Reinforcement Learning with Latent State Inference for Autonomous On-ramp Merging under Observation Delay

Amin Tabrizian[1], Zhitong Huang[2], and Peng Wei[1]

***Abstract*— This paper presents a novel approach to address the challenging problem of autonomous on-ramp merging, where a self-driving vehicle needs to seamlessly integrate into a flow of vehicles on a multi-lane highway. We introduce the Lane-keeping, Lane-changing with Latent-state Inference and Safety Controller (L3IS) agent, designed to perform the on-ramp merging task safely without the comprehensive knowledge about surrounding vehicles' intents or driving styles. We also present an augmentation of this agent called AL3IS that accounts for observation delays, allowing the agent to make more robust decisions in real-world environments with vehicle to vehicle (V2V) communication delays. By modeling the unobservable aspects of the environment through latent states, such as other drivers' intents, our approach enhances the agent's ability to adapt to dynamic traffic conditions, optimize merging maneuvers, and ensure safe interactions with other vehicles. We demonstrate the effectiveness of our method through extensive simulations generated from real traffic data and compare its performance with existing approaches. L3IS shows a 99.90% success rate in a challenging on-ramp merging case generated from the real US Highway 101 data. We further perform a sensitivity analysis on AL3IS to evaluate its robustness against varying observation delays, which demonstrates an acceptable performance of 93.84% success rate in 1-second V2V communication delay.**

I. INTRODUCTION

In the realm of modern transportation, the realization of autonomous driving systems has emerged as a promising avenue for enhancing road safety, traffic efficiency, and overall driving experience. One critical aspect of this advancement is the seamless integration of self-driving vehicles into complex and dynamic traffic scenarios, such as on-ramp merging to a busy highway. Highway on-ramp merging areas are considered bottlenecks ~~of traffic safety~~, because of their reduced capacity and frequent changing maneuvers by the vehicles in the traffic [1].

In a highway on-ramp merging scenario, to ensure a safe and efficient merging, the ego vehicle should make a sequence of real-time decisions on speed and lane changes based on surrounding vehicles' information and reasoning about their willingness to cooperate. Planning methods offer a crucial advantage in the context of reasoning about surrounding vehicles' information, as they enable the estimation of behavior models that are correlated with easily measurable states (e.g., velocity) [2].

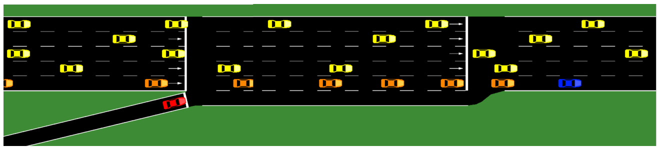

Fig. 1: Highway on-ramp merging scenario. Red: ego vehicle; Blue: cooperative vehicles; Orange: aggressive vehicles; Yellow: mainstream vehicles.

With the recent successes of reinforcement learning (RL) in board games [3] and Atari [4], researchers tried to apply RL-based algorithms to extended domains such as health care [5], industry automation [6], natural language processing [7], aviation [8], and autonomous driving [9]. In the case of on-ramp merging, numerous studies leverage RL techniques to ensure a safe and effective merging. An on-ramp merging optimization control framework based on a deep RL that optimizes lane keeping and lane-changing at the same time was proposed in [10]. The proposed approach had a shorter travel time and lower emergency braking rate compared to baseline methods. However, it should be noted that their framework could only accommodate a fixed number of surrounding vehicles, limiting its applicability to real-world scenarios. RL and model predictive control (MPC) were combined in [11] to leverage their strengths together. The authors claimed MPC solutions provide more robustness to out-of-distribution traffic patterns and RL techniques are better in terms of efficiency and passenger comfort. They considered a single-lane highway merging scenario with one default driving style for the surrounding vehicles. However, they did not address the inclusion of various driving styles for the surrounding vehicles or consider multi-lane highway scenarios.

Developing safe merging control algorithms was the aim of some other studies. A real-time bi-level control framework was proposed in [12] that ensures safety using control barrier functions (CBF). Combining learning-based control methods with CBF was not addressed in this work. CBFs always need a good approximation of the model of the system, which may not be applicable to all driving scenarios. A probabilistic CBF algorithm was proposed in [13] to account for model uncertainty. The proposed algorithm was tested based on a real dataset provided by the National Highway Traffic Safety Administration [14].

Instead of using direct vehicle-to-vehicle (V2V) communication, [15] focused utilizing realistic sensor data to obtain observations as input to its control framework. They also

[1] The George Washington University, {amin_tabrizian, pwei}@gwu.edu

[2] Saxton Transportation Operations Laboratory/Leidos, Inc, zhitong.huang@leidos.com

* This work is supported by National Science Foundation (NSF) Award # 2229885

employed a game-theoretic reasoning based on Monte Carlo RL in order to find a near-optimal policy. A passive actor-critic (pAC) technique was suggested in [16] for selecting a candidate spot for merging then used a multi-policy decision-making method to merge to candidate spots. They then validated their proposed framework on real-world data with a success rate of 92%.

Latent state inference plays an important role in autonomous driving. Knowing surrounding vehicles' intent to yield or not is crucial for ego vehicle's planning and control. The drivers' behavior or intention on other vehicles can be estimated by reasoning their underlying driving styles. A supervised learning approach for inferring the latent states of other vehicles in a T-intersection was proposed in [17]. They also adopted graph neural networks to model relational information of the neighboring vehicles. However, they assumed that the ego vehicle has access to the latent states of the other traffic participants during training which may not be a realistic assumption.

In this work, we propose Lane-keeping, Lane-changing with Latent-state Inference, and Safety Controller (L3IS) agent to fill the technical gaps mentioned above. The L3IS agent helps the ego vehicle make both acceleration and lane-changing decisions in highway on-ramp merging scenarios. It consists of multiple components to improve driving safety in practical scenarios. These components are as follows.

1) **Lane-keeping (LK) agent**: A proximal policy optimization (PPO) based deep RL agent for making acceleration changes.
2) **Lane-changing (LC) agent**: A deep Q-network (DQN) based agent for performing lane changes.
3) **Supervised-learning (SL) agent**: This agent is designed to predict the driving style of the surrounding vehicles.
4) **Safety controller**: To ensure the safety of the above learning-based controllers, a safety controller is deployed with a one-step look-ahead of the LC and LK agent actions.

In contrast to [17], we do not assume the LK and LC agents have access to the surrounding vehicles' driving styles during training. Finally, the augmented L3IS (AL3IS) agent is proposed as an extension of L3IS, which is shown to be effective when there exist V2V communication delays.

The main contributions of our work are:

- A principled way of learning and planning is proposed for performing lane-keeping and lane-changing actions that is capable of dealing with a variable number of surrounding vehicles.
- A systematic approach to addressing V2V communication delays is proposed using augmented states.
- Extensive simulations generated from real-world traffic data are constructed to evaluate the performance of L3IS and AL3IS agents against state-of-the-art baselines.

II. Preliminaries

A. Partially Observable Markov Decision Process

In a Markov decision process (MDP), an agent will choose an action a_t at time step t based on its current state s_t. Then, it will receive the reward of r_t, and the environment evolves to the state s_{t+1} probabilistically [18]. This process can be represented by a tuple of (S, A, T, R, γ), where S is the state space; A is the action space; T: $S \times A \times S \to \mathbb{R}$ is the transition model; $R : S \times A \to \mathbb{R}$ is the reward model; and γ is the discount factor.

When an agent state is uncertain, partially observable MDP (POMDP) which is an extension of MDP can be used for modeling. In this formulation an additional mapping function $\Omega : O \to s$ that maps an observation to a state will be needed [19]. The objective of this problem is to find a mapping from an observation to a probability distribution over actions $\pi : O \to \mathcal{P}(A)$ that maximizes the expected return:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \, \mathbb{E} \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \tag{1}$$

B. Policy Improvement

In RL, several methods have been proposed for policy improvement through interaction with an environment [20]. In this study, we will use Deep Q-network [21] and Proximal Policy Optimization [22] for making acceleration and lane change decisions respectively.

1) Deep Q-network (DQN): The objective of DQN is to iteratively update the Q-function using the Bellman equation:

$$Q(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') \right) \tag{2}$$

Here, $Q(s, a)$ denotes the current estimate of the Q-value for state s and action a, α is the learning rate, and r is the immediate reward. The Q-network is trained to minimize the loss function:

$$L(\theta) = \mathbb{E} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right] \tag{3}$$

Here, θ represents the neural network parameters, and θ^- represents the parameters of a target network, periodically updated with the current Q-network parameters. The expectation is taken over a minibatch of transitions (s, a, r, s') sampled from a replay buffer.

2) Proximal Policy Optimization (PPO): This algorithm introduces a clipped surrogate objective function to ensure limited policy updates. The objective function, denoted as $L(\theta)$, is defined as:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t [\min(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t],\tag{4}$$

where epsilon is a hyperparameter, $r_t(\theta)$ represents the ratio of the new policy to the old policy $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$, $\hat{A}_t$ is the advantage estimate at time t. The motivation for this objective function is to limit the policy update amount at each time step. To ensure proper state space exploration, a noise process is added to the actor's chosen action:

$$\mu'(s_t) = \mu(s_t | \theta_t^\mu) + \mathcal{N} \tag{5}$$

In all of the aforementioned equations s_t will be estimated using Ω mapping function in a POMDP setting.

III. Method

A. Latent State Inference with Supervised Learning Agent

As discussed in the introduction section, assuming that the ego vehicle has access to the surrounding vehicles' driving style is impractical. However, it is possible to leverage a supervised learning (SL) approach to infer the driving intent based on state information shared by V2V communication. Here, the task is to estimate $P(z_t^i|o_i)$ where z_t^i is the latent state (driving style or intent) of the driver of vehicle i at time step t [17].

For training the SL agent, all vehicle trajectories will be generated by performing a fixed number of simulations with true labels. In these simulations, to ensure the diversity of the generated training data, the ego vehicle employs a random policy. Then a multi-layer perception (MLP) classifier will be trained to minimize the following negative log-likelihood loss function.

$$L(\theta) = \mathbb{E}_{z_t^i, o_t}[\log P_\theta(z_t^i)|o_t] \tag{6}$$

The learning framework for the SL agent is represented in Fig 2.

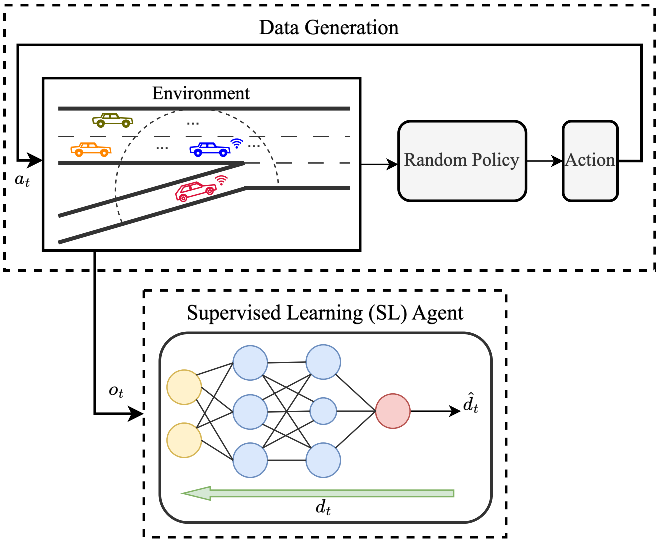

Fig. 2: The supervised learning (SL) agent training framework. A random policy will interact with the environment to generate training data. Here, d_t is the true driving style or latent state of the surrounding vehicles. Then, a classifier will be trained using the generated data to estimate surrounding vehicles' driving style $\hat{d}_t$.

B. RL with Augmented MDP

Dedicated short range communication (DSRC) ~~is a~~ V2V protocol used in autonomous vehicles for transmitting data [23]. This type of communication will help the vehicles to have access to information that is not easily computable by using onboard sensors. For the on-ramp merging scenario, vehicles already in the mainstream can communicate with the vehicles entering the highway using the V2V communication protocol. However, the assumption of perfectness for this type of communication is impractical for real-world scenarios. The effects of V2V communication delay are studied in [24]. They claimed that in the worst case of 50% communication loss, 5 out of 10 messages will be received which will result in an average delay of 300ms.

Regardless of the amount of delay, we can consider this as a learning problem in environments with constant delayed feedback. This problem is represented as a constant delayed MDP (CDMDP) with a 6-tuple (S, A, T, R, γ, k), where k is a non-negative integer that describes the number of time-step differences between when an agent reaches a state and the time it receives its information [25].

To further facilitate the learning process, it is possible to leverage the action history for estimating the agent's current state and then using this estimated state for developing a policy. By this approach, a policy can be formed by using $I_k \in S \times A^k$, the last observed state, and the k following actions [26].

For solving CDMPD, we will use an augmented MDP, in which a planning algorithm will be found for a larger state $S \times A^k$. The goal of this approach is to find optimal state values for this larger state space $(V^*_{I_k})$ [27]. This solution can tackle the packet loss problem in V2V communication which causes observation delay. An augmented state will be constructed that consists of the state of the agent in the past k time steps s_{t-k} and the k following actions $a_{t-k:t}$. For constructing s_{t-k}, the trained SL agent will be used for the estimation of the latent state.

In this study, we are dealing with constant delayed POMDP (CDPOMDP) as the driving styles of surrounding vehicles are latent. Most of the solution approach stays the same as CDMDP except we have to estimate the agent's last observed state $\hat{S}$ and then augment it with the following actions. The framework of CDMDP policy optimization is represented in Fig. 3. Unlike [17], we will not assume the ego vehicle has access to the latent states during training time. This assumption lacks practicality in real-world scenarios, where the driving styles of surrounding vehicles are consistently unknown to the ego vehicle.

IV. PROBLEM FORMULATION AND SOLUTION

The on-ramp merging scenario is framed as a POMDP which is then extended to a CDPOMDP for the AL3IS agent. The highway consists of a five-lane mainstream and an on-ramp. The surrounding vehicles on the mainstream have two different "aggressive" and "cooperative" driving styles, and all of them are run by Intelligent Driver Model (IDM) [28] provided by the SUMO simulator [29]. The vehicle that is being merged is under our control and is called the ego vehicle. A schematic of this scenario outlining the overall architecture is shown in Fig. 4.

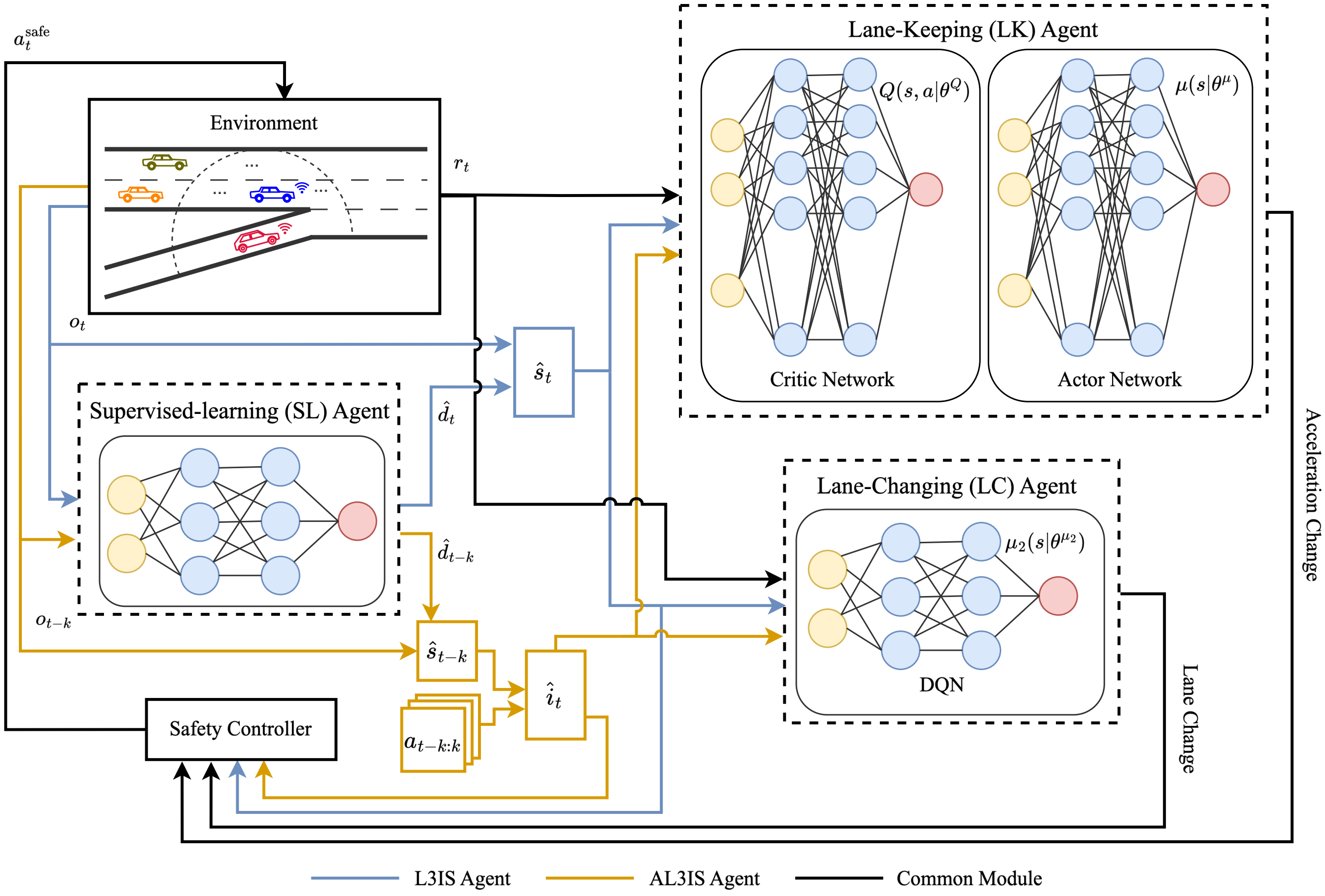

Fig. 3: Policy optimization framework. Common modules in black are shared by both L3IS and AL3IS agents. L3IS: Given an observation from the environment, SL agent will estimate the surrounding vehicles' driving styles. Then, the estimated latent state will be constructed and used as an input for both LK and LC agents. Finally, the agent's actions will be monitored and corrected if needed by the safety controller module. AL3IS: Delayed observation and estimated driving style will be augmented with the following actions and will adhere to the same process as the L3IS agent.

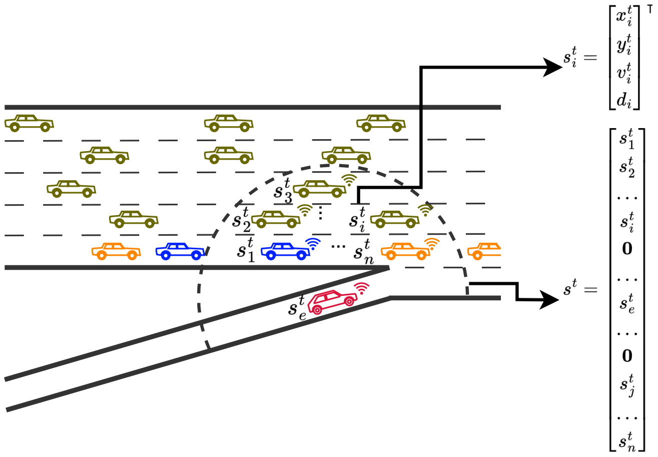
$s_i^t = \begin{bmatrix} x_i^t \\ y_i^t \\ v_i^t \\ d_i \end{bmatrix}^\intercal$
$s^t = \begin{bmatrix} s_1^t \\ s_2^t \\ \dots \\ s_i^t \\ \mathbf{0} \\ \dots \\ s_e^t \\ \dots \\ \mathbf{0} \\ s_j^t \\ \dots \\ s_n^t \end{bmatrix}$
s_1^t
s_2^t
s_3^t
s_i^t
s_n^t
s_e^t

Fig. 4: Highway merging scenario. The ego vehicle has access to the information of its surrounding vehicles (except for their driving style) within a certain radius centered at the ego.

A. POMDP Formulation

1) *State*: The environment state consists of the position and velocity of the ego vehicle and all of its surrounding vehicles within a certain radius centered at the ego vehicle.

$$s = (s^1, s^2, ..., s^i, 0, ..., s^e, ..., 0, s^j, ..., s^n), \qquad (7)$$

where $s^e = (x^e, y^e, v^e)$ is the state of the ego vehicle, and $s^i = (x^i, y^i, v^i, d)$ the i-th surrounding vehicles in which d represents the driving style. All surrounding vehicles' states are sorted firstly based on their lane number and then by their distance from the ego vehicle. The states of the vehicles behind the ego $(s^1, s^2, ..., s^i, 0, ...)$ will be sorted based on the mentioned criteria, zero-padded, and then put behind s_e. The same process is done for vehicles in front of the ego. Zero-padding is done to deal with a variable number of vehicles as demonstrated in Fig. 4.

2) *Action*: The environment is discretized into time steps of length Δt. In each time step, the ego vehicle will choose an acceleration between $(-a_{\min}, a_{\max})$ m/s^2

and whether to perform lane change or not.

3) *Transition Model*: After an action is taken, all of the vehicles' positions and velocities will be updated by the SUMO simulator.

4) *Reward Model*: To ensure safety, smoothness, and fuel efficiency, penalties are added for unsafe lane changes, speed changes, and collisions. The reward function for the LK agent is:

$$r_{lk} = \alpha_{lk} r_{sc} + \beta_{lk} r_c + \gamma_{lk} r_a, \tag{8}$$

where α_{lk}, β_{lk}, and γ_{lk} are hyperparamters. The rewards r_{sc}, r_c, and r_a represent the penalties for speed change, collision, and the reward of arriving at the end of the highway respectfully which are defined as follows.

$$r_{sc} = -|a_t|, \tag{9}$$

$$r_c = -p, \tag{10}$$

$$r_a = q. \tag{11}$$

The reward function for the LC agent only considers safe maneuvers.

$$r_{lc} = \alpha_{lc} r'_c + \beta_{lc} r'_a, \tag{12}$$

where

$$r'_c = -p', \tag{13}$$

$$r'_a = q'. \tag{14}$$

A feedback of the reward functions for LK and LC agents will be used to optimize their policies.

5) *Observation Model*: For the L3IS agent, the observation model is the same as the state model. However, as discussed in the previous sections, it is impractical to assume the information between vehicles will be transmitted instantly and without any delay. Therefore, for the AL3IS agent, a constant delayed observation model is assumed:

$$o_t = s_{t-k}, \tag{15}$$

where k denotes the observation delay.

B. Solution Algorithms and Network Architecture

The LK agent utilizes a PPO framework for its operations. Its actor and critic networks each comprise two hidden layers, with sizes of 128 and 64, respectively. Given the discrete nature of the LC agent's action space, a straightforward DQN serves its purpose effectively. This DQN features a single hidden layer with a size of 128.

C. Safety Controller Mechanism

Although deep RL shows a great performance in solving complex high dimensional problems, it is not sufficient to only rely on it for achieving safety [30]. We built a higher-level safety controller that monitors and corrects both LK and LC actions based on a one-step look-ahead kinematics.

1) LK Action Constraint: For ensuring safe acceleration changes, the ego and its target vehicle should maintain a safe distance. A constant speed is assumed for calculating the target vehicle's position and the ego vehicle's acceleration is known. Then, the acceleration will be corrected by the following formula:

$$a_e^{\text{safe}} = \begin{cases} a_{\text{min}} & \text{if } x'_e - x'_t \leq d_{\text{safe}} \\ a_e & \text{otherwise.,} \end{cases} \tag{16}$$

where

$$x'_t = v_t + x_t \tag{17}$$

$$x'_e = \frac{1}{2} a_e + v_e + x_e \tag{18}$$

Here, d_{safe} is the minimum safe distance, t and e are the subscripts for the target and ego vehicle respectively.

2) LC Action Constraint: To ensure safety in LC decisions, the same procedure is done for monitoring LC actions. The safety controller will judge the safety of LC actions and will terminate the action if it leads to unsafe states. By termination action, the ego vehicle will stay in the auxiliary lane for the next time step. The safety constraints are the same as LK except here both rear and front vehicles should have a safe distance from the ego after lane-changing. If lane-changing action is suppressed, the safety controller will change acceleration to a_{min} to allow the ego vehicle to have the desired safe distance.

V. Experiments

A. Environment Setup

All of the simulations are performed in the SUMO simulator. The simulation parameters are listed in Table. I. For the mainstream, vehicle flow rates and their desired maximum speed are obtained from the NGSIM US Highway 101 dataset [31]. The minimum headway (τ) of the aggressive and cooperative drivers is sampled from a uniform distribution between $(0.1, 0.7)$ and $(0.6, 0.8)$ respectively. All of the safety regulations for aggressive drivers are turned off in the SUMO setting to ensure their risky driving style. It should be emphasized that the safety regulations for the ego vehicle are also suppressed as our algorithm is completely in charge of controlling its actions. Each episode starts with inserting the surrounding vehicles with an aggressiveness probability of $P_{\text{cooperative}} = 0.5$. After 20 seconds the ego vehicle will be inserted. An episode will be finished if the ego vehicle exits the highway, encounters a collision, or the maximum allowed time steps have been passed.

B. Supervised Learning (SL) Agent Training

For training the SL agent, an agent with a random policy interacts with the environment for 1,000 episodes and the data for observations and true latent states is gathered using the framework described in Section III-A. The reason for using a random policy in data generation was to avoid generating biased data for training.

After data generation, a neural network is trained for more than 5,000 epochs to estimate the driving style of the vars

TABLE I: Experimental parameters

Vehicular Parameters	
Vehicle type	passenger
Car following model	IDM
Length	5 m
Max speed	aggressive: $(10, 13)\ \mathrm{m\,s^{-1}}$ cooperative: $(8, 11)\ \mathrm{m\,s^{-1}}$ second lane: $12.21\ \mathrm{m\,s^{-1}}$
Min gap	$2.5\ \mathrm{m\,s^{-1}}$
τ	aggressive: (0.1, 0.7) s cooperative: (0.6, 1.8) s second lane: 1 s
a_{max} accel	$2.6\ \mathrm{m/s^2}$
b_{max} decel	$4.5\ \mathrm{m/s^2}$
b_e emergency decel	$9\ \mathrm{m/s^2}$
Simulation Paramters	
Traffic demand	1st lane: 1,512 veh/hr 2nd lane: 1,692 veh/hr 3rd lane: 1,656 veh/hr 4th lane: 1,584 veh/hr 5th lane: 1,656 veh/hr
Main lane length	150 m
Merge lane length	80 m
Simulation time step	0.1 s

given the true targets. The test accuracy of the trained model is 96.34%. The learning curve of the SL agent is represented in Fig. 5.

C. L3IS Agent Training and Analysis

After training the SL agent, the driving style of the surrounding vehicles will be inferred using the trained agent. Apart from the SL agent, a safety controller is added to the L3IS agent to ensure safe maneuvers both for lane-keeping and lane-changing. For the baseline, we used a combination of LK and LC agents without latent state inference and safety controller. We also did an ablation study to evaluate the importance of the SL agent and the safety controller. All agents were trained for 360,000 steps and then tested for 1,000 episodes. The training and test results are represented in Fig. 6 and Table. II. It is apparent that L3IS significantly outperforms the baseline approach. It deserves mentioning that its collision rate is 0% and the reason for an incomplete success rate is the agent did not merge to the mainstream in a few episodes within the maximum allowed time steps.

TABLE II: Success rate and average reward analysis of L3IS.

Algorithm	Success rate	Average reward
Baseline	$90.88 \pm 0.59\%$	16.46 ± 0.29
L3IS	$\mathbf{99.90 \pm 0.17\%}$	$\mathbf{18.58 \pm 0.16}$

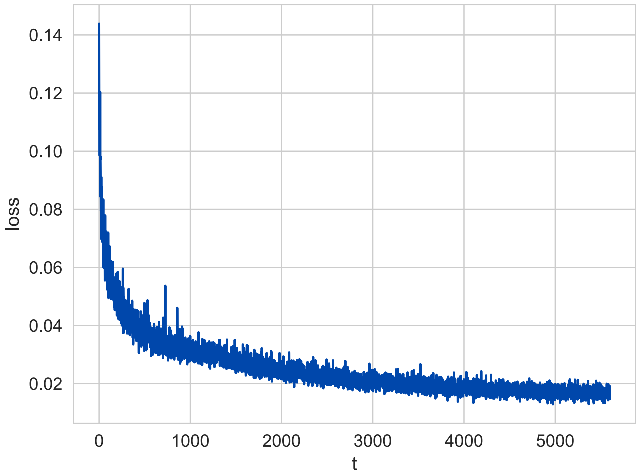

Fig. 5: Supervised Learning (SL) agent training loss.

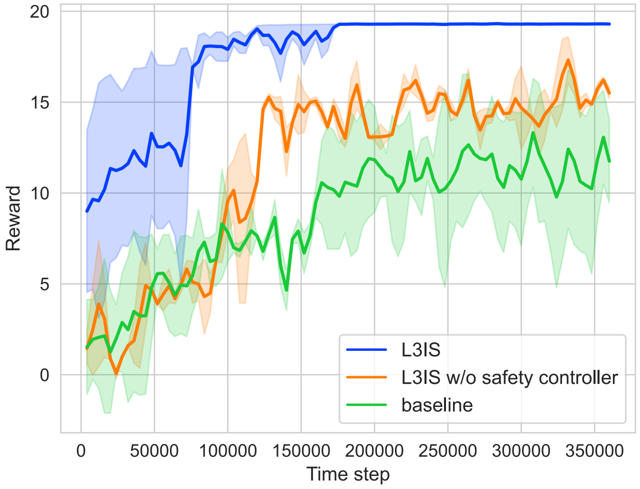

Fig. 6: Performance of L3IS with and without safety controller against baseline.

D. Observation Delay and Sensitivity Analysis

To evaluate the performance of the AL3IS agent, we performed several simulations with varying observation delays. The delays range from $1, 2, 3, 10$, and 15 seconds. The agent is trained under the same circumstances as L3IS. The results are represented in Fig. 7 and Table. III. As we can see, the collision rate increases significantly in the delays of more than 1 second. We believe the main reason for this is the safety controller no longer has access to the most recent observation and it does not have any mechanism for dealing with this situation. So it will lose its functionality under high delay amounts. Handling observation delays in the safety controller is left for future work. It is worth noting that the delay amount is $300ms$ in the worst case in which AL3IS shows an acceptable performance with only a 6.06% less success rate on average compared to L3IS with instant observation. A back-of-envelope calculation offers further insight into the challenge posed by observation delays in dense traffic scenarios. For instance, based on our analysis of US Highway 101 data, where the average car speed is $12\,\mathrm{m\,s^{-1}}$, a 1-second delay results in a positional change of 12 meters for each vehicle. This means that for a t-second delay, the positional shift becomes $12t$ meters, presenting significant hurdles for collision avoidance algorithms, particularly in dense traffic environments. To gain deeper insight into the significance of AL3IS, we conducted identical simulations employing the L3IS agent. The optimal performance of the L3IS agent is under the 2-second delay scenario, which is notably inferior to that of the AL3IS agent.

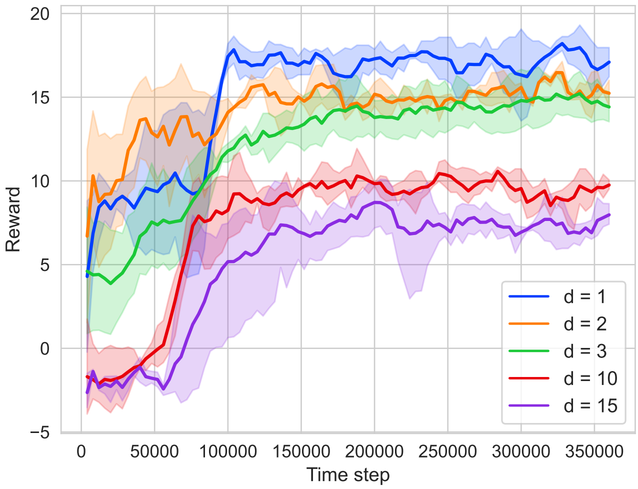

Fig. 7: AL3IS agent observation delay sensitivity analysis.

VI. Conclusion

In this paper, we proposed the L3IS agent for autonomous on-ramp merging in highway traffic scenarios. We demonstrated that the incorporation of latent state inferring and safety controllers significantly improves the success rate and overall performance of the merging maneuver. By using a combination of RL and supervised learning, our agent can accurately predict the driving style of surrounding vehicles, enabling it to make informed decisions even in the absence of complete information. Additionally, we extended the L3IS agent to account for observation delays, presenting the AL3IS agent. Our experiments showed that AL3IS maintains a high success rate even with delays of up to 1 second which was more than triple the amount of a worse-case delay in practical scenarios showcasing its robustness in real-world scenarios where communication latency is inevitable.

Overall, our work contributes to the advancement of autonomous driving systems in a connected vehicle setting, particularly in highway merging scenarios, by providing a principled approach that enhances safety, efficiency, and adaptability. Future research could focus on further improving the safety controller to handle communication packet loss and exploring additional factors such as adversarial driving intents and unseen driving styles to enhance the agent's robustness and generalization capabilities. Additionally, integrating real-world experimentation and validation would be essential steps toward deploying these techniques in practical autonomous vehicles.

References

[1] I. A. Ntousakis, I. K. Nikolos, and M. Papageorgiou, "Optimal vehicle trajectory planning in the context of cooperative merging on highways," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 464–488, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X16301413

[2] Z. Sunberg, C. Ho, and M. J. Kochenderfer, "The value of inferring the internal state of traffic participants for autonomous freeway driving," *2017 American Control Conference (ACC)*, pp. 3004–3010, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:6939535

[3] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[4] 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.

[5] C. Yu, J. Liu, S. Nemati, and G. Yin, "Reinforcement learning in healthcare: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–36, 2021.

[6] R. Nian, J. Liu, and B. Huang, "A review on reinforcement learning: Introduction and applications in industrial process control," *Computers & Chemical Engineering*, vol. 139, p. 106886, 2020.

[7] A. R. Sharma and P. Kaushik, "Literature survey of statistical, deep and reinforcement learning in natural language processing," in *2017 International conference on computing, communication and automation (ICCCA)*. IEEE, 2017, pp. 350–354.

[8] P. Razzaghi, A. Tabrizian, W. Guo, S. Chen, A. Taye, E. Thompson, A. Bregeon, A. Baheri, and P. Wei, "A survey on reinforcement learning in aviation applications," 2022.

[9] P. Wang and C.-Y. Chan, "Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[10] J. Zhao, T. Qu, and F. Xu, "A deep reinforcement learning approach for autonomous highway driving," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 542–546, 2020, 3rd IFAC Workshop on Cyber-Physical & Human Systems CPHS 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240589632100272X

[11] J. Lubars, H. Gupta, S. Chinchali, L. Li, A. Raja, R. Srikant, and X. Wu, "Combining reinforcement learning with model predictive control for on-ramp merging," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 942–947.

[12] Y. Lyu, W. Luo, and J. M. Dolan, "Probabilistic safety-assured adaptive merging control for autonomous vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 10 764–10 770.

[13] S. Udatha, Y. Lyu, and J. Dolan, "Reinforcement learning with probabilistically safe control barrier functions for ramp merging," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5625–5630.

[14] U.S. Department of Transportation, "Evaluation of adaptive cruise control interface requirements on the national advanced driving simulator," *in NHTSA's Vehicle Safety Research*, 2015.

[15] M. Garzón and A. Spalanzani, "Game theoretic decision making based on real sensor data for autonomous vehicles' maneuvers in high traffic," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 5378–5384.

[16] T. Nishi, P. Doshi, and D. Prokhorov, "Merging in congested freeway traffic using multipolicy decision making and passive actor-critic learning," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 287–297, 2019.

[17] X. Ma, J. Li, M. J. Kochenderfer, D. Isele, and K. Fujimura, "Reinforcement learning for autonomous driving with latent state inference and spatial-temporal relationships," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6064–6071.

[18] M. J. Kochenderfer, *Decision making under uncertainty: theory and application*. MIT press, 2015.

TABLE III: Performance evaluation of AL3IS and L3IS across varying time delays. Within the table, entries shaded represent the best performance in a particular delay, while text in **bold** signifies the highest performance of an agent.

	AL3IS		**L3IS**	
Delay $[s]$	Success Rate	Avg. Reward	Success Rate	Avg. Reward
1	$\mathbf{93.84 \pm 4.43\%}$	$\mathbf{17.39 \pm 0.83}$	$78.94 \pm 4.64\%$	14.40 ± 0.66
2	$86.36 \pm 1.86\%$	$\mathbf{15.87 \pm 1.86}$	$\mathbf{82.17 \pm 9.21\%}$	$\mathbf{16.68 \pm 0.35}$
3	$81.26 \pm 2.79\%$	14.69 ± 0.90	$79.19 \pm 0.71\%$	14.67 ± 0.21
10	$61.93 \pm 8.30\%$	10.72 ± 1.87	$64.39 \pm 1.93\%$	11.02 ± 0.50
15	$59.56 \pm 3.99\%$	9.26 ± 1.66	$56.97 \pm 4.00\%$	8.49 ± 1.46

[19] X. Ma, J. Li, M. J. Kochenderfer, D. Isele, and K. Fujimura, "Reinforcement learning for autonomous driving with latent state inference and spatial-temporal relationships," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6064–6071.

[20] I. Grondman, L. Busoniu, G. A. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1291–1307, 2012.

[21] 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, pp. 529–533, 2015.

[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[23] M. N. Ahangar, Q. Z. Ahmed, F. A. Khan, and M. Hafeez, "A survey of autonomous vehicles: Enabling communication technologies and challenges," *Sensors*, vol. 21, no. 3, p. 706, 2021.

[24] F.-C. Chou, S. E. Shladover, and G. Bansal, "Coordinated merge control based on v2v communication," in *2016 IEEE Vehicular Networking Conference (VNC)*, 2016, pp. 1–8.

[25] T. J. Walsh, A. Nouri, L. Li, and M. L. Littman, "Planning and learning in environments with delayed feedback," in *Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007. Proceedings 18.* Springer, 2007, pp. 442–453.

[26] D. Bertsekas, *Dynamic programming and optimal control: Volume I.* Athena scientific, 2012, vol. 4.

[27] J. L. Bander and C. C. White, "Markov decision processes with noise-corrupted and delayed state observations," *The Journal of the Operational Research Society*, vol. 50, no. 6, pp. 660–668, 1999. [Online]. Available: http://www.jstor.org/stable/3010623

[28] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, aug 2000. [Online]. Available: https://doi.org/10.1103%2Fphysreve.62.1805

[29] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility: an overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation.* ThinkMind, 2011.

[30] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Cooperation-aware reinforcement learning for merging in dense traffic," *CoRR*, vol. abs/1906.11021, 2019. [Online]. Available: http://arxiv.org/abs/1906.11021

[31] U.S. Department of Transportation, Federal Highway Administration, "Next generation simulation (ngsim) vehicle trajectories and supporting data," 2016, accessed 2023-08.