Using autonomous vehicles to improve traffic conditions

Sofia Afonso Campos de Carvalho
sofia.c.carvalho@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

***Abstract*—This work focuses on demonstrating the possible advantages of designing autonomous vehicles (AV) to improve traffic conditions. Given the constant improvement of the capabilities of autonomous vehicles and the lack of effective methods to counter traffic congestion, AVs have become a potential asset for finely managing traffic. Our work considers the traffic-critical setting of an open multi-lane highway with a merging single road, and presents a deep reinforcement learning strategy that trains the system of AVs on that road to minimize the congestion generated by the merge. This solution is tested in a traffic micro-simulator and proven to effectively improve the network's outflow and the vehicles' average speed at a 10% AV penetration rate. Replays of the simulations can be seen in SimulationReplays[1].**

***Keywords*—autonomous vehicles; traffic congestion; deep reinforcement learning**

I. Introduction

Traffic congestion is a serious problem in big cities around the world. The current methods to counter this problem (for instance, constructing better roads) are often costly and time-consuming, proving overall not appropriate for solving a worldwide problem. Therefore, there is an urgent need for alternative ways of countering traffic congestion.

On another note, autonomous vehicles (AVs) have been turning into valuable assets. The amount of AVs on the road has been quickly growing, and their capabilities have been continuously improving. AVs are currently simply designed as a means of transport, with the goal of getting the owner from one place to another. However, this approach might not be considering their full potential—in effect, people are sharing the roads with robots and could use them, for example, to benefit traffic conditions.

That said, this work is based on the premise that, with the right knowledge about the current state of traffic, there are optimal driving behaviors that can be followed to avoid traffic congestion. However, human drivers do not usually have access to this required information, and even if they did, they may favor their own goal (quickly reaching their destination) over improving traffic. AVs, on the other hand, can simply be programmed to follow the optimal behavior when in situations that commonly lead to congestion (like merges or intersections), and can easily assess the state of traffic by communicating with other AVs on the road. Additionally, the communication with other AVs allows them to create new traffic-beneficial strategies that require the cooperation of multiple vehicles.

It follows naturally that we could exploit AVs to improve traffic. To this end, we first need to identify which situations commonly lead to congestion, to then document which practices can be beneficial in those situations, and finally develop the AV control algorithms that implement those behaviors.

After researching some works on this topic, we found that the deep reinforcement learning (deep RL) approaches, that trained the AVs to learn an optimal behavior, were often very successful. We also noticed that there were no presented solutions for improving traffic on the critical scenario of an on-merge section in an open road with multiple lanes. Therefore, our work focuses on developing a deep RL-based AV control strategy that prevents traffic congestion in this scenario. Additionally, note that, since one AV cannot impact the traffic of an entire road, our strategy inevitably involves coordinating with other AVs and influencing the human-driven vehicles so that they all work towards the same goal.

To summarize, in this work we address the following research question: how can we design autonomous vehicles to improve traffic conditions?

We highlight the contributions of our project:

- The documentation of existing autonomous vehicle control algorithms that prevent traffic congestion;
- The development of an autonomous vehicle control strategy that improves traffic conditions on an open multi-lane highway with a single-lane on-merge section;
- The implementation of this solution in a mixed-autonomy traffic simulation, and the evaluation of its effectiveness.

The remainder of this document is organized as follows: In section II, we clarify some concepts that are used in this project and describe the works in the literature that inspired our solution. We then describe our envisioned strategy and detail the performed experiments in section III. In section IV, we present the obtained results for these experiments and comment on the effectiveness of our solution. Finally, in section V we reflect on the overall contributions and limitations of our project.

II. Background

In this section, we clarify some less trivial concepts that are used throughout this document and highlight some works

[1] https://drive.google.com/drive/folders/1sa5y2FAWpInOHvYFTHaO56dgzNgIEE-j?usp=sharing

developed in the scope of using autonomous vehicles to improve traffic conditions.

In the extended thesis document, we carry out an extensive discussion on the various solutions that have been developed on this topic, comparing solutions for multiple different scenarios and with different specific goals. Nevertheless, in this document, we limit this discussion to including solely the two works that proved especially relevant to the development of our solution.

A. Deep Reinforcement Learning and MDPs

Reinforcement Learning is one of the 4 basic categories of Machine Learning. *Machine Learning (ML)* is a type of artificial intelligence where a system uses previous data to "learn" and more accurately predict new outcomes, without being explicitly programmed to do so. *Reinforcement Learning (RL)* distinguishes itself by working through trial and error—an agent following an RL algorithm has a defined goal and a set of actions that it can perform to achieve the goal, and is rewarded or punished for performing an action that is, respectively, beneficial or prejudicial towards the goal. The agent then uses this feedback to update an internal policy. In the context of ML, a *policy* is a probability distribution over the possible actions (depending on the state), ultimately guiding how the agent should act. Therefore, an agent will attempt to define an optimal policy that maximizes its expected reward. On another note, *Deep Learning* is a family of ML strategies that use deep artificial neural networks (neural networks with multiple layers) to repeatedly process the input data into defining features. *Deep RL* is thus a family of techniques that combine deep learning with reinforcement learning, using neural networks to process the gathered information into actions, and ultimately representing the RL-learned behavior as neural networks.

There are two main types of RL algorithms: model-based and model-free. In model-based RL, the agent constructs an internal model of the problem given its experience to the moment, and then constructs its policy according to this model. In model-free RL, the agent uses its experience to directly learn its policy (or an action-value function, which informs how good each action is at a given state), without the use of a world model. A *Markov Decision Process (MDP)* is a mathematical framework often used in model-based RL problems, defined through a set of elements: the states an agent can be in; the actions it can take; for each action, the probability of transitioning from one state to another (called probability functions); the reward function for the agent's actions. Following this model, the agent's policy is first initialized to some value. Then, at each timestep, the agent is at some state, chooses an action according to its current policy, transitions to another state according to the probability functions, receives a reward, and updates its policy function according to what it did and the reward it received from doing so. This way, the agent's policy is eventually able to accurately guide it to the goal. A *Partially Observable Markov Decision Process (POMDP)* is a commonly used variation of an MDP that extends it by including information on what an agent senses at each time step (the observations), assuming that it is unable to know its own exact state. In a POMDP, the policy is calculated according to what the agent can observe, instead of the actual states it is in.

That said, a policy is updated according to a policy optimization algorithm, that strives to maximize the expected reward. Following the *Trust-Region Policy Optimization (TRPO)* algorithm, each policy update is bounded by a defined maximum difference between the new and old policies, expressed in terms of KL-divergence. The update thus corresponds to the largest possible improvement of the system's performance that satisfies this constraint. The *Proximal Policy Optimization (PPO)* algorithm is a simplification of the TRPO algorithm, using essentially first-order methods to keep the new policy close to the old, while the TRPO algorithm requires complex second-order methods.

In the scope of this work, a *centralized policy* is one that is calculated by a centralized controller after joining the information sensed by the set of agents that are within its reach. In contrast, a *distributed policy* is calculated by each agent, using only the knowledge obtained by itself.

B. Maximizing road utility

Given that our solution focuses on the context of multi-lane roads, we describe the work [3], which explores the idea of balancing the lane usage between the vehicles on a road to maximize the roads' capacity and utility. The goal of this work is to develop a centralized AV controller that, after finding the maximum affordable increase to a road's capacity, calculates the optimal vehicle configuration, and rearranges the vehicles according to this configuration.

To calculate the road's capacity (the number of vehicles that can simultaneously travel on the road) and the optimal configuration, the controller uses a helpful property of autonomous vehicles known as platooning. Platooning is defined as the ability of 2 sequential AVs to safely keep a smaller headway than the one that would be required if an HDV was involved. It follows that the optimal configuration is one where, in each lane, there will optimally be either only AVs or only HDVs. If a mixed lane exists, then the AVs should rearrange themselves so that no HDV is between them.

For the rearranging process, the controller makes use of the AVs' capability of influencing HDVs into taking a desirable action. The process is distributed to each AV, which individually interacts with the HDVs around it (making them change lanes, slow down, or speed up) to reach the configuration that is intended by the controller. This process is illustrated in fig. 1.

The other papers on the topic of our project usually consider settings where there is either only one lane per road or where lane changes are not allowed. In fact, very few of the works explored the possible advantages of lane-changing, which accentuates the findings in this paper. Besides the benefits that this strategy brings by itself, the techniques for lane balancing can also be combined with other approaches to

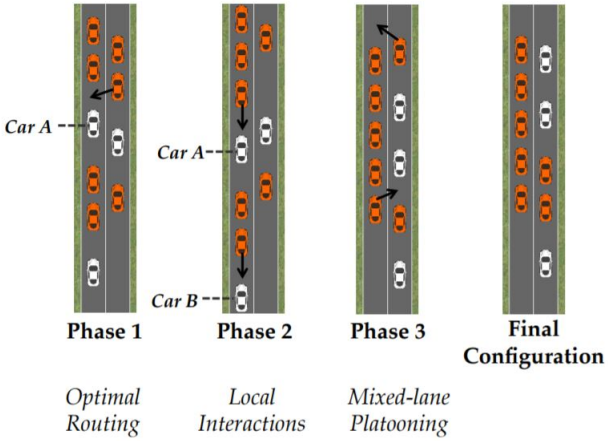

Fig. 1. Phase 1: the AVs follow optimal lane assignment. Phase 2: the AVs influence the HDVs to follow optimal lane assignment. The acting AVs pair with the HDVs A and B. Phase 3: the AVs platoon in the mixed lane. Taken from [3]

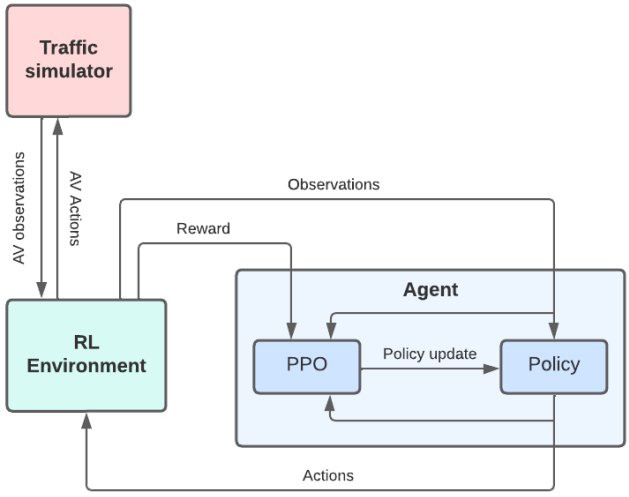

Fig. 2. Overview of the project setup

make them more efficient—the constant maximum usage of the road's capacity provides an additional efficiency gain in every scenario that involves a multi-lane road. Therefore, our solution takes inspiration from this algorithm, recognizing the multiple lanes as a resource to reorganize the vehicles in a way that is optimal for our goal.

C. Dissipating stop-and-go waves

Ideally, every vehicle would travel at its optimal performance, showing constant speed and spacing. However, this is not the case in natural highway traffic, as small disturbances inevitably form and propagate backward, eventually expanding and forming jams. This effect is known as stop-and-go waves and is commonly the cause of traffic congestion.

The work [2] focuses specifically on programming autonomous vehicles to mitigate these waves. It considers the scenario of an open single-lane highway with a merge section to generate the perturbations. The authors define a centralized RL controller with access to the AVs' observations and actions. The plan is that, after training the controller, the AVs that are at the beginning of the road learn to slow down or even stop in the event of a wave near the on-ramp, making the vehicles behind them also slow down prematurely, and this way smoothing the impact of the wave.

Note that, following this strategy, a higher AV penetration rate translates to needing a less drastic deceleration per AV, and since the vehicles can speed up again once the jam is cleared, a higher number of AVs will inevitably provide better results. Therefore, the authors test their solution for increasing AV penetration rates (the defined percentage of autonomous vehicles on the road). Surprisingly, the experiments show that a 2.5% AV penetration rate was sufficient to contribute greatly to dissipating the waves. Furthermore, at 10% the group of AVs was able to roughly dissipate the waves completely, with the vehicles moving twice as fast and with a 13% improvement in throughput.

This work showed multiple interesting conclusions. For one, it suggested that AV control algorithms may successfully replace the current less effective and more expensive ramp meters since, providing the same results, it is applicable to waves formed everywhere in the controlled area, instead of just the ramp. Additionally, it presented an AV control strategy for single-lane open roads with an on-ramp that is effective while not being too complex, which is commonly the issue with other solutions. For that reason, this solution allows itself to be expandable to more realistic conditions—our project takes great inspiration from this work, striving to expand it to the context of multi-lane roads.

III. Implementation

We address the problem of developing an autonomous vehicle control strategy that, in a mixed-autonomy context, prevents traffic congestion on an open multi-lane highway with a merge section, and that takes into consideration the possibility (and advantages) of lane-changing, using the lanes to optimally reorganize the vehicles on the main road.

A. Project overview and strategy

We define a centralized agent that has access to each AV's observations and can act through each AV's actions. Note that the agent does not have full access to the state of the network at each step, and instead only gets partial information from what the AVs can observe of their own surroundings. As such, the problem is modeled using a POMDP. We then use deep reinforcement learning to train this agent to learn an optimal policy that maximizes the network's outflow and the vehicles' average speed, since those are the best indicators of clear traffic.

Fig. 2 presents an overview of the project setup. The network and vehicles are represented in a traffic micro-simulator. At each step, the agent obtains the AVs' observations from

Fig. 3. Network configuration of the single-lane highway with merge section

the simulator and sends the AVs' next actions, based on the observations that it got and on an internal policy that it keeps. The agent then receives a reward depending on the state of the system and updates its policy according to that reward.

That said, we model our problem as the combination of two simpler problems:

1) a single-lane highway with a merge section, where the controller learns to maximize the network's outflow and dissipates stop-and-go waves;
2) a multi-lane highway without a merge section, where the controller learns to rearrange the AVs and the vehicles around them before the merge section into a distribution that would be optimal for handling the merge.

We decided to first experiment with training the AVs in these two sub-problems and then join the experiments to build the more complex scenario, to conclude whether these two behaviors can be joined to reach our initial goal.

That said, in scenario 1 we follow the approach described in [2], with the exception of some minor necessary changes to update the solution. For scenario 2, inspired by [3], the idea is to reorganize the vehicles into a configuration that is optimal for our initial goal. Since in [2] we concluded that a higher AV penetration rate allows for a smoother merge, we believe that the optimal configuration is one where the AVs are concentrated on the lane that suffers the merge. Therefore, the goal of scenario 2 is to make the AVs learn to reorganize that lane into having the largest possible AV rate. For the final scenario, we join these two strategies, expecting the AVs to manage between reorganizing the lane that suffers the merge and dissipating the waves to allow for a smooth merge.

B. Preliminary experiment 1

1) Network configuration: For the first preliminary experiment, we consider the single-lane highway with a merge section illustrated in fig. 3. The main road has a single lane of length $L_{hw} = 700$ m, with pre-merge length $L_{hw\_pre} = 600$ m and post-merge length $L_{hw\_post} = 100$ m. The merging road has length $L_m = 100$ m. We define the inflows of the main and merging roads to be, respectively, $f_{hw} = 2000$ veh/hr and $f_m = 100$ veh/hr, and the AV penetration rate to be $P_{AV} = 10\%$. We assume the AVs to be uniformly distributed, which means that every $\frac{100}{P_{AV}}$th vehicle that enters the network is autonomous.

2) Observation and action spaces: The observation space of the agent corresponds to what each AV can observe at each step. In this scenario, the observations are the AV's speed v_i, the speeds $v_{i,lead}$ and $v_{i,fol}$ of the vehicles directly in front and behind it, and the time headways $h_{i,lead}$ and $h_{i,fol}$ between the AV and those same vehicles. The action space is the bounded acceleration a_i of each AV.

3) Reward function: The used reward function is a weighted sum defined in eqs. 1 to 4. The first term R_v rewards the proximity of the system's speed $v(t)$ to the desired speed v_{des}, while the second R_{out} rewards high network outflows $out(t)$. We add a cost C_h for small AV space and time headways since these are indicators of congested traffic. In eq. 4, $h_{min,t}$ and $h_{min,s}$ are, respectively, the minimum desirable time and space headways, while $h_{i,t}(t)$ and $h_{i,s}(t)$ are the corresponding headways of AV i at time step t. For the experiments, we used $v_{des} = 25$ m/s, $out_{des} = 2100$ veh/hr, $h_{min,t} = 1$ s, $h_{min,s} = 7$ m, $a_1 = 0.1$ and $b_1 = 0.1$.

$$R_1 = a_1 \times R_v + (1 - a_1) \times R_{out} - b_1 \times \sum_{i \in AV} C_h \tag{1}$$

$$R_v = \|v_{des}\| - \|v_{des} - v(t)\| \tag{2}$$

$$R_{out} = \min[out(t)/out_{des}, 1] \tag{3}$$

$$C_h = \min[h_{i,t}(t) - h_{min,t}, h_{i,s}(t) - h_{min,s}, 0] \tag{4}$$

In the original solution presented in [2], only the average system speed was considered—however, we found that, in that case, the AVs ended up exploiting the reward function by stopping at the beginning of the main road (blocking the inflow) until the road was clear and then speeding through the road, this way managing to output high average speeds at the cost of lowering the network outflow. We added R_{out} as a workaround to this problem. Additionally, the original headway cost only accounted for time headways. We decided to account for space headways as well since the time headway can be uninformative at very low speeds.

C. Preliminary experiment 2

1) Network configuration: For the second preliminary experiment, the network is as shown in fig. 4, where the main road has two lanes instead of one, and there is no merging lane. As in the first experiment, $L_{hw} = 700$ m and $P_{AV} = 10\%$. Given the extra lane and since we want to test the controller's ability to learn the intended behavior in high-density traffic, the inflow of the main road is increased to $f_{hw} = 3500$ veh/hr. This is because, in the final experiment, the traffic is expected to have high density caused by the bottleneck. We assume the lane that suffers the merge to be the right lane (also called lane 0).

2) Observation and action spaces: The observation space includes some of the observations already described in III-B for each AV—v_i, $v_{i,fol}$ and $h_{i,fol}$—, with the addition of the AV's lane l_i and the type fol_i (AV or HDV) of the vehicle directly behind it. The action space consists of each AV's bounded acceleration a_i and their direction d_i.

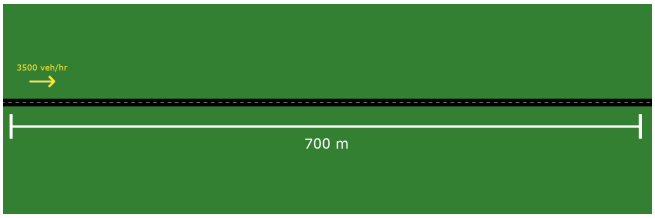

Fig. 4. Network configuration of the multi-lane highway

Fig. 5. Network configuration of the multi-lane highway with merge section

3) Reward function: The reward function is the weighted sum defined in eqs. 5 to 7 and 2. The first term R_v rewards proximity to the desired speed. The second term R_p rewards the proximity of the AV rate $p(t)$ on the right lane to the desired rate p_{des}. The final term B_{push} is a bonus that rewards AVs for influencing an HDV to change out of the right lane. For this scenario, we used the constants $v_{des} = 25$ m/s, $p_{des} = 0.25$, $a_2 = 0.75$ and $b_2 = 0.1$.

$$R_2 = a_2 \times R_v + (1 - a_2) \times R_p + b_2 \times \sum_{j \in HDV} B_{push} \quad (5)$$

$$R_p = \min[p(t)/p_{des}, 1] \quad (6)$$

$$B_{push} = \begin{cases} 1, & \text{if } lane_j(t-1) = 0 \wedge lane_j(t) \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

D. Main experiments

1) Network configuration: Our main experiments consider the open network scenario of a 2-lane highway of length $L_{hw} = 700$ m, with a merge section where the merging road is single-lane and has length $L_{hw} = 100$ m. As in III-B, the main road has pre-merge length $L_{hw\_pre} = 600$ m and post-merge length $L_{hw\_post} = 100$ m. The inflows on the main and merging road are set to $f_{hw} = 3500$ veh/hr and $f_m = 200$ veh/hr, and the AV penetration rate remains at $P_{AV} = 10\%$. Fig. 5 illustrates the network.

2) Observation and action spaces: The observation space for this problem combines the ones used in the preliminary experiments, including, for each AV: the AV's speed v_i and lane l_i, the speeds $v_{i,lead}$, $v_{i,fol}$ and headways $h_{i,lead}$, $h_{i,fol}$ of the vehicles directly in front and behind the AV, and the type fol_i of the vehicle directly behind it.

The action space is the same as in III-C, corresponding to each AV's bounded acceleration a_i and their direction d_i. Note that the direction corresponds to the intent or not of moving to the lane on the right/left of the current lane and that the AVs cannot be in between two lanes. Additionally, in this environment, any action chosen by the agent that would lead to immediate collisions (for example, changing lanes when there is no space on the target lane) is overwritten by the simulator.

Since the network is open, the number of AVs (and consequently, the number of observations and actions) varies throughout the experiment. Therefore, we define a fixed maximum size N_{AV} for the set of controlled AVs at each step and use zero padding for the observation and action space when there are fewer than N_{AV} AVs in the network.

3) Reward function: The reward function used to train the vehicles is a weighted sum of the reward functions used in the preliminary experiments, defined as follows:

$$R = \alpha \times R_1 + (1 - \alpha) \times R_2 \quad (8)$$

where R_1 is defined in eq. 1, R_2 is defined in eq. 5 and $\alpha \in [0, 1]$. R_1 rewards behaviors that lead to the dissipation of the waves that may form near the merge, while R_2 rewards behaviors that lead to an increase of the AV ratio on the right lane. We experiment with a set of different α values to conclude which proportion between these two terms allows for the best results. Regarding the first term, we set $v_{des} = 25$ m/s, $out_{des} = 3700$ veh/hr, $h_{min,t} = 1$ s, $h_{min,s} = 7$ m, $a_1 = 0.1$ and $b_1 = 0.1$. Regarding the second term, we set $p_{des} = 0.25$, $a_2 = 0.75$ and $b_2 = 0.1$.

We point out that we were unfortunately forced to limit the number of lanes to two due to the complexity of the experiments—the training duration proved proportional to the number of vehicles currently in the network since the environment required a running simulation of the network while training. This originated a significant difference in the training times when we added a second lane on the main road and ultimately rendered adding a third lane unfeasible.

E. Simulations

The experiments are implemented in Flow [5], an open-source framework developed by the authors of the work [2] to perform reinforcement learning experiments in traffic micro-simulators. Flow allows for the creation of various traffic-oriented RL tasks with the goal of developing control strategies for autonomous vehicles. Additionally, we use SUMO [1] for the execution of the simulations. SUMO is a renowned open-source traffic micro-simulator. Finally, the human-driven vehicles' behavior and dynamics are modeled using the Intelligent Driver Model (IDM) [4], a microscopic car-following model built into SUMO.

The simulations are executed with time steps of 0.2 s and a total duration of 3600 s. The RL agent receives observations and chooses new actions every 5 simulation time steps, repeating its actions in the meantime.

The agent is trained using the Proximal Policy Optimization (PPO) algorithm, with discount factor $\gamma = 0.999$ and a

TABLE I
RESULTS OF THE PRELIMINARY EXPERIMENT 1

$P_{AV,total}$ (%)	speed (m/s)	outflow (veh/hr)	# vehicles	$P_{AV,lane0}$ (%)
0	7.08	1440	49	-
10	14.25	1560	24	-

TABLE II
RESULTS OF THE PRELIMINARY EXPERIMENT 2

$P_{AV,total}$ (%)	speed (m/s)	outflow (veh/hr)	# vehicles	$P_{AV,lane0}$ (%)
0	20.2	3460	38	10*
10	16.4	3436	51	16

learning rate of 0.001. Furthermore, we use an MLP actor-critic policy with hidden layers (32, 32, 32) and relu activation function.

IV. RESULTS

In this section, we present the numerical results for each of the experimental settings described in section III and reflect on the learned behavior of the trained controllers.

Each simulation had a total duration of 3600s, and the results were averaged over 50 simulations to account for stochasticity between simulations. The considered metrics throughout the experiments are the average of the system's speed at each step ($speed$), the total outflow over the duration of the simulation ($outflow$), the average number of vehicles on the highway at each step ($\#vehicles$), and the average percentage of AVs on the right lane at each step ($P_{AV,lane0}$).

Videos of the simulations are available at SimulationReplays. The title of each video is informative of the experiment setting, the level of autonomy, and, in the case of the final experiments, the used value for α. Although some videos may be longer due to simulator processing delays, each video shows a total of 1200 simulation seconds. The current simulation time can be assessed in the top left corner. Regarding the colors of the vehicles, red vehicles correspond to AVs, white vehicles to HDVs, and blue vehicles to the HDVs that the AVs can currently observe. Finally, we recall that only a limited group of AVs can be controlled at each step—as such, in the event of a jam, there can be some uncontrolled AVs in the network that will act as HDVs.

A. Preliminary experiment 1

Table I shows the obtained results for the single-lane highway with merge section experiment (described in III-B), comparing the zero-autonomy simulations to the mixed-autonomy simulations performed using the trained controller. The final metric is not relevant in a single-lane scenario and is thus not included.

In the mixed-autonomy simulations, we notice an 8% increase in the outflow of the network, along with a 50% increase in the average speed, with the vehicles effectively traveling at double the speed. The number of vehicles on the network at each step also decreased to half in the case of mixed autonomy, showing a clear improvement in the road's efficiency.

From the simulation replay of the mixed autonomy setting[2], we observe that, as expected, the AVs at the beginning of the highway slow down in the event of a wave, forcing the line of vehicles behind them to slow down as well, and successfully dissipating the waves.

[2]Video titled *SingleLaneMerge_MixedAutonomy*

B. Preliminary experiment 2

In table II, we assess the results of the multi-lane highway experiment. To be able to evaluate the $P_{AV,lane0}$ evaluation metric (which represents the main goal of this experiment) on the zero-autonomy simulations, we perform the simulations using puppet AVs which, without a trained RL controller, simply act as HDVs while carrying the label of being an AV. That said, in the zero-autonomy simulations, this value corresponds to the set AV penetration rate of 10%. This is expected since the departure lane for each vehicle is random. In the mixed autonomy setting, we see that the controller was able to increase this percentage to 16%, but there is a consequent decrease of 19% in the average speed and also a slight decrease in the outflow.

From the replay of the simulation[3], we see that the AVs move to the right lane when possible, and sometimes slow down in front of an HDV, eventually influencing it to change to the left lane, as intended. However, since the inflow is high, the traffic is inevitably very dense, and thus the repeated lane changes in the mixed-autonomy scenario end up generating jams (as seen toward the end of the video), whereas in the zero-autonomy scenario there are no disturbances and the vehicles travel at their free-flow speed. Additionally, we note that, after these jams are formed, the attempts of the AVs to change to the right lane are overwritten given the lack of space on that lane. This justifies the decrease in the average speed and why the AV percentage increase on the right lane was not larger.

In any case, our insight is that, in the multi-lane merge scenario where both solutions are joined, the obtained increase in the AV penetration rate on the right lane should compensate for its consequences, since the controller in that environment has more resources to dissipate the waves.

C. Main experiment

For the final scenario, we show in table III the results of the experiments for different values of the α scalar in eq. 8. We compare the experiments for $\alpha = \{1, 0.75, 0.50, 0.25, 0\}$. To understand the results, we start by looking at the experiment where $\alpha = 0.50$, which is the one that achieved the best performance.

1) Best case strategy: Surprisingly, in the experiment where $\alpha = 0.50$, the $P_{AV,lane0}$ was actually lower than the original AV penetration rate on the inflow of 10%. From the replay of the simulation[4], we noticed that the controller had naturally developed a different strategy for dissipating the waves in this multi-lane scenario. In the replay, the AVs end up staying

[3]Video titled *MultiLaneHighway_MixedAutonomy*

[4]Video titled *MultiLaneMerge_MixedAutonomy_alpha050*

TABLE III
RESULTS OF THE MAIN EXPERIMENTS FOR DIFFERENT α VALUES

α	speed (m/s)	outflow (veh/hr)	# vehicles	$P_{AV,lane0}$ (%)
1	4.10	1955	113	11
0.75	6.43	2614	99	9
0.50	13.1	3045	53	5
0.25	8.08	2801	87	10
0	7.82	2823	89	11

mostly on the left lane, where they behave similarly to the first preliminary experiment, slowing down periodically to open space in the lane. Then, in the event of a wave near the merge, they quickly alternate between the two lanes, surprising the vehicles and thus forcing them to brake. Although this behavior effectively resulted in decreasing the impact of the wave on both lanes, we did not plan or anticipate it.

We first note that, in the event of a wave in a multi-lane scenario, the vehicles in the lane that suffers the merge are more likely to change into the adjacent lane to avoid stopping, rather than staying in their current lane. Thus, the wave on the right lane only aggravates once there is no more space on the adjacent lane for the vehicles to change into, that is after another wave has formed on the adjacent lane. The video of the zero-autonomy simulation for this setting[5] shows this behavior. This effectively translates to a second parallel merge of the right lane into the left lane—and this merge is much more alarming than the original since the number of vehicles on the right lane is almost 10 times the number of vehicles on the actual merging road. With this in mind, we understand why the left lane should be the most closely monitored, and consequently the one with the higher AV percentage. Additionally, we note that the HDVs have a slight resistance to overtaking on the right, meaning that when an AV brakes to prepare the left lane for the merge, it indirectly manages to slightly affect the right lane as well.

Finally, we realize that, for increasing values of α in our reward function, the term that rewards high AV percentages on the right lane (originally defined in equation 6) ends up amounting to very small values. Additionally, since there is a constant inflow and outflow of vehicles, and since this term is recalculated in every time step, it frequently varies regardless of the AVs actions. Therefore, we believe the controller ultimately interprets this term as an incentive for changing into the right lane, rather than as a continuous reward for staying in that lane, justifying the repeated changes between lanes.

2) Remaining results: For every other value of α, we get significantly worse results. Note that, when α is 0, the reward function becomes equal to $R2$ (equation 5), and the experiment is similar to the one described in section III-C, however applied to the more complex scenario of an on-merge section. Accordingly, in the simulation video[6] the AVs change to the right lane as soon as possible. However, since they lack the incentive to dissipate the waves, they do not slow down to prepare their lane for the merge, and allow the waves to quickly expand.

Likewise, when α is 1, the reward function is $R1$ (equation 1) and the controller is rewarded solely for behaviors that lead to the dissipation of eventual waves, as in the experiment of section III-B. From the simulation [7], we can see that the AVs rarely change lanes, and simply focus on braking extensively in their current lane, a strategy that ends up generating its own jams. We believe that, without the incentive to change to the right lane, the controller is unable to learn the working strategy of alternating lanes, and instead chooses a defensive strategy that is ineffective.

In the replay of the experiments where α is 0.25[8] and 0.75[9], we recognize some efforts to follow the same strategy as in the $\alpha = 0.50$ experiments—we see that the AVs change lanes repeatedly, trying to control both lanes, and also brake in their lane, managing to create small gaps. However, the controller is unable to balance these two behaviors and the waves inevitably propagate. That said, when $alpha = 0.75$, the strategy works slightly better in the beginning than when $alpha = 0.25$, as the AVs are able to dissipate the first waves. Nevertheless, when a wave is finally able to expand, the efforts of the controller to mitigate the wave are counterproductive and end up generating additional jams behind the AVs, whereas when $alpha = 0.25$ the waves simply propagate as in the zero-autonomy setting.

This detail, along with the overall worse results in table III for $alpha < 0.5$ than for $alpha > 0.5$, leads us to conclude that the strategy used to mitigate stop-and-go waves in the preliminary experiment III-B is only effective in the multi-lane scenario if it is joined with the incentive to change lanes, and is otherwise actually harmful to traffic.

3) Achieved traffic improvement: Finally, we evaluate the performance of the best-case experiment, $\alpha = 0.50$, by comparing it against the zero-autonomy setting. Table IV shows the results for both simulations. We notice that, using the RL controller, there is a 28% increase in the average speed of the vehicles, along with a 5% increase in the network's outflow and a 44% decrease in the density of traffic.

We conclude that the trained AV controller, although not following the strategy we envisioned, effectively improved traffic conditions on a two-lane highway with a merge section.

TABLE IV
PERFORMANCE OF THE BEST MAIN EXPERIMENT AGAINST THE ZERO-AUTONOMY SIMULATION

$P_{AV,total}$ (%)	speed (m/s)	outflow (veh/hr)	# vehicles	$P_{AV,lane0}$ (%)
0	10.2	2902	94	10*
10	13.1	3045	53	5

V. CONCLUSIONS

This project highlights the possible advantages of autonomous vehicles to traffic. We propose a deep RL-based

[5]Video titled *MultiLaneMerge_ZeroAutonomy*
[6]Video titled *MultiLaneMerge_MixedAutonomy_alpha0*
[7]Video titled *MultiLaneMerge_MixedAutonomy_alpha1*
[8]Video titled *MultiLaneMerge_MixedAutonomy_alpha025*
[9]Video titled *MultiLaneMerge_MixedAutonomy_alpha075*

autonomous vehicle control mechanism that improves traffic conditions near a merge section on an open two-lane highway.

Since the envisioned strategy is the composition of two sub-strategies, we additionally present implementations for each—we update an existing AV control strategy that dissipates stop-and-go waves near a merge section in a single-lane highway, and develop a strategy that reorganizes the vehicles on a multi-lane highway—and prove their effectiveness in a mixed-autonomy traffic micro-simulation. We then study the proportion between the two behaviors that allows for the best performance in the joined scenario, noticing the AVs' best-learned behavior to be different from our envisioned strategy. As such, we document this alternative behavior and how it developed from our implementation, so that in the future it may be reproduced and possibly optimized.

From the simulations, this autonomous vehicle control solution, at a 10% AV penetration rate, is shown to allow for a 28% increase in the average vehicle speed and a 5% increase in the network's outflow, with the AVs effectively minimizing the consequences of the merge section and utilizing both lanes to better prepare the vehicles for the merge. Therefore, this work effectively presents an autonomous vehicle control approach that minimizes traffic congestion on a multi-lane merge, proving the potential benefits of designing autonomous vehicles to improve traffic conditions.

Our solution is unfortunately limited from only being tested and proved to work on a two-lane scenario, and therefore we see it as future work to expand it to the context of more lanes. Furthermore, it does not account for every human being unique and ultimately assumes that every human's behavior in traffic will be similar, which leaves a significant gap between our solution's effectiveness in the simulator and its true effectiveness in real-life traffic. Knowing that the optimal learned strategy was actually very different from our envisioned one, we additionally suggest re-implementing the solution to reward those optimal behaviors, simplifying it and possibly even achieving better results.

REFERENCES

[1] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of sumo-simulation of urban mobility. *International journal on advances in systems and measurements*, 5(3&4), 2012.

[2] A. R. Kreidieh, C. Wu, and A. M. Bayen. Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, page 6. IEEE, 2018.

[3] D. A. Lazar, R. Pedarsani, K. Chandrasekher, and D. Sadigh. Maximizing road capacity using cars that influence people. In *2018 IEEE Conference on Decision and Control (CDC)*, page 8. IEEE, 2018.

[4] M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

[5] C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen. Flow: A modular learning framework for mixed autonomy traffic. *IEEE Transactions on Robotics*, 2021.