Intent-Aware Autonomous Driving: A Case Study on Highway Merging Scenarios

Nishtha Mahajan[1] and Qi Zhang[2]

***Abstract*— In this work, we use the communication of intent as a means to facilitate cooperation between autonomous vehicle agents. Generally speaking, intents can be any reliable information about its future behavior that a vehicle communicates with another vehicle. We implement this as an intent-sharing task atop the merging environment in the simulator of highway-env, which provides a collection of environments for learning decision-making strategies for autonomous vehicles. Under a simple setting between two agents, we carefully investigate how intent-sharing can aid the receiving vehicle in adjusting its behavior in highway merging scenarios.**

I. Introduction

Autonomous vehicles (AVs) hold the promise of improving the driving of not only individual vehicles but also those with whom they interact. Of special interest is AVs' ability to communicate with both infrastructure and other AVs that allows them to act in ways that human drivers cannot. This work seeks to leverage the communication capability of AVs to facilitate cooperation among them. Specifically, we focus on investigating how by sharing some information about their future behavior, which we formally call *intent*, AVs can influence each other's behavior positively.

A. What is Intent-Sharing among AVs and Why?

As per SAE (Society of Automotive Engineers) Standard on Cooperative Driving Automation [1], intent-sharing is a type of cooperative driving automation facilitated by machine to machine communication. Here, an intent is defined to be some information about the sending entity's future actions that it shares to aid the decision-making of the receiving entity. Intent-sharing is different from status-sharing in the sense that the status only contains information about what the sending entity observes at present, while the intent contains information about the future. The Standard also notes that the receiving entity does not have to accept the information from the sender. The sender executes its planned trajectory irrespective of how the receiver responds. It is different from other forms of cooperation where interaction between entities may be required to arrive at an agreed cooperation plan.

Although the receiver might have limited influence on the sender's intent choice, the receiver gains better prediction of the sender's prolonged future trajectory and therefore, by planning accordingly, can potentially achieve cooperative driving behavior that is beneficial for both entities. As an example that we will investigate carefully in this work, consider a scenario where a vehicle aims to merge into a highway from the entrance ramp, where there is another incoming vehicle on the highway that can potentially block the entrance. The incoming vehicle can either stay at the same speed so that the merging vehicle has to yield, or change its lane or speed or both so that the merging vehicle can safely enter without hesitation. Either way, for better cooperative driving, it is crucial for the merging vehicle to know the intent of the incoming vehicle.

B. How to Implement Intent-Aware Autonomous Driving?

It is non-trivial to implement the aforementioned concept, what we call intent-aware autonomous driving (AD), for effective cooperative driving, as the following three questions have to be addressed. Q1: How to computationally define and represent intents for AVs? Q2: Given a specific intent, how does its sender comply with the intent (i.e., plan and execute a trajectory consistent with the intent), and how does its receiver utilize the intent (e.g., whether or not to wait at the entrance ramp)? Q3: How to choose an intent to induce the optimal cooperative driving behavior (e.g., to yield to the merging vehicle, or not)?

Here, Q1 aims to provide a computational framework as the foundation for intent-aware AD, whereas Q2 and Q3 are coupled together to provide an algorithmic solution under the framework.

C. Our Contributions

In this work, we view AVs as autonomous decision-making agents and formulate a decision-theoretic framework for intent-aware AD (Section III). As a case study, we implement intent-aware AD for highway merging scenarios in the simulator of highway-env [2] (Section IV). Our implementation answers Q1 and Q2: we design a representation of intents for the highway-env scenarios that facilities the intent-sender to comply with its intent, while the intent-receiver adopts a reinforcement learning (RL) approach to utilize the intent. Our results demonstrate that our implementations achieve more reliable and more efficient driving behavior (Section VI). Finally, we leave Q3 as our future work, along with several other future directions that can be built on our intent-aware AD framework (Section VII).

II. Related Work

A. Intent and Intent-Sharing

Past research has focused both on *inferring* the intent of other agents [3], [4] and *explicitly communicating* an agent's

[1]Nishtha Mahajan is an independent researcher, work done at the University of South Carolina nmahaja@alumni.purdue.edu

[2]Qi Zhang is with The Artificial Intelligence Institute, University of South Carolina, Columbia, SC 29208, USA qz5@cse.sc.edu

MAD-Games: Multi-Agent Dynamic Games Workshop, IROS 2023

own intent [5], [6]. While inferred intent is usually an agent's goal or plan recognized based on the history of its activity, shared intent is future-directed that the agent is yet to reveal through its actions. For instance, Qi and Zhu [4] use agents' goal locations to represent intents, which are inferred based on an observation history. On the other hand, Kim et al. [6] communicate intentions between agents by generating each agent's imagined trajectory and encoding them as intention messages using an attention mechanism. We are interested in the latter case of explicit communication, which, as we will show, allows the intent-receiver agent to adapt its behavior to the communicated intent right from the start.

In the AD space, intent-sharing has been shown to allow AVs to undertake less conservative but safe driving maneuvers [7]. To this end, Wang et al. use reachability theory and conflict charts to demonstrate the potential of sharing velocity and acceleration bounds over a given time horizon to prevent conflicts during lane changes. Beyond AV-AV settings, Matthews et al. [8] choose between the decisions to display and not display the intention of an AV to pedestrians, and show both through in-field experiments and simulations that intent-communication has a positive impact on the behavior of the receiving entity.

The decision in intent-sharing is mostly considered to be the choice between whether to share or not share one's intent. Compliance with a given intent is not necessarily considered. On the contrary, we seek to treat intent as a decision variable that the intent-sender agent has to ensure compliance with.

B. RL for Highway Merging

Highway merging is considered a challenging traffic scenario for AD, owing to the complexity of interactions between vehicles. Several variations of this problem have been considered in literature where behavior of AVs either on the entrance ramp and/or the highway is learned using RL.

For instance, Tang [9] use RL to learn merging policies for AVs using a technique called self-play, wherein they iteratively update the set of AV agents in their training using previously learned policies. While their method achieves a high merge-success rate, it is not collision-free and they state that unobservability of intentions, among other reasons, might be a contributing factor. On the other hand, Liu et al. [10] focus on safety and equip their RL method with a motion predictive safety controller to reject unsafe actions that their RL policy might learn for an AV to merge into a stream of simulated human-driven vehicles. In contrast, Triest et al. [11] use real human-driving data for merge scenarios and train an RL-based merging vehicle to choose target vehicles it should keep distances with, in order to merge successfully at a fixed merge-point. They find that RL performs better with high-level actions such as theirs as compared to low-level acceleration actions in such cases.

Cooperative AD has also been explored for merging scenarios. Bouton et al. [12] achieve this by first assigning a cooperation level to human-driven vehicles on the highway. Then they use a belief-state RL method to make the merging AV merge at a fixed point, by inferring the cooperation level of the vehicles it observes. Contrarily, Toghi et al. [13] use multi-agent RL to determine optimal driving behavior of AVs on the highway such that a merging vehicle can successfully merge. They use the concept of social value orientation to control the degree of altruism of AVs.

III. Formulation of Intent-Aware AD

As a starting point, we focus on the problem of intent-aware AD in the setting of two-AV, one-way intent-sharing between an intent-sender AV (indexed by 2) and an intent-receiver AV (indexed by 1). We view the AVs as decision-making agents and formulate the problem with the notion of two-agent dynamic game. Let $\mathbf{s}_t$ denote the world state at (discrete) time step t, which we assume can be factored as $\mathbf{s}_t = [\mathbf{s}_t^e, \mathbf{s}_t^1, \mathbf{s}_t^2]$, where $\mathbf{s}_t^1, \mathbf{s}_t^2$ are the two agents' local states and $\mathbf{s}_t^e$ is part of the world state external to the two agents. The world state's (probabilistic) dynamics is denoted as $p(\mathbf{s}_{t+1}|\mathbf{s}_t, a_t)$, where $a_t = [a_t^1, a_t^2]$ consists of both agents' actions. The agents are self-interested in the sense that they aim to optimize their respective cumulative reward. We let $r_t^j = r^j(\mathbf{s}_t^j)$ denote the reward at time t for agent $j \in \{1, 2\}$, assuming reward function $r^j(\cdot)$ only depends on agent j's local state.

For agent 2, we denote its intent as $\mathbf{i}^2$, which intuitively encodes information about its admissible future trajectories. Formally, we assume agent 2's control over its local state is independent of agent 1, of which the dynamics is denoted as $p^2(\mathbf{s}_{t+1}^2|\mathbf{s}_t^e, \mathbf{s}_t^2, a_t^2)$, such that it can optimize its trajectory without considering agent 1. Formally, given an intent $\mathbf{i}^2$, agent 2 aims to choose its policy $\pi^2 : [\mathbf{s}_t^e, \mathbf{s}_t^2, \mathbf{i}^2] \mapsto a_t^2$ that optimizes its expected cumulative reward while ensuring the future trajectory up to a certain time horizon H is admissible to intent $\mathbf{i}^2$:

$$\begin{aligned} & \max_{\pi^2} \ \mathbb{E}\left[\sum_{t=0}^{H} r_t^2 \middle| \mathbf{s}_0^e, s_0^2\right] \\ s.t. \quad & \mathbf{s}_{t+1}^2 \sim p^2(\mathbf{s}_t^e, \mathbf{s}_t^2, a_t^2), \quad t = 0, \dots, H-1 \\ & a_t^2 \sim \pi^2(\mathbf{s}_t^e, \mathbf{s}_t^2, \mathbf{i}^2), \quad t = 0, \dots, H-1 \\ & (s_t^2)_{t=0}^H \in \mathbf{i}^2 \end{aligned} \tag{1}$$

where in the last constraint we slightly abuse the notation to let $\mathbf{i}^2$ denote the admissible set of agent 2's trajectories.

Agent 1 conditions its action selection on the world state, as well as the intent received from agent 2, i.e., $\pi^1 : [\mathbf{s}_t, \mathbf{i}^2] \mapsto a_t^1$, to optimize its expected cumulative reward:

$$\begin{aligned} & \max_{\pi^1} \ \mathbb{E}\left[\sum_{t=0}^{H} r_t^1 \middle| \mathbf{s}_0\right] \\ s.t. \quad & \mathbf{s}_{t+1} \sim p(\mathbf{s}_t, a_t), \ a_t^1 \sim \pi^1(\mathbf{s}_t, \mathbf{i}^2), \ t = 0, \dots, H-1 \\ & (s_t^2)_{t=0}^H \in \mathbf{i}^2 \end{aligned} \tag{2}$$

where the last constraint means agent 1 knows that agent 2 will comply with its intent.

Eq. (1) and (2) thus provide a formalism for Q1 and Q2. We next describe our implementation of this formalism for the highway merging scenarios.

Fig. 1: Initial configuration of our highway merge environment. The yellow vehicle is the AV capable of sharing its intent. The green vehicle is the merging AV that receives the former's intent. The blue vehicles are human-driven.

IV. An Implementation for Highway Merging

Under the problem setting formulated in Section III, we now conduct a case study on highway merging scenarios to provide an implementation of intent-aware AD and demonstrate its benefits. The implementation is built upon highway-env [2], a microscopic highway simulator developed for learning-based AV control. Figure 1 shows the initial configuration of the highway merge setup, where the intent-sender AV 2 starts in the rightmost lane of the highway while the intent-receiver AV 1 starts on the entrance ramp. As the merging AV approaches the highway, it has to interact with the intent-sender AV in order to merge successfully. Other vehicles are human-driven.

Specifically, 1) in Section IV-A, we provide a representation for several intents for AV 2, each intent characterizing a reasonable type of its future trajectories; 2) in Section IV-B, we design rule-based policies for AV 2 to comply with its intent, which can viewed as feasible solutions to its policy optimization in Eq. (1); and 3) in Section IV-A, we propose an RL approach to AV 1's policy optimization in Eq. (2) that aims to utilize the received intent, with a reward design for safe and efficient merge maneuver.

Our goal with this case study is to demonstrate the benefit of intent-aware AD for the merging AV 1: knowing the intent of AV 2 on the highway, AV 1 is hypothesized to be able to perform a better merge maneuver than it would without knowing the intent. We will conduct experiments in Sections V and VI to test this hypothesis.

A. Representation of Intents

As described in Section III, an intent encodes information about future behavior of an AV, where the AV that shares its intent also commits to complying with it. In this case study, we represent an intent as a restricted set of actions that an intent-sender AV commits to taking for a period of time. In highway-env, we consider the actions to be the following five discrete high-level maneuver decisions (as opposed to low-level controls for acceleration and steering angle), $\mathcal{A} = \{\texttt{IDLE}, \texttt{LANE\_LEFT}, \texttt{LANE\_RIGHT}, \texttt{FASTER}, \texttt{SLOWER}\}$. Each action corresponds to an update in either the target lane or speed of the AV agent. Once an action is chosen, the simulator's built-in low-level controllers determine acceleration and steering commands to execute that action.

Then, an intent is specified by a subset of actions, $\mathcal{A}' \subseteq \mathcal{A}$, that the intent-sender AV is restricted to choose from, which can be represented as a vector of indicators:

$$\mathbf{i}_{\mathcal{A}'} := [\mathbb{1}_{a \in \mathcal{A}'}]_{a \in \mathcal{A}} \tag{3}$$

where $\mathbb{1}_E$ is the indicator function of event E. We also require that each committed action $a \in \mathcal{A}'$ has to be chosen at least once in the committed period of time, since otherwise the intent will reduce to a smaller action subset and cause unnecessary ambiguity.

In this case study, we let the intent-sender AV commit to complying with an intent for the *entire duration* of a simulation episode.

B. AV 2's Compliance with the Intents

Since the committed action set $\mathcal{A}'$ is a subset of the original action set $\mathcal{A}$, there are technically $2^5 = 32$ possible intents. However, not all of these possibilities may be feasible or meaningful. For instance, an empty $\mathcal{A}'$ is not feasible because no action (IDLE) is also an action in our action set. Further, when only one action is committed to, it may not be feasible for the AV to keep choosing LANE_LEFT, LANE_RIGHT, FASTER or SLOWER. Hence, in our setup, we always include IDLE action in $\mathcal{A}'$. When we scale up to two committed actions, we see four possibilities. Out of these, $\mathcal{A}' = \{\texttt{IDLE}, \texttt{LANE\_RIGHT}\}$ is not feasible in our case because the intent-sender AV is already on the rightmost lane of the highway. This results in the following four intents that we will consider:

$$\begin{aligned}
\mathbf{i}_{\texttt{IDLE}} &= [1, 0, 0, 0, 0] \\
\mathbf{i}_{\texttt{LANE_LEFT}} &= [1, 1, 0, 0, 0] \\
\mathbf{i}_{\texttt{FASTER}} &= [1, 0, 0, 1, 0] \\
\mathbf{i}_{\texttt{SLOWER}} &= [1, 0, 0, 0, 1].
\end{aligned}$$

We manually design rule-based policies for the intent-sender AV 2 to comply with one of the intents. First, we set its initial speed to 30 m/s and assign it an intent randomly sampled from a uniform distribution on $\{\mathbf{i}_{\texttt{IDLE}}, \mathbf{i}_{\texttt{LANE\_LEFT}}, \mathbf{i}_{\texttt{FASTER}}, \mathbf{i}_{\texttt{SLOWER}}\}$. Then AV 2 acts according to the behavior policy specific to its intent. For intent $\mathbf{i}_{\texttt{IDLE}}$, the AV 2 just chooses IDLE throughout an episode. For other intents, it also needs to decide when the other action will be taken. We facilitate this decision by assigning an action-trigger position for the AV to take the committed action. Concretely, AV 2 chooses IDLE till it reaches the assigned action-trigger position, where it chooses the committed action, namely LANE_LEFT, FASTER, or SLOWER. Thereafter, it continues to choose IDLE. The action-trigger position is randomly chosen from three options along AV 2's trajectory before the end of the merging zone, which introduces diversity in admissible future trajectories.

C. AV 1's RL-based Intent Utilization

We now describe our RL-based approach to AV 1's utilization of the received intent, for which we specify the state structure, and reward function below.

1) State: AV 1 conditions its policy on the world state $\mathbf{s}$ and AV 2's intent $\mathbf{i}^2$, where the world state is further decomposed into AV 1's local state $\mathbf{s}^1$, AV 2's local state $\mathbf{s}^2$, and the external state $\mathbf{s}^e$ (Section III). In our case study, human-driven vehicles $\mathcal{K}$ constitute the external state.

To implement this, we use vehicles' kinematics to represent their state vectors:

$$\mathbf{s}^e = \left[[x^k,\, y^k,\, v_x^k,\, v_y^k]\right]_{k\in\mathcal{K}} \tag{4}$$

$$\mathbf{s}^1 = [x^1,\, y^1,\, v_x^1,\, v_y^1] \tag{5}$$

$$\mathbf{s}^2 = [x^2,\, y^2,\, v_x^2,\, v_y^2] \tag{6}$$

where x^u and y^u are x and y positions of vehicle u, and v_x^u and v_y^u are x and y components of vehicle u's velocity, respectively.

To interpret and use the received intent information from AV 2, we consider an auxiliary state due to communication:

$$\mathbf{z}^{1\leftarrow 2} = \begin{cases} \mathbf{i}^2, & \text{if AV 2 shares its intent with AV 1} \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{7}$$

AV 1's policy is now conditioned on the world state $\mathbf{s}$ and the auxiliary state $\mathbf{z}^{1\leftarrow 2}$. The introduction of an auxiliary state allows us to also investigate cases when AV 2 does not share its intent, as we will discuss in Section V.

2) Reward: Our reward structure for AV 1 is composed of four components:

$$r_t^1 = r_t^s + r_t^l + r_t^c + r_t^m. \tag{8}$$

Component r_t^s encourages high speed:

$$r_t^s = \beta^s \frac{v_t^1 - v_{\min}^1}{v_{\max}^1 - v_{\min}^1}$$

where v_t^1, $v_{\min}^1$, and $v_{\max}^1$ are respectively the current, minimum, and maximum speeds of AV 1. We set $v_{\min}^1 = 20$ m/s, and $v_{\max}^1 = 30$ m/s.

Component r_t^l encourages AV 1 to seek and stay in the rightmost lane of the highway:

$$r_t^l = \begin{cases} \beta^l, & \text{if AV 1 is in the rightmost lane} \\ 0, & \text{otherwise.} \end{cases}$$

Component r_t^c penalizes collision with other vehicles and road objects:

$$r_t^c = \begin{cases} \beta^c, & \text{if AV 1 has crashed} \\ 0, & \text{otherwise.} \end{cases}$$

Component r_t^m is the reward that AV 1 obtains when it merges successfully, which is further decomposed into four terms:

$$r_t^m = \begin{cases} r^q + r^f + r^r + r^e, & \text{if successful merge at } t \\ 0, & \text{otherwise.} \end{cases}$$

These terms scale the merge reward per the quality of merge maneuver:

- Term r^q encourages quick merging via $r^q = \beta^q \frac{1}{t_m}$, where t_m is the time at which merge occurred.
- Terms r^f and r^r are gap-based rewards that encourage safe merging: $r^f = \beta^f \min\left\{\log \frac{\Delta d_m^{f,1}}{t_*^h\, v_m^1},\, 0\right\}$ and $r^r = \beta^r \min\left\{\log \frac{\Delta d_m^{1,r}}{t_*^h\, v_m^r},\, 0\right\}$. Here, r^f and r^r correspond to gaps with the front (f) and rear (r) vehicles respectively. $\Delta d_m^{f,1}$ and $\Delta d_m^{1,r}$ are relative distances of AV 1 with vehicles f and r, respectively, at merge-time t_m. t_*^h is the desired time-headway, which we set to 1.2 s. v_m^1 and v_m^r are speeds of AV 1 and vehicle r at t_m. The structure of these rewards is based on [10] and seeks to aggressively penalize AV 1 when it merges with a time-headway less than t_*^h. If there is no vehicle in front of or behind AV 1, these rewards are set to 0.
- Term r^e is the component of r_t^m that rewards AV 1 for an efficient merge, by penalizing it for merging with speeds deviating from its target speed: $r^e = \beta^e \left|\frac{v_*^1 - v_m^1}{v_*^1}\right|$, where v_*^1 and v_m^1 are AV 1's target speed and speed at t_m, respectively. We set $v_*^1 = 30$ m/s.

β's in the above equations are scaling coefficients to scale their corresponding rewards. We set $\beta^s = 0.275$, $\beta^l = 0.1$, $\beta^c = -5.0$, $\beta^q = 2.0$, $\beta^f = 0.5$, $\beta^r = 0.5$, and $\beta^e = -1.0$ after manual tuning so as to encourage reasonable and safe merging behavior.

We set the initial speed of AV 1 to 20 m/s.

V. Experimental Setup

We design our environment with one intent-sender AV (AV 2) and four human-driven vehicles on the highway, and one merging AV (AV 1) on the entrance ramp (Figure 1). We use human-driven vehicles in our experiments to create a realistic traffic scenario for the merging AV to learn from. However, these vehicles are incapable of intent-sharing, and their behavior is executed as a part of the environment. In highway-env, the behavior of a human driver is modeled using the Intelligent Driver Model (IDM) [14] for longitudinal motion and Minimizing Overall Braking Induced by Lane change (MOBIL) model [15] for lateral motion.

A. Scenarios

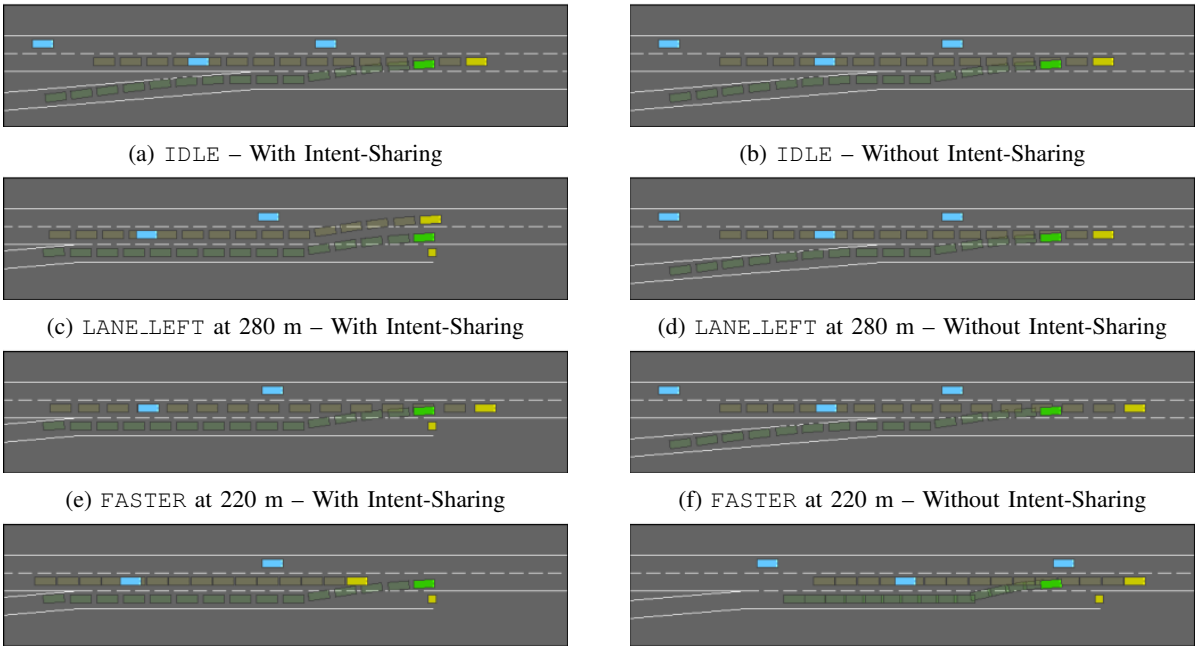
Intuitively, the merging AV should be able to use extra information about the intent-sender AV (which we also call mainstream AV) to better characterize its state, which should in turn help it make better decisions about merging. To test this conjecture, we consider two scenarios for our experiments: with and without intent-sharing. In the first case, the merging AV can receive the intent of the mainstream intent-sender AV. In the second case, however, intent-sharing is blocked, and the merging AV does not get information about the mainstream AV's intent. However, it is important to note that the mainstream AV still behaves as per its intent in both the scenarios, irrespective of whether it shares that intent or not. For both scenarios, we randomly choose an intent out of the four candidates to start a training episode.

B. RL Algorithm Details

While our problem setting consists of multiple agents, we have pre-decided policies for our intent-sender AV. Therefore, we can use single-agent RL methods to learn an optimal behavior policy for the merging AV. Specifically, we use Stable Baselines3's [16] implementation of Deep Q Network (DQN) [17] using multi-layer perceptrons (MLP) for both the scenarios of our experiments. We use two

TABLE I: Performance of the Learned Merging Policies with and without Intent-Sharing.

Intent	Action-Trigger Positions (m)	With Intent-Sharing		Without Intent-Sharing	
		Cumulative Reward	Crash Rate (%)	Cumulative Reward	Crash Rate (%)
$\mathbf{i}_{\texttt{IDLE}}$	N/A	$\mathbf{2.725 \pm 0.087}$	**0.0**	1.416 ± 2.566	20.0
$\mathbf{i}_{\texttt{LANE\_LEFT}}$	220	$\mathbf{3.616 \pm 0.000}$	**0.0**	3.144 ± 0.338	**0.0**
	250	$\mathbf{3.481 \pm 0.000}$	**0.0**	2.855 ± 0.577	**0.0**
	280	$\mathbf{3.481 \pm 0.000}$	**0.0**	2.701 ± 0.514	**0.0**
$\mathbf{i}_{\texttt{FASTER}}$	190	$\mathbf{3.149 \pm 0.000}$	**0.0**	2.848 ± 0.389	**0.0**
	220	$\mathbf{2.993 \pm 0.000}$	**0.0**	2.660 ± 0.354	**0.0**
	250	$\mathbf{2.767 \pm 0.000}$	**0.0**	2.587 ± 0.272	**0.0**
$\mathbf{i}_{\texttt{SLOWER}}$	160	$\mathbf{2.957 \pm 0.742}$	**0.0**	1.490 ± 3.072	20.0
	190	$\mathbf{2.898 \pm 0.562}$	**0.0**	1.436 ± 2.894	20.0
	220	$\mathbf{2.736 \pm 0.486}$	**0.0**	-0.121 ± 3.010	40.0



(a) IDLE – With Intent-Sharing

(b) IDLE – Without Intent-Sharing

(c) LANE_LEFT at 280 m – With Intent-Sharing

(d) LANE_LEFT at 280 m – Without Intent-Sharing

(e) FASTER at 220 m – With Intent-Sharing

(f) FASTER at 220 m – Without Intent-Sharing

(g) SLOWER at 190 m – With Intent-Sharing

(h) SLOWER at 190 m – Without Intent-Sharing

Fig. 2: Snapshots at merge with visible past trajectories. For $\mathbf{i}_{\texttt{IDLE}}$, the merging AV learns the same policy both with and without intent-sharing (2a, 2b). It positions itself to merge behind the mainstream AV, whose intent indicates that it is going to stick to its current trajectory without making changes to accommodate the merging AV. For $\mathbf{i}_{\texttt{LANE\_LEFT}}$, Figure 2c shows that based on its training, the merging AV has learned that the mainstream AV will definitely change its lane by 280 m. So it triggers a lane change at the same time-step. Without intent-sharing, however, the merging AV doesn't know about the lane change, so it positions itself behind the mainstream AV and does not wait for it to change lanes (2d). For $\mathbf{i}_{\texttt{FASTER}}$, when intent is shared, the merging AV merges towards the end of the merging zone, in anticipation that the mainstream AV will speed up (2e). On the other hand, without intent-sharing, the merging AV just positions itself behind the mainstream AV irrespective of what the mainstream AV does (2f). For $\mathbf{i}_{\texttt{SLOWER}}$, when the merging AV knows the mainstream AV's intent, it is able to merge in front of this slow-moving AV while moving at its target speed (2g). However, when it does not receive the latter's intent, it is forced to slow down and merge behind the already slow-moving mainstream AV (2h).

hidden layers with 512 neurons each and adjust the following hyperparameters: `learning_rate` = 5e-4, `buffer_size` = 15000, `learning_starts` = 1000, `batch_size` = 32, `gamma` = 0.95, `train_freq` = 1, `gradient_steps` = 1, and `target_update_interval` = 50. We learn for 40,000 steps for each of five different seeds.

VI. Results and Discussion

Table I shows the performance of policies learned by the merging AV. First, we report the average cumulative reward that the merging AV receives by the end of a simulation episode. We also report the standard error. Second, we report the percentage of cases (seeds) in which the merging AV is unable to learn a crash-free policy. We also decompose performance with respect to the four intent cases and their respective action-trigger positions. It may be noted that action-trigger positions are set as: (a) 220 m, 250 m, and 280 m for $\mathbf{i}_{\texttt{LANE\_LEFT}}$, (b) 190 m, 220 m, and 250 m for $\mathbf{i}_{\texttt{FASTER}}$, and (c) 160 m, 190 m, and 220 m for $\mathbf{i}_{\texttt{SLOWER}}$.

These values correspond to cases for early, intermediate and late intent-realization. For context, the merging zone in our setup extends from 230 m to 310 m. Further, action-trigger positions vary across intents because we observed that not all positions of intent-realization allow sufficient time for the merging AV to utilize that intent, and that these positions vary across different intents.

First of all, we see that the merging AV is able to get higher cumulative rewards when intent-sharing is on as compared to when it is turned off (better values are in bold). The variation in rewards is also significantly lower in the intent-sharing case, especially with $\mathbf{i}_{\texttt{LANE\_LEFT}}$ and $\mathbf{i}_{\texttt{FASTER}}$ where the same merging policy is learned for all the seeds used in our experiments. On the other hand, merging policies without intent-sharing show high variation in cumulative rewards. This makes sense intuitively because the merging AV is uncertain about the behavior of the intent-sender mainstream AV and thus, tries to learn a general policy if it is unaware of the mainstream AV's intent. Another point to note is that the policies learned with intent-sharing are crash-free, while crashes are observed for some seeds in the $\mathbf{i}_{\texttt{IDLE}}$ and $\mathbf{i}_{\texttt{SLOWER}}$ cases without intent-sharing.

A closer look at the cumulative rewards obtained for different intents reveals more interesting trends. We see that, for the initial configuration used in our experiments, the merging AV performs best when the mainstream AV follows $\mathbf{i}_{\texttt{LANE\_LEFT}}$. Furthermore, even within a given intent, the position at which intent-related actions are triggered influence the cumulative rewards that the merging AV is able to get. This indicates that the mainstream AV also has a choice to make – the intent it should pursue and the sequence of actions it should take while complying with that intent. This decision can be influenced by multiple factors such as the configuration of vehicles on the highway and the entrance ramp, and the relative importance that the intent-sender AV gives to the merging AV's interests as compared to its own.

For visual inspection of the learned merging behavior, Figure 2 provides snapshots of the rendered environment at the instant the merging AV successfully merges, using policies from one set of our learned models. The snapshots also show past positions of the intent-sender mainstream AV and the merging AV using transparent (faded) yellow and green colors respectively. We see that, with intent-sharing, the merging AV is able to learn different policies for each intent. However, in the absence of intent-sharing, it tries to learn a general policy while trying to handle uncertainty in the mainstream AV's behavior. We further discuss individual intent cases in detail in the caption.

VII. Conclusion and Future Work

In this work, we formulate intent-aware AD as a multi-agent decision problem. Using a case study with two AVs in a highway merging scenario, we further demonstrate that sharing of intent facilitates learning of robust and crash-free behavior policies for the intent-receiver AV agent by effectively adapting to the intended behavior of an intent-sender AV. As future work, several directions can be explored. A reasonable next step is then to extend our framework to allow choosing of intent via joint learning of the intent-sender and intent-receiver agents. Another direction is to scale beyond the two-agent-one-intent setting and allow communication of intents from multiple intent-sender agents to multiple intent-receiver agents. It is also possible to extend to scenarios beyond merging.

Acknowledgment

This work is supported in part by NSF IIS-2154904, CNS-2213731.

References

[1] O.-R. A. D. Committee, "Taxonomy and definitions for terms related to cooperative driving automation for on-road motor vehicles," SAE International, Tech. Rep., 2021.

[2] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.

[3] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus, "Intention-aware motion planning," in *Algorithmic Foundations of Robotics X: Proceedings of the Tenth Workshop on the Algorithmic Foundations of Robotics*. Springer, 2013, pp. 475–491.

[4] S. Qi and S.-C. Zhu, "Intent-aware multi-agent reinforcement learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 7533–7540.

[5] J. Wu, X. Sun, A. Zeng, S. Song, S. Rusinkiewicz, and T. Funkhouser, "Spatial intention maps for multi-agent mobile manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8749–8756.

[6] W. Kim, J. Park, and Y. Sung, "Communication in multi-agent reinforcement learning: Intention sharing," in *International Conference on Learning Representations*, 2021.

[7] H. M. Wang, S. S. Avedisov, O. Altintas, and G. Orosz, "Multi-vehicle conflict management with status and intent sharing," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 1321–1326.

[8] M. Matthews, G. Chowdhary, and E. Kieson, "Intent communication between autonomous vehicles and pedestrians," *arXiv preprint arXiv:1708.07123*, 2017.

[9] Y. Tang, "Towards learning multi-agent negotiations via self-play," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[10] Q. Liu, F. Dang, X. Wang, and X. Ren, "Autonomous highway merging in mixed traffic using reinforcement learning and motion predictive safety controller," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 1063–1069.

[11] S. Triest, A. Villaflor, and J. M. Dolan, "Learning highway ramp merging via reinforcement learning with temporally-extended actions," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1595–1600.

[12] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Cooperation-aware reinforcement learning for merging in dense traffic," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3441–3447.

[13] B. Toghi, R. Valiente, D. Sadigh, R. Pedarsani, and Y. P. Fallah, "Social coordination and altruism in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 791–24 804, 2022.

[14] 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.

[15] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, 2007.

[16] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013.