Appendix

A. Summary of Notations

Notation	Description	Notation	Description
E	a set of agents	i	an agent i
$\mathcal{S}$	a state space	s_t	a state at time t
$\mathcal{O}_i$	an observation space of agent i	$o_{t,i}$	an observation of agent i at time t
$\mathcal{A}$	an action space	$a_{t,i}$	an action of agent i at time t
$a^c_{t,i}$	a continuous action of agent i at time t	$a^d_{t,i}$	a discrete action of agent i at time t
$\bar{a}_{t,i}$	a proto-action of agent i at time t	$a^*_{t,i}$	a true action agent i at time t
$\mathbf{a}_t$	a joint action at time t	$\mathbf{a}_{t,-i}$	a joint action except agent i at time t
$\mathcal{T}$	a state transition probabilities	Ω_i	an observation transition probabilities of agent i
$R_{t,i}$	a reward of agent i at time t	γ	a temporal discounted factor
$\mathcal{C}$	a character space	$\mathcal{D}$	a dynamic model
$\mathbf{c}_i$	a character vector of agent i		

B. System specification

CPU	AMD Ryzen 9 3950X 16-core
GPU	GeForece RTX 2080 Ti
RAM	128 GB
SSD	1T

C. Hyperparameters

C.1. Algorithm 1

Hyperparameter	Value	Hyperparameter	Value
total episodes (K)	3500	total timesteps (T)	3000
policy delay (d)	2	target noise variance ($\bar{\sigma}$)	0.2
replay buffer size ($\lvert\mathcal{B}\rvert$)	4×10^6	train batch size (B)	128
discount factor (γ)	0.99	soft update rate (τ)	1×10^{-3}
exploration variance 1 (σ_1)	0.1	exploration variance 2 (σ_2)	0.6
actor learning rate	5×10^{-4}	critic learning rate	5×10^{-4}
actor hidden node	$[64, 64]$	critic hidden node	$[64, 64]$
activation function of actor hidden layer	ReLU	activation function of critic hidden layer	ReLU
activation function of actor output layer	tanh	activation function of critic output layer	linear

C.2. Algorithm 2

Hyperparameter	Value	Hyperparameter	Values
optimizer	Adam	learning rate	10^{-3}
the number of iterations (L)	200	the number of samples (N)	3000

D. Post-processor Function in (2)

To build a post-processor function $g(\cdot)$, we first allocate the continuous action space $\mathcal{A}^d = [-W, W]$ into $|\mathcal{A}^d| = 2W + 1$ discrete action values. In other words, a continuous number lies in the range $\bar{a}_t^d \in \left[w - \frac{W+w}{2W+1}, w + \frac{W-w}{2W+1}\right]$ is assigned to a discrete action value $w \in \mathcal{A}^d \subset \mathbb{Z}$, i.e.,

$$a_t^d = w, \text{ if } w - \frac{W+w}{2W+1} < \bar{a}_t^d \leq w + \frac{W-w}{2W+1}.$$

The condition can be written as the range of $a_t^d = w$,

$$\frac{2W+1}{2W}\left(\bar{a}_t^d - \frac{W}{2W+1}\right) \leq w < \frac{2W+1}{2W}\left(\bar{a}_t^d + \frac{W}{2W+1}\right), \tag{5}$$

and it can be reformulated as

$$w = \min\left(\left\lfloor \frac{2W+1}{2W}\left(\bar{a}_t^d + \frac{W}{2W+1}\right)\right\rfloor, W\right),$$

where $\min(\cdot, W)$ hinders w from being outside of action space $[-W, W]$. Here, the floor function is used on the right side of the inequality equation (5). But the ceiling function on the left side of the inequality equation (5) can be an alternative with the max function $\max(\cdot, -W)$.

The post-processor function $a_t^d = g(\bar{a}_t^d, W)$ is finally formulated as follows.

$$g(\bar{a}_t^d, W) = \min\left(\left\lfloor \frac{2W+1}{2W}\left(\bar{a}_t^d + \frac{W}{2W+1}\right)\right\rfloor, W\right)$$

E. Derivation of (3)

$$
\begin{aligned}
\hat{\mathbf{c}}_j &= \arg\max_{\mathbf{c}} \ln P(o_{1:T,j}, a_{1:T,j}|\mathbf{c}) \\
&= \arg\max_{\mathbf{c}} \ln \int P(s_{1:T}, o_{1:T,j}, a_{1:T,j}|\mathbf{c}) ds_{1:T} && (6) \\
&= \arg\max_{\mathbf{c}} \ln \int P(s_{1:T}|o_{1:T,j}, a_{t:T,j}) \times \frac{P(s_{1:T}, o_{1:T,j}, a_{1:T,j}|\mathbf{c})}{P(s_{1:T}|o_{1:T,j}, a_{t:T,j})} ds_{1:T} && (7) \\
&= \arg\max_{\mathbf{c}} \int P(s_{1:T}|o_{1:T,j}, a_{t:T,j}) \times \ln \frac{P(s_{1:T}, o_{1:T,j}, a_{1:T,j}|\mathbf{c})}{P(s_{1:T}|o_{1:T,j}, a_{t:T,j})} ds_{1:T} && (8) \\
&= \arg\max_{\mathbf{c}} \int P(s_{1:T}|o_{1:T,j}, a_{t:T,j}) \times \ln P(s_{1:T}, o_{1:T,j}, a_{1:T,j}|\mathbf{c}) ds_{1:T} + H(s_{1:T}|o_{1:T,j}, a_{t:T,j}) && (9) \\
&= \arg\max_{\mathbf{c}} \int P(s_{1:T}|o_{1:T,j}, a_{t:T,j}) \times \ln P(s_{1:T}, o_{1:T,j}, a_{1:T,j}|\mathbf{c}) ds_{1:T} && (10)
\end{aligned}
$$

The equality of (6) and (7) is because of multiplying the same value on the numerator and denominator. The inequality of (7) and (8) is based on Jensen's inequality, which means $f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)]$ is satisfied when $f(\cdot)$ is a concave function (in our case, $f(\cdot)$ is $\ln(\cdot)$). Subsequently, we can rewrite $-P(\cdot)\ln P(\cdot)$ as a entropy $H(\cdot)$. The inequality of (9) and (10) is because the entropy $H(\cdot)$ is always a positive value.

$$
\begin{aligned}
\hat{\mathbf{c}}_j &= \arg\max_{\mathbf{c}} \int P(s_{1:T}|o_{1:T,j}, a_{t:T,j}) \times \ln P(s_{1:T}, o_{1:T,j}, a_{1:T,j}|\mathbf{c}) ds_{1:T} \\
&= \arg\max_{\mathbf{c}} \int P(s_{1:T}|o_{1:T,j}, a_{1:T,j}) \Bigg[\ln P(s_1) + \sum_{t=1}^{T} \ln \Omega_j(o_{t,j}|s_t) + \sum_{t=1}^{T} \ln \pi(a_{t,j}|o_{t,j}; \mathbf{c}) \\
&\quad + \int \sum_{t=1}^{T} \ln \mathcal{T}(s_{t+1}|s_t, a_{t,j}, \mathbf{a}_{t,-j}) d\mathbf{a}_{1:T,-j} \Bigg] ds_{1:T} && (11) \\
&= \arg\max_{\mathbf{c}} \sum_{t=1}^{T} \ln \pi(a_{t,j}|o_{t,j}; \mathbf{c}) \times \int P(s_{1:T}|o_{1:T,j}, a_{1:T,j}) ds_{1:T} && (12) \\
&= \arg\max_{\mathbf{c}} \sum_{t=1}^{T} \ln \pi(a_{t,j}|o_{t,j}; \mathbf{c}) && (13) \\
&= \arg\max_{\mathbf{c}} \sum_{t=1}^{T} [\ln \pi(a^c_{t,j}|o_{t,j}; \mathbf{c}) + \ln \pi(a^d_{t,j}|o_{t,j}; \mathbf{c})] && (14)
\end{aligned}
$$

We can decompose (10) as (11) by the Markov property. Next, we can ignore the $\Omega(\cdot)$ and $\mathcal{T}(\cdot)$ of (11) because these terms are not related to $\mathbf{c}$. Likewise, we can ignore the $P(s_{1:T}|o_{1:T,j}, a_{1:T,j})$ of (12). Consequently, (13) can be decomposed as the probabilities with respect to both continuous and discrete action as (14) because we consider the hybrid action space.

F. Loss Function for Character Inference

If $\pi(a^c_{t,j}|o_{t,j};\mathbf{c})$ is the Gaussian distribution and $\pi(a^d_{t,j}|o_{t,j};\mathbf{c})$ is the Dirac delta distribution, each term of the equation $\mathcal{U}(\mathbf{c})$ is defined as follows:

$$\ln \pi(a^c_{t,j}|o_{t,j};\mathbf{c}) = \frac{1}{2}\ln 2\pi\sigma^2_\pi + \frac{|a^c_{t,j} - a^{*,c}_{t,j}|}{2\pi\sigma^2_\pi},$$

$$\ln \pi(a^d_{t,j}|o_{t,j};\mathbf{c}) = \mathbb{1}[a^d_{t,j} \neq a^{*,d}_{t,j}](|a^{*,d}_{t,j} - \bar{a}^d_{t,j}|),$$

where $a^{*,c}_{t,j}$ and $a^{*,d}_{t,j}$ mean the actual action value sampled by observing the target agent, and $\mathbb{1}[\cdot]$ means the indicator function. When the estimated deterministic action $a^d_{t,j}$ is different to the actual action $a^{*,d}_{t,j}$ (i.e., $a^d_{t,j} \neq a^{*,d}_{t,j}$), indicator function becomes 1; Conversely, when $a^d_{t,j} = a^{*,d}_{t,j}$, indicator function becomes 0. If inferred character parameter $\hat{\mathbf{c}}$ is similar to the actual character parameter $\mathbf{c}$, the errors between the action produced by $\hat{\mathbf{c}}$ and the observed actual action would decrease.

G. Experiments: Autonomous Driving

To deal with a continuous state space, a hybrid action space, and the agents' characters, we consider the autonomous driving simulator.

In the demonstration task, the agents, the autonomous vehicles, drive the L-lane roundabout road. The agents are randomly deployed on the road in every episode. The agents' goal is to drive as close to the desired velocity as possible, and the agents should control the acceleration and lane changes to reach the goal. To address this task, we set the POMDP. Here, the state includes the velocity and position of all vehicles, and the observation includes information about neighboring vehicles. The action includes acceleration and lane change control in continuous and discrete space, respectively. The reward function comprises three terms: considering the desired velocity, safety distance, and meaningless lane change. We provide the specific POMDP model in the following subsection.

G.1. State

state $s_t \in \mathcal{S}$ is defined as

$$s_t = [\mathbf{v}_t^T, \mathbf{p}_t^T, \mathbf{k}_t^T]^T.$$

The state s_t means the total information of all vehicles on the road. Here, $\mathbf{v}_t = [v_{t,1}, v_{t,2}, \cdots, v_{t,N}]$ represents the velocity of all vehicles, $\mathbf{p}_t = [p_{t,1}, p_{t,2}, \cdots, p_{t,N}]$ denotes the positions of the vehicles, and $\mathbf{k}_t = [k_{t,1}, k_{t,2}, \cdots, k_{t,N}]$ denotes the lane position of all vehicle at a given time t.

G.2. Observation

The observation $o_t \in \mathcal{O}$ comprises the partial state information that the agent can observe. We assume that an agent i can observe the following and leading vehicles located in the same and next lanes. Thus, we set the observation $o_{t,i}$ as follows:

$$o_{t,i} = [v_{t,i}, \Delta\mathbf{v}_{t,i}, \Delta\mathbf{p}_{t,i}, k_{t,i}]^T,$$

where $v_{t,i}$ denotes the velocity of an agent i, $\Delta\mathbf{v}_{t,i}$ is relative velocity between the agent i and observable vehicles, $\Delta\mathbf{p}_{t,i}$ is relative position, and $k_{t,i}$ denotes the lane number at given time t. Here, $\Delta\mathbf{v}_{t,i} = [\Delta v_{t,lL}, \Delta v_{t,lS}, \Delta v_{t,lR}, \Delta v_{t,fL}, \Delta v_{t,fS}, \Delta v_{t,fR}]$, and $\Delta\mathbf{p}_{t,i} = [\Delta p_{t,lL}, \Delta p_{t,lS}, \Delta p_{t,lR}, \Delta p_{t,fL}, \Delta p_{t,fS}, \Delta p_{t,fR}]$, where subscripts l and f mean leading and following vehicles, and subscripts L, S, and R signify located left, same, and right lane, respectively.

G.3. Action

The action $\mathbf{a}_{t,i} \in \mathcal{A}$ consists of a continuous action $a_{t,i}^c \in \mathcal{A}^c$ and a discrete action $a_{t,i}^d \in \mathcal{A}^d$ at time t. In this framework, a continuous action is acceleration control, and a discrete action is a lane change. Acceleration control space $\mathcal{A}^c$ is defined as a space from maximum acceleration to minimum acceleration $[a_{min}, a_{max}]$; Lane change space $\mathcal{A}^d$ is defined as $\{-1, 0, 1\}$. In $\mathcal{A}^d$, $a_{t,i}^d = -1$ means the agent moves a lane outwards (right side), conversely $a_{t,i}^d = 1$ means the agent moves a lane inwards (left side), and $a_{t,i}^d = 0$ means the agent keeps the same lane.

G.4. Reward

As discussed in section 3.1, the character-based reward function is defined as $R_{t,i} = R_i(s_t, a_{t,i}, s_{t+1}; \mathbf{c}_i)$. In this experiment, the reward function $R_{t,i}$ is defined as:

$$R_{t,i} = c_1\mathcal{R}_1 + c_2\mathcal{R}_2 + c_3\mathcal{R}_3 + r_{fail},$$

where $\mathbf{c} = \{c_1, c_2, c_3\}$ denotes a vector of the character coefficients and $\{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$ denotes a vector of the reward terms, and r_{fail} means a penalty for the unfeasible actions (i.e., trial to move a non-existence lane and a lane where other vehicles are located.).

We use r_{fail} term for punishing unfeasible action, which is designed for safety learning purposes. By introducing this penalty, an agent can learn about unsafe decisions without experiencing an accident. In other words, it allows the agent to use the safety assistant system fewer times, such as the ADAS (Advanced Driver Assistance System).

Subsequently, detailed equations of the reward terms are as follows.

The first reward term is defined as follows:

$$\mathcal{R}_1 = 1 - \left| \frac{v_{t+1,i} - v_i^*}{v_i^*} \right|,$$

where v_i^* denotes the target velocity of the agent i. We consider that the agent can drive close to the target velocity. When $v_{t,i} = v_i^*$, the reward term is maximized as the highest value 1; when $v_{t,i} \neq v_i^*$ the reward term is lower than 1.

Next, the second reward term is defined as follows:

$$\mathcal{R}_2(\Delta p_{t+1,fS}) = \min \left[0, 1 - \left(\frac{s^*}{\Delta p_{t+1,fS}} \right)^2 \right],$$

where s^* denotes the safety distance between the vehicles, and we design this reward term to induce the agent to drive with the following vehicle in mind when the agent changes the lane. In this reward term, s^* is defined as follows.

$$s^* = s_0 + \max \left[0, v_{t+1,fS} \left(t^* + \frac{\Delta v_{t+1,fS}}{2\sqrt{|A_{min} \times A_{max}|}} \right) \right],$$

where s_0 denotes the minimum gap between vehicles, t^* denotes the minimum time headway, the minimum time gap between two sequential vehicles required to arrive at the same location. This safety distance is based on the Intelligent Driving Model (IDM) controller, which is one of the adaptive vehicular control systems [1]. If $s^* \leq \Delta p_{t+1,fS}$ (i.e., the agent keeps the safety distance with a following vehicle when moving the lane), $\mathcal{R}_2$ becomes the 0; on the other hand, $\mathcal{R}_2$ becomes the negative value.

The third term is defined as follows:

$$\mathcal{R}_3 = |a_{t,i}^d| \Delta p_{t,lS} \times \min[0, \Delta p_{t+1,lS} - \Delta p_{t,lS}].$$

This reward term is related to unnecessary lane changes, which is a movement to lanes with less driving space than the current lane. When the agent changes the lane $|a_{t,i}^d| = 1$ and $\Delta p_{t,lS} < \Delta p_{t+1,lS}$ or keeps the lane $|a_{t,i}^d| = 0$, this penalty term can be neglected (i.e., $\mathcal{R}_3 = 0$). Conversely, when the agent changes the lane $|a_{t,i}^d| = 1$ and $\Delta p_{t,lS} \geq \Delta p_{t+1,lS}$, this penalty term becomes the negative value.

H. Behavioral Pattern over Character Coefficients

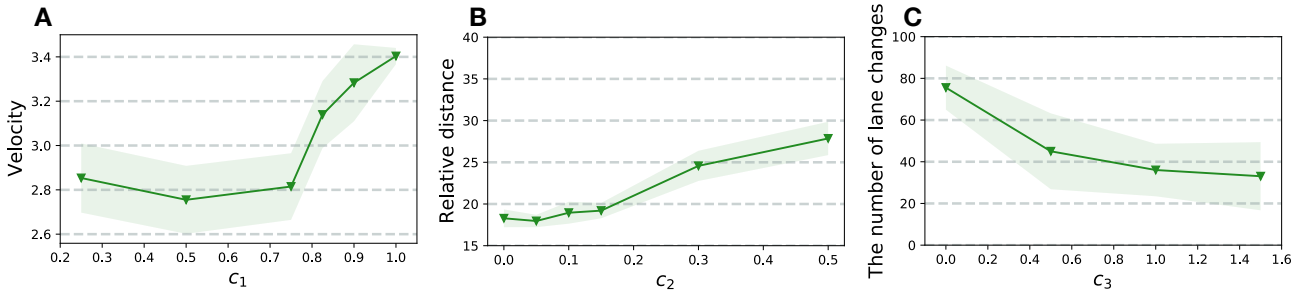

Figure 7. Behavioral pattern of the agent over character coefficient c_n. **A**: Tendency of the average velocity of the agent over character c_1 ($c_2 = c_3 = 0$). **B**: Tendency of the relative distance to the following vehicle over character c_2 ($c_1 = c_3 = 0$). **C**: Tendency of lane-changing frequency over c_3 increases($c_1 = c_2 = 0$).

To confirm behavioral differences over the character coefficient, we perform ablation studies on reward function by isolating the independent effect of each character coefficient. It can provide insight into how these characters impact the resulting trajectories. The behavioral differences resulting from character coefficients' changes are illustrated in Figure 7. The markers and shaded areas represent the average value and confidence interval with two standard deviations, respectively.

As described in Appendix G, the reward function is defined as $R_{t,i} = c_1\mathcal{R}_1 + c_2\mathcal{R}_2 + c_3\mathcal{R}_3 + r_{fail}$, where $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$ is related to desired velocity, safe distance and, lane-changing, respectively. Therefore, changes in each character coefficient affect average velocity, relative distance, and the number of lane changes.

Figure 7**A** shows the average velocity of the agent as increasing c_1. This result verifies that the autonomous vehicle drives closer to the desired velocity ($v_i^* = 3.5m/s$). Furthermore, the lower c_1 widens the dispersion area of velocity.

Figure 7**B** represents the relative distance between the autonomous vehicle and the surrounding vehicle over c_2. The result confirms that the relative distance increases as c_2 grows. This character coefficient is straightforwardly related to a safe distance. The agent would pursue safe driving by securing a larger driving space as c_2 grows.

Figure 7**C** shows the number of lane changes as c_3 increases. In the reward function, c_3 puts weights on the unnecessary lane-changing penalty. The unnecessary lane-changing implies movement to lanes with less driving space than the current lane. As c_3 decreases, the agent performs lane-chaining action more frequently.

I. Performance of Character Inference

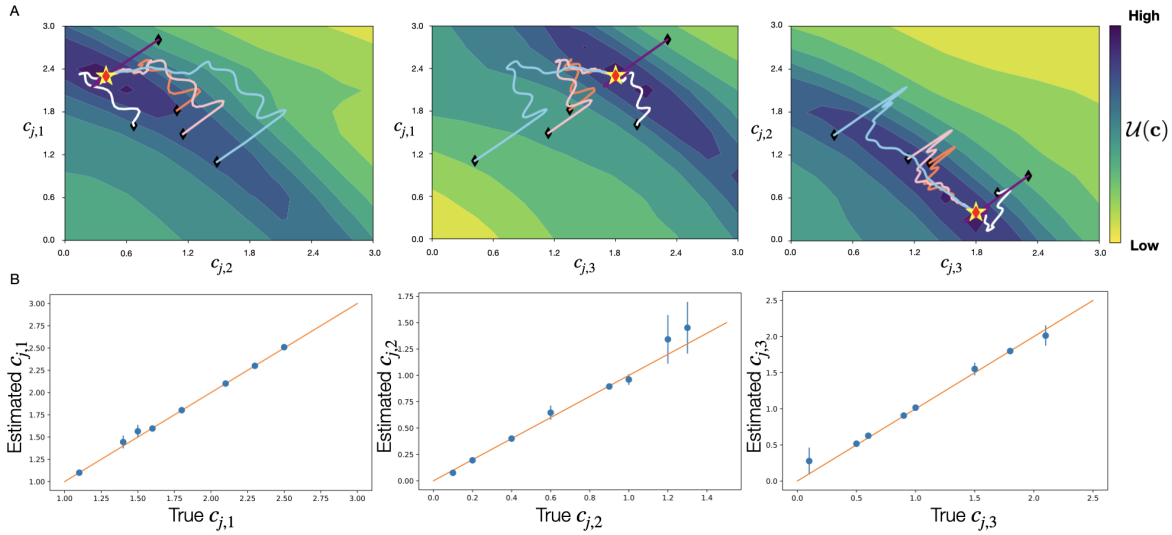

Figure 8. **A**. The converging trajectories of the character parameters. A black diamond indicates the initial points, a red diamond indicates the estimated points, and a yellow star means the true point. **B**. The estimated character parameters of the agent versus true character parameters. The orange line represents the identity line meaning perfect estimation, the blue circles depict the estimated values, and the blue line presents the confidence interval for three standard deviations.

Figure 8A presents the contour plots of the log-likelihood function for the combination of character parameters $\mathbf{c}_{j,k}$, where $k \in [1, 2, 3]$. It shows that the true value is well inferred no matter where the initial value is located. The yellow star, red and black diamonds in these diagrams represent the true, estimated, and initial points, respectively; the curve line presents the character inference trajectory from an initial point to an estimated point.

Figure 8B shows the estimated character value by the agent i versus the true character value of the target j. Each blue point and bar is the average value and the three-standard deviation considering ten experiments. The orange line indicates that the estimated and true values are identical. It represents that the character inference is successful without a large error between the estimated and true value, and in particular, $c_{j,1}$ and $c_{j,3}$ are overall accurate with a small standard deviation. Conversely, the inference about $c_{j,2}$ becomes inaccurate when $c_{j,2} \geq 1.2$. We conclude that the character inference module generally infers the agent's characters well over the observation-action trajectory of the target agent.

References

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