CSCRS Road Safety Fellowship Report: A Human-Machine Collaborative Acceleration Controller Attained from Pixel Learning and Evolution Strategies

Fangyu Wu[1]Alexandre M. Bayen[2]

Abstract— **Autonomous driving has been a major area of research in transportation since the mid-1980s. A central motivation of this new paradigm is that it promises greater safety and efficiency compared to human drivers. However, despite recent remarkable progresses, current autonomous driving systems still require improvements. Inspired by the success of autopilot systems in flight control, we propose to approach the problem from the perspective of vehicle control augmentation. Specifically, we adapt a state-of-the-art reinforcement learning (RL) algorithm called Evolution Strategies (ES) to train an auxiliary acceleration controller on a pixelated local observation space. By augmenting the actions of a human driver modeled by the Intelligent Driver Model (IDM), we show that the RL controller is able to help humans drive more efficiently than they would without augmentation by more than 50%. This preliminary study suggests that this type of hybrid approach may be used to develop user-specific adaptive driving controllers and provably safe autonomous vehicles. For reproducibility, the source code of this project is released at https://github.com/flow-project/flow.**

I. Introduction

With the recent rapid development of autonomous vehicles, there has been a wave of technological innovations in the automobile industry, such as Google's Waymo, Tesla's Autopilot, and General Motor's Super Cruise. While these companies aim to develop driving systems that could ultimately replace human drivers, recent news on fatal traffic accidents involving those experimental automated vehicles suggest that progress remains to be made in automation [1].

The social implications of full vehicle automation are also concerning. One of the fundamental assumptions of the current traffic rules is that a vehicle must be operated with complete human supervision. Hence, certification of autonomous vehicles will certainly involve a long and challenging legal process.

Motivated by the technological and societal challenges mentioned above, we draw inspirations from the aviation industry. Aircraft autopilot systems are among the safest methods of transportation. However, their high performance and robustness were not built overnight. Instead, they initially started as assisted pilot systems and were only slowly upgraded over time. Introduction of new features was incremental and always tested. Such a conservative development path ensures a safe and steady transition from early manual control systems to modern autonomous control systems.

Inspired by the success of flight autopilot systems, we propose to adopt a framework called *augmented driving* to facilitate the transition from human-operated driving to fully autonomous driving. Augmented driving is a vehicle control system that enhances the performance of the existing driving control system, such as a human driver or an adaptive cruise controller.

In the short term, by combining the strength of the host and the assistant, one of the goals of automation is to develop a vehicle controller safer and more efficient than if used in isolation. In the long term, such a framework provides a tangible path for technological development and social changes. With it, automotive companies would be able to focus on realistic incremental improvements. Over time, as the assistant is steadily incrementally improved, the host will delegate more and more tasks to the assistant. Eventually, as

[1] Fangyu Wu is with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94709, USA, fangyuwu@berkeley.edu

[2] Alexandre M. Bayen is with the Department of Electrical Engineering and Computer Sciences and the Institue of Transportation Studies, University of California, Berkeley, Berkeley, CA 94709, USA, bayen@berkeley.edu

the auxiliary controller becomes powerful enough, it will effectively take over the role of the host. Since augmented driving systems will not be able to replace humans in the transition phase, society has a chance to adjust its legal structures to this technological shift.

In this article, we formally define the problem of driving augmentation and propose a solution based on a state-of-the-art evolutionary RL algorithm named ES [2]. The key contributions of this article are as follows:

- We develop an end-to-end learning algorithm that can train a vehicle acceleration controller with only pixel inputs and system-evaluated rewards.
- We demonstrate that the image-based assistive acceleration controller improves the baseline human driving model IDM by over 50%.
- We suggest that in the long run *augmented driving* can be a viable strategy to design adaptive and safe driving controllers.

The remainder of this article is organized as follows. We begin with a brief overview of the literature in Section II. In Section III, we formulate the problem in mathematical terms. Section IV proposes a solution based on a state-of-the-art RL algorithm, called ES, coupled with a car following model, named IDM. The numerical experiments are demonstrated in Section V and discussed in Section VI. Finally, we conclude our study in Section VII.

II. Background

Automated vehicle control has been an important research area in robotics and control. Over the past ten years, there has been remarkable technological progress in this field, like the early success of Team Stanley in the DARPA Grand Challenge [3], the release of Autopilot assisted driving system by Tesla [4], and the release of the autonomous taxi program by Waymo [5].

Among the technologies that enable the high-performance autonomous vehicles, RL has been significant. By imitating the behaviors of human drivers, researchers from NVIDIA show that a neural network can learn to drive in simple scenarios directly from camera pixels [6], [7]. With more advanced RL algorithms such as asynchronous actor critic [8], deep Q-learning [9], and evolution strategies [2], one may be able to train a functional driving controller with higher sample efficiency and lower generalization errors than the plain imitation learning approach.

Among the existing RL-based autonomous driving methods, a common theme is to train a vehicle to obey the traffic rules [7] and to arrive at destination as fast as possible [10]. While these goals are essential for a vehicle to be able to operate in the real world, they do not answer the other half of the promises of vehicle automation, i.e., using autonomous vehicles to reduce road fatality and to improve transportation efficiency. To see this, one may find that under this fundamental objective, an autonomous driving controller may learn to drive selfishly, e.g., frequent passing and aggressive tailgating. Although behaving as such might let the vehicle arrive the destination faster, it will put public safety at risk and impact other drivers' travel time. Therefore, to reduce accidents and increase throughput, one needs to look beyond the paradigm of learning to drive and learning to drive fast.

A next-level question to ask is how to train autonomous vehicles to drive in a mixed-autonomy traffic with the goal of maximizing navigation safety and road throughput. To this end, recent field experiments have demonstrated that it is possible to use a small percentage of autonomous vehicles to improve overall traffic flow. In one of the experiments, the researchers show that a manually designed proportional integral saturation controller can significantly improve the overall traffic flow on a single-lane circular track with only one autonomous vehicle [11].

Recently, numerical studies have shown that model-free RL can achieve the same level of performance as the hand-tuned controller [12], [13], [14], [15] in and beyond the original setup of [11], i.e., the idealistic traffic on a single-lane circular track. Using the state-of-the-art RL algorithms, it is demonstrated that model-free RL is capable of finding interesting driving strategies in more complex traffic networks, including intersections, merge points, and roundabouts.

One of the characteristics of the work in [12], [13] is that it assumes perfect information about the system in terms of vehicle speeds, spacings, and positions during the test time. However, in real world applications, this data is not easy to acquire and measurements are often noisy when driving in high speed or in dense urban areas.

In addition to complications arising from imperfect information, feature engineering becomes difficult in complex urban settings, where the topology of the road network becomes complicated and the state of the drivers becomes high-dimensional. For example, it is challenging to define the notion of leaders and followers in multi-lane traffic: A taxi driver who is looking for customers may not necessarily follow the front vehicles; A careful truck driver who is about to change lane may pay more attention to the vehicles in the target lanes than the vehicle behind.

To address the issues of measurement difficulties and feature engineering, a natural solution is to work directly with raw sensor inputs. For many autonomous driving systems, such inputs take the form of N-dimensional images centered at the vehicle's current position. Such representation is convenient since it directly captures everything within the system without explicit state estimation and feature engineering.

Given the inputs as images, a good choice for the RL algorithm is the ES method. Image-based end-to-end RL has been proven a viable strategy since the success of deep Q-learning [16]. Later work by [2] has shown that the ES method has comparable performance to deep Q-learning but is more convenient for parallel computing on clusters. Given the availability of a massive cloud computing infrastructure, we hence choose ES as our RL algorithm.

Motivated by the discussion above, we choose to study the performance of augmented driving controllers in the image-based state space using an end-to-end learning process based on ES method. We formally formulate the problem in the following section.

III. Problem Formulation

As commonly adopted in RL, we define the problem as a discrete-time *partially observed Markov decision process* (POMDP). By definition, a POMDP consists of state space $\mathcal{S}$, action space $\mathcal{A}$, observation space $\mathcal{O}$, transition probability $\mathcal{T}$, emission probability $\mathcal{E}$, and reward r. In our specific problem, the POMDP $\{\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{E}, r, \rho_0\}$ takes the following form.

- $\mathcal{S} \in \mathbb{R}^{h \times w \times c \times t}$ is the *state space* with height h, width w, channel c, and memory t. As we are using a grayscale image, $c = 1$. The memory $t = 50$ since memory buffer is set to be the past 5 seconds at a temporal resolution of 0.1 second.
- $\mathcal{A} : \mathcal{O} \to \mathbb{R}$ is the *action space*, corresponding to the *correction* to vehicle's baseline longitudinal acceleration. Hence, the resulting combined longitudinal acceleration a is a linear combination of the baseline acceleration a_0 and the RL correction term a_+ weighted by a controller parameter α

$$a = (1 - \alpha)a_0 + \alpha a_+,$$

where α is the augmentation coefficient where $\alpha \in [0, 1]$; a_0 and a_+ are bounded control inputs between -5 m/s^2 and 3 m/s^2.
- $\mathcal{O} : \mathcal{S} \to \mathbb{R}^{h' \times w' \times c' \times t'}$ is the *observation space* with height h', width w', channel c', and memory t'. Specifically, we choose $h' = 100$, $w' = 100$, $c' = 1$ and $t' = 5$. See Figure 1 for an example about the geometric relation between observation space $\mathcal{O}$ and state space $\mathcal{S}$. Note that on the right, the images are rendered at two pixels per meter and the radius of the observation space is set to 25 meters. As shown in Figure 2, the observation space stores the temporal information from the past five-second at a temporal resolution of one second.
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the *transition probability* that maps a state-action pair to the next state. Here POMDP assumes that the next state is only dependent on the current state and action.
- $\mathcal{E} : \mathcal{S} \times \mathcal{O} \to [0, 1]$ is the *emission probability* that maps the state to the observation.
- $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the *reward function*. The specific form of reward function is problem-dependent, but it should capture the two fundamental aspects of driving, i.e., safety and efficiency. For this study, we choose

$$r = \frac{1}{v_{\max}} \left(\beta v_{\text{avg}} - (1 - \beta) v_{\text{std}} \right),$$

with

$$v_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} v_i, \quad \text{and} \quad v_{\text{std}} = \sqrt{\sum_{i=1}^{N}\left(v_i - \frac{1}{N}\sum_{i=1}^{N} v_i\right)^2},$$

where v_{avg} is the average velocity, v_{std} is the velocity standard deviation, v_i is the current velocity of the ith vehicle, N is the total number of vehicles in the POMDP, and $v_{\max}$ is the maximum allowable velocity, which is set to 30 m/s^2, and β is an user-defined weighting parameter which in this article is set to be 0.8. For training efficiency, v_i is obtained directly from the simulator, rather than estimated from the images. Note that we use v_{avg} to measure the system throughput and v_{std} to approximate the risks of collisions [17].

- Finally, $\rho_0 : \mathcal{S} \longrightarrow [0, 1]$ is the *initial state distribution*.

In RL, our goal is to find a series of actions that maximizes the total reward. We formalize this notion of reward maximization below.

Denote $p(s_{t+1}|s_t, a_t)$ as the transition probability from the state-action pair (s_t, a_t) to the next state s_{t+1}. Let the τ be the trajectory of the system from the time t to event horizon T, that is, $\tau = (s_t, a_t, s_{t+1}, a_{t+1}, \ldots, a_{T-1}, s_T)$. Note that event horizon T is defined to be from the initial time to the time when the POMDP terminates. In our problem, we set $T = 1500$ s.

The objective of the problem is then to find a function $\pi^*(a_t|o_t) : O \longrightarrow \mathcal{A}$, also known as *policy*, such that the expected future reward is maximized,

$$\underset{a_t,\cdots a_T}{\text{argmax}} \quad \mathbb{E}_\tau \sum_{i=t}^{T} r_i. \tag{1}$$

The function π^* that produces the solution to Equation (1) is defined as the *optimal policy*.

If we can represent π^* with some θ-parametrized function approximator $\hat{\pi}_\theta$, then solving Equation (1) is equivalent to solving the following optimization problem:

$$\underset{\theta}{\text{argmax}} \quad \mathbb{E}_\tau \sum_{i=t}^{T} r_i. \tag{2}$$

With problem formally stated in Equation (2), we propose our solution in Section IV.

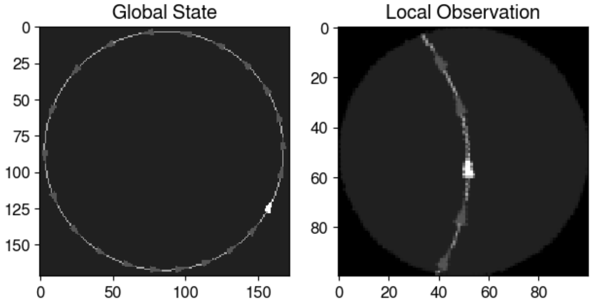


Fig. 1: Comparison between global state space and local observation space. Note that the state space and observation space only contains raw pixels.

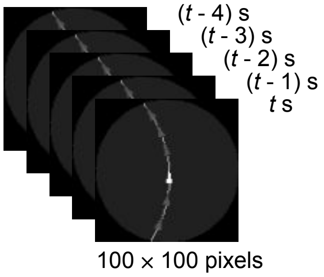


Fig. 2: Pixelated local observation space with a five-second memory. Note that the memory buffer increments at 1 s while the simulation steps at 0.1 s.

IV. Proposed Solution

We apply the ES method to solve the problem defined in Section III. To achieve this, we need a data collection procedure and an implementation of the ES method. We delineate the technical details of the two processes as follows.

A. Data collection

We demonstrate that our method can learn an effective traffic controller with pure simulated data. To this end, we select the open-sourced traffic simulation software *Simulation of Urban Mobility* (*SUMO*) [18] and its corresponding RL library *Flow* [12].

To approximate human driver's behaviors, we choose the well-known human-driving model IDM [19] as defined below:

$$\frac{d^2x}{dt^2} = a\left(1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v, \Delta v)}{s}\right)^2\right),$$

with $s^*(v, \Delta v) = s_0 + vT + \frac{v\Delta v}{2\sqrt{ab}}$, where v_0 is the desired velocity, s_0 is the minimum desired spacing, T is the minimum desired headway, a is the maximum vehicle acceleration, b is the deceleration coefficient, δ is the exponent coefficient. In this study, we choose to use the default IDM parameters in [20]: $v_0 = 30$ m/s, $s_0 = 2$ m, $T = 1$ s, $a = 1$ m/s^2, $b = 1.5$ m/s^2, and $\delta = 4$.

Note that the use of IDM as the human-driving model is not a necessity. In fact, one can pick any other human-driving model, or even use an actual human driver. The procedure of training a controller will stay the same. In our case, we simply opt for IDM, as it is one of the most acknowledged models in the community.

Since both *SUMO* and *Flow* do not support real-time image-based state information query, we have developed a rendering plugin in *Flow* based on a python graphics library called Pyglet [21]. It is capable of rendering any *SUMO* traffic scene at approximately 15 frames per second on a modern computer. With it, one can train a functional RL controller with ES in a 16-CPU machine in less than 4 hours. To facilitate future research, the source code of this plugin is released at https://github.com/flow-project/flow.

B. Algorithm implementation

For the RL algorithm, we use an off-the-shelf implementation of ES from a python RL library called *RLlib* [22]. The library is designed for accelerated RL training through distributing the computations across multiple CPUs. In this study, all the numerical experiments are parallelized on a 16-CPU AWS EC2 instance of type c4.4xlarge. The source code of the implementation is hosted on https://github.com/flow-project/flow and the EC2 AMI hash is ami-04b53bf506e95394f.

For hyperparameter settings, we decide the optimal parameters through a parameter sweep. Through grid searching in the parameter space, we find the most effective ES hyperparameters are as follows:

- Evaluation probability: 0.05
- Noise standard deviation: 0.01
- Step size: 0.01
- Iterations per trial: 50
- Number of trials: 3

The training is repeated for three times with different random seeds. The resulting trained model is selected to be the one with the highest rewards among the three trials.

Lastly, we use a neural network as the underlying parameterized model. As for the neural network architecture design, we use a two-layer convolutional neural network followed by a two-layer multilayer perceptron. The details of the architectural design of the neural network are illustrated in Figure 3.

V. Numerical Experiments

We choose to investigate the effectiveness of augmented driving in two sets of numerical experiments. The first set of experiments is conducted on a circular track inspired by the work in [23], as shown in Figure 4a. The second experiment set is set to be a cyclical cross following the design in the work of [12], as illutrated in Figure 4b. The circular track approximates the single-lane highway driving scenario, while the cyclical intersection resembles an one-way intersection in a dense urban setting. Despite its abstraction from real-world traffic scenarios, these two environments provide a controlled environment to study the feasibility of pixel-based driving augmentation. For brevity, in the following paragraphs, we denote the circular track as *Circle*, and the cyclical intersection as *Cross*.

To test the effectiveness of the driving augmentation, for each environment, we train a RL agent for various augmentation coefficient α ranging from 10% to 100% in increments of 10%. The training processes for the 10 agents on *Circle* are illustrated in Figure 5a, while the training processes for the 10 agents on *Cross* are shown in Figure 5c. As indicated in the color bars, the darker greens represent higher rewards, while the lighter greens represent the lower rewards. Note that pure white color on the left edges of the plots indicate NaN (not a number), which is caused by collisions during early stage of policy exploration. The figures indicate that high rewards are attained with a augmentation factor coefficient between 40% to 70%.

As implied by the experiments, RL augmentation can improve driving efficiency of the baseline

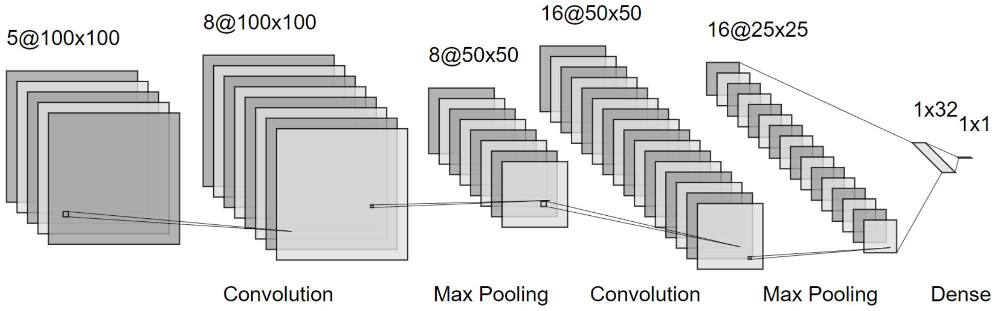
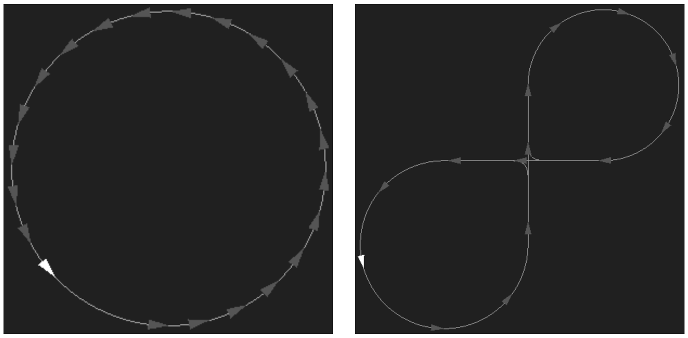
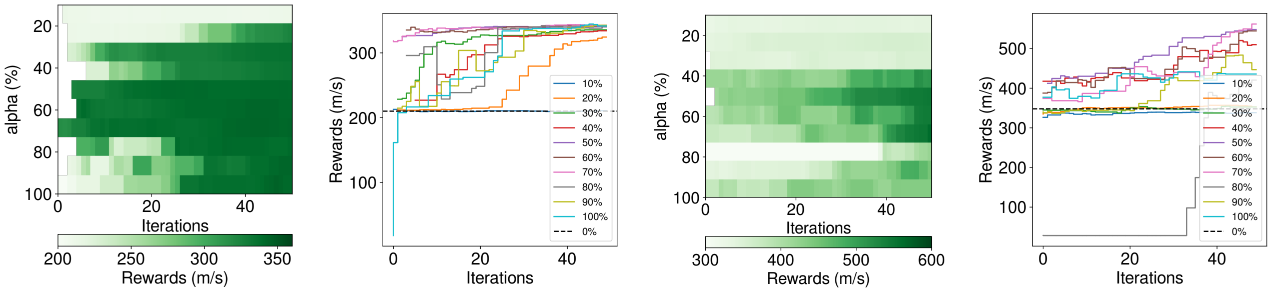


Fig. 3: Neural network model architecture. The network takes a five-channel image as input and produce a single number as output which corresponds to the correction to human driver's default acceleration.



(a) *Circle* environment. (b) *Cross* environment.

Fig. 4: Experiment environments. *Circle* environment represents a single-lane circular track. *Cross* environment represents a self-connected single-lane intersection.



(a) *Circle* learning heatmap. (b) *Circle* learning curves. (c) *Cross* learning heatmap. (d) *Cross* learning curves.

Fig. 5: Training learning heatmaps and learning curves. Both environments achieve better performance at an augmentation coefficient between 40% and 70%.

	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Circle	210	210	324	336	334	340	342	**343**	340	336	342
Cross	348	336	353	350	510	548	545	**562**	420	447	436

TABLE I: Training results of ES on *Circle* and *Cross*. The best performance is attained at 70% as shown in bold.

IDM driver by more than 50% as shown in Figure 5b and Figure 5d. The baseline controllers are marked with black dash lines while the augmented controllers are displayed in solid colored curved. To be precise, the best performance of each experiment is tabulated in Table I.

Furthermore, note that in the *Cross* environment pure RL agent is outperformed by a few human-machine-collaborated systems. We hypothesize that such improvement may be diminished if we train the agents with more iterations.

In addition, the results indicate that the training is robust to the choice of the augmentation coefficient. For *Circle* environment, any augmentation above 20% can approximately achieve the same level of performance. For *Cross* environment, any augmentation between 40% to 70% can attain the optimal value.

VI. Discussion

The numerical results presented above have practical implications on the development of vehicle automation.

- First and foremost, the results indicate that driving automation is a viable path to a more effective transportation. As shown in the numerical experiments, the addition of a RL augmentation controller improves the system performance by a large margin. The fact that the optimal augmentation is attained at 70% in *Cross* also indicates that there exists a set of constrained problems where human-machine collaboration is no worse than full machine control.
- Furthermore, we argue that such augmentation system is also cognitively safe. Because the augmentation term requires active participation of the human driver throughout the course of the driving, the human driver will always stay engaged in the control process. Should an emergence happens on the road that requires human control, the driver will be ready to react. Such human-in-the-loop control should demonstrate a shorter reaction time compared to a human-out-of-the-loop control system and thus will be more likely to avoid potential dangers.
- Moreover, augmented driving controller may also be modified to account for the differences of every individual driver. Because many RL algorithms, such as ES, actor-critic method, and policy gradient method, are capable of learning in online setting, we can continue improving the performance of the RL controllers after the deployment. Through interacting with the human driver in real-time, it may further be adapted to the characteristics of that human driver. For example, the controller may learn to provide more guidance to a risky driver but only exert minimum intervention to a skilled driver. The additional level of adaptation may allow for a new level of driving safety and efficiency.
- Lastly, the results also indicate a new direction in vehicle controller design. In this article, we augment a human driver with a RL agent. However, one can augment any controller with a RL controller. This sheds light on a new type of driving control, a system consisting of a model-driven baseline controller with a data-driven RL controller. In the hybrid system, the baseline controller should demonstrate provably safe behaviors given any perturbation up to the magnitude of the RL controller output; the RL agent will be trained through driving data and microscopic simulations to attain statistically significant improvement in efficiency. As the result, the combined system will be both fail-safe as guaranteed by the hand-designed baseline controller and highly efficient as enabled by the RL controller.

VII. Conclusions

In this study, we show that the state-of-the-art RL algorithm ES can be applied to augment human in the control of vehicle acceleration by taking directly the raw pixel inputs without any state estimation during the test time. In an end-to-end fashion, we train a convolutional neural network to improve a human driver's acceleration control. The results indicate that the augmented driver can operate 50% more effectively than without augmentation. Moreover, we find that the system where human and machine collaborate outperforms both purely human-operated and machine-operated systems. Such controller design is also less likely to induce cognitive inactivity, keeping the human driver more alert during driving, and thus reduces the likelihood of traffic accidents.

The results suggests many promising directions for future research. Researchers may investigate how to adapt the RL controller to learn from a specific human driver in an online settings. One can also attempt to couple such RL agent with a fail-safe manually designed controller such that the combined system is both verifiably safe and statistically high performant. To build from this research, one can find the source code at https://github.com/flow-project/flow.

VIII. Future Work

This section includes the extension to the main research topic presented above, where I delineate my recent progress on (1) the development of end-to-end pixel learning in a much more complex road network called the *University of Delaware Scaled Smart City* (UDSSC) and (2) my preliminary theoretical work on statistical verification of RL algorithms based on the principles of *randomized control trials* (RCT).

A. UDSSC Environment

Towards the first goal, I am working on scaling the proposed RL method from idealistic single-agent traffic such as *Circle* and *Cross* to more realistic multi-agent traffic. To this end, I am adapting the UDSSC environment for multiagent autonomous vehicle control.

The UDSSC environment is presented in Figure 6. The network, seen in Fig. 6, is a scaled implementation of a physical robotic navigation testbed at the University of Delaware. It consists of distinct elements of actual road network design, including: (1) roundabouts, (2) four-way intersections, (3) T-shaped intersections, (4) multilanes, and (5) merges.

The goal of RL agent is to control and coordinate a subset of vehicles in the traffic to increase the average velocity of all road segments without incurring traffic accident. In this environment, the number of agents can be greater than one but each one of them has the same observation space and action space as the single agent in previous study presented above. In Figure 6 we illustrate a simulated traffic flow on the road network. The objective of the RL algorithm is therefore to make the traffic flow as fast and stable as possible.

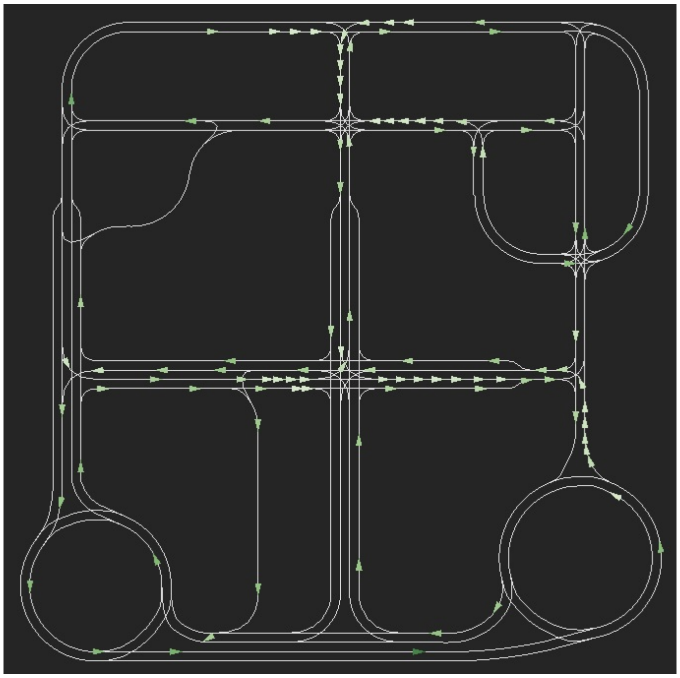


Fig. 6: UDSSC environment. Traffic instabilities such as queues and stop-and-go waves form primarily near the intersections and roundabouts of the network.

As a semi-realistic testbed for vechicle navigation and coordination in real traffic, this environment set possesses sufficient details for its trained agent to be applied to real world, yet still maintains computational tractability for it to be trained within reasonable amount of time.

B. RL Algorithm Verification

In this section, I discuss the most commonly used RL design verification method and propose my changes to it to make it robust to confounding factors. Specifically, this last section serves as a response to a problem that I encountered during literature review as detailed in [24]. In that study, the authors introduced two new algorithmic improvements to A3C algorithm. The first addition is the introduction of a goal-driven hierarchical structure similar to that of [25]. The second addition is a custom reward function. The authors were able to achieve better performance after adding these two features. Thus they (prematurely) concluded that both features were meaningful augmentations to the original A3C method [8]. Later they found out that it was only the first feature, the custom reward function, that was helping the performance. Consequently, their previous claim on the effectiveness of the hierarchical structure was rendered invalid.

Indeed, formal causality tests should be required in the proposal of new RL algorithms. In the following paragraphs, I first formalize the verification approach that was taken in [24], as it is also quite commonly used in the research community. Next, I show how one might avoid similar research accidents, i.e., the problem of confounding factors, through marginalization on potential confounders.

1) Popular Verification Procedure: I would like to point out the importance of experiment design in the establishment of causal relations. The authors of a new algorithm always need to show through experiments that their design *results in* better performance than a RL algorithm without such augmentations. However, between the current practice in the RL research community and the formal method of causal inference, I find there is a significant gap. In current RL literature, a new algorithm A' is often proposed based on the following verification procedure:

1) Test a sample of the most performant algorithms $\{A_1, A_2, \cdots, A_n\}$ on certain benchmark tasks $\{B_1, B_2, \cdots, B_m\}$ to produce a baseline score $E_{i,j}[s(A_i, B_j)]$.
2) Identify potential augmentations to an existing RL algorithm A_0. Denote the set of candidate algorithmic changes as $C = \{c_1, c_2, \cdots, c_l\}$. Denote the new RL algorithm produced from A_0 and a selection of candicate changes $\tilde{C}$ as $A_0 \bigoplus \tilde{C}$, where $\tilde{C} \subseteq C$. Note we usually do, but not necessarily require:
$$E_j[s(A_0, B_j)] > E_{i,j}[s(A_i, B_j)].$$
3) To verify the effectiveness of the algorithmic changes $\tilde{C}$, compute the *expected* performance of $A_0 \bigoplus \tilde{C}$ over $\{B_1, B_2, \cdots, B_m\}$. In other words, compute:
$$E_j[s(A_0 \bigoplus \tilde{C}, B_j)].$$
4) If the following condition holds:
$$E_{\tilde{N},j}[s(A_0 \bigoplus \tilde{C}, B_j)] > E_{i,j}[s(A_i, B_j)|\forall i, j],$$
then the algorithmic design $\tilde{C}$ is an valuable addition to the original algorithm A_0.

The above procedure essentially argues that addition of $\tilde{C}$ *causes* $A_0 \bigoplus \tilde{C}$ to outperformance the state of the art $E_{i,j}[s(A_i, B_j)]$ by showing that addition of $\tilde{C}$ to A_0 and the *observed* improvements are *correlated*.

However, as we know, *correlation does not imply causation*. Claiming causality based solely on observational study will put the new discoveries at the risk of fallacy. To avoid the trap, I propose to adopt tools from causal inference.

2) Proposed Verification Procedure: Among various techniques from the field causal inference, I present *a* way to verify RL algorithmic designs based on the principles of RCT. The proposed method is delineated below.

1) Test a sample of the most performant algorithms $\{A_1, A_2, \cdots, A_n\}$ on certain benchmark tasks $\{B_1, B_2, \cdots, B_m\}$ to produce a baseline score $E_{i,j}[s(A_i, B_j)]$.
2) Identify potential changes pertained to the augmentations of an existing RL algorithm A_0. Denote the set of candidate algorithmic changes as $C = \{c_1, c_2, \cdots, c_l\}$. Denote the new RL algorithm produced from A_0 and a selection of candicate changes $\tilde{C}$ as $A_0 \bigoplus \tilde{C}$, where $\tilde{C} \subseteq C$. Note we usually do, but not necessarily require:
$$E_j[s(A_0, B_j)] > E_{i,j}[s(A_i, B_j)].$$
3) To verify the effectiveness of the algorithmic changes $\tilde{C}$, compute the *expected* performance of $A_0 \bigoplus (\tilde{C} \bigcup \tilde{N})$ over $\{B_1, B_2, \cdots, B_m\}$ and $\tilde{N} \subseteq C \setminus \tilde{C}$, where $\tilde{N}$ is a set containing elements randomly selected from $C \setminus \tilde{C}$. In other words, compute:
$$E_{\tilde{N},j}[s(A_0 \bigoplus (\tilde{C} \bigcup \tilde{N}), B_j)].$$
4) If the following condition holds:
$$E_{\tilde{N},j}[s(A_0 \bigoplus (\tilde{C} \bigcup \tilde{N}), B_j)] > E_{i,j}[s(A_i, B_j)|\forall i, j],$$
then the algorithmic design $\tilde{C}$ is an valuable addition to the original algorithm A_0.

Unlike the previously mentioned technique, this approach carefully marginalizes out the effect of potential confounding variables in $\tilde{N}$ such that the demonstrated performance, if any, can be asserted, with greater certainty, to be *caused* by the proposed algorithmic design $\tilde{C}$.

Acknowledgement

Funding for this project was provided by UC Berkeley Safe Transportation and Research Education Center (SafeTREC) and the Collaborative Sciences Center for Road Safety (CSCRS), a U.S. Department of Transportation-funded National University Transportation Center led by the University of North Carolina at Chapel Hills Highway Safety Research Center.

References

[1] G. Staff and Agencies, "Tesla car that crashed and killed driver was running on autopilot, firm says," Mar 2018. [Online]. Available: https://www.theguardian.com/technology/2018/mar/31/tesla-car-crash-autopilot-mountain-view

[2] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[3] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[4] S. Ingle and M. Phute, "Tesla autopilot: semi autonomous driving, an uptick for future autonomy," *International Research Journal of Engineering and Technology*, vol. 3, no. 9, 2016.

[5] C. News, "First look inside self-driving taxis as waymo prepares to launch unprecedented service," Oct 2018. [Online]. Available: https://www.cbsnews.com/news/first-look-inside-waymo-self-driving-taxis/

[6] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2641–2646.

[7] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[8] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016, pp. 1928–1937.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[10] L. Fridman, B. Jenik, and J. Terwilliger, "Deeptraffic: Driving fast through dense traffic with deep reinforcement learning," *arXiv preprint arXiv:1801.02805*, 2018.

[11] R. E. Stern, S. Cui, M. L. Delle Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, H. Pohlmann, F. Wu, B. Piccoli, *et al.*, "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 205–221, 2018.

[12] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," *arXiv preprint arXiv:1710.05465*, 2017.

[13] E. Vinitsky, K. Aboudy, L. Le Flem, N. Kheterpal, K. Jang, F. Wu, R. Liaw, E. Liang, and A. Bayen, "Benchmarks for reinforcement learning inmixed-autonomy traffic," *Conference on Robot Learning*, 2018.

[14] E. Vinitsky, K. Parvate, A. R. Kreidieh, C. Wu, Z. Hu, and A. Bayen, "Lagrangian control through deep-rl: Applications to bottleneck decongestion," *IEEE Intelligent Transportation Systems Conference*, 2018.

[15] A. R. Kreidieh and A. Bayen, "Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning," *IEEE Intelligent Transportation Systems Conference*, 2018.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[17] J. Carmona, F. García, D. Martín, A. d. l. Escalera, and J. M. Armingol, "Data fusion for driver behaviour analysis," *Sensors*, vol. 15, no. 10, pp. 25 968–25 991, 2015.

[18] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility," in *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, vol. 42, 2011.

[19] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[20] M. Treiber and A. Kesting, "Traffic flow dynamics: data, models and simulation," *Physics Today*, vol. 67, no. 3, p. 54, 2014.

[21] A. Holkner, "Pyglet: Cross-platform windowing and multimedia library for python," *Google Code*, 2008.

[22] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stoica, "Rllib: Abstractions for distributed reinforcement learning," in *International Conference on Machine Learning*, 2018, pp. 3059–3068.

[23] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamical model of traffic congestion and numerical simulation," *Physical review E*, vol. 51, no. 2, p. 1035, 1995.

[24] N. Dilokthanakul, C. Kaplanis, N. Pawlowski, and M. Shanahan, "Feature control as intrinsic motivation for hierarchical reinforcement learning," *arXiv preprint arXiv:1705.06769*, 2017.

[25] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, "Feudal networks for hierarchical reinforcement learning," *arXiv preprint arXiv:1703.01161*, 2017.