Improving Urban Planning Through Agent Based Modeling and Q-Learning

Tristan Kalvoda
kalvodatristan@gmail.com
Division of Science and Mathematics
University of Minnesota, Morris
Morris Minnesota USA

Abstract

This paper explores the use of agent-based modeling (ABM), enhanced by reinforcement learning techniques such as Q-learning, to optimize urban transportation systems. The focus is on modeling and improving aspects such as traffic design, vehicular flow, and pedestrian mobility. The central research question is how to effectively simulate realistic agent behavior in order to develop models that can inform and support policy-making for more efficient and adaptive urban planning. The paper presents and analyzes simulation-based case studies that demonstrate how learning agents can reproduce realistic movement patterns and provide insights for larger-scale urban systems.

Keywords: Agents, Agent-Based Models, City Planning, Urban Systems, Q-Learning

1 Introduction

Imagine driving toward a city on a major highway, expecting to arrive in a few hours. Suddenly, the cars ahead slow to a crawl. Soon you hear incessant honking and see a vast sea of cars, trucks, and buses extending beyond the horizon. Your car inches forward a few dozen feet per hour, as you find yourself trapped in an enormous traffic jam. You are stuck on this road for what feels like days, and it is.

This historic event unfolded in August 2010 on China Highway 110, a road linking resource rich regions in Inner Mongolia to the metropolis of Beijing. The jam lasted 12 whole days, with some people being stuck for 5 days. It was caused by years of steadily increasing traffic, nearly 40% annually, and a surge of heavy coal trucks transporting resources from the mines, which overwhelmed the region's railway capacity [4].

Today, the stakes are much higher—51% of all global commutes are made by car, with regions like the United States and Canada reaching as high as 92% of commutes being by cars [7]. This represents an immense volume of daily movement. With hundreds of millions of vehicles navigating roads, intersections, and transit networks, there is enormous pressure to ensure that infrastructure, traffic systems, and regulations are not only efficient, but adaptive.

Such complex systems demand more than just static planning. They require tools capable of simulating change and capturing emergent behavior–system-wide patterns arising from the interactions of individual components that cannot be predicted by examining the components alone. In urban systems, these interactions often stem from the decisions and behaviors of individuals. *Agent-Based Modeling (ABM)* offers a powerful framework to test these changes and simulate their effects by representing individual "agents" and observing how their actions influence the larger system. In the Highway 110 traffic example, these agents correspond to individual vehicles or drivers, each following their own rules and making local decisions that collectively produce large-scale traffic patterns.

For example, imagine a city planning to introduce a new bike lane along a major commuter route. Traditional models might estimate traffic volumes and their changes by using averages derived from historical data. These models are unable to capture a lot of the complexity in systems, like individual choices and interactions. In contrast, ABMs allow planners to simulate how individual drivers, cyclists, pedestrians, and even local businesses may respond to the change. In this scenario, the drivers, cyclists, and pedestrians act as agents whose behaviors can be modeled and observed.

Some drivers may decide to reroute their commute to avoid the areas with the reduced car lanes, shifting congestion to neighboring streets. Others may see the bike lane as a safer, more convenient option and switch from driving to biking, reducing the number of cars on the road. Pedestrian traffic may increase too, when there are fewer cars on the streets and more bike and pedestrian-friendly intersections. These individual decisions interact and compound, creating ripple effects that extend throughout and even beyond the areas neighboring the bike lane.

By simulating this large amount of individual behaviors and interactions, ABMs help planners anticipate both the positive outcomes and the unintended consequences. This deeper insight enables more informed decision-making and supports the design of smart, efficient, and adaptable cities.

2 Background

2.1 Agents

An agent is an autonomous actor that can observe its surroundings, make decisions, and take actions based on a set of rules or objectives. Agents can represent people, vehicles,

businesses, or any other individual component of a system that can act independently.

Consider the example of a driver in a traffic simulation, the driver:

- Perceives their surroundings, including nearby vehicles, roads, and traffic devices
- Makes decisions such as slowing down, turning, or changing lanes
- Acts on those decisions by looking for an exit, checking if adjacent lane is clear, and switching when lanes when safe

Each agent behaves according to its own logic (Figure 1). Figure 1 illustrates the basic loop that a pedestrian agent follows in a simulation: the agent observes its surroundings, interprets that information as states and rewards, and then chooses an action that affects the environment. This decision process can be based on simple rules like "If traffic in a lane is slow then swap lanes" or more complex logic like "this road is busy at this time, so take an alternate route." Even though each agent operates independently, their actions influence each other. These actions lead to *emergent behavior*, where the system as a whole behaves in complex, unexpected patterns that arise from the interactions of many individuals.

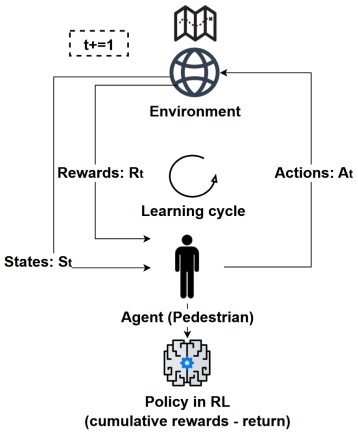


Figure 1: Pedestrian agent schematic [6].

2.2 Agent-Based Models

Agent-Based Models (ABMs) are computational simulations that represent and track the behaviors and interactions of many individual agents within a shared environment. You define how individual agents behave, and then observe what happens when they interact.

This framework is particularly useful in urban planning, where complex patterns such as congestion or neighborhood development emerge from countless local decisions. Each agent, whether representing a driver, pedestrian, or household, acts based on its own goals and perception of the environment. The collective results of these interactions allow researchers to study how small behavioral adjustments can lead to major system-wide effects, such as shifts in traffic flow or land use.

ABMs therefore serve as a bridge between micro-level decision-making and macro-level urban dynamics, providing a way to test changes and interactions without making the real-life changes.

2.3 Reinforcement Learning

Reinforcement Learning (RL) is a computational approach that enables an agent to improve its behavior through experience. Unlike traditional programming, where all decisions are pre-defined, RL allows an agent to dynamically adjust its behavior in order to achieve a goal based on feedback from its environment. [5]

It is part of the broader field of machine learning, which focuses on developing systems that can automatically adapt or make decisions without being explicitly told what to do. The agent learns through a process of trial and error, where actions that lead to positive outcomes are reinforced, and those that don't are discouraged. Over time, the agent develops a strategy for making better choices in new or changing situations.

While reinforcement learning involves agents improving their behavior through experience, real drivers and pedestrians do not consciously perform such learning in the formal sense. In our model, RL serves as a practical framework for constructing agents whose decisions approximate realistic behavioral patterns. The purpose is not to suggest that individuals are optimizing rewards in real time, but to use RL as a mechanism for generating consistent, data-driven responses within the simulated environment.

2.4 Q-Learning

Q-Learning is a framework in which an agent interacts with an environment by observing a state s, taking an action a, and receiving a reward R. The goal is to learn a policy π – a rule that tells the agent which action to choose in each state–that selects the action in each state that maximizes long-term reward. For any state s, the set of all actions the agent is allowed to take is denoted by $A(s)$. Each state–action pair (s, a) must satisfy $a \in A(s)$, meaning the Q-table — a table storing the learned values $Q(s, a)$ for each valid state–action pair—only contains values for actions that are actually available in each state.

Q-Learning estimates a function called the Q-value, which represents the expected future reward of taking a certain action in a given state and then following the optimal policy. These values are stored in a Q-table, where each entry corresponds to a state–action pair [2]. At each decision step, the agent typically selects the action with the highest Q-value (unless it is exploring, as seen in Section 2.4.1) . After taking

an action and observing the resulting reward and next state, the agent updates the corresponding Q-value entry $Q(s, a)$ in the table. This update incorporates the new information and improves the agent's future decisions, gradually refining the policy as learning progresses.

The specific way $Q(s, a)$ is updated is determined by the Q-Learning update rule, which is derived from the Bellman optimality equation (see Section 2.4.1). The algorithm adjusts the current estimate $Q(s, a)$ toward a new target value based on the immediate reward and the highest-valued action in the next state.

2.4.1 Bellman Equation. The Q-Learning update rule is derived from the Bellman optimality equation, which expresses the value of an action as the immediate reward plus the discounted value of the best possible future action [2]. In practice, an update occurs every time the agent takes an action, observes the resulting reward, and transitions to a new state. The new information is then used to update $Q(s, a)$ toward a target value based on the observed reward and the highest estimated value of the next state.

$$Q(s, a) \leftarrow Q(s, a) + \alpha\Big(R + \gamma \max_{a'} Q(s', a') - Q(s, a)\Big)$$

- s: the current state.
- a: the action taken by the agent.
- s': the next state after taking action a.
- R: the reward received for taking action a in state s.
- $\max_{a'} Q(s', a')$: the highest currently estimated action-value for the next state.
- γ: the discount factor balancing immediate and future rewards.
- α: the learning rate, controlling how much new information revises old estimates. Smaller values reduce the influence of noisy or fluke rewards, promoting stability.

The expression in parentheses, $R + \gamma \max_{a'} Q(s', a') - Q(s, a)$, is known as the *temporal-difference (TD) error*. It measures how different the current estimate $Q(s, a)$ is from the value suggested by the new information the agent has just received. The term $R + \gamma \max_{a'} Q(s', a')$ is called the *target*. It represents the updated estimate of what the true value of taking action a in state s should be, based on the immediate reward and the highest currently estimated future value from the next state.

For example, suppose the agent takes action a in state s, receives a reward $R = 10$, and the highest currently estimated future Q-value in the next state is 20. In this case, the target is $10 + \gamma \times 20$. The TD error is the difference between this target and the agent's current estimate $Q(s, a)$. The update then moves $Q(s, a)$ a fraction α of the way toward the target, meaning that larger TD errors produce larger adjustments, while smaller errors result in more modest updates.

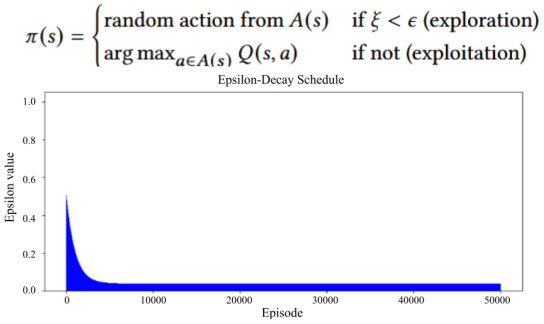


Figure 2: Epsilon-greedy action selection and ϵ-decay schedule. [6].

2.4.2 Epsilon-Greedy Policy. To balance exploration (trying new actions) and exploitation (choosing the best known action), Q-Learning typically uses an ϵ-greedy policy. At each decision step, the agent selects one action: with probability ϵ it chooses a random action, and with probability $1 - \epsilon$ it selects the action with the highest Q-value.

The authors change the value of ϵ over time; this process, known as ϵ-decay, shifts the agent toward exploiting accumulated knowledge by focusing on actions more likely to yield higher rewards. Starting with a high ϵ encourages the agent to explore the environment thoroughly and gather diverse experiences early in training.

Figure 2 illustrates this process, showing both the ϵ-greedy action-selection mechanism and the ϵ-decay schedule. This dynamic ensures that the agent avoids becoming trapped in suboptimal strategies while still converging toward an effective policy.

In this context, the policy π is the rule that selects an action for each state. The full expression for $\pi(s)$ is given in Figure 2, but conceptually, an ϵ-greedy policy means the agent chooses a random action with probability ϵ, and chooses the action that maximizes $Q(s, a)$ with probability $1 - \epsilon$.

2.5 Combining ABMs and Q-Learning

When combined, ABMs provide the dynamic environment in which Q-learning agents operate, while Q-learning supplies each agent with a learning mechanism that adapts through feedback. As agents act within a simulated environment, the ABM updates environmental conditions such as traffic density, pedestrian flow, or traffic signal states, which in turn influence the rewards that guide future decisions. This feedback loop allows agents to learn context-dependent strategies that emerge from experience, rather than simply following fixed, pre-programmed behaviors. While agents may still respect rules encoded in the environment, such as traffic laws, their actions adapt based on observed outcomes and rewards.

For example, in a traffic simulation, a driver agent may initially choose routes randomly. As the simulation progresses,

the agent observes congestion levels, wait times at intersections, and changes in traffic signals. These observations update the agent's Q-values, reinforcing actions that lead to shorter travel times or smoother traffic flow. Simultaneously, the ABM updates the environment based on all agents' behaviors, such as increased congestion on certain streets, creating a realistic feedback loop. Over time, agents develop context-aware strategies, such as selecting alternative routes during peak hours or adjusting speed to match traffic patterns, resulting in emergent traffic behavior that mirrors real-world conditions. [3]

3 Case Studies and Framework

3.1 Ayse Glass Example (Hamburg Pedestrian Study)

The study by Glass et al. (2023) [6] explored how reinforcement learning can improve the behavioral realism of ABMs in an urban mobility context. Their research focused on the HafenCity district of Hamburg, Germany—one of Europe's largest ongoing urban redevelopment projects, transforming former industrial docks into a mixed-use neighborhood.

The model incorporated ten representative categories of locations, including parks, offices, residential areas, and public spaces. Each location type was assigned a happiness score derived from a survey of 130 residents, reflecting their reported "*happiness in the different urban environments.*" These raw scores were transformed into a scaled probability P to serve as the reward function for the Q-learning agents: agents:

$$reward(r) = P(\text{happiness} \mid \text{go}) = k,$$

where "go" indicates that the agent chooses to travel to or visit that location type, and k is the happiness score of the chosen location. For example, the "healthcare" category received a score of 50 points, while "entertainment, art, and culture" received 450 points. Higher-valued locations thus offered stronger rewards, incentivizing agents to seek routes that led to more satisfying environments. At the beginning of the simulation, pedestrians moved randomly with no prior knowledge of which paths would yield higher cumulative happiness. As the simulation progressed, Q-learning updated each agent's Q-table via the Bellman equation, allowing agents to estimate not only the immediate reward of visiting a location but also the long-term expected reward of traversing an entire route. Over time, the agents shifted from short-term decision-making toward maximizing cumulative happiness, favoring sequences of consistently rewarding paths rather than simply pursuing a single high-value destination. [6]

Exploration and exploitation were balanced using an ϵ-greedy strategy (see Section 2.4.2). A high initial value of ϵ encouraged broad exploration, preventing agents from prematurely settling on suboptimal routes. As ϵ decayed over time, agents increasingly exploited the best-performing routes in their Q-tables. The decay schedule, along with the minimum and maximum values of ϵ, is illustrated in Figure 2. Despite relying only on a simple tabular implementation, the agents converged toward realistic walking patterns without requiring significant computational resources [6].

Glass et al. examined how several hyperparameters shaped agent behavior. The learning rate α governed how strongly new experiences updated Q-values: high values reacted too strongly to individual experiences, producing unstable behavior, while low values failed to adapt within the duration of the simulation. The discount factor γ controlled how much agents valued future happiness relative to immediate reward. Higher values of γ produced more goal-oriented, long-term planning behavior, whereas lower values led to agents focusing on short-term rewards and more random wandering. The number of training episodes controlled how much total experience each agent accumulated. Finally, the decay rate of the exploration parameter ϵ determined how quickly agents transitioned from exploration to exploitation. Figure 3 summarizes the relative importance of these hyperparameters, showing which parameters had the strongest influence on overall Q-values and cumulative rewards. The most effective pedestrian trajectories emerged from a combination of moderate learning rates, high discount factors, and gradual ϵ decay, ensuring both sufficient exploration and stable long-term behavior [6].

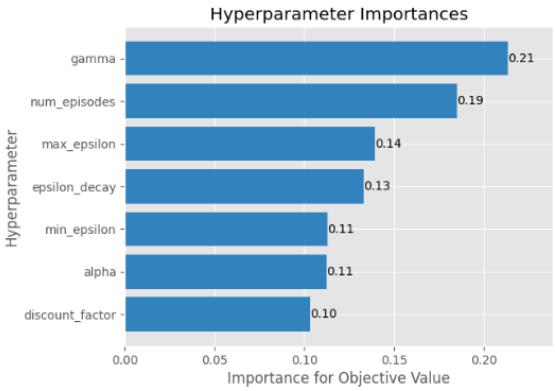


Figure 3: Importance ranking of Q-learning hyperparameters based on their influence on agent learning outcomes [6].

Although the model is limited by its fixed set of ten location types, a single pedestrian agent class, and an environment that does not respond dynamically to congestion or crowding, it demonstrates that even a simple reinforcement learning architecture can substantially enhance behavioral realism in pedestrian agent-based models. By grounding the reward structure in human survey data, the model produces emergent movement patterns that should more closely resemble real-world behavior, providing a clear proof of concept for integrating data-driven reinforcement learning into urban simulation frameworks.

3.2 Case Study: Traffic Dispersion from Bridge Construction (Sejong City)

While the Hamburg model illustrates how reinforcement learning can capture realistic pedestrian behavior at a micro scale, large-scale urban systems often require models capable of representing entire multimodal transportation networks. In such cases, reinforcement learning may be complemented or replaced by empirically calibrated decision rules. The Sejong City bridge study by Yun et al. (2022) exemplifies this broader application of agent-based models to infrastructure planning and traffic optimization.

Yun et al. (2022) [8] developed an agent-based traffic simulation for Sejong City, South Korea, to evaluate the effects of proposed bridge locations on city-wide traffic dispersion. The model used real datasets, including residence addresses, demographic microdata (MDIS), time-use surveys, GIS road networks, and public transit schedules, to construct a realistic synthetic population and urban infrastructure environment for evaluating infrastructure policies. To determine how agents moved through the city, the model used the A* (A-star) shortest-path algorithm [1]. The A* algorithm is a widely used pathfinding method that evaluates potential routes by combining the cumulative cost to reach a location with a heuristic estimate of the remaining distance to the destination, allowing agents to efficiently navigate the road network while minimizing computational effort. This approach provides stable and interpretable route predictions for infrastructure planning. The original study offers more implementation details, but these are beyond the scope of this paper.

Figure 4 shows the locations of existing bridges (red) and proposed candidate bridges (blue), providing a visual overview of the study's area of interest.

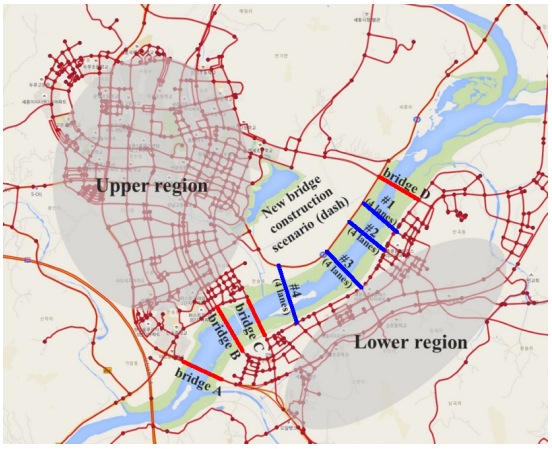
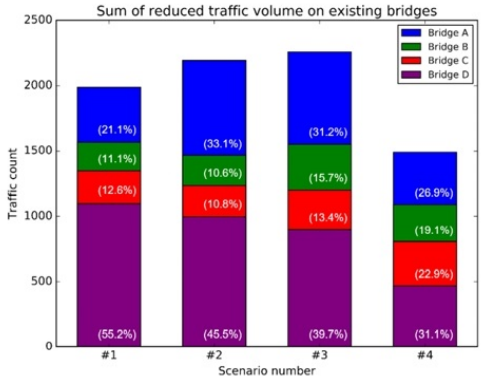


Figure 4: Existing and candidate bridge locations in Sejong City, South Korea. Adapted from [8].

The central experiment involved testing several candidate sites for a new bridge across the city. For each candidate bridge, the model measured three key outcomes:

1. Changes in traffic volume on existing bridges,
2. Flow and demand on the new bridge, and
3. Overall changes in traffic distribution across the network.

Scenario comparisons revealed which bridge location most effectively alleviated congestion while balancing infrastructure costs and urban access. Figure 5 shows the total reduction in traffic volume on existing bridges for each candidate site. The bars indicate the average reduction across runs, differentiated by which color for which bridge it is reducing traffic for. Bridge 3 produced the largest reduction overall, particularly in locations B and C, demonstrating that this site most effectively redistributed traffic away from congested areas.



(b) Contribution of existing bridges to total reduction

Figure 5: Total reduction in traffic on existing bridges for each candidate bridge location in Sejong City [8].

To ensure the model reflected reality, Yun et al. validated agent-generated traffic demand against observed data for Sejong City. Figure 6 compares simulated traffic volumes to real-world measurements, showing that the calibrated model reproduces traffic patterns with sufficient accuracy to support city planning decisions.

The simulation results clearly demonstrated that bridge location has a substantial impact on traffic dispersion. Some candidate sites produced a more balanced redistribution of traffic across the network, while others merely shifted congestion to alternate routes, highlighting how individual route choices respond to large-scale infrastructure changes. These findings underscore the limitations of aggregate traffic models, which can overlook important local effects.

The authors note several caveats. Their model assumes static agent decision rules, meaning agents do not learn or adapt via reinforcement learning, and it does not explicitly

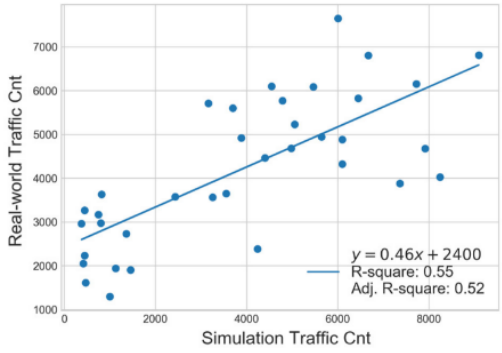


Figure 6: Comparison of real-world traffic and simulated traffic volumes [8].

simulate real-time traffic control such as traffic signals. Furthermore, while the microdata is detailed, time-use surveys and demographic distributions may not fully capture future behavioral changes. Despite these limitations, the study provides a compelling proof-of-concept: agent-based models can offer urban planners a bottom-up approach for evaluating infrastructure policies and anticipating nuanced impacts of new developments.

4 Results

Across both case studies, several broader insights emerge about the capabilities and limitations of agent-based models for evaluating urban infrastructure and movement patterns.

In the Hamburg pedestrian study [6], agents initially moved randomly but gradually learned routes that maximized their cumulative happiness. Q-learning updates caused their trajectories to resemble realistic walking behavior, showing that even a simple reinforcement learning mechanism can enhance ABM behavioral realism and allow agents to adapt dynamically.

The Sejong City bridge study [8] illustrates how individual-level decisions scale to city-wide outcomes. By simulating route choices of a synthetic population, the model showed that bridge effectiveness depends on how agents redistribute across the network, with some locations alleviating congestion and others merely shifting it elsewhere. These findings indicate that infrastructure evaluations are more reliable when they account for actual movement patterns.

Taken together, the studies highlight complementary strengths of different frameworks. The Hamburg model demonstrates adaptive, learning-based behavior, while Sejong relies on deterministic routing via A*, which reproduces mobility patterns precisely. This comparison underscores a trade-off: learning-based systems capture behavior evolution in new environments, whereas deterministic routing excels at reproducing current patterns.

Finally, both projects expose current limitations of ABM practice. The Sejong model lacks adaptive learning or real-time traffic controls, while the pedestrian model uses a simplified tabular Q-learning algorithm without heterogeneous agent types or dynamically changing environments. These constraints limit the generalizability of each system but also point toward productive directions for future research, particularly integrating reinforcement learning into larger, infrastructure-scale ABMs.

5 Conclusion

Agent-based modeling, when augmented with reinforcement learning techniques such as Q-learning, offers a powerful framework for understanding and improving urban systems. ABMs provide the dynamic environment in which agents operate, while Q-learning equips agents with the ability to adapt to feedback, resulting in context-dependent strategies that closely mimic real-world behaviors.

These findings suggest that future urban planning initiatives can benefit from the integration of ABMs and reinforcement learning. By simulating the interactions of autonomous agents within realistic environments, planners can test policy decisions, evaluate infrastructure projects, and anticipate emergent consequences, contributing to more efficient and adaptive urban design.

As computing power and data availability continue to grow, integrating reinforcement learning into city-scale ABMs could support more responsive digital counterparts to real cities. Such systems would allow planners to iteratively refine urban design based on simulated feedback, leading toward more efficient and resilient planning practices.

However, several limitations to ABMs remain. First, ABMs are inherently context-dependent: models calibrated for one city may not generalize easily to another without substantial adjustment. For example, a pedestrian model developed for Hamburg would not seamlessly transfer to New York City, as population density, urban environments, and behavioral patterns differ significantly. Second, the choice of metrics can influence conclusions; optimizing for "happiness" may not always align with efficiency or broader measures of urban performance, and different individuals may prioritize different outcomes. Finally, compared to real-world routing systems like Google Maps or Apple Maps, which provide drivers with dynamic directions and traffic-based redirections, ABMs lack access to real-time data and cannot replicate dynamic guidance.

Acknowledgments

I thank my advisors, Peter Dolan and Elena Machkasova, for their guidance and support.

References

[1] [n. d.]. A* Algorithm Explained. https://www.datacamp.com/tutorial/a-star-algorithm. Accessed: 2025-11-24.

[2] [n. d.]. Q-Learning in Reinforcement Learning - GeeksforGeeks — geeksforgeeks.org. https://www.geeksforgeeks.org/machine-learning/q-learning-in-python/. [Accessed 27-10-2025].

[3] R. Amor and T. Allard. 2019. Hierarchical Agent-Based Modeling for Improved Traffic Routing. *Applied Sciences* 9, 20 (2019). doi:10.3390/app9204376

[4] Wikipedia contributors. 2025. China National Highway 110 traffic jam Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=China_National_Highway_110_traffic_jam&oldid=1306150964 [Online; accessed 22-October-2025].

[5] GeeksforGeeks. 2025. *Reinforcement Learning*. https://www.geeksforgeeks.org/machine-learning/what-is-reinforcement-learning/ Last updated 07 Nov 2025.

[6] Ayse Glass, Jorg R. Noennig, Burak Bek, Roman Glass, Eylul K. Menges, Iryna Okhrin, Pramod Baddam, Mariela Rossana Sanchez, Gunalan Senthil, and René Jäkel. 2024. Innovative Urban Design Simulation: Utilizing Agent-Based Modelling through Reinforcement Learning. In *Proceedings of the 2023 6th International Conference on Computational Intelligence and Intelligent Systems* (Tokyo, Japan) *(CIIS '23)*. Association for Computing Machinery, New York, NY, USA, 20–25. doi:10.1145/3638209.3638213

[7] Rafael Prieto-Curiel and Juan P. Ospina. 2024. The ABC of mobility. *Environment International* 185 (2024), 108541. doi:10.1016/j.envint.2024.108541

[8] Tae-Sub Yun, Dongjun Kim, Il-Chul Moon, and Jang Won Bae. 2022. Agent-Based Model for Urban Administration: A Case Study of Bridge Construction and its Traffic Dispersion Effect. *Journal of Artificial Societies and Social Simulation* 25, 4 (2022), 5. doi:10.18564/jasss.4923