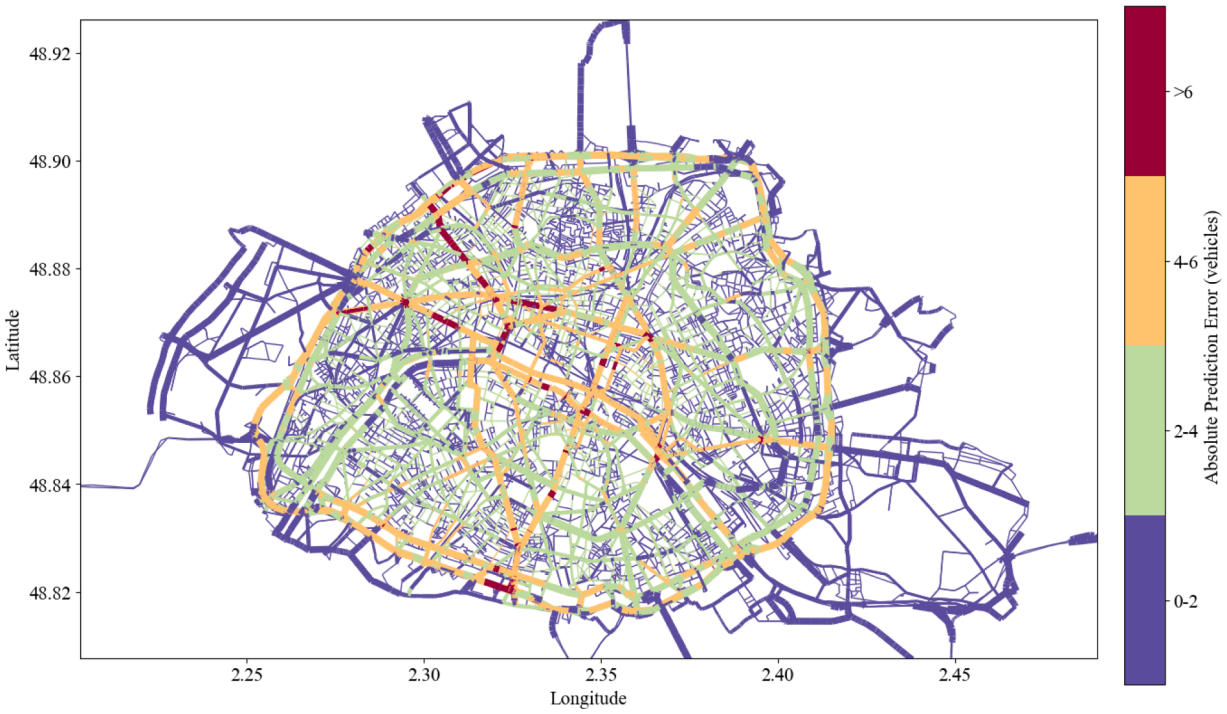


Fig. 12. Absolute difference (MAE) between predicted and actual over all roads, over all observations in the test set.

of model performance across the entire test set:

$$\delta_e = \sum_{s \in S} \frac{|y_{e,p,s} - \hat{y}_{e,p,s}|}{y_{e,p,s}} \tag{34}$$

The absolute differences map visualizes raw prediction errors in vehicle counts, with discrete thresholds at 2, 4, and 6 vehicles to highlight varying levels of prediction accuracy. Meanwhile, the relative differences map normalizes errors by the base traffic volume of each street segment, using 10%, 25%, and 50% thresholds to account for differences in traffic scale across various street types and locations.

This dual visualization approach reveals an interesting spatial pattern: while absolute errors are higher in central areas (decreasing from over 6 to 0 - 2 vehicles from center to periphery), relative errors show an inverse relationship, increasing from 0 - 10% on the main roads to over 50% in peripheral areas.

This can be explained by the very low traffic volumes in peripheral areas: since fewer vehicles travel these roads, absolute prediction errors remain small. However, due to the inherently higher stochasticity in low-volume traffic, relative deviations become more pronounced. Thus, the model's relative performance appears worse in the outskirts, even though the magnitude of the error is low.

In both representations (relative and absolute), it is evident that the model achieves high accuracy on the main street network. This aligns with the results in Section 8.7, indicating that roads with higher traffic volumes show generally higher accuracy (across all metrics, see Section 8.6).

8.8. *Histograms of prediction errors*

To evaluate the distributional characteristics of model prediction errors, we compute residuals as $\hat{y}_e - y_e$ for each road segment e in the test set. Fig. 14a presents the resulting histogram of these residuals. The distribution is approximately symmetric and centered near zero, indicating that the model exhibits minimal systematic bias overall.

To further explore these patterns, we disaggregate the residuals by road type. Fig. 14 displays histograms for three representative categories: trunk, primary and living streets. Other road types were omitted due to their visual and statistical similarity. Across categories, the histograms exhibit generally consistent patterns.

Trunk and primary roads show broader error distributions ($\sigma = 4.56$ and $\sigma = 4.52$, respectively), while living streets show narrower error distributions ($\sigma = 1.29$). This indicates that model uncertainty is not uniform across street types; specifically, roads with higher traffic volumes tend to have larger residual prediction errors.

This analysis complements the network-level error analysis presented in Section 8.7.

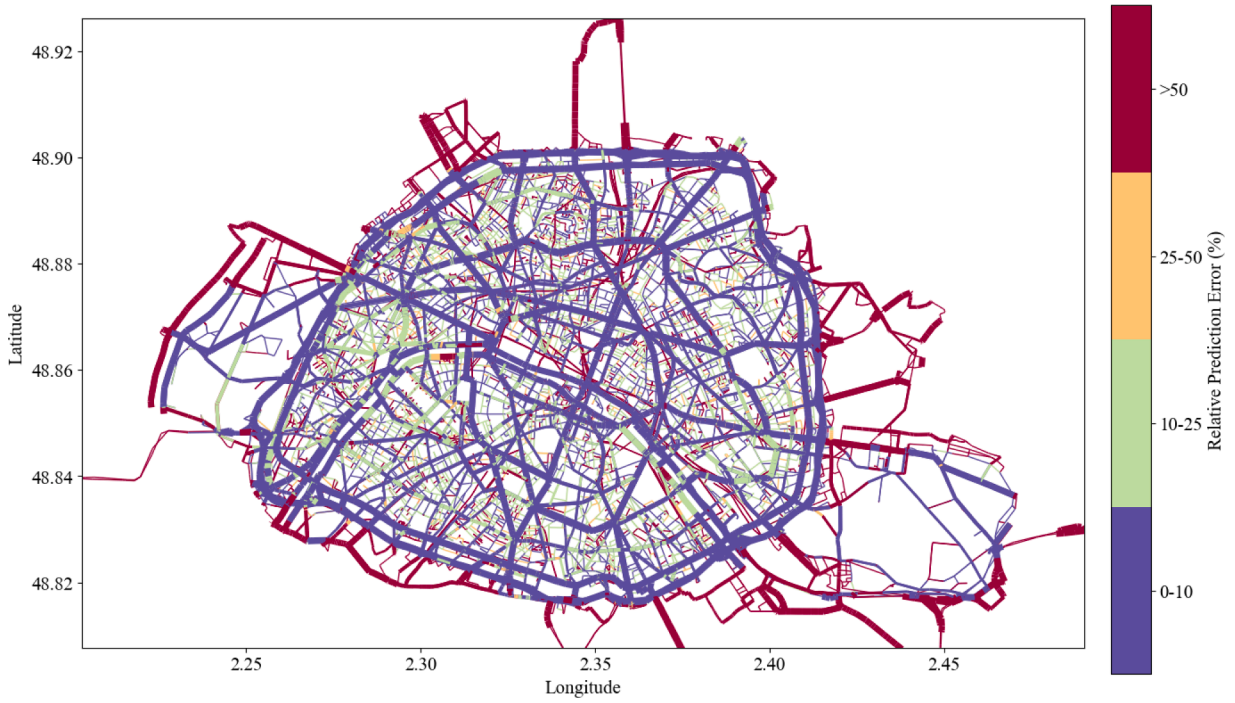
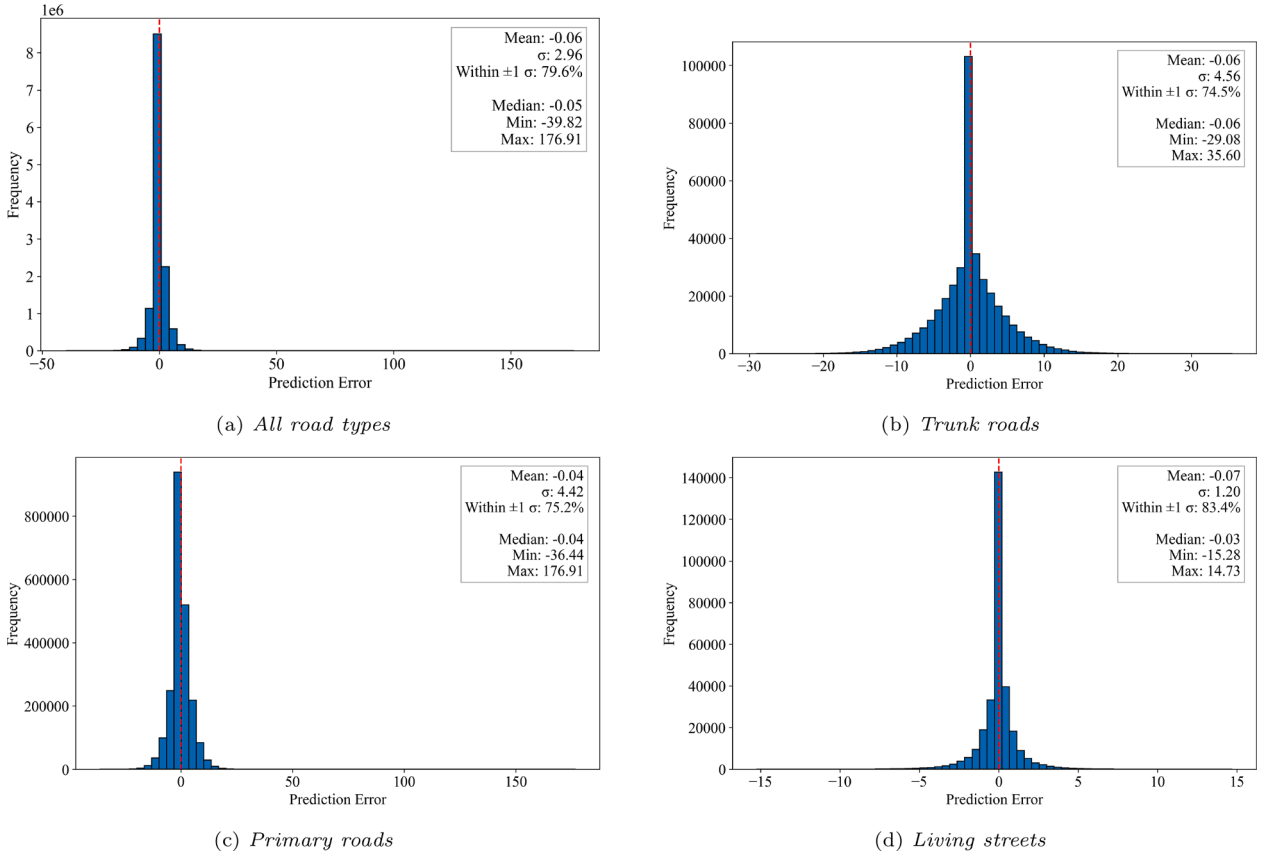


Fig. 13. Relative difference (MAE) between predicted and actual over all roads, over all observations in the test set.



(a) *All road types* (b) *Trunk roads*

(c) *Primary roads* (d) *Living streets*

Fig. 14. Histograms of prediction errors for different road types.

9. Conclusion

This study presents a ML-based surrogate modeling framework for large-scale agent-based transport simulations. By transforming the MATSim street network into a dual graph, we enable link-level predictions of traffic volume changes under structural policy interventions. Our approach is designed to capture spatial dependencies across connected streets and support scalable evaluation of traffic policies.

In a case study using the MATSim model of Paris, which includes more than 30,000 street segments, we apply 10,000 scenarios involving road capacity reductions across varying combinations of urban districts. The key findings are as follows:

- **Surrogate Model Performance:** The ML surrogate accurately predicts the effects of capacity reduction policies at the street-segment level, achieving an R^2 of 0.91 and strong correlation metrics on the test set. On primary roads with capacity reduction policies, it achieves an R^2 of 0.98.
- **Stochasticity and Model Error:** We find that simulation noise from the agent-based model - due to the inherent randomness of agent-based interactions, and also due to population downsampling - plays a dominant role in surrogate error. Base-case variance serves as a reliable proxy for aleatoric uncertainty, and its contribution to MSE remains consistent across street types.

10. Discussion

10.1. Daily vs. hourly traffic volumes: assumptions and outlook

This study focuses on average daily traffic volumes as the primary optimization target. While we acknowledge that peak-hour traffic often presents the most critical challenges in transportation planning - and that daily averages may obscure important intra-day variations - our goal was to develop a broadly applicable surrogate model capable of capturing the overall effect of traffic policy interventions. Daily aggregates provide a stable and interpretable signal for assessing whether a given policy leads to increased or decreased traffic volumes, even if short-term behavioral fluctuations occur throughout the day. That said, we are aware that this approach may not fully reflect hourly traffic dynamics, and it remains theoretically possible that peak-period flows are not accurately represented, even when daily totals are. We therefore see hourly flow modeling, including the analysis of peak-hour impacts, as a valuable direction for future work - especially given its significance for practical traffic management and infrastructure design.

10.2. Generalization to other transportation policies and cities

While this study focuses on capacity reduction policies in Paris, the surrogate modeling framework is flexible and applicable to a wide range of link-level transportation policies, such as speed reductions, congestion pricing, tolls, and access restrictions. Applying the framework to other policies requires conducting new agent-based simulations that reflect the policy-specific scenarios and spatial configurations. The surrogate model must then be retrained on this data to capture the resulting traffic dynamics and behavioral adaptations. Feature inputs should likewise be adapted to reflect the intervention type (e.g., speed limits, toll rates). As with capacity reductions, it remains essential to carefully define the spatial scope of potential interventions to enable efficient optimization.

The framework is also transferable to other cities, but each new application demands generating appropriate simulation data and retraining the model to maintain predictive accuracy.

10.3. Applicability across simulation platforms

Although this study employs MATSim as its simulation environment, the proposed surrogate modeling framework is applicable to any traffic simulation tool - open-source or commercial - as long as it produces link-level traffic outputs under varying policy scenarios. This includes platforms such as SUMO, AIMSUN, or VISSIM. The core requirement is the availability of consistent input-output data that links spatial policy rollouts to resulting traffic volumes, independent of the underlying simulation architecture. However, our methodology for quantifying the impact of simulation stochasticity on surrogate model error is specific to agent-based models, where inherent randomness in agent decisions can lead to variable outcomes across runs. This component does not generalize to deterministic simulations, but the surrogate learning framework itself does.

10.4. Limitations and future work

Despite the strong performance of the surrogate model, several limitations remain that should be addressed in future research.

- *Computational Cost and Limited Transferability*
  While the trained ML surrogate enables rapid policy evaluation, its initial training process is computationally intensive. In our case, 10,000 full-scale ABM simulations - each requiring approximately one hour on a high-performance computing cluster - were necessary before deployment. This substantial preprocessing effort limits adaptability, as the model must be retrained for each new city or policy setting. Furthermore, its applicability is restricted to cities where an agent-based simulation exists, making it less feasible for cities without pre-existing ABMs or with limited computational resources. Future work should explore inductive graph learning and domain adaptation techniques to enhance generalizability across different urban settings without requiring full retraining.

- *Neglect of Physical Traffic Principles*
  The surrogate model relies on learned representations rather than explicitly incorporating physical traffic flow constraints. While this allows flexibility and efficiency, it may overlook fundamental principles such as flow conservation, congestion spillover, capacity saturation, and queue propagation. Future research should explore hybrid approaches that integrate physics-based constraints with data-driven learning to enhance robustness.
- *Impact of Population Scaling*
  To balance computational efficiency and practical applicability, we trained the model using simulations scaled to 1% of households. While this approach enables feasible training, it introduces limitations. Smaller samples may distort travel time distributions and underrepresent certain travel behaviors, potentially affecting prediction accuracy. Studies suggest that a 5% or 10% scaling factor would improve fidelity, though at a higher computational cost. Future work should explore strategies to mitigate these effects, such as bias correction techniques or hybrid approaches that combine multiple scaling factors.
- *Interpretability and Transparency*
  Interpreting ML-based surrogate models remains a challenge, particularly in policy-sensitive applications. While GNNs effectively capture complex spatial dependencies, their decision-making processes lack transparency, making it difficult for policymakers to validate predictions. Enhancing model explainability through interpretable architectures or post-hoc analysis techniques could improve trust and adoption.
- *Dependence on Simulation Fidelity*
  The surrogate model inherits all assumptions and limitations of the underlying agent-based simulation. If the simulation is poorly calibrated or fails to reflect real-world behavior accurately, the surrogate model will replicate these shortcomings. Therefore, reliable use of the surrogate for policy analysis requires that the agent-based model be carefully validated and calibrated against empirical data before generating training data.

Addressing these challenges will improve the adaptability and reliability of ML-based surrogate models, ensuring their broader applicability in transport policy evaluation.

Despite its current limitations, the surrogate model's rapid evaluation capabilities could enable a range of applications in the long run. First, it could facilitate simulation-based optimization, allowing policymakers to efficiently explore large solution spaces and identify optimal policy setups. Second, its speed makes real-time control applications viable, enabling swift scenario evaluations in time-sensitive decision-making contexts. These capabilities can then help planners to systematically refine policy options to identify the most effective interventions and also provide a computationally efficient framework for objective evaluation in optimization-based approaches.

CRediT authorship contribution statement

Elena Salomé Natterer: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation, Conceptualization; **Saini Rohan Rao:** Visualization, Software; **Alejandro Tejada Lapuerta:** Methodology, Conceptualization; **Roman Engelhardt:** Writing – review & editing, Methodology, Conceptualization; **Sebastian Hörl:** Writing – review & editing, Conceptualization; **Klaus Bogenberger:** Writing – review & editing, Funding acquisition.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used *Chat-GPT 4.0* in order to help with the writing process. Additionally, they used *perplexity.ai* and *answerthis.io* during the literature review. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Data availability

The MATSim simulation used to generate the training data for this study is publicly available at: https://github.com/enatterer/matsim-ile-de-france.

To support reproducibility and further research, we provide an open-source Python library for building and benchmarking surrogate models for agent-based simulations: https://github.com/enatterer/ml_surrogates_for_agent_based_transport_models. The library includes:

- A modular framework for data preprocessing, graph construction, and model evaluation,
- Implementations of the used hybrid architecture and nine additional algorithms using the PyTorch deep learning framework,
- Scripts to reproduce all experiments presented in this paper.

Declaration of interests

These resources are made available to foster further research and advancement of surrogate modeling in transport simulation and planning. The simulation data can be shared upon request. The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Elena Natterer reports financial support was provided by German Federal Ministry of Transport. The other authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the German Federal Ministry of Transport for providing funding through the project “MINGA” with grant number 45AOV1001K. We remain responsible for all findings and opinions presented in the paper. We are grateful to Florian Dandl for insightful discussions on the methodology and to Ramandeep Singh for her valuable feedback. The authors gratefully acknowledge the computational and data resources provided by the Leibniz Supercomputing Centre (www.lrz.de). We thank the anonymous reviewers for their valuable comments, which helped to improve the quality of the paper and its results.

Appendix A. Analysis of Base Case Volume Patterns and Sampling Bias

Our analysis reveals a systematic pattern: roads that receive capacity reductions show higher base case volumes and variances compared to roads that never receive reductions. This pattern appears consistently across all road types (primary, secondary, and tertiary) and emerges despite our random district selection process. Here, we explain the underlying mechanisms that create this sampling bias.

A.1. The pattern we observe

For all road types, we consistently observe higher base case variance ($\sigma^2_{t,b}$) in roads with capacity reduction:

- Primary roads: $\sigma^2_{t,b} = 23.55$ vs 17.81
- Secondary roads: $\sigma^2_{t,b} = 16.44$ vs 11.61
- Tertiary roads: $\sigma^2_{t,b} = 12.39$ vs 10.20

A.2. District selection process

Our methodology involves:

- We had 8,308 different scenarios available
- In each scenario, we randomly select about four districts
- All roads in a selected district with road type “primary”, “secondary” and “tertiary” receive capacity reduction
- A road is labeled as “with capacity reduction” if its district is selected in any scenario
- A road is labeled as “without capacity reduction” if its district is never selected

A.3. Analysis of district characteristics

Our analysis of district-level data reveals several key patterns:

A.3.1. Volume distribution within districts

- Districts show highly skewed volume distributions (mean skewness = 3.64)
- On average, only 27% of roads in a district have volumes above the district mean
- Volume distributions vary significantly between districts

A.3.2. Selection frequency patterns

When we rank districts by how often they are selected, we find clear patterns:

- Districts selected most frequently (top 25% by selection frequency):
  - Lower mean volumes (47.44)
  - Lower standard deviation (70.63)
  - More balanced distribution (skewness = 2.08)
  - Higher proportion of high-volume roads (30%)
- Districts selected least frequently (bottom 25% by selection frequency):
  - Higher mean volumes (60.37)
  - Higher standard deviation (158.44)
  - Highly skewed distribution (skewness = 5.76)
  - Lower proportion of high-volume roads (21%)

A.4. Key statistical relationships

Our analysis reveals two important correlations:

- Strong negative correlation (-0.762) between district volume skewness and selection frequency
- Positive correlation (0.605) between proportion of high-volume roads and selection frequency

A.5. How the sampling Bias emerges

The sampling bias develops through the following sequence:

1. **Initial District Selection:**
   - Districts with more balanced volume distributions tend to be selected more frequently
   - These districts have a higher proportion of high-volume roads
   - When selected, all roads in the district receive capacity reduction
2. **Cumulative Effect Over Many Scenarios:**
   - Roads from more balanced districts are more likely to receive capacity reduction
   - These districts contribute more high-volume roads to the "with capacity reduction" group
   - This systematically increases both the average volume and variance in the capacity reduction group

A.6. Conclusion

The higher base case volumes and variances we observe in roads with capacity reduction are not a direct result of the capacity reductions themselves. Instead, they emerge from how our random selection process interacts with the underlying patterns of traffic volumes in different districts. The key insight is that districts with more balanced volume distributions are selected more frequently, and these districts tend to have a higher proportion of high-volume roads. This leads to their roads being overrepresented in the "with capacity reduction" group, creating the observed pattern in base case volumes and variances.

References

Auld, J., Hope, M., Ley, H., Sokolov, V., Xu, B., Zhang, K., 2016. Polaris: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations. Transp. Res. Part C: Emerg. Technol. 64, 101–116. https://doi.org/10.1016/j.trc.2015.07.017

Balac, M., Ciari, F., Axhausen, K.W., et al., 2017. Modeling the impact of parking price policy on free-floating carsharing: Case study for Zurich, Switzerland. Transp. Res. Part C: Emerg. Technol. 77, 207–225. https://doi.org/10.1016/j.trc.2017.01.022

Balmer, M., Meister, K., Rieser, M., Nagel, K., Axhausen, K.W., 2008. Agent-based simulation of travel demand: Structure and computational performance of MATSim-t. Arb. Verk. Raumpl. 504. https://doi.org/10.3929/ethz-a-005626451

Bastarianto, F.F., Hancock, T.O., Choudhury, C.F., Manley, E., 2023. Agent-based models in urban transportation: review, challenges, and opportunities. Eur. Transp. Res. Rev. 15 (1), 19. https://doi.org/10.1186/s12544-023-00590-5

Ben-Dor, G., Ben-Elia, E., Benenson, I., 2021. Population downscaling in multi-agent transportation simulations: A review and case study. Simul. Modell. Pract. Theory 108, 102233. https://doi.org/10.1016/j.simpat.2020.102233

Ben-Dor, G., Ogulenko, A., Klein, I., Ben-Elia, E., Benenson, I., 2024. Simulation-based policy evaluation of monetary car driving disincentives in Jerusalem. Transp. Res. Part A Policy Pract. 183, 104061. https://doi.org/10.1016/j.tra.2024.104061

Bogenberger, K., Weikl, S., 2012. Quality management methods for real-time traffic information. Proc. Social Behav. Sci. 54, 936–945. Proceedings of EWGT2012 - 15th Meeting of the EURO Working Group on Transportation, September 2012, Paris. https://doi.org/10.1016/j.sbspro.2012.09.809

Cervellera, C., Macciò, D., Rebora, F., 2021. Deep learning and low-discrepancy sampling for surrogate modeling with an application to urban traffic simulation. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. https://doi.org/10.1109/IJCNN52387.2021.9533357

Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR) , 77–85 https://doi.org/10.1109/CVPR.2017.16

Horni, A., Nagel, K., Axhausen, K.W., 2016. The Multi-Agent Transport Simulation MATSim. Ubiquity Press. https://doi.org/10.5334/baw

Hörl, S., Balac, M., 2021. Synthetic population and travel demand for Paris and Île-de-France based on open and publicly available data. Transp. Res. Part C: Emerg. Technol. 130, 103291. https://doi.org/10.1016/j.trc.2021.103291

Hörl, S., Becker, F., Axhausen, K.W., 2021. Simulation of price, customer behaviour and system impact for a cost-covering automated taxi system in Zurich. Transportation Research Part C: Emerging Technologies 123, 102974. https://doi.org/10.1016/j.trc.2021.102974

Jiang, W., Luo, J., 2024. Graph Neural Network for Traffic Forecasting: A Survey. Expert Systems with Applications 207, 117921. https://doi.org/10.1016/j.eswa.2022.117921

Kagho, G.O., Balac, M., Axhausen, K.W., 2020. Agent-Based Models in Transport Planning: Current State, Issues, and Expectations. Proc. Comput. Sci. 170, 726–732. https://doi.org/10.1016/j.procs.2020.03.164

Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR). https://openreview.net/forum?id=SJU4ayYgl.

Laudan, J., Heinrich, P., Nagel, K., 2025. High-Performance Mobility Simulation: Implementation of a Parallel Distributed Message-Passing Algorithm for MATSim. Information 16 (2), 116. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/info16020116

Liu, Y., Zou, B., Ni, A., Gao, L., Zhang, C., 2020. Calibrating microscopic traffic simulators using machine learning and particle swarm optimization. Transportation letters https://doi.org/10.1080/19427867.2020.1728037

Llorca, C., Moeckel, R., 2019. Effects of scaling down the population for agent-based traffic simulations. Proc. Comput. Sci. 151, 782–787. https://doi.org/10.1016/j.procs.2019.04.106

Lu, Q.-L., Qurashi, M., Antoniou, C., 2023. Simulation-Based Policy Analysis: The Case of Urban Speed Limits. Transp. Res. Part A Policy Pract. 175, 103754. https://doi.org/10.1016/j.tra.2023.103754

Macal, C.M., North, M.J., 2010. Tutorial on agent-based modeling and simulation. J. Simul. 4 (3), 151–162. https://doi.org/10.1057/jos.2010.3

Manley, E., Cheng, T., Penn, A., Emmonds, A., 2014. A framework for simulating large-scale complex urban traffic dynamics through hybrid agent-based modelling. Comput. Environ. Urban Syst. 44, 27–36. https://doi.org/10.1016/j.compenvurbsys.2013.11.003

Marmaras, C., Xydas, E., Cipcigan, L., 2017. Simulation of electric vehicle driver behaviour in road transport and electric power networks. Transp. Res. Part C Emerg. Technol. 80, 239–256. https://doi.org/10.1016/j.trc.2017.05.004

Müller, J., Straub, M., Stubenschrott, M., Graser, A., 2023. Simulation of a full-scale implementation of Superblocks in Vienna: 15th ITS European Congress. Proceedings of the 15th ITS European Congress https://publications.ait.ac.at/en/publications/simulation-of-a-full-scale-implementation-of-superblocks-in-vienn.

Narayanan, S., Makarov, N., Antoniou, C., 2024. Graph neural networks as strategic transport modelling alternative - a proof of concept for a surrogate. IET Intell. Transp. Syst. 1-19. https://doi.org/10.1049/itr2.12551

Natterer, E.S., Loder, A., Bogenberger, K., 2025. Effects of Paris' Cycling Policies on Vehicular Flow: An Empirical Analysis. Transportation Research Record 0(0). https://doi.org/10.1177/03611981251356507

Reza, S., Campos Ferreira, M., Machado, J.J.M., Tavares, J. M. R.S., 2022. A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks. Expert Syst. Appl. 206, 117275. https://doi.org/10.1016/j.eswa.2022.117275

Roman, O., Maheshwari, T., Do, C., Adey, B., Fourie, P., Ye, Q., Bansal, P., et al., 2025. A model-based adaptive planning framework using surrogate modelling for urban transport systems under uncertainty. https://doi.org/10.2139/ssrn.5020996

Saprykin, A., Chokani, N., Abhari, R.S., 2019. Gemsim: A gpu-accelerated multi-modal mobility simulator for large-scale scenarios. Simul. Modell. Pract. Theory 94, 199–214. https://doi.org/10.1016/j.simpat.2019.03.002

Shi, Y., Huang, Z., feng, S., Zhong, H., Wang, W., Sun, Y., 2021. Masked label prediction: Unified message passing model for semi-supervised classification. https://doi.org/10.48550/arXiv.2009.03509

Smilovitskiy, M., Olmez, S., Richmond, P., Chisholm, R., Heywood, P., Cabrejas, A., van den Berghe, S., Kobayashi, S., 2025. Overcoming computational complexity: A scalable agent-based model of traffic activity using FLAME-GPU. In: Mathieu, P., De la Prieta, F. (Eds.), Advances in Practical Applications of Agents, Multi-Agent Systems, and Digital Twins: The PAAMS Collection. Springer Nature Switzerland, Cham, pp. 240–251. https://doi.org/10.1007/978-3-031-70415-4_21

Sroczyński, A., Andrzej, C., 2023. Road traffic can be predicted by machine learning equally effectively as by complex microscopic model. Sci. Rep. 13 (1), 14523. https://doi.org/10.1038/s41598-023-41902-y

Tanaka, F., Amano, T., Uchiyama, A., Hiromori, A., Nakamura, Y., Yamaguchi, H., 2024. Policy optimization for pedestrian traffic management by surrogation of simulation models. In: 2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS), pp. 203–211. https://doi.org/10.1109/MASS62177.2024.00036

Triebke, H., Kromer, M., Vortisch, P., 2023. Bridging the gap between mesoscopic transport planning and microscopic traffic simulation: An analytical and numerical analysis of traffic dynamics. Transp. Res. Rec. 2677 (5), 62–76. https://doi.org/10.1177/03611981221128284

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I., 2017. Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2017. Graph attention networks. https://doi.org/10.48550/arXiv.1710.10903

Xing, W., Ruihao, Z., Fumin, Z., Lyuchao, L., Faliang, H., 2023. Sttf: An efficient transformer model for traffic congestion prediction. Neural Computing and Applications . https://doi.org/10.1007/s44196-022-00177-3

Yin, B., Diallo, A.O., Seregina, T., Coulombel, N., Liu, L., et al., 2024. Evaluation of Low-Traffic Neighborhoods and Scale Effects: The Paris Case Study. Transp. Res. Rec. 2678 (1), 88–101. Publisher: SAGE Publications Inc. https://doi.org/10.1177/03611981231170130

Yousefzadeh, N., Sengupta, R., Karnati, Y., Rangarajan, A., Ranka, S., 2025. Graph attention network for lane-wise and topology-invariant intersection traffic simulation. IEEE Trans. Intell. Transp. Syst. 26 (4), 5082–5093. https://doi.org/10.1109/TITS.2025.3546810

Zhang, L., Liu, Q., Zhou, Y., et al., 2025. An improved transformer based traffic flow prediction model. Sci. Rep. 15 (1), 4165. https://doi.org/10.1038/s41598-025-92425-7