Safety21
INNOVATING FOR SAFETY

US DOT National
University Transportation Center for Safety

Carnegie Mellon University

Penn
University of Pennsylvania

The Ohio State University

Morgan State University

CCAC Community College of Allegheny County

The University of Texas Rio Grande Valley

Evaluating Autonomous Vehicles' Pedestrian Safety Benefits in Road Networks

Carlee Joe-Wong (OrcID: 0000-0003-0785-9291)

Osman Yağan (OrcID: 0000-0002-7057-2966)

Final Report – August 31, 2026

DISCLAIMER

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle	5. Report Date
Evaluating Autonomous Vehicles' Pedestrian Safety Benefits in Road Networks	August 31, 2026
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Carlee Joe-Wong (https://orcid.org/0000-0003-0785-9291) Osman Yağan (https://orcid.org/0000-0002-7057-2966)	#604

9. Performing Organization Name and Address	10. Work Unit No.
Carnegie Mellon University 5000 Forbes Avenue Pittsburgh, PA 15213	**11. Contract or Grant No.** Federal Grant # 69A3552344811 / 69A3552348316

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
US Department of Transportation	Final Report (July 1, 2025 – July 31, 2026)
	14. Sponsoring Agency Code USDOT

15. Supplementary Notes

Conducted in cooperation with the U.S. Department of Transportation.

A more detailed description of some of the research conducted during this project can be found online at: https://arxiv.org/pdf/2605.07171, https://arxiv.org/abs/2505.15101, and https://arxiv.org/abs/2608.13212

16. Abstract

The purpose of this work was to investigate how CAV (connected autonomous vehicle) deployments affect pedestrian safety and the tradeoffs between safety and traffic congestion that are induced by CAV navigation around a city, particularly in the context of mixed human-driven vehicle and CAV populations. The project focused on developing methods to choose routing and control strategies that navigate this tradeoff, addressing scalability challenges in operating over city-wide road networks as well as exploring the use of large language models (LLMs) to reduce the need to gather new data on the effectiveness of different routing strategies. The main results include an algorithm that provably optimizes traffic congestion subject to a safety constraint, a characterization of the efficiency tradeoffs involved in partitioning a city into different regions for navigation control, techniques for making LLMs more efficient in generating navigation strategies for vehicles over time, and an analysis of how routing strategies must be adapted to local road topology in order to effectively account for traffic congestion.

17. Key Words	**18. Distribution Statement**
Autonomous vehicles, actuated traffic signal controllers, pedestrians	No restrictions.

19. Security Classif. (of this report)	**20. Security Classif. (of this page)**	**21. No. of Pages**	**22. Price**
Unclassified	Unclassified	14	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Problem Description

Though vehicles with varying degrees of autonomy and connectivity are being increasingly deployed in multiple different cities, their effects on road traffic patterns and pedestrian and vehicle safety have yet to be fully understood. Previous work has shown that the presence of connected autonomous vehicles (CAVs) can alter traffic patterns in mixed-autonomy settings, e.g., if CAVs can anticipate traffic incidents or road closures and coordinate to take alternate routes around these blockages, thus ensuring that traffic does not continue to accumulate around them and cause a cascade of congestion throughout a road network. However, the corresponding effects on safety have not yet been well-studied, particularly the effects on pedestrians.

There is reason to believe that having more CAVs on the road may improve safety: such vehicles, for example, may have faster reaction times and be equipped with more comprehensive environment sensors than human drivers, allowing them to avoid collisions with pedestrians or other vehicles and thus reducing injuries and fatalities. At the same time, there have been several recent stories in the news media of CAVs' involvement in traffic incidents that resulted in injuries or fatalities to bystanders and other vehicles, often due to their inability to properly react to rare or unexpected environment conditions. These safety effects can moreover vary based on different road types, times of day, etc.—meaning that an accurate prediction of autonomous vehicles' safety effects should include predictions of where and when they will be deployed, which is in turn affected by how and whether autonomous vehicles seek to minimize traffic congestion throughout a road network.

This project seeks to quantify how the presence of autonomous vehicles within a road network affects pedestrian safety, and consequently how autonomous vehicles should be routed within such networks so as to improve safety, while also aiming to relieve traffic congestion. The project in particular focuses in mixed-autonomy settings, where both human-driven vehicles and CAVs operate within a given road network, which also includes pedestrians and bystanders.

Approach and Methodology

Our approach follows three directions. We first design algorithms that can effectively choose vehicle routing strategies so as to trade off between the objectives of increasing safety (e.g., as measured by fewer expected injuries or fatalities) and reducing traffic congestion (e.g., as measured by queue lengths at traffic lights or average travel times). We then address two significant challenges to deploying these algorithms in practice: first, the need for coordination across large, city-wide road networks; and second, the need to balance adapting to current road network conditions with deploying policies based on prior knowledge or road network data. We detail our efforts along each research challenge below before discussing our main findings, conclusions, and recommendations.

Routing strategies to trade off safety and congestion

Since routing strategies that solely focus only on improving safety or congestion are likely to negatively affect the other objective, choosing a good routing algorithm requires predicting both its safety and congestion effects. Analytically quantifying either is often difficult, due to the scale and complexity inherent in road networks and the cascading effects that traffic changes in one area can have on other areas at later times. The two main strategies for handling this uncertainty are to use historical data to predict these effects, or to incorporate new data observed in real time. Since historical data under different routing strategies may be difficult to obtain, we initially consider using online and simulated observations to find a desirable tradeoff between safety and congestion.

We formalize this problem by modeling each routing strategy as a potential decision. Each strategy is associated with both a (known, fixed) cost and a (stochastic, unknown) reward. The "cost" can be thought of as the expected congestion and the reward as a safety statistic like number of fatalities prevented. The expected congestion can be predicted from offline simulators of road traffic networks, while the number of fatalities is more likely to be unknown due to lack of prior knowledge in how CAV deployments affect safety in different regions of a city, particularly as congestion levels (which may be correlated with safety concerns) change throughout the city. Future work might consider modifications of our proposed algorithms in which both congestion and safety are unknown and stochastic.

Our goal within this formal framework is to find the routing strategy that minimizes congestion, subject to a constraint on safety. To do so, we will try different strategies, observe their results (both in terms of safety and congestion) and choose a new strategy accordingly. Thus, our selection algorithm must balance between exploring low-cost, as-yet-unobserved strategies that may be feasible and exploiting strategies that are known to be feasible based on prior observations. We handle this tradeoff by proposing a cost-ordered feasibility algorithm that orders the potential decisions by their expected costs and then assesses their feasibility, starting with the lowest-cost decision. The idea is to successively eliminate decisions by observing their results until we find the lowest-cost, feasible decision. The main challenge in doing so is that the safety constraint is often specified in terms of a safety-optimal decision. For example, one might search for a routing strategy that achieves at least 80% of the reduction in fatalities that would be achieved by the routing strategy that optimizes solely for fatality reduction. This comparison would be straightforward if we knew the safety-optimal routing strategy; however, in practice we do not. Thus, we must simultaneously estimate the safest strategy as well as whether the current candidate strategy meets our safety threshold.

We solve this challenge by maintaining an estimate of the maximum possible safety based on past observations. We maintain a set of candidate "safest/highest reward" strategies and, for each strategy in this set, we estimate the reward (i.e., safety criterion) by

averaging our observations and adding an uncertainty factor that decreases with the number of observations for each strategy. The uncertainty factor effectively increases the estimated reward of these highest-reward candidates, ensuring that we obtain a conservative "best" safety estimate. At the same time, we average the observed safety statistics of our candidate lowest-cost strategy and subtract an uncertainty factor that depends on the number of observed outcomes of this strategy. We then compare this deflated estimate with the conservative "best" estimate: if the deflated estimate of our candidate strategy is higher than our conservative best safety estimate, we can conclude with high probability that our candidate strategy does indeed meet the desired safety threshold and should be employed. If not, we must collect more data samples to improve our safety estimates.

We have evaluated this proposed strategy both theoretically and in simulations. Our primary performance metrics are the cost and quality regrets, defined as the total incurred of the cost or reward, respectively, of our cost-ordered feasibility algorithm, less the cost/reward of the optimal (i.e., cost-minimizing subject to a reward constraint) strategy. On the theoretical side, we derive both upper and lower bounds for the cost and quality regrets. Our lower bounds, which lower bound the achievable regret for any algorithm that solves this optimization problem, are in terms of the number of samples of each decision that are required for finding the lowest-cost feasible decision, and thus inform our cost-ordered feasibility algorithm. We also upper bound the cost and quality regret achieved by our cost-ordered feasibility algorithm, showing that our upper bound is smaller than that achieved by other decision-making algorithms. Our cost-ordered feasibility's upper bound does not quite match our derived lower bound, indicating that future performance improvements are possible, and we suggest modifications that may improve algorithm performance.

Experimentally, we compare our cost-ordered feasibility algorithm with previously proposed algorithms for solving this optimization problem. Across multiple datasets and experiment settings, our algorithm has the lowest cost and quality regret (Figure 1). For example, we use the parameter alpha to quantify the strictness of the reward constraint, i.e., the chosen strategy should have at least (1 – alpha) fraction of the reward of the reward-optimal strategy. As alpha increases and more decisions become feasible, our algorithm's regret also decreases. For moderately low values of alpha, we obtain the lowest average regret; the TS-CS (Thompson Sampling-Cost Subsidy) algorithm does sometimes attain lower regret, but TS-CS does not have provable upper bounds on the regret, with counterexamples showing it can perform arbitrarily poorly on some problem instances.

Coordinating multiple decision-making agents

CAV deployments are often large-scale, spanning an entire city's road network with a fleet of multiple vehicles. Thus, using the same routing strategy for all, or even a strategy for

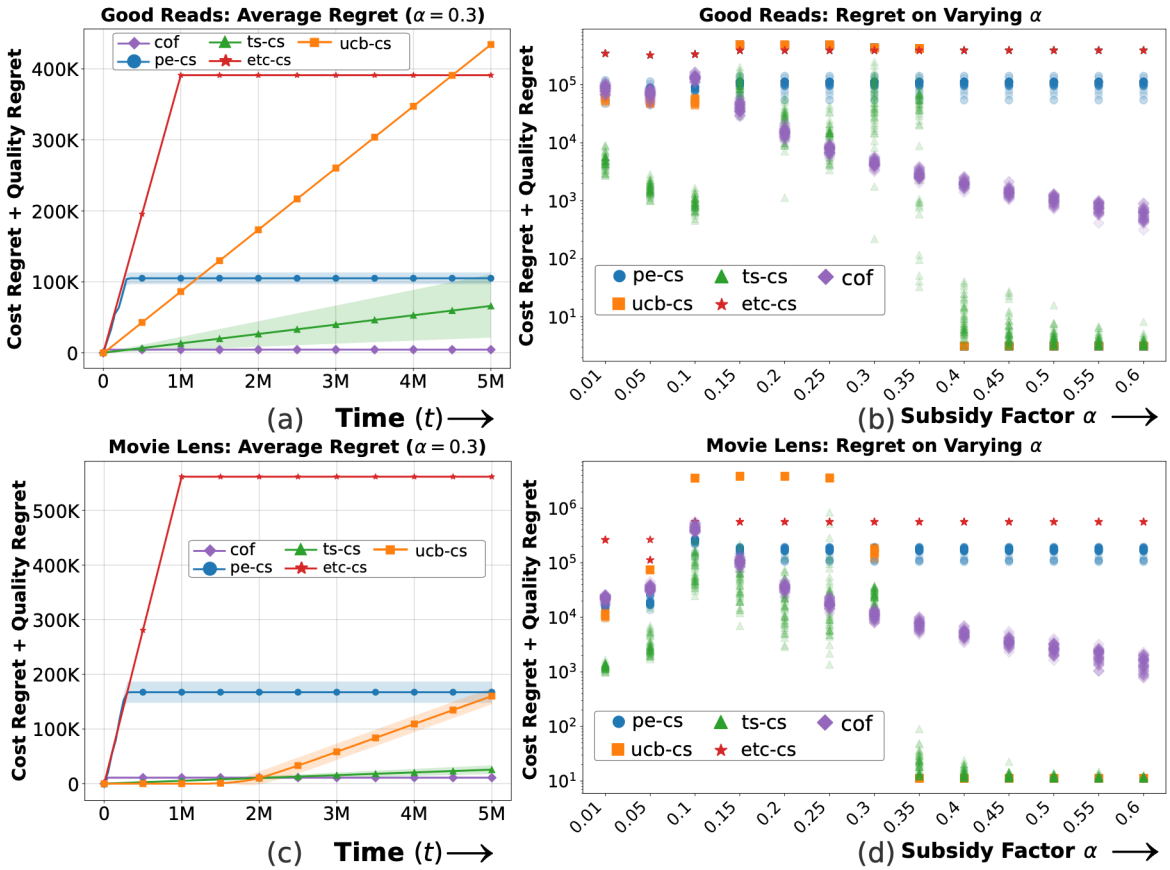


Figure 1: Our cost-ordered feasibility (COF) algorithm has generally lower cost and quality regret than previously proposed baselines, as the length of the experiment increases (a,c) and for different subsidy factors on the reward constraint (b,d).

CAVs in different areas of the city, introduces significant scalability challenges. Thus, our second goal was to develop techniques for multiple agents, e.g., multiple agents whose decisions route CAVs, to coordinate in routing CAVs around a city.

Because the prevalence of pedestrians can greatly vary in different areas of a city, our initial approach was to divide the city into geographical regions and choose a routing strategy within each of them. Our approach is therefore flexible to cases where a single controller is in charge of CAVs in each region, or when multiple CAVs, traffic lights, etc. control CAV movement in each region. We therefore focused on evaluating strategies for dividing a city into regions. To do so, we searched for city-wide datasets that captured traffic congestion at the street level; our goal was then to divide the streets into different regions and evaluate our approaches within different cities. Our dataset, taken from the CityFlow simulator used in the paper *CoLLMLight: Cooperative Large Language Model Agents for Network-Wide Traffic Signal Control* published in the International Conference on Learning Representations (ICLR) 2026, includes the road topology from New York City as well as several cities in China such as Jinan and Hangzhou.

We tried two different partitioning strategies: symmetrized Louvain and InfoMap. The Louvain algorithm is based on community detection; it tries to find groups of nodes with high connectivity to each other. We treat each intersection as a "node" and quantify their connectivity by the number of cars that travel between them. InfoMap is instead based on tracking the areas in which vehicles travel, grouping regions according to whether vehicles tend to stay within the same region. We implemented both of these algorithms on the New York road network, which includes 196 intersections. Each algorithm has two variations: the "hard" algorithm and the "topology" variant that requires each intersection to have at least two neighbors in its region.

To evaluate the resulting partition, we used large language models (LLMs) to control traffic in each region; one LLM agent was in charge of each of the regions. The table below shows the resulting AWT (average wait time), AQL (average queue length), and ATT (average travel time) for these partitioning strategies across all vehicles, compared to a baseline in which each intersection is its own region. The "size balanced" baseline tries to divide the network into regions of equal size. We observe that a higher number of regions generally leads to better results (lower AWT, AQL, and ATT) due to having more flexibility in the LLM agent actions at different intersections, although the Louvain partitions are slightly worse than the InfoMap partitions, despite having more regions.

Strategy	# Regions	Avg Size	AWT (s)	AQL	ATT (s)
Baseline	196	1	82.94	1929.7	1025.7
InfoMap Hard	15	13	206.0	2290.0	1167.3
InfoMap Topology	11	18	206.0	2290.0	1167.3
Louvain Hard	23	8.5	210.2	2320.7	1168.6
Louvain Topology	23	8.5	205.3	2332.0	1170.3
Size Balanced	7	28	323.8	2600.0	1350.0

The table below shows that there is a tradeoff between having more regions, and thus needing more coordination between agents, does introduce a tradeoff with the number of tokens. Each token represents a unit of input or output to the LLM; thus, a higher number of tokens indicates longer delays and greater monetary and computational expense. The table below shows that the token requirements, as a percentage of a fixed token budget, change roughly proportionately with the number of regions.

Region Size	Estimated Number of Tokens	% of Token Budget
8	~18.5K	14%
16	~37K	28%
32	~74K	57% -> S*
64	~148K	>100%

Based on these findings, our next research focus was to reduce the token consumption of using LLMs to make decisions for CAV and vehicle routing around the road network.

Large language model (LLM)-based decision making for CAVs

Our final research focus for this project was to incorporate large language models (LLMs) into CAV decision-making. While our initial focus on safety and congestion tradeoffs assumed the ability to gather new data as different strategies are tried, in practice such data collection may be limited: real-world data collection with unknown pedestrian safety implications is potentially dangerous, and even simulator-based data collection can be computationally intensive, particularly when evaluating strategies at a city-wide scale. Incorporating LLMs allows us to speed up learning by leveraging the knowledge embedded in LLMs during their training process, even if they have not been specifically trained to make traffic management decisions.

We used the same traffic dataset as we used to examine division of the city into regions. Instead of controlling each CAV individually, we considered the problem of controlling traffic lights at each road intersection. This formulation allowed us to abstract away from directly routing CAVs, which would require accounting for their origins and intended destinations, thus allowing us to generalize our evaluation better to different cities' road networks. At each traffic light, the LLM decides the frequency and timing of when the light in each direction turns green, with built-in safety templates to ensure opposing directions do not have a green light at the same times. Our goal was to adapt the traffic light timing according to the current number of cars at each intersection; thus, the LLM is given both the layout of the road network and the current number of cars waiting in each direction as part of its context. The traffic lights change every 10-30 seconds, while the LLM decides the traffic light strategy every 5 minutes. Figure 2 illustrates the resulting traffic light control problem.

A significant challenge that emerged as we began to evaluate this paradigm was the cost, in both time and resources, of using the LLM every few minutes. Since the LLM needed to use a significant amount of context in its decision making, it needed to consume thousands of input tokens for every decision, as was further shown in the table above. We therefore focused on reducing this cost by taking advantage of the fact that much of the data in the cache, aside from current vehicle statistics, does not change significantly from

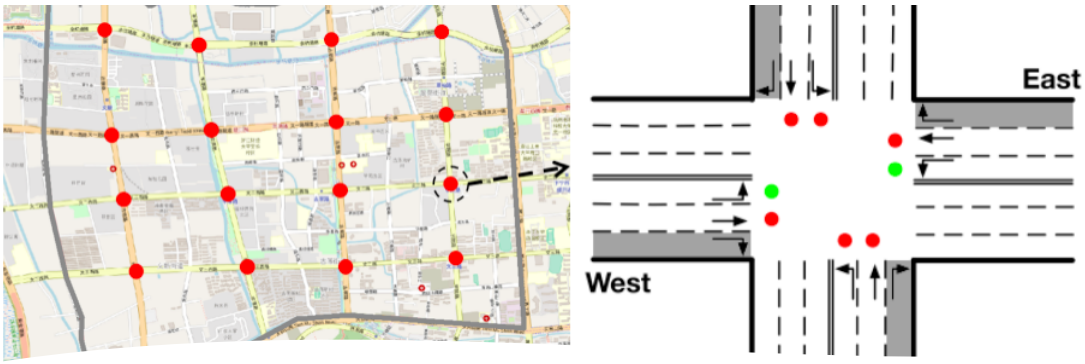


Figure 2: Network of road intersections (left) and an example intersection showing red and green lights for different directions of traffic (right).

timeslot to timeslot. Thus, we can cache some of the intermediate LLM outputs and reuse them in subsequent timeslots, thus saving us from regenerating some tokens and reducing the LLM's inference time. As the vehicle statistics change, we refresh the cache; the refresh frequency is then a hyperparameter that allows us to control the amount of savings (with fewer refreshes leading to more savings) versus the performance of the LLM.

Figure 3 shows the time to first token (a measure of LLM inference latency) and prefill time when we refresh the cache at a frequency of 5 (left) and 15 (right) timesteps. We observe that in both cases, the time to first token is limited to about 0.2 seconds, which is well within the tolerance of a LLM that updates the traffic control every few minutes. The prefill latency spikes whenever the cache is refreshed, as expected, and is overall slightly lower for a refresh frequency of 15. We also evaluated the average queue lengths, travel times, and vehicle waiting times in the Jinan topology for different cache refresh frequencies, as shown in the table below. Lengthening the refresh interval worsens each of these metrics, though only by a few percentage points. We therefore conclude that such a refresh method is viable for LLM control and can help to make such control more efficient and feasible when used over time.

Jinan (3x4)	Baseline	Refresh@5	Refresh@10	Refresh@20	No Refresh
Average queue length	187.68	197.80	205.65	217.83	230.8
Average travel time (s)	293.89	299.8	306.04	313.99	321.69
Average waiting time (s)	43.11	44.31	44.78	46.45	46.47

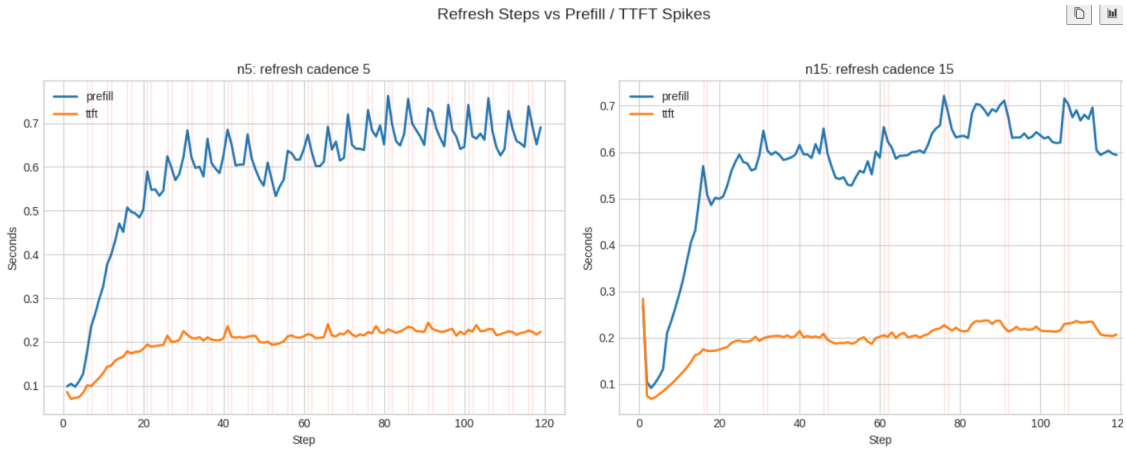


Figure 3: Prefill latency and time to first token (TTFT) for cache refresh cadences of 5 (left) and 15 (right).

Adaptive LLM routing for CAV decision-making

While caching reduces the repeated processing required by a single LLM, a complementary way to reduce decision-making cost is to adaptively choose which model, or combination of models, should be queried for each traffic state. Traffic-control decisions can vary considerably in difficulty. Routine conditions with stable vehicle and pedestrian patterns may be handled by a smaller, faster model, while unusual conditions, such as abrupt queue growth, conflicting observations, or increased pedestrian exposure, may require a more capable model or agreement among several models. In our related work on LLM routing (Elumar et al. NeurIPS 2025), we formulate this selection problem as an online contextual and combinatorial bandit problem. The current state is used to estimate the reliability and cost of each available model, and the controller selects the least costly set of models expected to meet a desired confidence level.

Such a controller can also operate sequentially. It can begin with one or more low-cost models and stop when their agreement and estimated confidence are sufficiently high. When confidence is low or the models disagree, the controller can escalate the decision to stronger models. For CAV and traffic-signal control, the context could include the road-network topology, current vehicle queues, pedestrian counts or exposure estimates, recent changes in traffic, and information from nearby controllers. Safety templates, such as prohibiting conflicting green phases, would remain hard constraints independent of the selected model. This approach complements the cache-refresh mechanism described above: caching determines how much previous computation can be reused, while routing determines how much new computation and model capability are needed for the current state. It also aligns with the cost-ordered feasibility framework developed earlier in this report, since the objective is again to use the lowest-cost option that satisfies a quality or safety requirement.

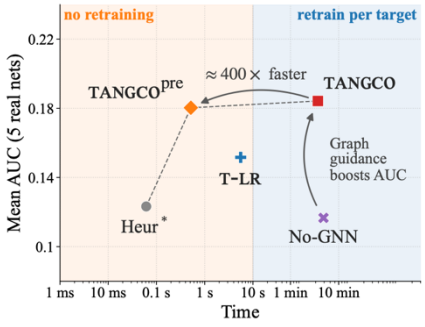
Topology-aware allocation for robustness against cascading congestion

Road-network control also has a robustness dimension. A crash, road closure, signal failure, or abrupt rerouting decision can shift traffic to neighboring intersections, overload their available capacity, and produce a cascade of congestion. In related work, we developed TANGCO (Topology-Aware Neural Graph-Guided Capacity Optimization), which treats this problem as capacity allocation in a flow network. Given a network, its initial loads, and a fixed total capacity budget, TANGCO uses a graph neural network (GNN) policy trained through a cascade simulator to distribute spare capacity across nodes to improve robustness against local load redistribution.

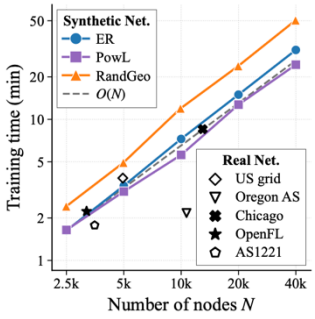
TANGCO improved on the best of four hand-designed allocation heuristics in all 450 synthetic-network instances and in 40 of 45 conditions involving real-network topologies, with relative robustness gains ranging from 1.6% to 246%. These networks included a Chicago road-network topology, as well as power, air-transportation, and Internet networks. A version pre-trained on diverse synthetic graphs also transferred to unseen real networks without network-specific retraining. To apply this framework to CAV control, nodes could represent intersections or regions, load could represent vehicle demand or queue length, and spare capacity could represent signal-timing flexibility, rerouting slack, or coordination resources. Pedestrian exposure and conflict risk could be included as node or edge features and incorporated as safety constraints or penalties. The current TANGCO model does not directly represent pedestrian safety, but it provides a promising foundation for allocating limited control resources to locations where congestion spillovers and pedestrian consequences are jointly most significant.

Network	Heur.*	TANGCO^{pre}	TANGCO
Uniform load			
US grid	0.022	0.042	0.034
Oregon AS	0.237	0.336	0.319
Chicago	0.019	0.023	0.020
OpenFlights	0.076	0.261	0.262
AS1221	0.263	0.239	0.286
Pareto load			
US grid	0.020	0.060	0.040
Oregon AS	0.243	0.321	0.326
Chicago	0.012	0.011	0.012
OpenFlights	0.082	0.213	0.240
AS1221	0.260	0.250	0.282

(a) Robustness (AUC)



(b) Robustness vs. deployment cost



(c) Training time vs. network size

Figure 4: (a) Robustness on five real networks including the Chicago road network; TANGCO^{PRE} is pre-trained on synthetic graphs and applied with no per-network training, while TANGCO trains per network. (b) Robustness versus time to first allocation on a new network. (c) Median training time versus number of nodes.

Findings

The detailed findings from our research activities are covered in the "Approach and Methodology" section below. In this section, we highlight the main findings and their implications for CAVs and pedestrian safety.

- **Algorithms should be carefully designed to balance pedestrian safety and traffic congestion.** Our results show that naively designed algorithms that aim to balance these objectives may have arbitrarily bad performance in some special cases. In transportation scenarios, such performance may lead to increased fatalities, so care is needed when choosing an algorithm to solve these optimization problems.
- **Centralized CAV control is not scalable, but some coordination is necessary.** Our results show that distributed control often leads to superior performance compared to centralized control, due to its increased flexibility, but at the cost of increased coordination overhead, e.g., through communication messages. Partitioning the road network into regions may offer a good tradeoff between coordination benefits and overhead.
- **LLMs can effectively make decisions, but they must be used carefully to ensure scalability.** Naïve usage of LLMs can incur high token costs and latency. This project demonstrates the feasibility of two alternatives: one that takes advantage of the repeated information across time, and another that carefully selects lower-cost LLMs when possible to do so without compromising decision quality.
- **Scalable CAV control requires topology-aware coordination.** The effects of CAV control decisions depend on local road topology and traffic patterns. Thus, an effective control strategy requires accounting for this local structure, and in particular the effects of cascading traffic congestion throughout a road network. This finding further supports the idea of road network partitioning.

Conclusions and Recommendations

Taken together, our results suggest that scalable CAV control that takes both traffic congestion and pedestrian safety into account will require both computational adaptation and topology-aware coordination. At the decision-making layer, adaptive LLM routing can reserve more expensive models for uncertain or safety-critical traffic states. At the network layer, GNN-based allocation can identify where limited control capacity should be placed to reduce spillovers and cascading congestion. A natural next step is to combine these ideas with the cost-ordered feasibility framework: candidate signal or routing actions would be generated and evaluated using the least-cost reliable set of models, while the final policy would minimize congestion subject to an explicit pedestrian-safety requirement. Such a system should first be evaluated in simulation using pedestrian counts, conflict points, near-miss indicators, and other crash-surrogate measures before being considered for field deployment.