Impact of Collective Behaviors of Autonomous Vehicles on Urban Traffic Dynamics: A Multi-Agent Reinforcement Learning Approach

Ahmet Onur Akman*[1] onur.akman@uj.edu.pl
Anastasia Psarou[1] anastasia.psarou@uj.edu.pl
Zoltán György Varga[1] zoltan.varga@uj.edu.pl
Grzegorz Jamróz[2] grzegorz.jamroz@uj.edu.pl
Rafał Kucharski[2] rafal.kucharski@uj.edu.pl

[1] Doctoral School of Exact and Natural Sciences, Jagiellonian University, Kraków, Poland
[2] Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland

Abstract

This study examines the potential impact of reinforcement learning (RL)-enabled autonomous vehicles (AV) on urban traffic flow in a mixed traffic environment. We focus on a simplified day-to-day route choice problem in a multi-agent setting. We consider a city network where human drivers travel through their chosen routes to reach their destinations in minimum travel time. Then, we convert one-third of the population into AVs, which are RL agents employing Deep Q-learning algorithm. We define a set of optimization targets, or as we call them behaviors, namely selfish, collaborative, competitive, social, altruistic, and malicious. We impose a selected behavior on AVs through their rewards. We run our simulations using our in-house developed RL framework PARCOUR. Our simulations reveal that AVs optimize their travel times by up to 5%, with varying impacts on human drivers' travel times depending on the AV behavior. In all cases where AVs adopt a self-serving behavior, they achieve shorter travel times than human drivers. Our findings highlight the complexity differences in learning tasks of each target behavior. We demonstrate that the multi-agent RL setting is applicable for collective routing on traffic networks, though their impact on coexisting parties greatly varies with the behaviors adopted.

1 Introduction

With the recent advancements in autonomous driving technology, one can argue that we are moving towards a future where traffic systems are populated with self-driving autonomous vehicles (AV). Existing studies argue that the full integration will be costly and is still far down the road, but also suggest that AVs could significantly lower accident rates, increase accessibility, and improve overall traffic flow through optimized and coordinated route selection policies [9, 13]. The current feasibility and anticipated prospects of integration of AVs are strongly promoted by parties with financial interests [7, 6]. However, managing this integration requires rigorous analysis of the multifaceted outcomes of such change, involving identifying and quantifying its impact on various stakeholders.

One of the main affected parties in this transformation may be the current inhabitants of our traffic systems: human drivers. Apart from the aforementioned potential benefits, one of the reservations regarding autonomous driving technology is its potential to negatively impact the experience for

17th European Workshop on Reinforcement Learning (EWRL 2024).

human drivers, possibly exacerbating their travel times. Previous research has highlighted how this can occur naturally or consequently to a joint strategy [15, 12]. This research aims to contribute to these discussions by examining the interaction between human drivers and AVs within a simulated traffic environment. We demonstrate how different behavioral strategies of AVs can influence the traffic flow and how this can be simulated with multi-agent RL (MARL).

In mixed traffic environments, the adopted behavior of a group of AVs may influence not only the efficiency of the group itself but also the overall traffic efficiency. Studies have shown that AVs can be programmed with various behavioral strategies and how different behaviors can affect traffic flow and congestion [11, 3]. The interplay between these strategies and human driving behavior is an open field for investigation, as it determines to what extent AVs can enhance or disrupt existing traffic systems or compete with alternative modes to become the most attractive commuting option.

Among the various challenges presented by the integration of AVs, we address the problem of route choice in mixed traffic environments. The route choice behaviors of the drivers can significantly influence the traffic flow. Human drivers typically base their route choices on experience, sometimes paired with real-time information, aiming to minimize their travel time [2]. In contrast, AVs can employ more sophisticated learning algorithms and real-time data to make optimized route choices that consider both individual and collective benefit [18, 8]. This divergence in decision-making necessitates a detailed understanding of mixed route choice problems where humans and AVs coexist.

In this study, we represent the discrete route choice problem within an RL framework by conceptualizing it as a multi-agent decision-making process. We construct a traffic environment, and we support the dynamics of this environment with traffic simulation software to reflect close-to-reality transport dynamics. In this traffic environment, a group of driver agents, consisting of human drivers and AVs, select routes to minimize their delays. We simulate the mobility of this driver population at a peak hour on a random day. The traffic system initially consists of only human drivers, modeled as selfish self-utility maximizers with no external guidance. After a predefined number of episodes, a portion of the human drivers is replaced by a group of AVs, which are RL agents employing Deep Q-Networks (DQN) [16]. AVs learn a selected behavior imposed on them through their rewards. The reward formulation of an agent reflects the specific agent's behavior, such as minimizing one's own travel time or maximizing the delay for others. We investigate the outcomes in scenarios where AVs are imposed a selected behavior of each of six predefined behaviors: selfish, social, malicious, cooperative, competitive, and altruistic. We assess the attainability of these optimization targets for AVs in the given experimental setting and examine how they influence overall traffic efficiency.

In the following, we provide more in-depth specifications for our problem formulation. Next, we introduce PARCOUR, our reinforcement learning framework that facilitated our experiments. We then present our findings, highlighting the interesting trends observed in different scenarios. Our contributions can be summarized as follows:

1. Definition of a day-to-day route-choice problem in a traffic network and formulation of this problem as a MARL problem involving a varied set of agents.
2. Development of PARCOUR, our reinforcement learning tool designed for custom learning scenarios like the one presented in this paper.
3. Simulation of the proposed route-choice problem with different AV strategies and an analysis of our observations.

2 Methodology

2.1 Scenario

We design a simplified transport system involving route choice and congestion. We simulate traffic movement at a peak hour on an arbitrary day. We provide each driver with a set of paths to choose from and a utility to maximize. We observe their experiences over the days and how they learn to optimize their returns.

We create a population of 1200 human drivers intending to travel from their origins to their destinations through one of the three generated paths, by starting their commute on their individual start time. Each driver's start time is chosen randomly from the interval $(0, 3600)$, representing every second within an hour. The origin, destination, and start time information is initially assigned to each driver

at random and remains unchanged throughout the simulation. Each human driver aims to reach their destination with minimal travel time. We allow this population to interact with the environment, learn about the traffic dynamics, and refine their knowledge about their options for 1000 episodes. We name this time period *Phase Settle*.

In *Phase Shock*, starting in episode 1000, we replace approximately one-third of this population with AVs, modeling a sudden shift to autonomous driving in the city. We pause the learning process for humans and allow AVs to interact with this environment. By this, we aim to model the AVs' offline training procedure with a realistic environment model. Therefore, the remaining human drivers are still present in the environment with their frozen knowledge. We allow AVs to explore their options, and we once again allow the system to reach a loosely defined balance.

Finally in *Phase Adapt*, which is from episode 4000, we resume the learning process for humans, while AVs are still learning. This is the time that AVs refine their offline-learned knowledge in the actual city traffic. Therefore, we now have two sets of agents in a shared system, adapting to each other's existence. We continue simulating for 2000 more episodes. We observe the implications of this coexistence for these two groups of drivers.

Episodes In each episode, we simulate driver movements with start times spread across an hour. Drivers learn from their route choices and the delays they experience upon these choices. They take turns to select routes, with the order determined by their start times. Before making their decisions, AVs receive information about routes chosen previously by others, which helps construct their agent state. Whereas human drivers decide solely based on their cost expectations associated with their options. After all route choices are made, we simulate the travel using traffic simulator software and collect each driver's travel time. Agents then derive their rewards from this information according to their behavior definitions. For instance, every human driver is selfish; their rewards (to minimize) are their own travel times.

Difference between AVs and humans Humans are modeled using a behavioral model described in Section 2.3, while AVs employ Deep Q-Learning from the RL framework. We define a set of reward-induced behaviors for AVs: selfish, collaborative, competitive, malicious, altruistic, and social. In each simulation run, we choose a different behavior to learn, but under the condition that the entire group of AVs will bear the same behavior. However, each of the behaviors are individual optimization targets and they do not constitute a collective objective, as we assume no direct communication or shared knowledge amongst our agents, including the AVs.

2.2 Traffic Network

We use the traffic network of Csömör, a small town in Hungary, shown in Fig. 1 [5]. This network is complex enough to demonstrate our case, with a well-organized grid-like layout with frequent junctions, facilitating necessary grounds for strategy development for our agents.

We choose two origin and two destination nodes from this network and generate three paths connecting each of the four origin and destination combinations. We use a custom heuristic-based path-finding algorithm to create an action space that remains fixed across episodes for all agents. The resulting paths are not guaranteed to be optimal but sufficiently distinct, with a few intersections on critical nodes. We show all three paths generated to connect OD (0, 0) in Fig. 2.

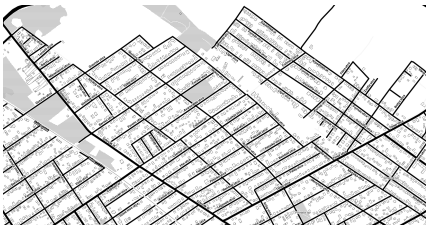

Figure 1: Csömör traffic network

Figure 2: Generated paths for OD (0, 0)

Congestion We sample the agent start times from a Gaussian distribution centered on the half-hour mark within our one-hour range. This way, we observe congestion build up and incentivize the use of routes with higher free-flow travel time. The congestion on a selected junction over different time steps within the same episode is displayed in Figure 3.

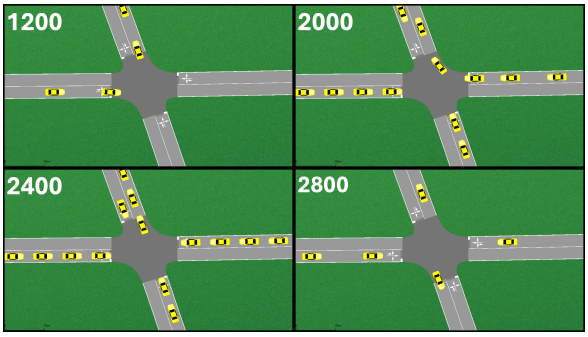

Figure 3: Congestion on a selected junction within a single episode on indicated time steps

2.3 Human Modeling

We use the behavioral model presented in the literature [4] for modeling our human agents' learning and decision-making procedure. This model is derived from random utility theory, a widely utilized theoretical paradigm for modeling transport-related choices among discrete alternatives. In each episode, the probability of an agent j to choose a route option i is:

$$p_j(i) = \frac{\exp\left(\beta_j \overline{c}_{i,j}\right)}{\sum_{i' \in I^j} \exp\left(\beta_j \overline{c}_{i',j}\right)}\,, \tag{1}$$

where $\overline{c}_{i,j}$ is agent j's cost expectation for option i. β_j represents the impact of human driver j's personal traits on decision-making, which is assigned to each human agent from a predetermined negative range at random. This approach reflects our realistic assumption that not every human driver will have the same level of uncertainty involved in their decision-making.

For each human agent j, we define a memory of cost expectations for each available action. Each value in this memory is iteratively updated with perceived feedback as follows:

$$\overline{c}_{i,j}^{t+1} = (1 - \alpha_j) \cdot \overline{c}_{i,j}^{t} + \alpha_j \cdot c_{i,j}^{t}\,, \tag{2}$$

where α_j is the fixed step size, $c_{i,j}^t$ is agent j's perception upon taking action i at time t, $\overline{c}_{i,j}^t$ and $\overline{c}_{i,j}^{t+1}$ are the agent j's expectations of costs associated with action i at time t and $t+1$, respectively. In our case, $c_{i,j}^t$ will be the time spent in route i from the start time until the arrival to the destination.

In contrast to the RL algorithms, this model does not aim to represent an optimal decision-maker but rather to portray rational decision-making with some impact of personal traits and the human-like process of iterative expectation refinement. This model allows us to frame human driver behavior in a traditional agent-environment interaction loop as we have it for the RL agents.

2.4 Reward function and AV behaviors

The reward function imposes a selected behavior on the agent. For an agent k with behavioral parameters $\varphi^k \in \mathbb{R}^4$, reward is obtained as:

$$r_k = \varphi_1^k \cdot T_{own}^k + \varphi_2^k \cdot T_{group}^k + \varphi_3^k \cdot T_{other}^k + \varphi_4^k \cdot T_{all}^k\,, \tag{3}$$

where T^k is a vector of travel time statistics provided to agent k, which contains:

- **Own travel time** T_{own}: The amount of time the agent has spent in traffic.
- **Group's travel time** T_{group}: The average travel time of agents within the same group as the given agent (e.g., AVs for an AV agent).

- **Other group's travel time** T_{other}: The average travel time of agents within other groups than the given agent (e.g., humans for an AV agent).
- **System-wide travel time** T_{all}: The average travel time of all the agents in the traffic.

We formulate each behavior with behavioral parameters listed in Table 1. We refer to behaviors with a positive φ_1 as *self-serving* behaviors. We care to maintain the same scale for the rewards across behaviors by choosing values with an absolute sum equal to 1. This allows us to observe comparable reward curves for different behaviors and maintain a consistent range of gradients in each experiment, providing comparable learning processes of deep networks.

Table 1: Behavioral strategies and their objective weightings

Behavior	φ_1	φ_2	φ_3	φ_4	**Interpretation**
Altruistic	0	0	0	1	Minimize delay for everyone
Collaborative	0.5	0.5	0	0	Minimize delay for oneself and one's own group
Competitive	2	0	-1	0	Minimize self-delay and maximise for other group
Malicious	0	0	-1	0	Maximise delay for other group
Selfish	1	0	0	0	Minimize delay for oneself
Social	0.5	0	0	0.5	Minimize delay for oneself and everyone

We assume that an agent's action can only impact a certain time window, and any events occurring outside of this window are not informative for the agent. Although the consequences of the agent's decision might extend beyond this time window, we assume these effects are negligible. To avoid distracting factors, we calculate T^k for an agent k with start time t_k by considering only the traffic flow observed in the time window $[t_k - L_r, t_k + L_r]$, where L_r is a hyperparameter.

2.5 Observations and agent states

We provide each agent with a partial statistic about the current traffic condition in the form of observations, and we derive it from other vehicles' decisions, leveraging the turn-based structure of our episodes, as detailed in Section 2.1. The environment emits an observation o_t during each turn, which includes the selected actions in the previous turns within the same episode. We assume that an agent k with start time t_k can only observe a limited time window preceding t_k. Therefore, we constrain the observations with an observation time window of a length L_o, where L_o is a hyperparameter. We also limit the observations to include information only about those with the same OD pair as the observing agent.

The observation o_t is used to derive the agent state $s^a_{k,t}$ for agent k. The agent state s^a_t embeds useful statistics about the current traffic conditions, leveraging the notion of action *warmth* to maintain the temporal order of previous actions.

Upon receiving the route choices of prior agents, agent k constructs its agent state $s^a_{k,t}$ as a vector of size $2 * |A|$. $|A| = 3$ is the number of route options for an agent and the same for all agents in our experiments. The first half of the values are filled with the *warmth* of each action, but only considering prior agents from the same group. The second half is filled in the same way, but by considering the prior agents from other groups. The *warmth* of an action j concerning the other group for an AV agent k with start time t_k is obtained as:

$$\textit{warmth}_j^{\text{other}} = \sum_{i \in N_{\text{other}, w_{t_k}}} (t_i - \min w_{t_k}) \cdot \mathbb{1}_{\{\text{OD}_i = \text{OD}_k\}} \cdot \mathbb{1}_{\{\text{action}_i = j\}} \,, \tag{4}$$

where

- w_{t_k} is the observation time window of length L_o defined for start time t_k. It contains all time steps ranging in $[t_k - L_o, t_k]$.
- $N_{\text{other}, w_{t_k}}$ is the set of prior non-AV agents whose start times fall within the observation window w_{t_k}.
- OD_i and OD_k are the OD pairs of agents i and k, respectively.

- t_i and action_i are the start time and chosen action of agent i, respectively.
- $\mathbb{1}_{\{\text{OD}_i=\text{OD}_k\}}$ is an indicator function that is 1 if the OD pair of agent i is the same as agent k's, and 0 otherwise.
- $\mathbb{1}_{\{\text{action}_i=j\}}$ is an indicator function that is 1 if the action of the agent i is j, and 0 otherwise.

Following this, an agent state $s^a_{k,t}$ of AV agent k is constructed as:

$$s^a_{k,t} = \begin{pmatrix} \mathit{warmth}^{\text{AV}} \\ \mathit{warmth}^{\text{other}} \end{pmatrix}, \tag{5}$$

where $\mathit{warmth}^{\text{AV}} = \begin{pmatrix} \mathit{warmth}^{\text{AV}}_1 \\ \mathit{warmth}^{\text{AV}}_2 \\ \cdots \\ \mathit{warmth}^{\text{AV}}_{|A|} \end{pmatrix}$ and $\mathit{warmth}^{\text{other}} = \begin{pmatrix} \mathit{warmth}^{\text{other}}_1 \\ \mathit{warmth}^{\text{other}}_2 \\ \cdots \\ \mathit{warmth}^{\text{other}}_{|A|} \end{pmatrix}$. In our presented scenario, for an AV agent, the set of other groups contains only the group of human drivers and vice versa.

We should highlight that start times and OD pairs are intrinsically contained in the agent state. However, as these factors remain unmodified throughout the simulation runtime, we choose not to explicitly represent them. In other cases, one should modify the agent state accordingly.

2.6 PARCOUR

We conduct our experiments using our very own multi-agent reinforcement learning framework: PARCOUR. PARCOUR (Playground for Agents with Rationality Competing for Optimal Urban Routing) is designed to allow researchers to define and test different behavior and learning models in custom multi-agent route-choice scenarios and observe agent interactions in a shared environment.

In PARCOUR, we ensure modularity by modeling each component with separate lifecycles and minimal dependencies. PARCOUR exemplifies the principle of separation of concerns (SoC), as each component has distinct responsibilities. Moreover, it relies solely on a small set of core libraries like PyTorch [17] and pandas [14] to provide broader compatibility. A Unified Modeling Language (UML) class diagram illustrating the structure of PARCOUR is provided in Appendix A.

PARCOUR is for discrete route choice problems, like the one we present in this study. A rigorous investigation of this type of problem requires an accurate modeling of the traffic flow. We capture the realistic transport dynamics by enabling the integration with an external traffic simulator. In our experiments, we choose to use Simulation of Urban MObility (SUMO) [10], an open-source, microscopic, and continuous traffic simulation software [1]. However, users can integrate any external traffic simulator with a Python interface by extending the provided `BaseSimulator` class.

The `ScenarioRunner` object serves as the central orchestrator, connecting and managing the interactions between various components. Users can model any scenario using their custom `ScenarioRunner` class. A scenario in PARCOUR is constructed as a series of *phases*. The events to occur in each phase, the number of phases, and their spans are configurable by the user.

The framework allows users to integrate an external RL library, or implement their own learning algorithms, just like the ones we utilize in this study. PARCOUR is still in early development and it is available as open-source software.[1]

3 Results

We implement our scenario (Sec. 2.1) in Csömör network (Sec. 2.2) using PARCOUR, integrated with SUMO (Sec. 2.6). We run experiments with six reward-induced AV behaviors (Sec. 2.4) in identical settings. Each case was repeated three times, and the results were averaged to provide a clearer representation of the repeated trends and to capture the variability across trials. This section presents our findings and highlights the contrast between the impacts of different AV behaviors.

[1] Available at: https://github.com/COeXISTENCE-PROJECT/parcour

Demand and Action Space Our action space comprises 12 paths connecting every four OD pairs. Each route's free flow travel times are given in Table 2. Free flow travel time is the time it takes for a vehicle to complete its journey under no traffic congestion. The provided values are calculated by SUMO.

Table 2: Free flow travel time (in minutes) for different routes

Origin	Destination	Route 0	Route 1	Route 2
0	0	2.08	1.39	3.22
0	1	0.78	2.95	2.36
1	0	0.69	0.92	1.88
1	1	1.91	3.02	1.99

Each OD pair is associated with a demand load, measured by the number of drivers intending to travel between them at any timestep of the simulation. While we aim to maintain similar loads for each pair, perfect equality in demand is hardly the case in real-world traffic and is also not reflected in our generated driver population. Demand levels for each OD pair and AV to human driver ratios are shown in Figure 4. Note that up to Phase Shock, all demand is created by human drivers and the levels correspond to green and red bars combined.

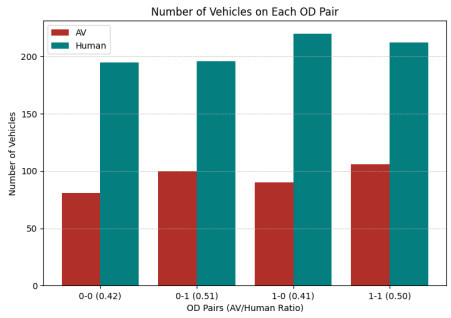

Figure 4: Number of drivers traveling on each OD

Comparing behaviors We test six cases, in each imposing a different behavior on the AV fleet. We compare the mean travel times of humans and AVs in each experiment, with human driver data (origins, destinations, start times, if they will mutate to an AV) remaining consistent. This consistency yields similar curves in Phase Settle for humans. The mean travel times of AVs and humans over the episodes for each scenario are visualized in Figure 5.

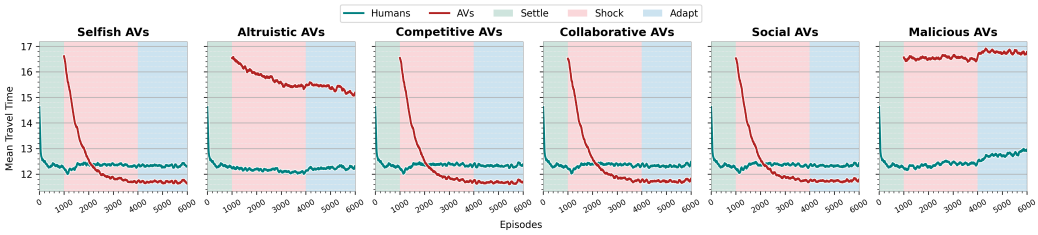

Figure 5: For each case, mean travel times of humans and AVs over episodes

We show the consequences of different AV behaviors on their travel time and other drivers in Table 3, where we compare the mean travel times averaged on the last 100 episodes of Phase Settle and the last 100 episodes of Phase Adapt. Traffic efficiency measures the change in travel time of the entire driver population in the traffic system.

These results demonstrate the contrast in the impact of different AV behaviors on the traffic flow. The negative impact on the human commuting experience is relatively small but existent in competitive, collaborative, selfish, and social cases. All self-serving behaviors yield lower delays for AVs and increase overall traffic efficiency. In the malicious case, AVs establish a noticeable disadvantage for human drivers, at the cost of significantly more delays for themselves. Interestingly, malicious AVs manage to assert their malicious strategy more effectively once humans are allowed to react to their existence. In the altruistic case, a minor enhancement in human driving experience costs 23.06% more delays for AVs. In our experimental setting, despite our definition of the altruistic objective, altruistic AVs not only fail to enhance traffic efficiency but worsen it.

Table 3: Impact of different AV behaviors. Positive impact indicates lowered travel time.

Behavior	On AV T.T.(%)	On Human T.T. (%)	On Traffic Efficiency (%)
Altruistic	-23.06	0.27	-7.06
Collaborative	4.28	-0.71	0.86
Competitive	4.59	-0.75	0.93
Malicious	-36.29	-5.43	-15.12
Selfish	4.57	-0.70	0.96
Social	4.21	-0.66	0.87

Learning efficiency of DQNs We show the learning processes of DQNs of AVs in each case in Figure 6 using Mean Squared Error (MSE) loss. Our results show a generally downward trend in losses in all cases, highlighting an effective learning procedure in our simulations. Surprisingly, we observe that Phase Adapt is marked by further loss reduction and stabilization in most cases. However, depending on the AV objective, achieved losses and fluctuations vary greatly.

Altruistic, collaborative, and social AVs exhibit a more consistent and effective learning process, converging steadily to a below-one MSE loss. In the malicious case, we see an early convergence to a near-one MSE loss from early on, but apart from a minor fluctuation at Phase Adapt, this level of loss is more or less preserved until the end of the simulation. Selfish AVs struggle relatively more to converge to an acceptable loss value, and we see that the loss curve is noisier, ending up with a below-one MSE loss. In contrast, competitive AVs exhibit a less stable learning curve, reflecting the increased complexity in the task of learning such a strategy in our multi-agent setting.

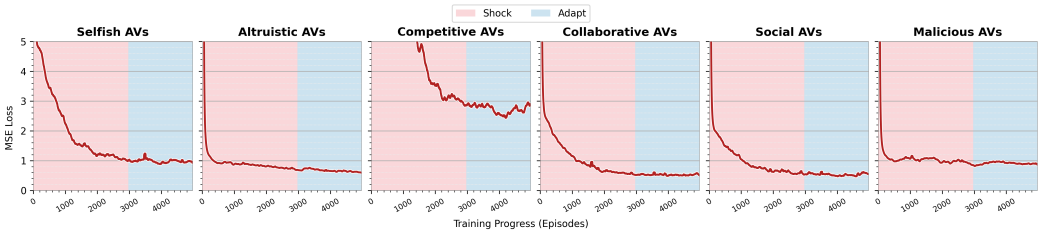

Figure 6: Average MSE loss of deep-RL agents over episodes

To emphasize learning stability with reward acquisition, we show the reward curves for each case in Figure 7. In the altruistic case, despite the stable loss curve, there is the least steady reward minimization over the episodes. This was anyway evident in Table 3. This hints at the difficulty of implementing such behavior in our experimental setting. We observe a similar situation in the malicious case, although Phase Adapt greatly aids the reward minimization (as also apparent in Figure 5). Other cases exhibit satisfactory reward minimization over the episodes. Interestingly, despite the unsteady loss curve, the competitive case exhibits the steepest improvement over the episodes, highlighting the contrast in difficulties in learning and implementing this behavior.

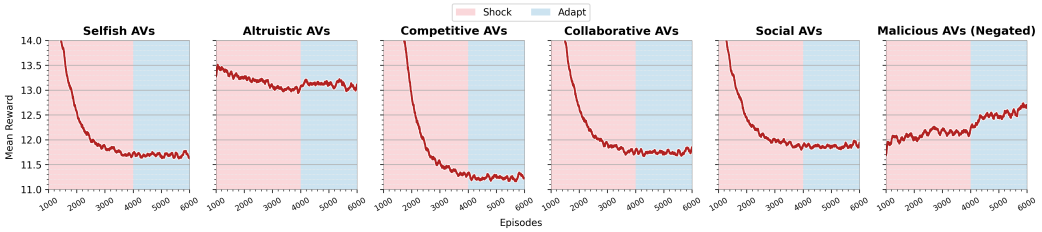

Figure 7: Average rewards of deep-RL agents over episodes observed in each experiment. Malicious rewards are negated for visualization.

Impact on human experience Thus far we considered the changes in traffic dynamics as a whole and disregarded the differences between different parts of the traffic. However, to accurately assess the

implementation and impact of each autonomous vehicle (AV) behavior, a more detailed, fine-grained analysis is required. This necessitates examining the differences across various subnetworks within the traffic system. Each set of routes connecting individual OD pairs constitutes a distinct subnetwork, and these subnetworks converge at key intersections. These subnetworks vary in terms of the number of nodes, path lengths, and the complexity of interactions at shared intersections, which are governed by priority-based turning rules. Therefore we expect to see variations in how AVs affect the driving experience in different parts of the traffic.

Firstly, we focus on how human driver travel times change over the episodes based on their origin and the destination they are traveling to. The shifts in human travel times for each OD pair are visualized in Figure 8. It is noticeable that the impact of AV behavior greatly varies across OD pairs. It is visible in all cases, and quite obvious in the malicious case, that human drivers in some subnetworks get affected negatively, while others either do not get affected or even experience a positive change in their travel times. We also notice that human travel times are more unpredictable in some ODs than others, indicated by wide error bars for ODs (0, 0) and (0, 1), while others are relatively consistent.

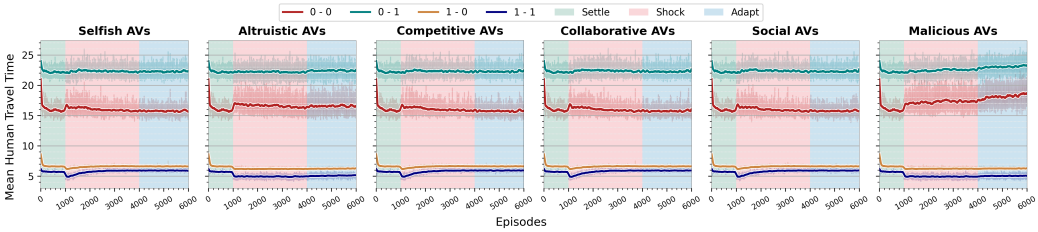

Figure 8: Change in human travel times for each OD pair. Error bars show the standard deviation across experiment runs.

Next, we have a closer look at the changes that occurred in OD (0, 0) (which was displayed in Figure 2). In Figure 9, we show the portion of each driver group choosing one of the three routes connecting OD (0, 0) over the episodes. This figure nicely visualizes the way AVs implement their strategies, and how humans adapt to them. The first thing we notice is that in all cases where AVs adopt a self-serving behavior, they optimize their travel time by simply populating the most favorable routes. We notice a slight increase in the number of human drivers in route 2, which is the least viable option. We interestingly observe quite similar changes in the cases of self-serving AVs and social AVs. In the altruistic case, we see that the AV domination in Route 0 is not as drastic, and AVs also populate less favorable routes as well. This makes Route 1 more attractive than how it was, and human drivers partially shift their preference from Routes 0 and 2 to Route 1.

In the malicious case, AVs populate each route in a more balanced manner. After the Phase Adapt initiates, they favor Route 1 more, but the route preferences are still not as contrasted as in other cases. Interestingly, malicious AVs prefer Route 2 more than AVs in other cases, which is the longest path (as shown in Table 2), but we understand that this is for a good reason. Figure 2 shows the critical points where three routes intersect. In a few spots, particularly at the intersection leading to the destination link, Route 2 has the priority. This may be consequential in congestion, like in the case at timestep 2400 in Figure 3, as this may form long queues on the lanes with low priority. Given that malicious AVs focus solely on maximizing human travel times rather than minimizing their own delays, it becomes clear why selecting Route 2 can be a viable option for their strategy.

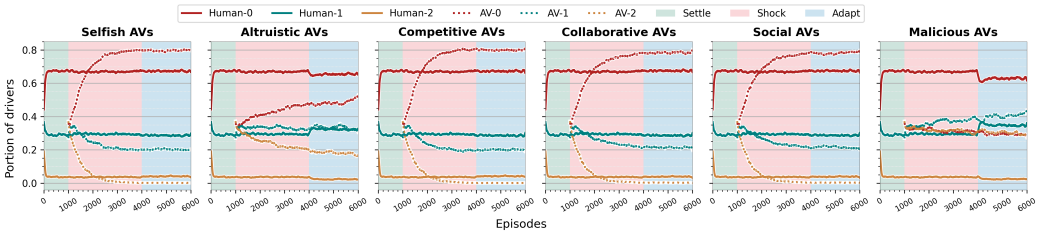

Figure 9: Portions of drivers choosing each route connecting origin 0 to destination 0

To understand how these preference changes affect travel times, we reveal the human travel time distributions in OD (0, 0) in Figure 10. We observe noticeable changes in the spread of travel times, which indicates how AVs implement their strategies in Phase Shock and how humans react to these changes in Phase Adapt. For instance, we see that all self-serving behaviors cause tighter distributions of human travel times at the end of Phase Shock. Generally, we notice an increased spread for the cases where humans shift to Route 2, which is the least favorable route, and denser distributions for the cases where humans shift to Route 1, which is the middle ground in terms of attractiveness.

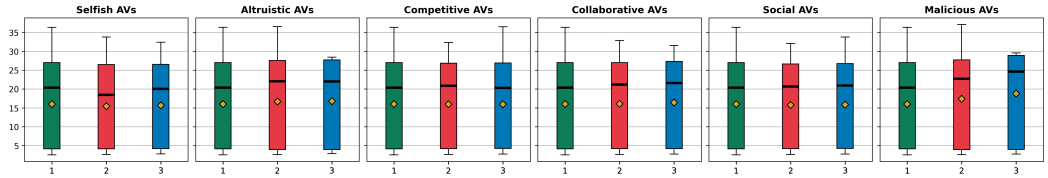

Figure 10: Distribution of travel times of human drivers traveling on OD (0, 0) in the final episode of (1) Phase Settle, (2) Phase Shock, (3) Phase Adapt. Black lines show the median and the golden markers show the mean points.

4 Conclusions

We have conducted an experimental investigation of the potential implications of a shift to autonomous driving in transport systems. We defined a multi-agent route choice problem in a traffic network involving a driver population made up of humans and RL-enabled AVs. We established distinct learning and decision-making methods for both kinds, which enabled us to simulate their experiences in a unified RL agent-environment interaction loop.

We introduced PARCOUR, our RL framework for route choice problems, which facilitated our experiments. PARCOUR enables the simulation of traffic flow in realistic transport dynamics by enabling integration with an external traffic simulator, which, in our case, was SUMO.

We defined six target behaviors for AVs, each with a distinct impact on traffic flow. We formed a unified parameterization for formulating these behaviors through reward functions. We experimented with six scenarios where, in each case, the group of AVs adopted one of the defined behaviors. We analyzed the attainability of each of these behaviors in a MARL setting. We provided a clear overview of our findings using different data visualization techniques paired with our interpretations.

Our results indicate that different reward definitions for AVs imply contrasted consequences for the other drivers in the traffic, as well as the users of the autonomous driving technology. We observed that each behavior can influence the overall traffic flow with different rates, while this is generally a positive influence for the self-serving AVs. We investigated the way AVs enforce their strategies, and we showed that the implications of their behaviors for different parts of the network vary greatly. We analyzed the preference changes of humans in adapting to the existence of AVs and showed that a portion of the human driver population needed to give up on their preferred routes.

The multi-agent route choice problem concerning heterogeneous driver populations remains an open field of research. Drawing conclusions concerning real-world applications requires analyzing realistically scaled problems. With this study, we have provided a solid foundation for prospective studies, which will address more complex problem definitions, the use of more sophisticated learning models, the integration of a centralized control mechanism for AVs, and what other implications this may have on the human driving experience.

Author contributions AOA - conceptualization, methodology, development, experimentation, data analysis, visualization, writing, review, editing; AP - methodology, development; ZGV - methodology, development; GJ - conceptualization, methodology, review; RK - conceptualization, methodology, review, project administration, funding acquisition.

Acknowledgement This research is financed by the European Union within the Horizon Europe Framework Programme, ERC Starting Grant number 101075838: COeXISTENCE.

References

[1] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.

[2] Moshe E Ben-Akiva and Steven R Lerman. *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT press, 1985.

[3] Erdem Bıyık, Daniel A Lazar, Ramtin Pedarsani, and Dorsa Sadigh. Altruistic autonomy: Beating congestion on shared roads. In *Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics 13*, pages 887–904. Springer, 2020.

[4] Ennio Cascetta. *Transportation systems analysis: models and applications*, volume 29. Springer Science & Business Media, 2009.

[5] Stamen Design. Map tiles by stamen design, 2024. Under CC BY 4.0. Data by OpenStreetMap, under ODbL. Accessed: 2024-05.

[6] Centre for Connected and Autonomous Vehicles. Connected and automated mobility 2025: Realising the benefits of self-driving vehicles. Technical report, UK Government, 2022.

[7] Charlie Johnson and Jonathan Walker. Peak car ownership: the market opportunity of electric automated mobility services. *Rocky Mountain Institute and Mobility Transformation*, 2016.

[8] Xiaohui Li, Zhenping Sun, Dongpu Cao, Zhen He, and Qi Zhu. Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications. *IEEE/ASME Transactions on mechatronics*, 21(2):740–753, 2015.

[9] Todd Litman. Autonomous vehicle implementation predictions: Implications for transport planning. Technical report, Victoria Transport Policy Institute, 2020.

[10] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[11] Michal Maciejewski and Joschka Bischoff. Large-scale microscopic simulation of taxi services. *Procedia Computer Science*, 52:358–364, 2015.

[12] Michal Maciejewski and Joschka Bischoff. Congestion effects of autonomous taxi fleets. *Transport*, 33(4):971–980, 2018.

[13] Margarita Martínez-Díaz and Francesc Soriguera. Autonomous vehicles: theoretical and practical challenges. *Transportation research procedia*, 33:275–282, 2018.

[14] Wes McKinney. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

[15] David Metz. Developing policy for urban autonomous vehicles: Impact on congestion. *Urban Science*, 2(2):33, 2018.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[18] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:187–210, 2018.

EWRL Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We define a multi-agent route choice scenario (see Section 2.1) that provides a ground for making observations regarding the scope we form in the abstract and introduction. Then, we formulate our scenario as a decision-making process and implement it in the MARL setting. In Section 3, we make observations aiming to contribute to the discussions we refer to in Section 1. Our findings match with what we present in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Section 2, we describe and visualize precisely the complexity of the used traffic network. In Section 3, we interpret our results by pointing out that they are valid for our experimental setting and do not imply that they may generalize to other settings. In Section 4, we point out the need for more realistically scaled problem definitions for making observations applicable to real-world applications.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We do not present any theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the hyperparameters and experimental setting details in the Appendix, in Section B. We conduct our experiments with the mentioned settings using our framework PARCOUR, which is now available as open-source software, and which we share the UML class diagram of in the Appendix in Section A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While EWRL does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We conduct our experiments with the mentioned settings using our framework PARCOUR, which is now available as open-source software.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the EWRL code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the EWRL code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We share experimental setting specifications in detail in the Appendix in Section B, and we describe all new concepts we introduce in Section 2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Our paper reports variability in data by reporting distributions and variance of travel times (Section 3). We use various visualization techniques to provide insights into the underlying patterns of our high-level observations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide the computational resources we used and the runtime of our experiments in the Appendix, in Section C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the EWRL Code of Ethics `https://ewrl.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We do our best forth to fully comply with the EWRL Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the EWRL Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We show in our experimental setting what would be the implications of adapting autonomous driving with adversarial or social strategies (as we define them) in transport systems, and we share our results revealing the effect on human commuting experience. We note that our problem definition is not extensive enough to generalize our findings to rigid societal conclusions. We believe that extrapolating our observations, given our constraints, would be misleading for an uninformed reader. We aim our findings to present a foundation for prospective research that investigates the impact of autonomous driving on a more realistic scale.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The novel work is a product of the named authors only. Used assets are appropriately referred to by citing the related references.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: Even though it was sufficient to facilitate our experiments, our framework (Sec. 2.6) is still in the stage of early development. We are actively enhancing its functionality and documentation.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the EWRL Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the EWRL Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

A UML class diagram of PARCOUR

Figure 11 displays a UML class diagram illustrating the structure of PARCOUR. This diagram presents only the classes we used in our experiments.

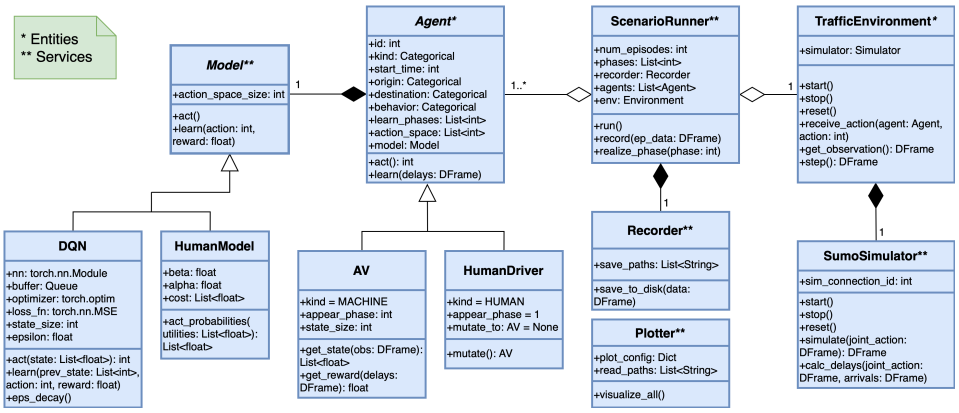

Figure 11: UML Class Diagram of PARCOUR

B Reproducibility

The results we present in this paper are obtained through experiments with 1200 driver agents and 6000 episodes. Generated driver population data is provided in the same repository as PARCOUR. We construct the scenario we describe in Section 2.1 with 3 phases, namely: Settle, Shock, and Adapt. These phases start in episodes 1, 1000, and 4000, respectively. We conducted each experiment three times, and the results presented are the averages of these repetitions.

Phase Settle is where all driver agents are humans, implemented using the model we describe in Section 2.3. Phase Shock is where 377 randomly selected human drivers are replaced by AVs. AVs are RL agents utilizing the (Single) Deep Q-learning algorithm. We turn off the learning for humans in Phase Shock. Phase Adapt is when we enable learning for everyone and let both parties adjust to each other's existence.

Start time and OD assignments to agents are done randomly. OD assignment distributions are provided in Figure 4. Start times range in a period of one hour, with seconds precision. We sample start times from a Gaussian distribution as justified in Section 2.2.

AVs all have their individual Q-Networks, trained solely using their own experiences. Each network has an identical architecture, implemented using `nn.Module` from PyTorch. Each network consists of an input layer, two hidden layers, and an output layer. It processes the state input through the input layer to 32 units, followed by a hidden layer to 64 units and another hidden layer to 32 units before mapping values to the action through the output layer. ReLU activation is applied after each layer except the output. The learning rate is the same for all agents, fixed to 0.003. We use MSE loss and Adam optimizer, which are implemented in `PyTorch`. Each agent has a replay buffer of size 256 and a batch size 32. The exploration is ensured by an epsilon-greedy policy with ε set to 0.99 and a decay rate of 0.998, applied in every episode.

The observation window size L_o (Sec. 2.5) is set to 300, which corresponds to a 5 minutes range. The reward window size L_r (Sec. 2.4) is set equal to L_o.

Humans have their individual cost tables (Sec. 2.3), which we initially populated with each route option's free flow travel times. They share the same α value, set to 0.2. They have randomized β, as reasoned in Section 2.3, randomly determined from the range $[-0.8, -0.2]$.

The network we used is included within the same repository as PARCOUR. We use nodes from our traffic network `279952229#0` and `115604053` as origins, and `-115602933#2` and `-441496282#1` as destinations (Sec. 2.2). We generate the action space using our path generation algorithm with logit β set to -0.1, creating three routes linking every origin-destination combination. Generated paths and the path generation code are also included in our repository.

PARCOUR has an integrated plotting functionality, which automatically saves useful plots to the disk once an experiment concludes. However, the plots we present in this work are produced using a different code from the data generated in our experiments. All line plots in Section 3 are smoothed by 50 steps using `uniform_filter1d` from `scipy`.

C Computing Resources

The experiments (Sec. 3) were conducted on our faculty's high-performance computing cluster. The resources allocated for these experiments included a single GPU (NVIDIA Tesla V100), 64 GB of RAM, and 4 CPU cores per task. The total execution time for each experiment was around 18 hours.