Machine Learning Surrogates for Optimizing Transportation Policies with MATSim

Presented at the MATSim User Meeting 2025, June 12, 2025, Munich, Germany

Elena Natterer[1], Roman Engelhardt[1],

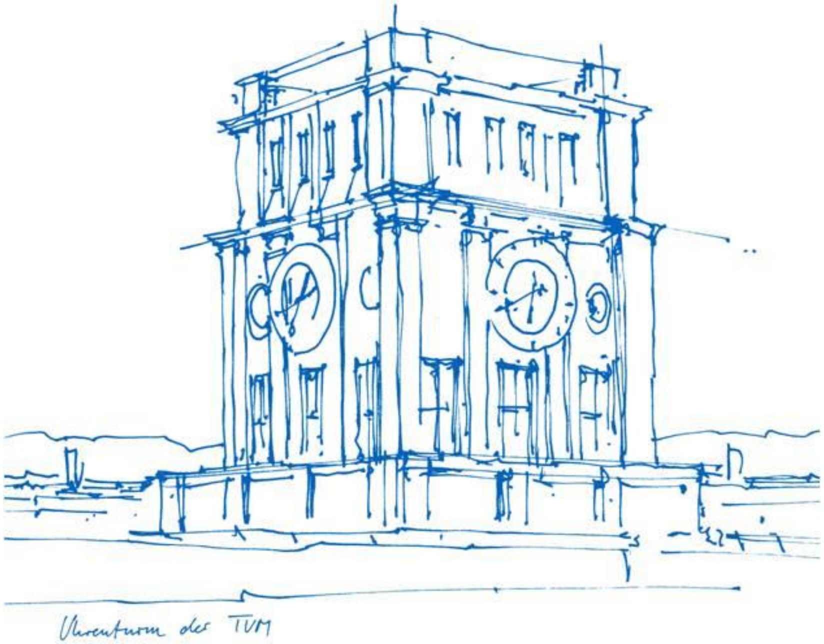
Sebastian Hörl[2], Klaus Bogenberger[1]

1: Chair of Traffic Engineering and Control,

Technical University of Munich, Munich, Germany

2: Institut de Recherche Technologique SystemX,

Palaiseau, France



Agenda

1. Motivation
2. Constructing the Surrogate
3. Case Study
4. Results
5. Conclusion + Discussion

Urban traffic planning is a multi-objective challenge

Cities pursue diverse, sometimes conflicting traffic goals

- Reduce emissions
- Lower congestion
- Reduce noise
- Reduce pollution
- Improve safety

There are solutions to these!

- Reduce emissions
- Lower congestion
- Reduce noise
- Reduce pollution
- Improve safety
- …

The real problem: Which policies to pick?

- Capacity reduction
- Speed reduction
- Bicycle lane expansion
- Congestion charges
- …

Policy	*Possible extent*	*Spatial extent*
Capacity reduction	[20%, 50%, …]	Link level
Speed reduction	Tempo 30/50?	Link level
Bicycle lane expansion	With what size?	Link level
Congestion charges	1/2/3 €/h?	…
…		

How big is the policy solution space?

Let n be the number of streets in a city.

- Capacity reduction
- Speed reduction
- Congestion charges
- …

Possible extent

[20%, 40%, 60%]
[30 km/h, 50 km/h, hybrid]
[1€/h, 2€/h, 3€/h]

Number of options per link

4^n
4^n
4^n

$$4^n * 4^n * 4^n = 4^{3n}$$

Example: Paris, France, with $n = 30{,}000$:

$4^{3*30{,}000} \sim 10^{54{,}185}$

# of atoms in the universe: 10^{80}

This number is by a factor of $10^{54{,}105}$ bigger!

If we considered only one policy, on district-level:

Let d be the number of districts in a city.

- Capacity reduction

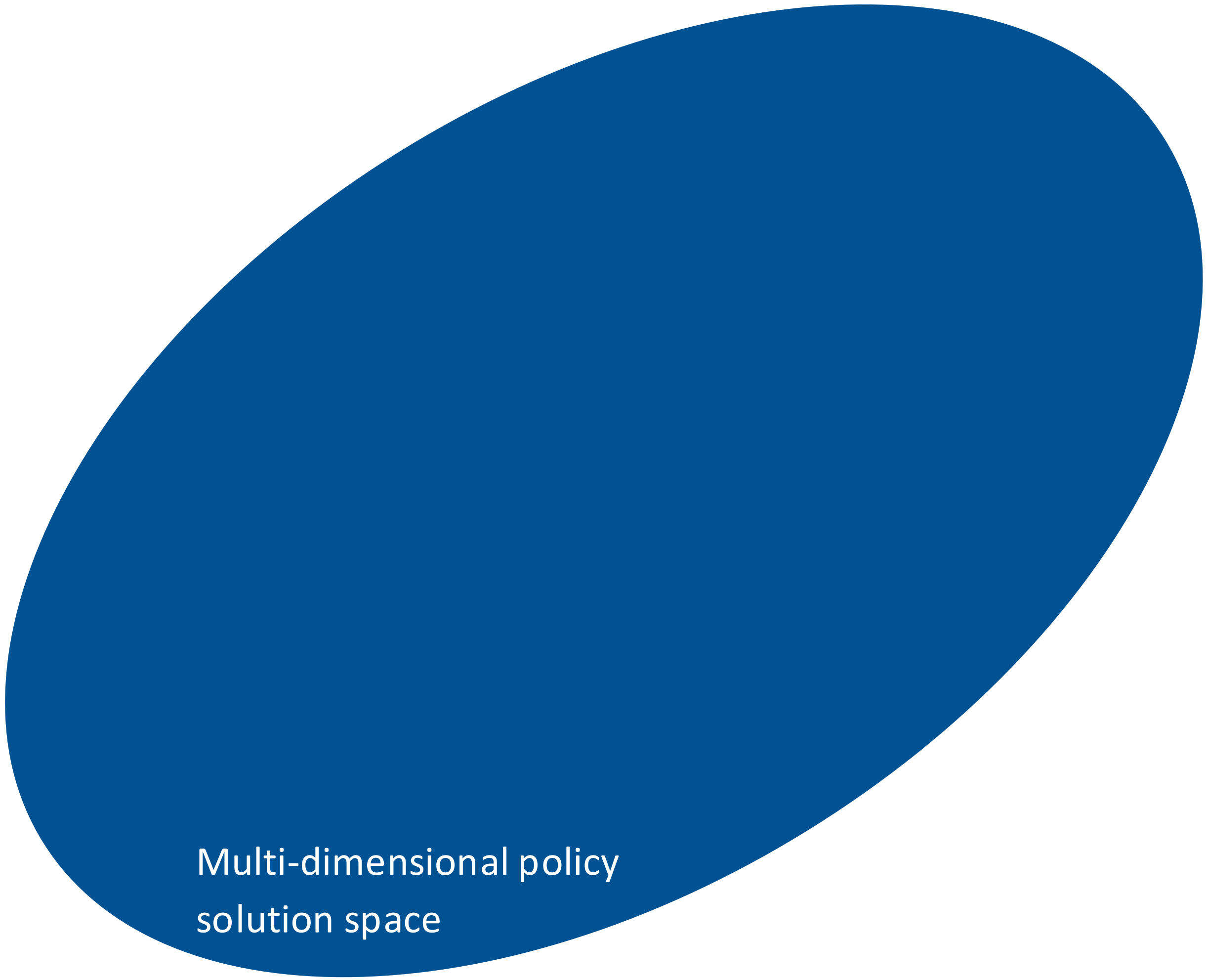
Possible extent	*Number of options*
[50%]	2^d

Example: Paris, France, with *d = 20*:

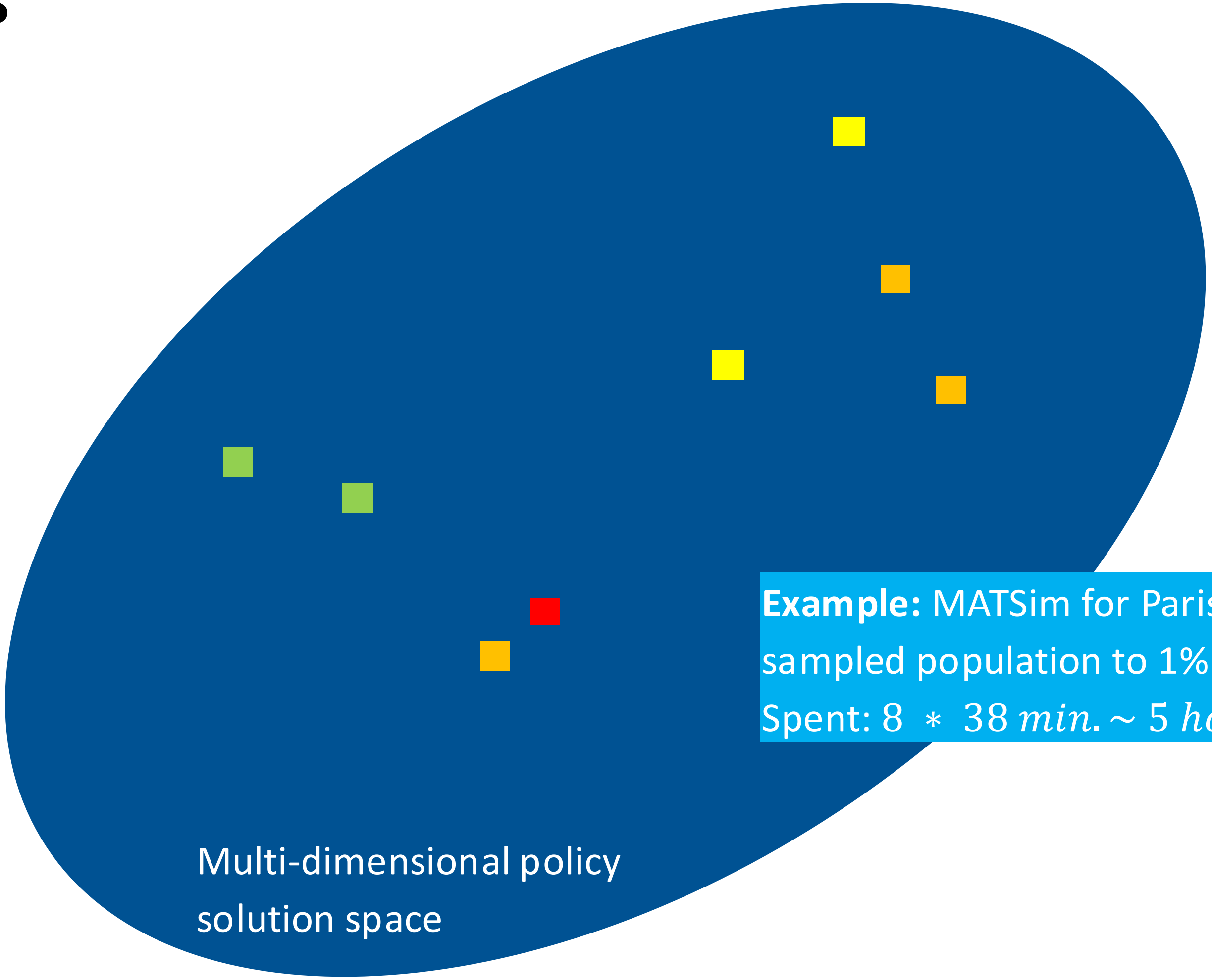
$$2^{20} = 1{,}048{,}576$$

Still, quite a big number.

The real problem: Which solution to pick?

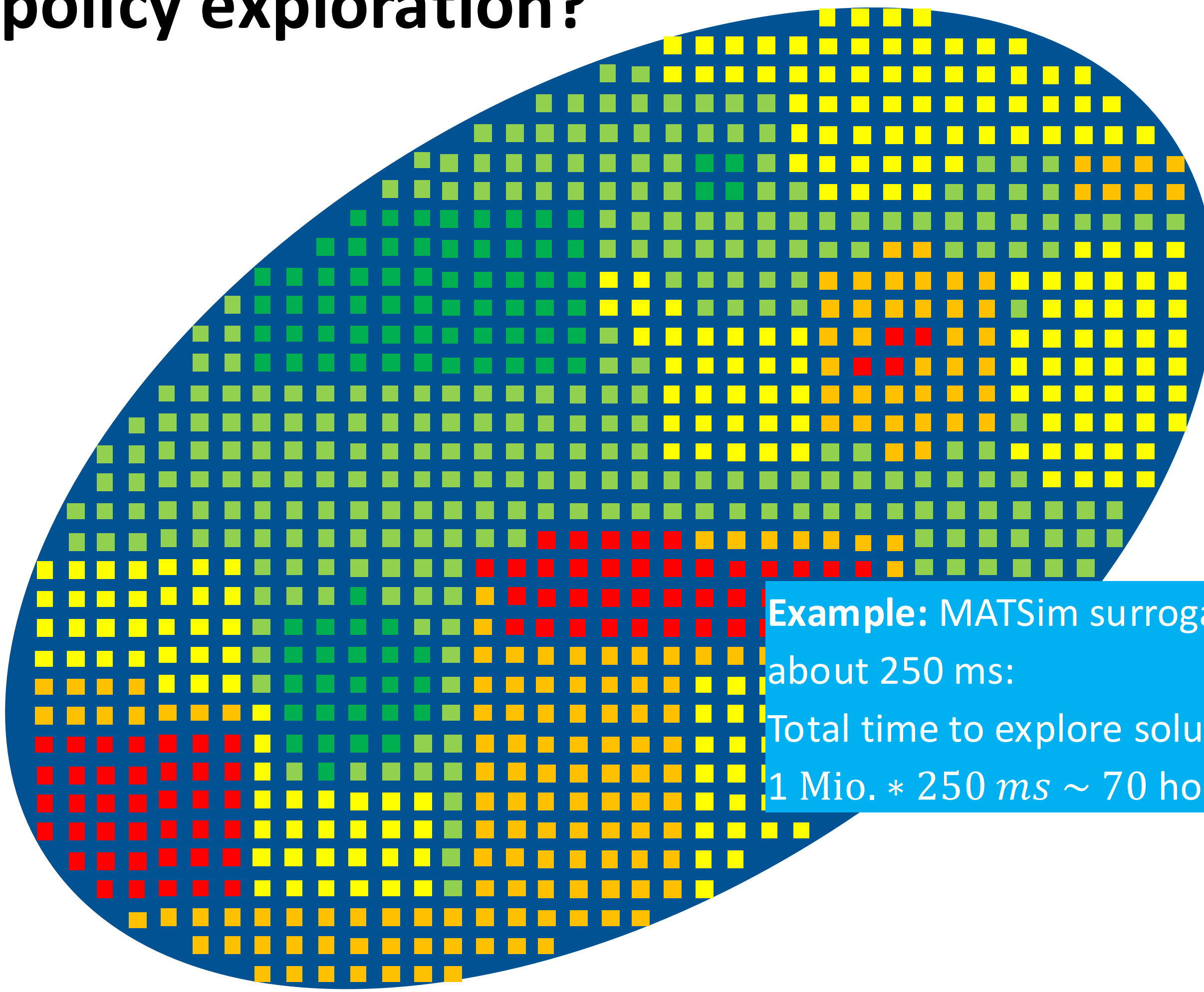


Procedure so far: Use agent-based models to test some solutions



Example: MATSim for Paris, with down-sampled population to 1% takes 38 min: Spent: $8 * 38\, min. \sim 5\, hours$

What if we could use a fast surrogate model to unlock efficient policy exploration?

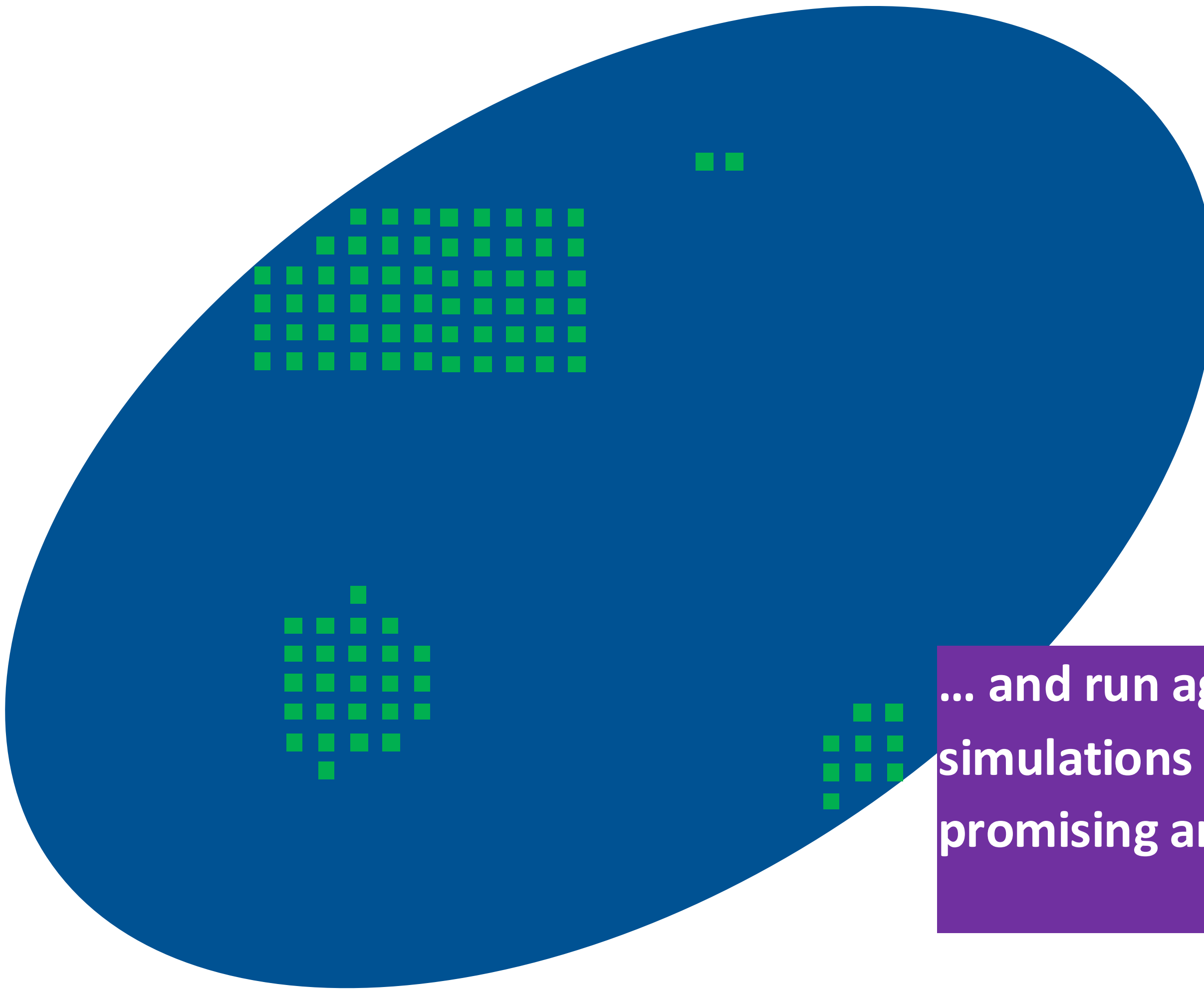


Example: MATSim surrogate for Paris takes about 250 ms:

Total time to explore solution space:

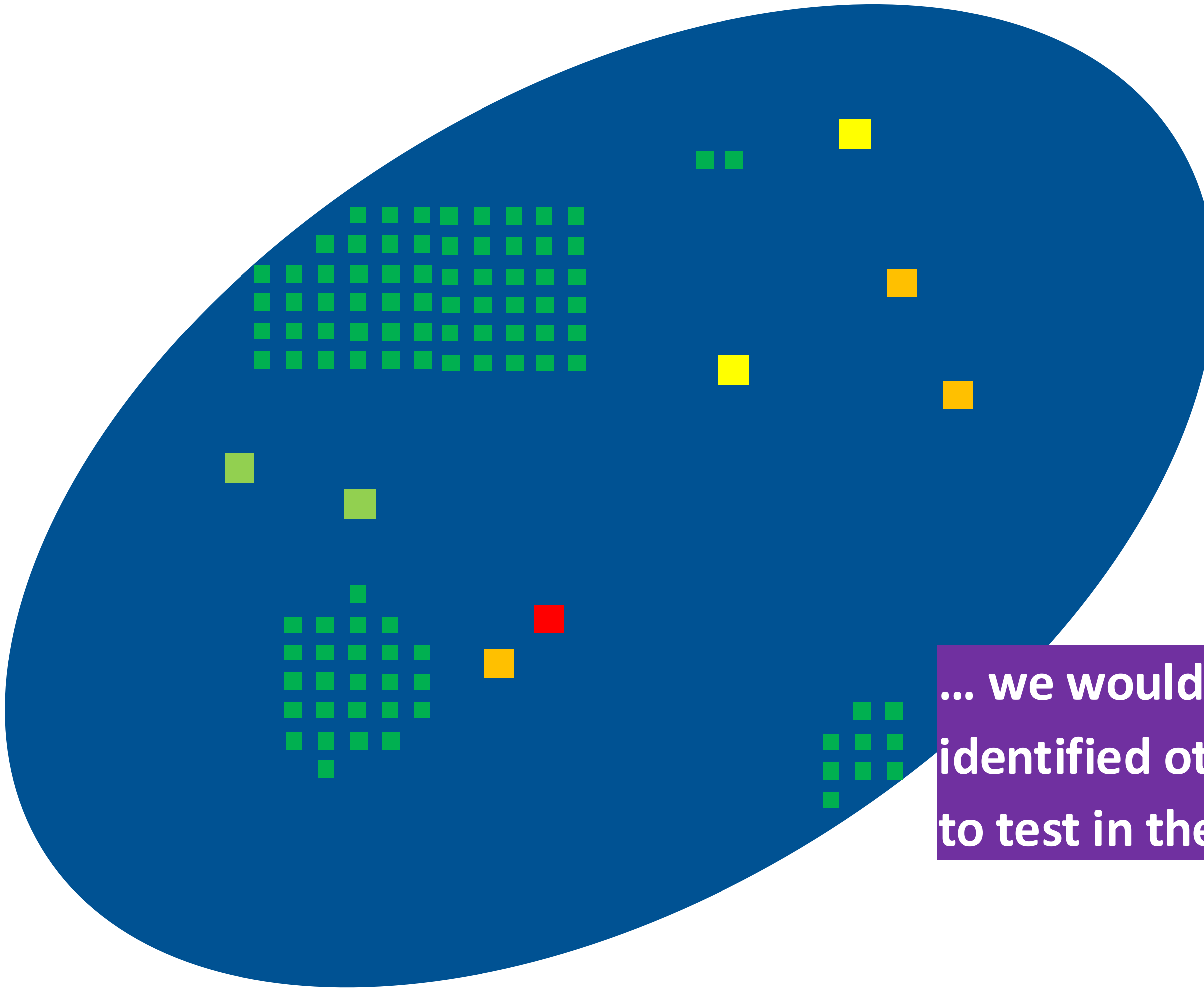
$1 \text{ Mio.} * 250\, ms \sim 70$ hours

We could identify the areas that *optimize* the cities’ goals



… and run agent-based simulations in the most promising areas!

Comparing it with our initial guesses …



… we would have identified other policies to test in the simulations!

Replicating Agent-Based Models Outcomes with Machine Learning Surrogates

- A machine learning surrogate is a data-driven model that serves as a proxy for a high-fidelity model or simulation, enabling faster evaluation of the system under study

- Agent-Based models (ABM) can be understood as functions that map policy interventions to their corresponding effects

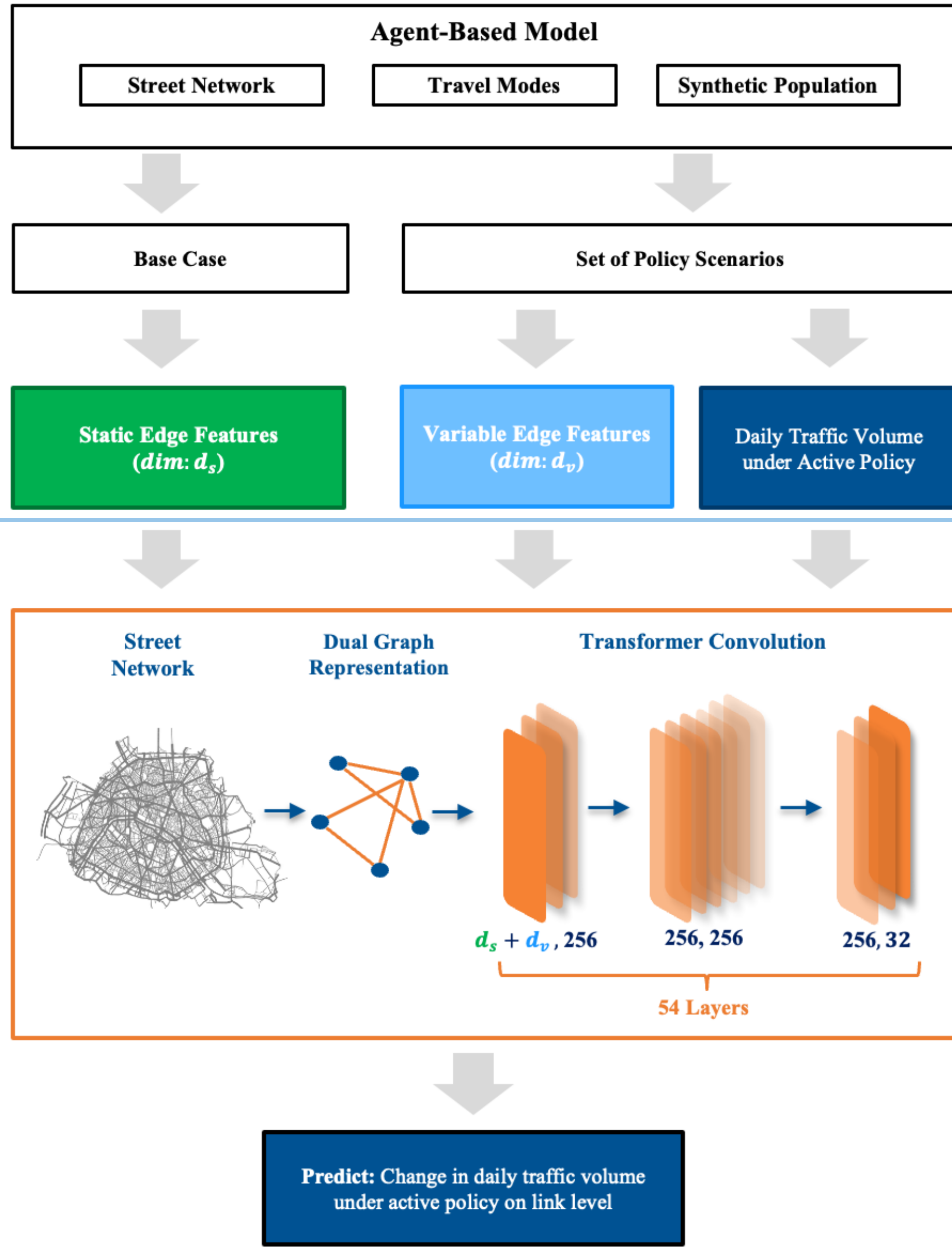
- Machine Learning (ML) has emerged as a powerful tool for approximating complex functions when trained on sufficient data

- Graph Neural Networks (GNNs) are machine learning models specifically designed to handle graph-structured data

ML Surrogates can help unlock the vast solution space of traffic policies, **if accurate!** ☺

Constructing the Surrogate

Method overview



MATSim
Multi-Agent Transport Simulation

Case Study

We validate the surrogate with Paris' MATSim simulation

MATSim model:

- Model used: *Synthetic population and travel demand for Paris and Île-de-France, based on open and publicly available data*, Hörl, S. and Balac, M., 2021
- For performance reasons, 1% of households (of 12 million people) are selected

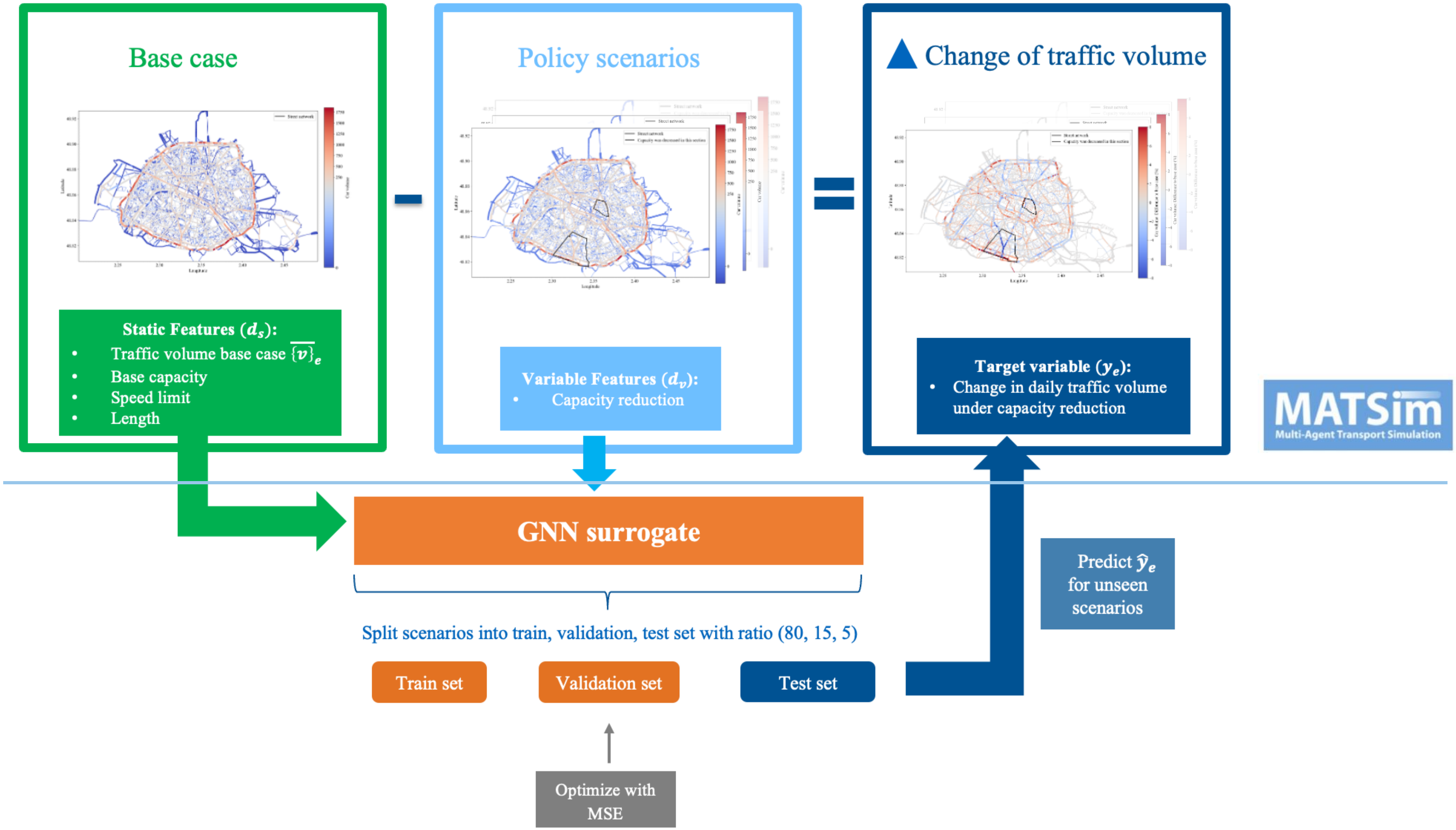
Tested policy:

- Capacity reduction by 50% on all streets classified as Primary, Secondary or Tertiary in OSM in selected districts

Base Case Traffic Volume:

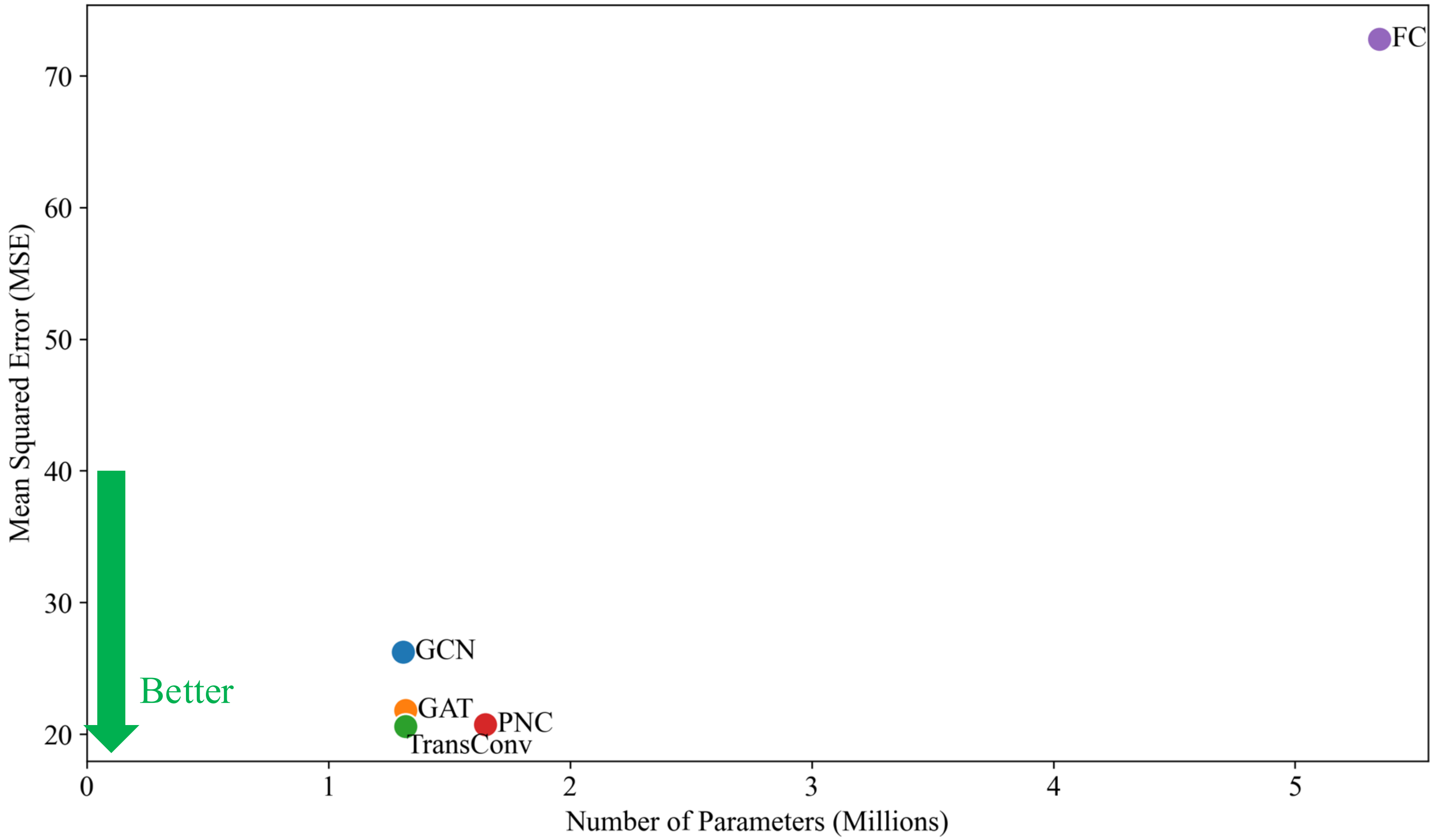
- Average over 100 MATSim simulations without policy intervention, and different random seeds

➡ **Create 10,000 scenarios, each defined by a specific combination of districts where the policy is implemented. Split the dataset into training (85%), testing (10%), and validation (5%) sets. Predict the change in traffic volume on link level for scenarios in the test set due to the capacity reduction.**

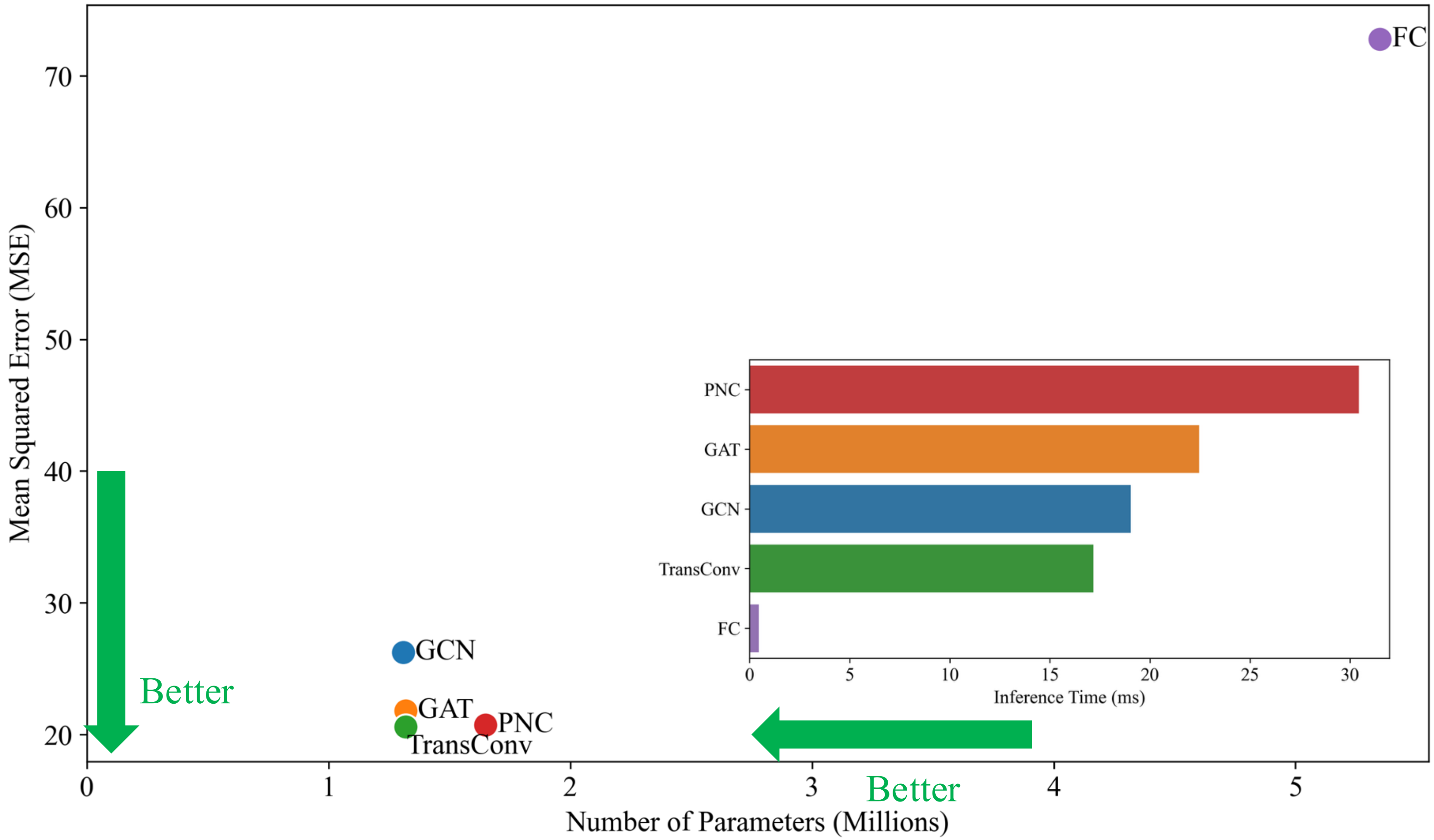


Base case
Static Features (d_s):
• Traffic volume base case $\overline{\{v\}}_e$
• Base capacity
• Speed limit
• Length
Policy scenarios
Variable Features (d_v):
• Capacity reduction
Change of traffic volume
Target variable (y_e):
• Change in daily traffic volume under capacity reduction
MATSim
Multi-Agent Transport Simulation
GNN surrogate
Split scenarios into train, validation, test set with ratio (80, 15, 5)
Train set
Validation set
Test set
Predict $\hat{y}_e$ for unseen scenarios
Optimize with MSE

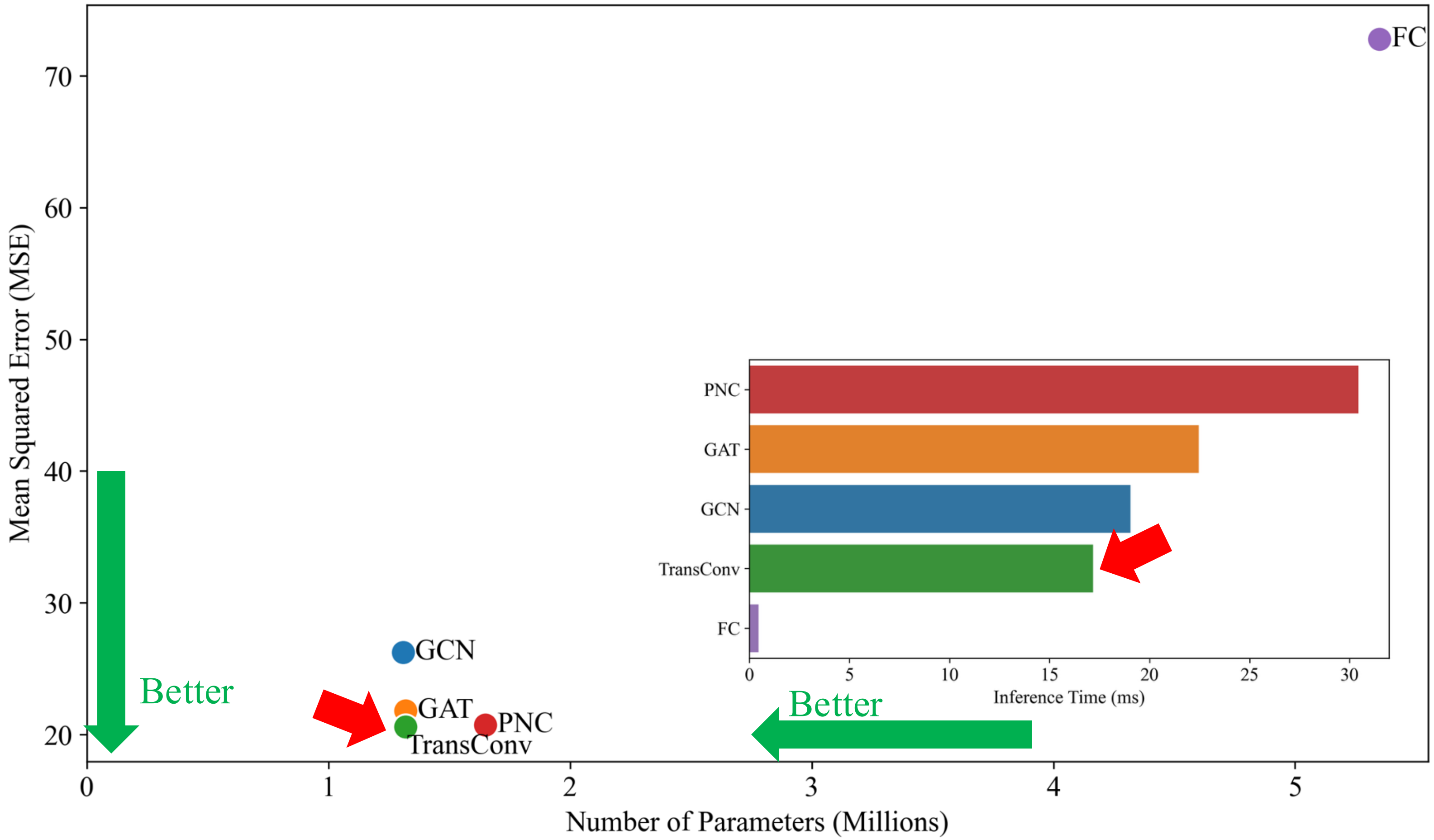
Compact Architecture Evaluation: Complexity vs. MSE



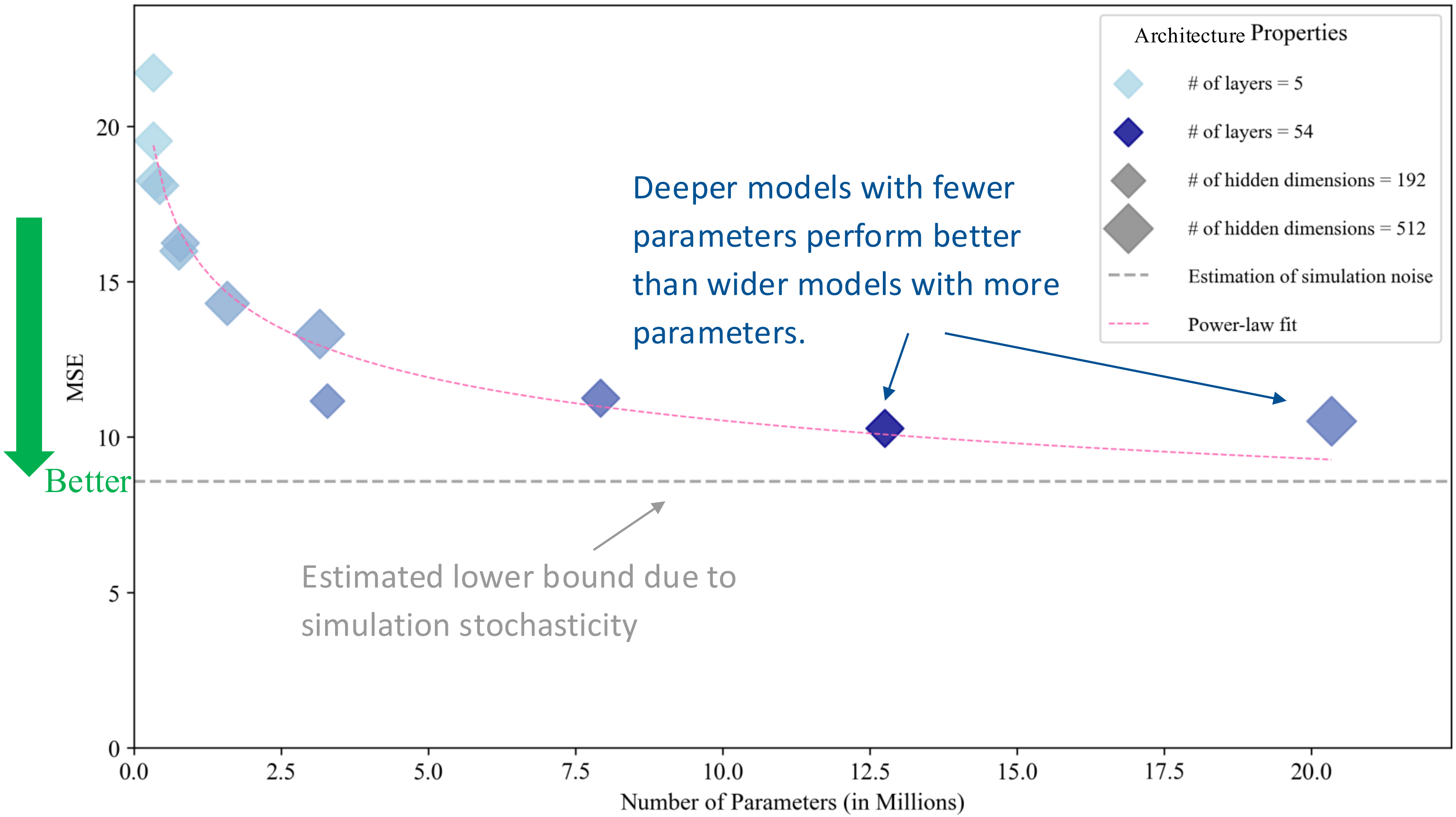
Compact Architecture Evaluation: Complexity vs. MSE with Inference Times



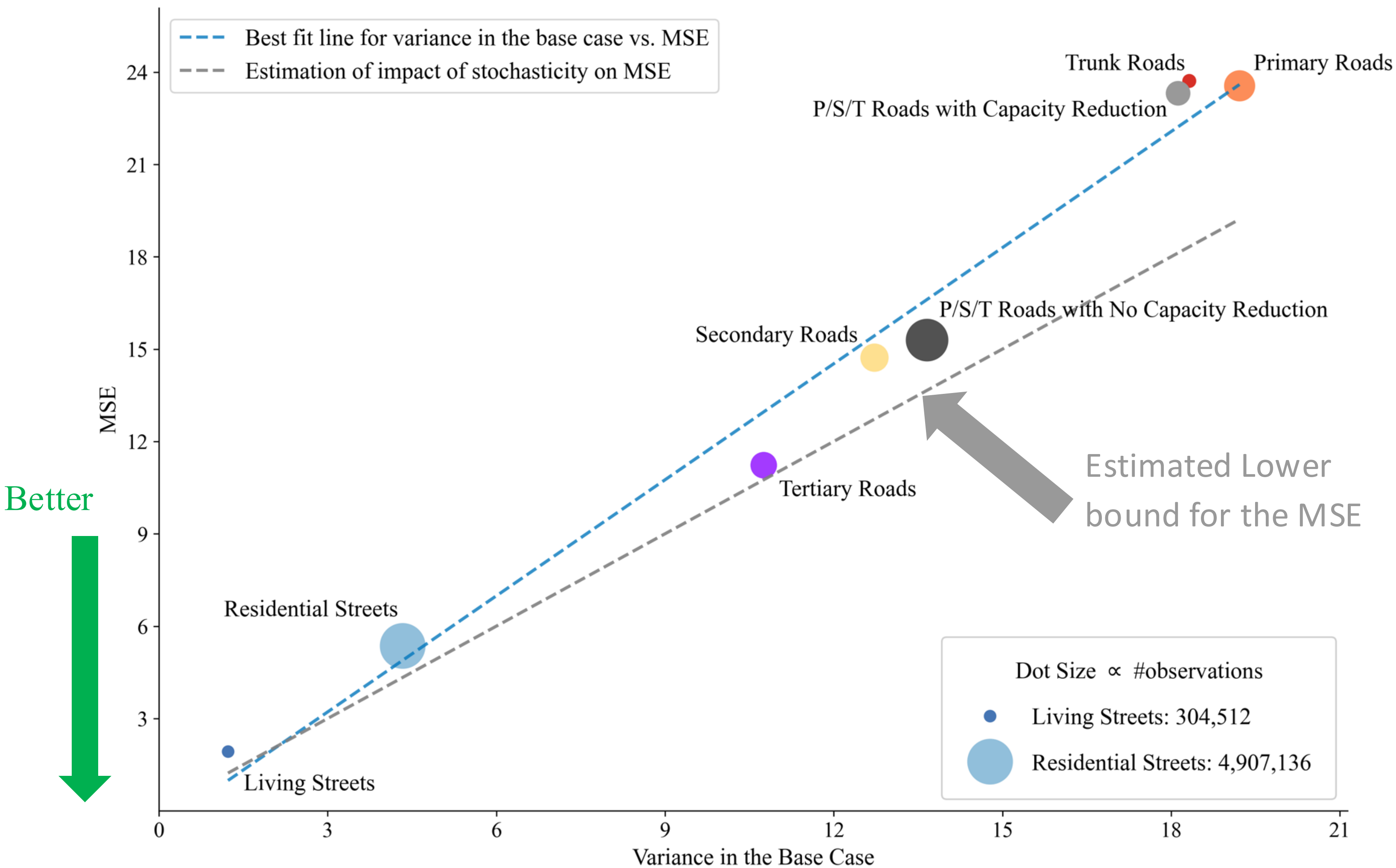
We identify Transformer Convolution as the best architecture in terms of accuracy and inference time



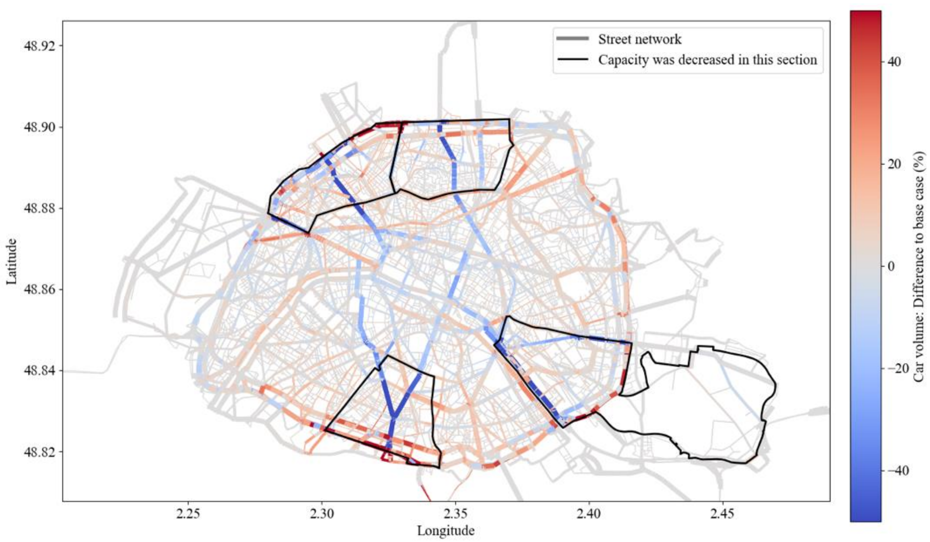
The best Transformer architecture favors greater depth over wider layers



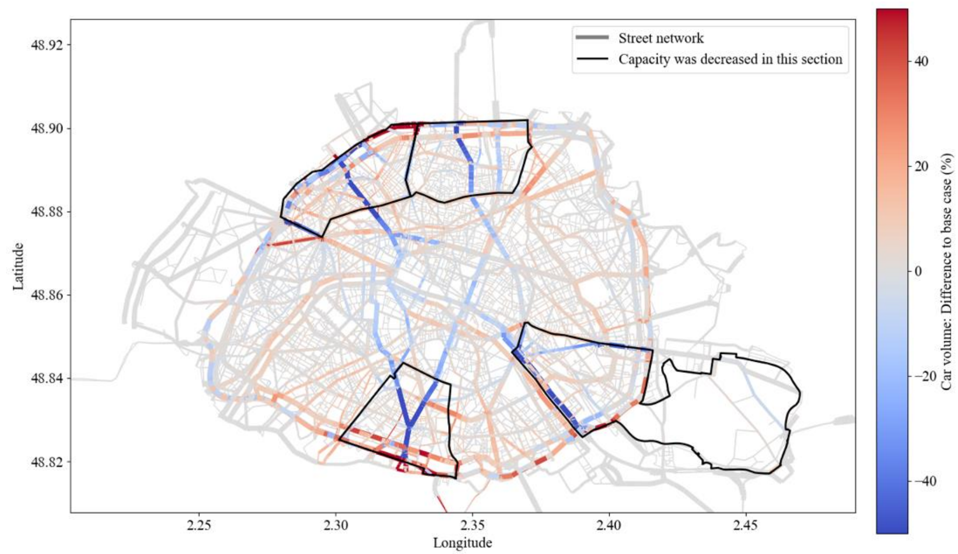
The model predictions vary by street type



Network-level evaluation show effective learning

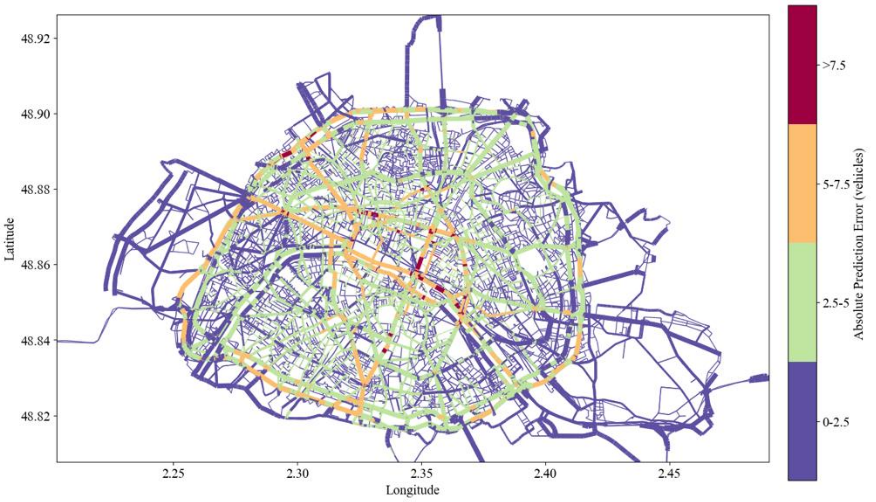


MATSim simulated change in traffic volume

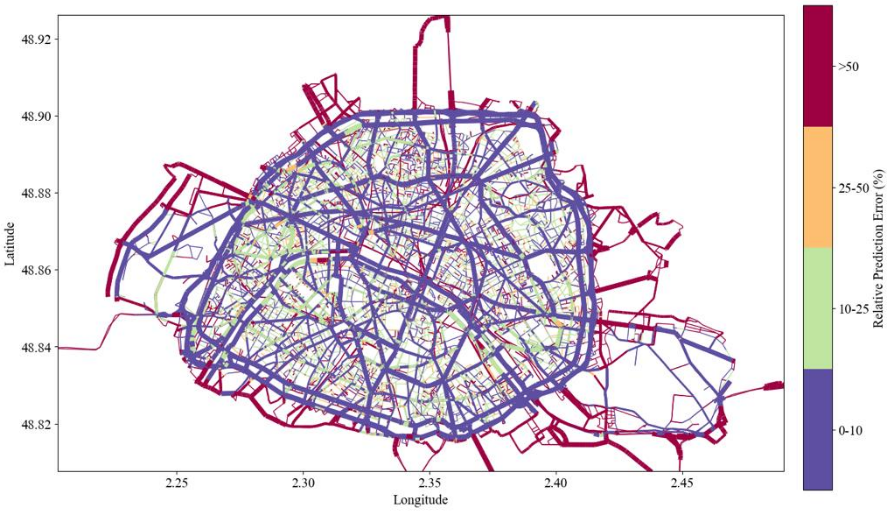


Surrogate predicted change in traffic volume

Across the full test set, the surrogate achieves strong performance across the main road network



Absolute prediction errors



Relative prediction errors

Conclusion + Discussion

Main Takeaways

- The GNN surrogate accurately predicts the effects of capacity reduction policies at the link level, achieving an R^2 of 0.90.

- Simulation noise from the agent-based model plays a dominant role in surrogate model error.

- GNN surrogates for traffic simulations enable efficient exploration of the vast solution space of traffic policies

Current limitations

- Computational Cost for generating the Surrogate
- Limited Transferability
- Neglection of physical traffic principles
- Interpretability and Transparency

Next steps

- Apply inductive graph learning to enable generalization across cities.
- Incorporate the prediction of hourly traffic volumes to enable the estimation of dynamic effects, such as congestion patterns.
- Test whether the model can predict the effect of a combined policy A + B, even though it was only trained on A and B individually.

Thank you for your attention! ☺

Full paper

(in revisions):

Elena Natterer

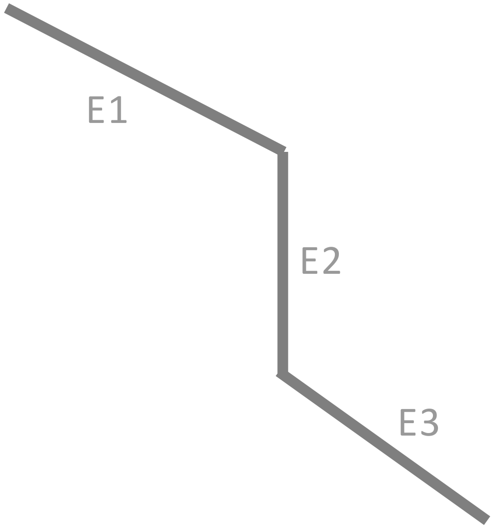
elena.natterer@tum.de

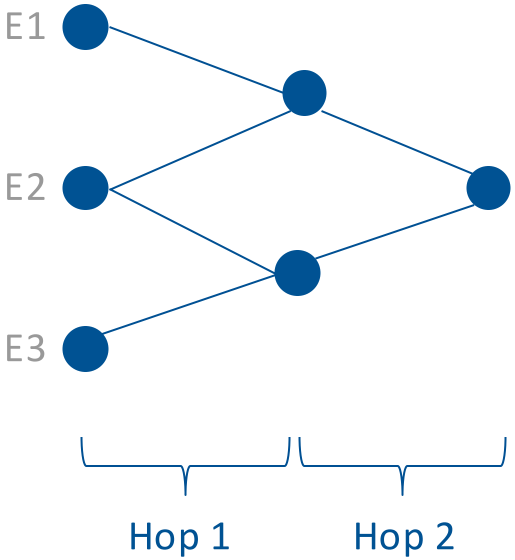
LinkedIn:

Backup

Why do the number of layers matter?

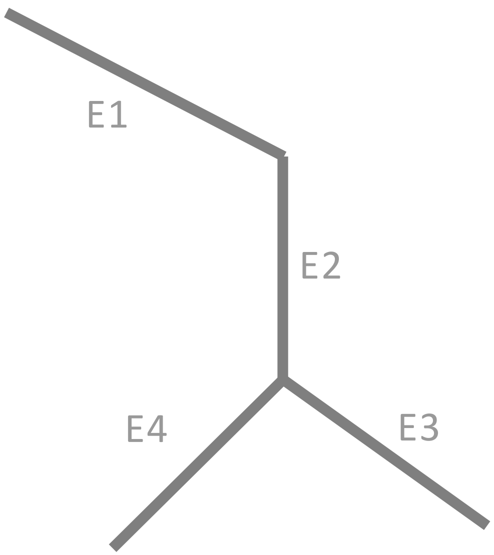
Example for a receptive field: In a graph with 3 edges, it takes 2 hops for all edges to exchange messages with one another through message passing.

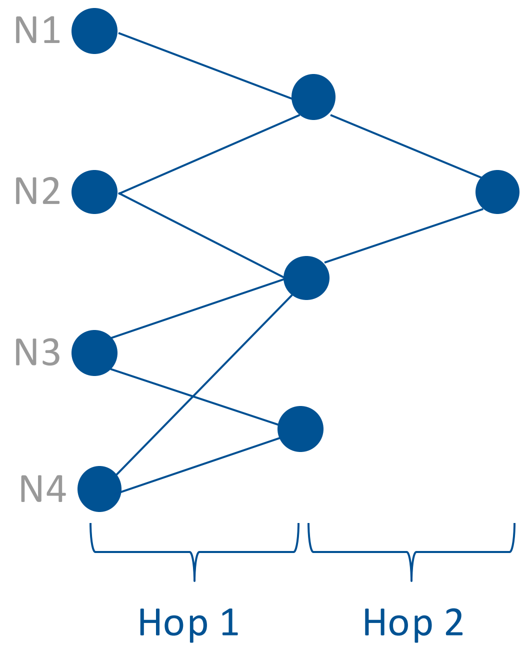




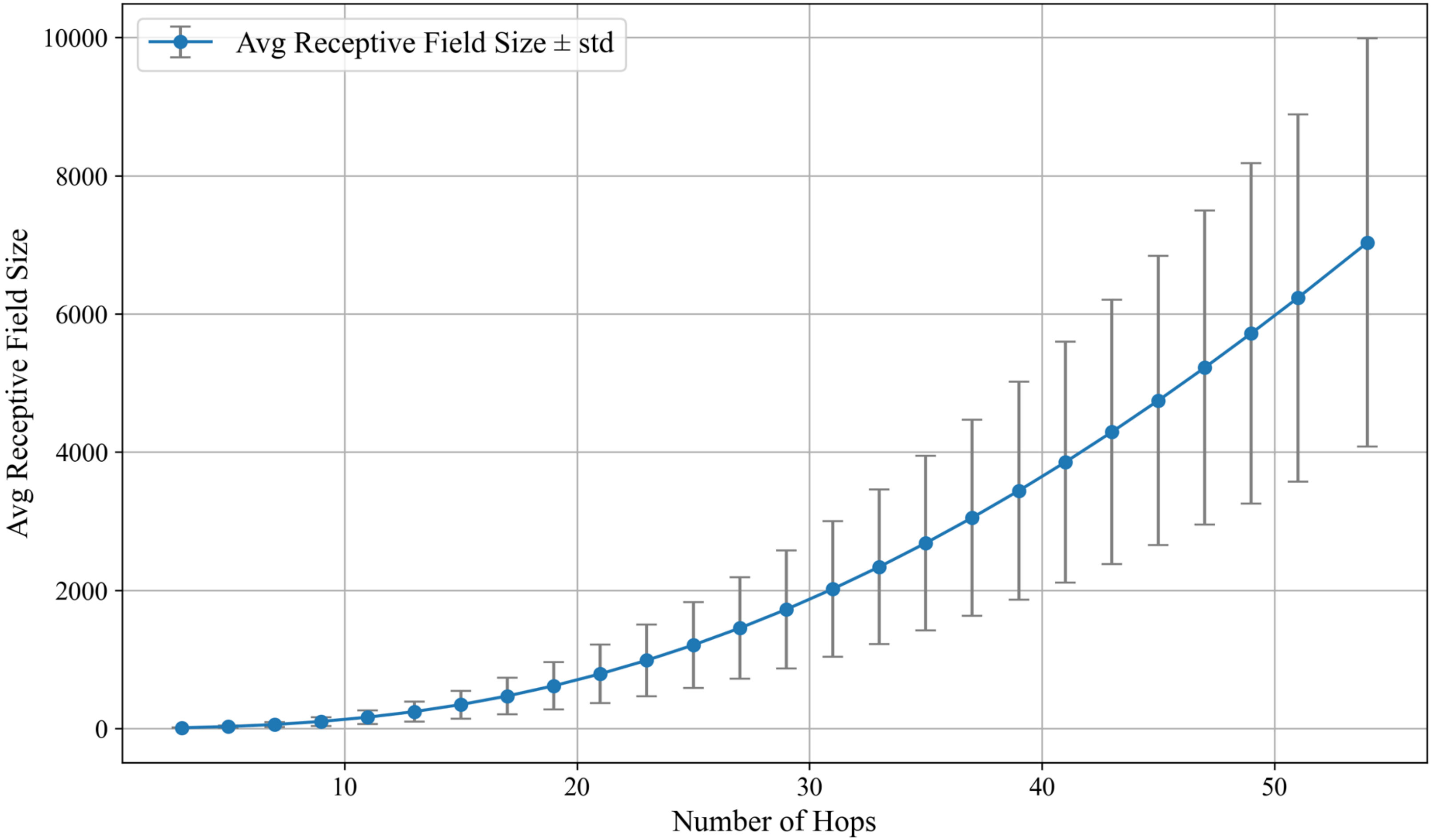
Why do the number of layers matter?

Example for a receptive field: In a graph with 3 edges, it takes 2 hops for all edges to exchange messages with one another through message passing.





With 54 layers, the receptive field spans roughly 7,000 nodes – about 25% of the graph



Disentangling Model Error from Simulation noise

The Bias-Variance-Decomposition of the MSE yields:

$$\text{MSE} = \underbrace{\sigma^2}_{\substack{\text{Aleatoric Variance} \\ \text{(Noise due to Simulation Stochasticity)}}} + \underbrace{\text{Bias}^2}_{\text{Squared Bias}} + \underbrace{\text{Var}}_{\text{Model Variance}}$$

→ *The inherent Simulation Stochasticity sets a lower bound to the Mean Squared Error.*

The Aleatoric Variance cannot be eliminated with the given data, only quantified - and even that is hard.

$y_{e,p}^r$ MATSim (actual) change in traffic volume on edge *e*, under policy *p*, in simulation run *r*

$$\overline{y}_{e,p} = \frac{1}{|R|} \sum_{r=1}^{|R|} y_{e,p}^r$$

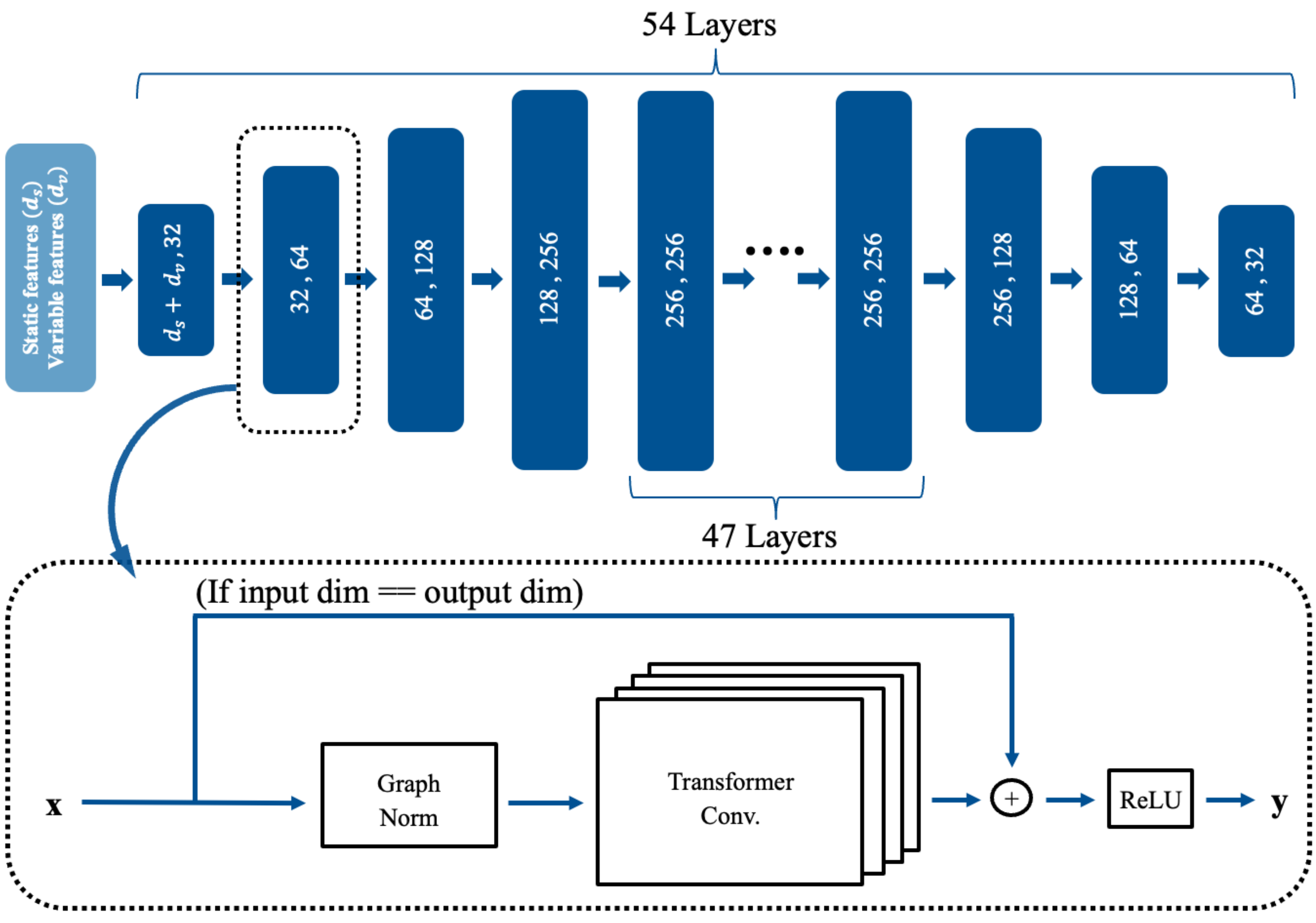
The Aleatoric Variance is: $\sigma_{e,p}^2 = \frac{1}{|R|} \sum_{r=1}^{|R|} (y_{e,p}^r - \overline{y}_{e,p})^2$

But this would require computing $y_{e,p}^r$ for many simulation runs r! Infeasible

We know the base case variance: $\sigma_{e,b}^2 = \frac{1}{|R|} \sum_{r=1}^{|R|} (v_e^r - \overline{v}_e)^2$

***Assumption:* $\sigma_{e,b}^2$ *serves as a good enough approximation of* $\sigma_{e,p}^2$**

The GNN consists of 54 layers Transformer Convolutions

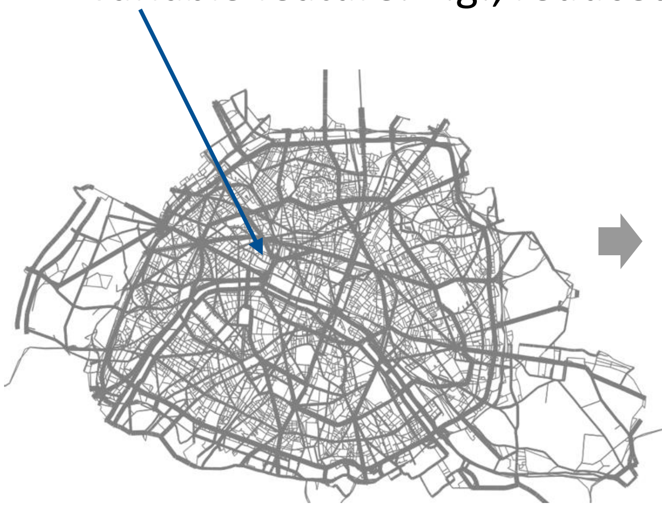


Transformer Convolution: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.TransformerConv.html

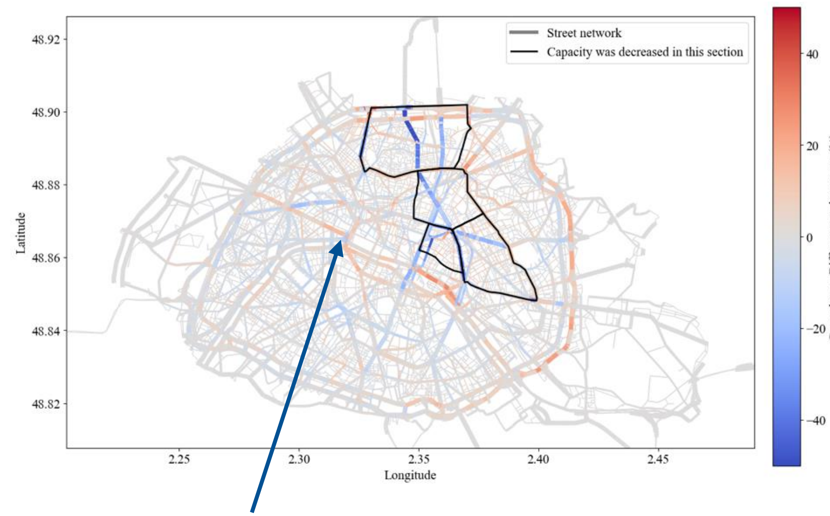
Input and Output of the GNN Surrogate

For each street:

- Static features: Traffic volume, capacity, speed limit, segment length
- Variable feature: E.g., reduced capacity



Surrogate



For each street: Change of traffic volume

Input Dimension per Scenario: 30,000 x 5

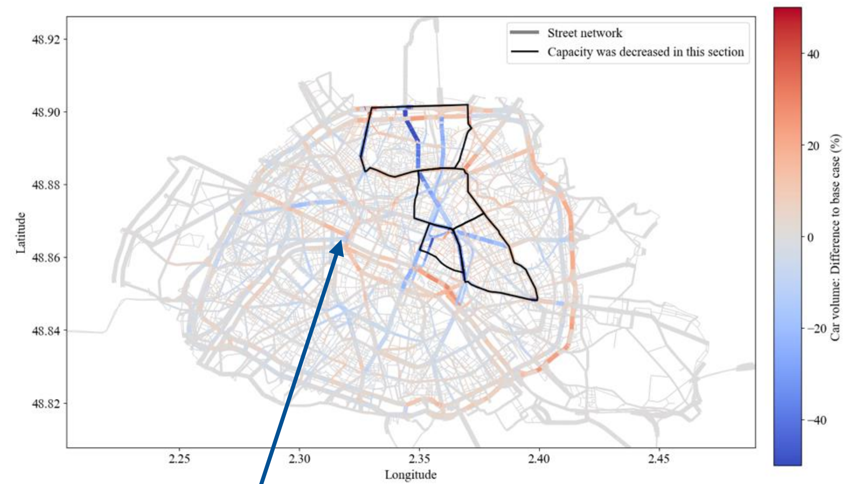
Output Dimension per Scenario: 30,000 x 1

Input and Output of the GNN Surrogate

For each street:

- Static features: Traffic volume, capacity, speed limit, segment length
- Variable feature: E.g., reduced capacity

Surrogate

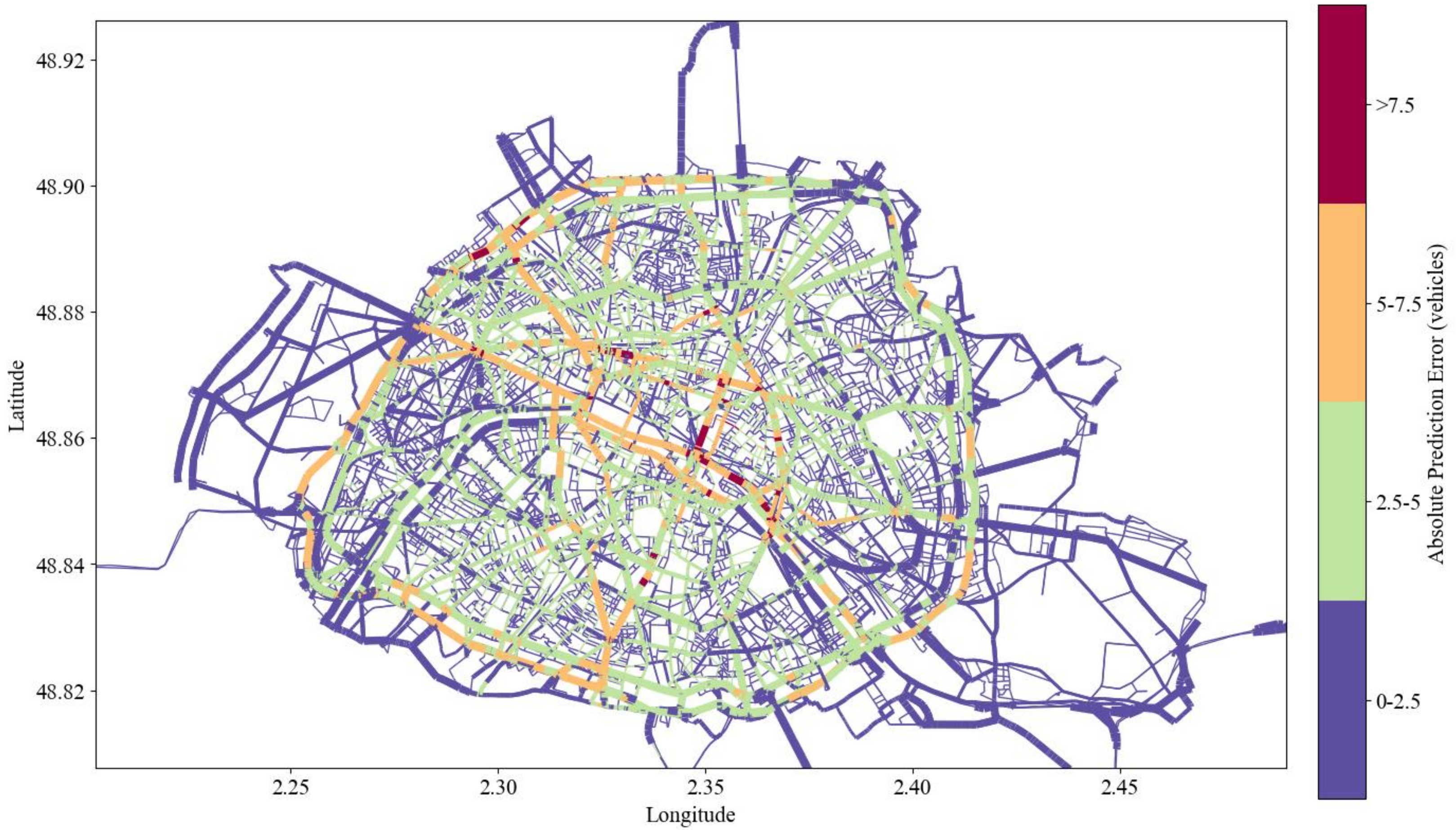


For each street: Change of traffic volume

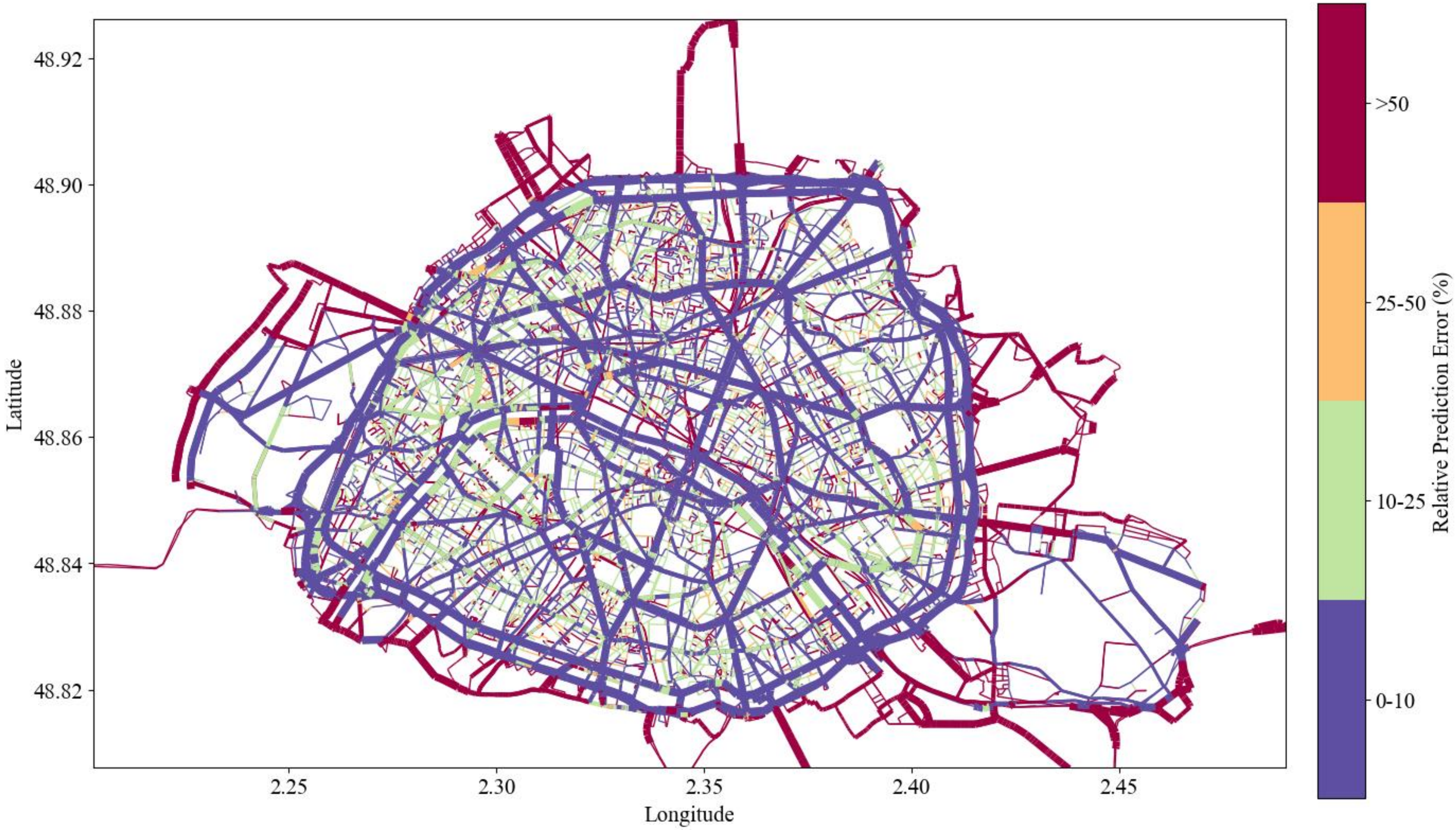
Input Dimension per Scenario: 30,000 x 5

Output Dimension per Scenario: 30,000 x 1

Absolute difference between predicted and actual (MAE)

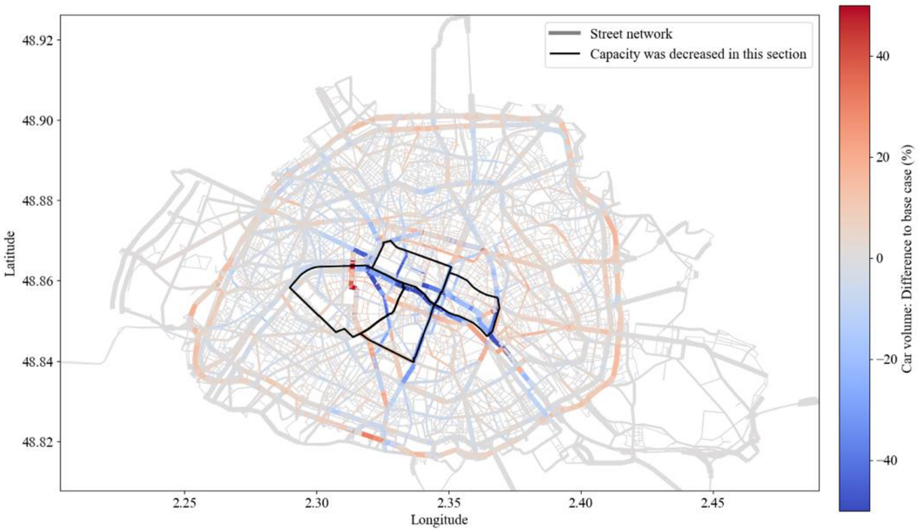


Relative difference between predicted and actual (MAE)

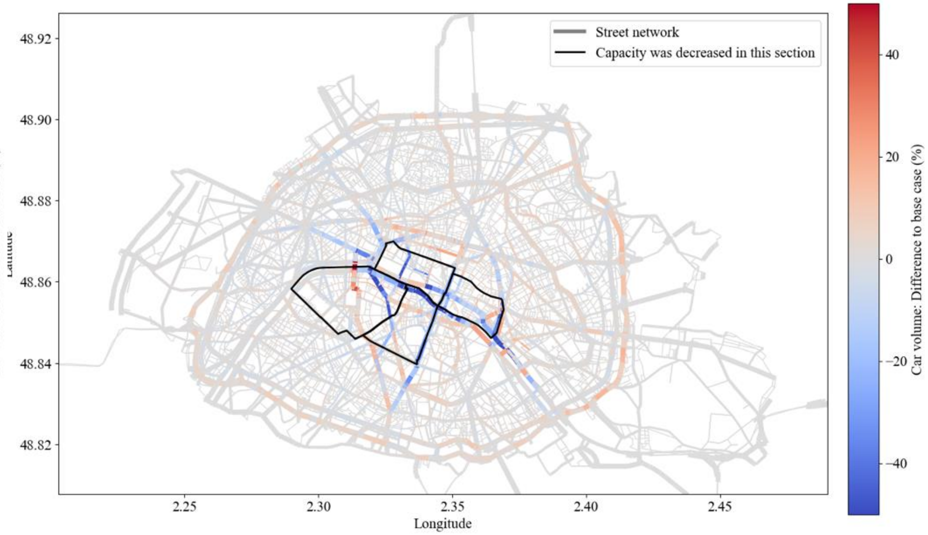


Network-Level Evaluation Show Effective Learning

Red/Blue: On those streets there was an increase/decrease of up to 50% in car volume

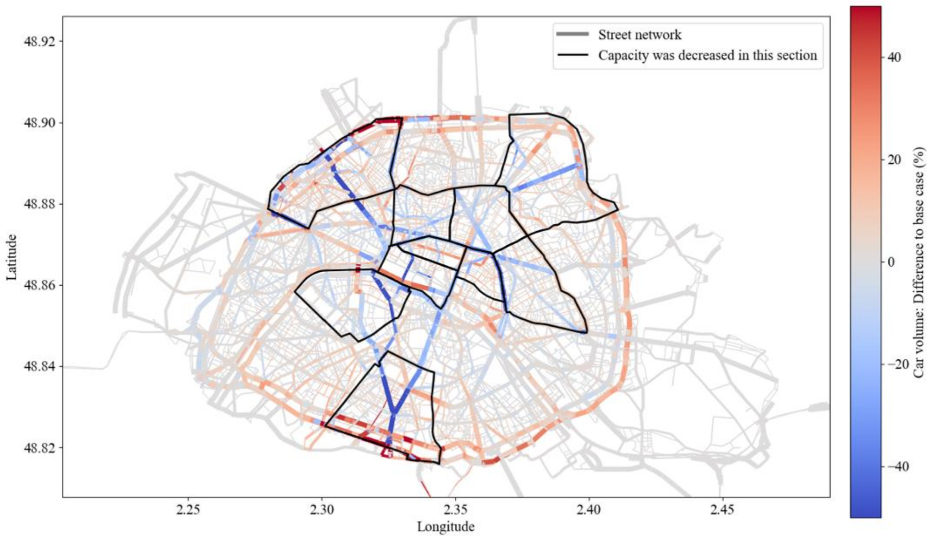


MATSim simulated change in car volume

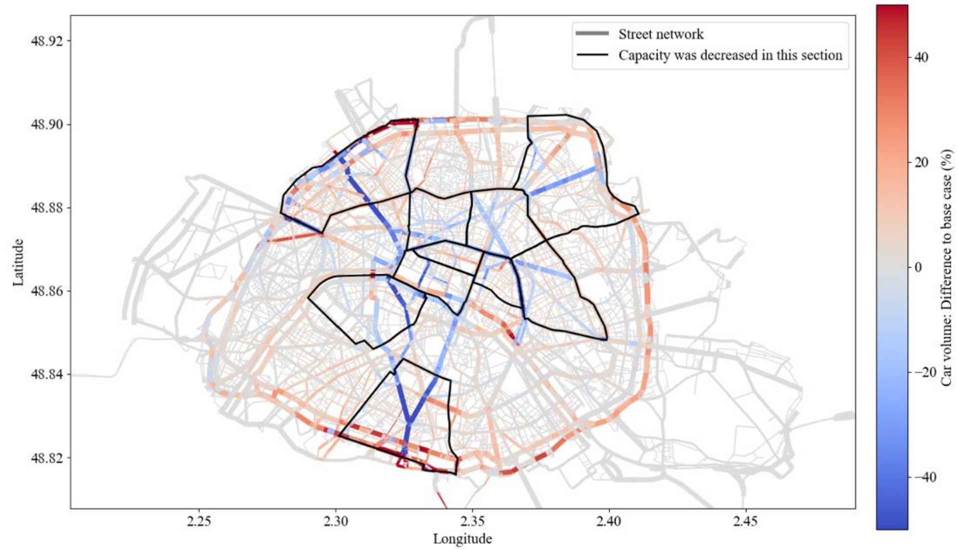


GNN surrogate predicted change in car volume

Some more examples (2/2)



MATSim simulated change in car volume



GNN surrogate predicted change in car volume

Receptive Field