Prioritized Experience-based Reinforcement Learning with Human Guidance for Autonomous Driving

Jingda Wu, *Student Member, IEEE*, Zhiyu Huang, *Student Member, IEEE*, Wenhui Huang, and Chen Lv, *Senior Member, IEEE*

***Abstract*— Reinforcement learning requires skillful definition and remarkable computational efforts to solve optimization and control problems, which could impair its prospect. Introducing human guidance into reinforcement learning is a promising way to improve learning performance. In this paper, a comprehensive human guidance-based reinforcement learning framework is established. A novel prioritized experience replay mechanism that adapts to human guidance in the reinforcement learning process is proposed to boost the efficiency and performance of the reinforcement learning algorithm. To relieve the heavy workload on human participants, a behavior model is established based on an incremental online learning method to mimic human actions. We design two challenging autonomous driving tasks for evaluating the proposed algorithm. Experiments are conducted to access the training and testing performance and learning mechanism of the proposed algorithm. Comparative results against the state-of-the-art methods suggest the advantages of our algorithm in terms of learning efficiency, performance, and robustness.**

***Index Terms*—Reinforcement learning, priority experience replay, human demonstration, autonomous driving**

I. Introduction

REINFORCEMENT learning (RL) has substantially contributed to numerous fields [1-4] by solving control and optimization problems. As a branch of machine learning methods, RL improves the capability of controlling agents in black-box environments through the exploratory trial-and-error principle [5]. Recent popular RL algorithms, e.g., rainbow deep Q-learning [6], proximal policy optimization (PPO) [7], and soft actor-critic (SAC) [8], have shown ability in handling high-dimensional environment representation and generalization, due to the introduction of deep neural networks. Albeit RL can achieve good performance in complex tasks, its drawback emerges that their interactions with the environment are very inefficient [9]. Thus, using RL to solve a problem needs skillful definitions and settings and consumes remarkable computational resources [10].

Combining human guidance with RL can be a promising way to mitigate the above drawback [11]. First, human intervention has been used to improve RL performance. Intervention is triggered by unfavorable actions and should be avoided by RL. Then, the human demonstration is a powerful tool to enhance RL's ability [12]. In this context, the objective functions are generally reshaped compatible with supervised learning to improve efficiency [13].

Despite the above human guidance-based methods, RL needs to process numerous data from its self-explorations. The existing methods do not particularly optimize the utilization of human guidance data, consequently, they still need great human workloads to avoid submersion of guidance in exploratory data. Additionally, human guidance, which is variant to proficiency, mental and physical status of participants, should not be equally treated since some low-quality guidance can even impair the RL performance.

We propose a priority-based experience replay method on human guidance and put forward the associated human guidance-based RL algorithm to bridge the abovementioned gap. Our approach is off-policy, which leverages the experience replay mechanism [14] to maximize the utilization efficiency of self-exploratory data. The proposed priority replay mechanism can further improve the utilization efficiency of human guidance data by quantifying their values and weighing their utilized probability, which ultimately augments the RL performance. As a result, the efficiency can be improved by over seven times under the adopted task. The schematic diagram of our algorithm is depicted in Fig. 1. To evaluate the training and testing performance of our proposed method, we design two challenging autonomous driving scenarios. The experimental results suggest the advance of the proposed algorithm compared to state-of-the-art baselines in learning efficiency, practical performance, and robustness.

The contribution of this paper can be summarized into three aspects. 1) we propose a novel prioritized experience utilization mechanism regarding human guidance in the RL process to improve performance. 2) we establish a comprehensive and holistic framework of human guidance-based RL by integrating the human-RL action switch scheme, behavior cloning-based objective function, human-demonstration replay method, and human-intervention reward shaping mechanism. 3) we validate the superior performance of the proposed algorithm in solving

J. Wu, Z. Huang, W. Huang, and C. Lv are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, 639798. (e-mails: {jingda001,zhiyu001}@e.ntu.edu.sg, {huang.wenhui, lyuchen}@ntu.edu.sg)

Corresponding author: C. Lv.

challenging autonomous driving tasks comprehensively.

The remainder is organized as follows: a review of related work is provided in Section II, preliminaries for the proposed algorithm is introduced in Section III, Section IV provides the proposed human guidance-based reinforcement learning algorithm, a human behavior model for substituting real human participant is established in Section V. Section VI presents the problem formulation for the adopted autonomous driving tasks, Section VII provides the experimental results, and the conclusion is drawn in Section VIII.

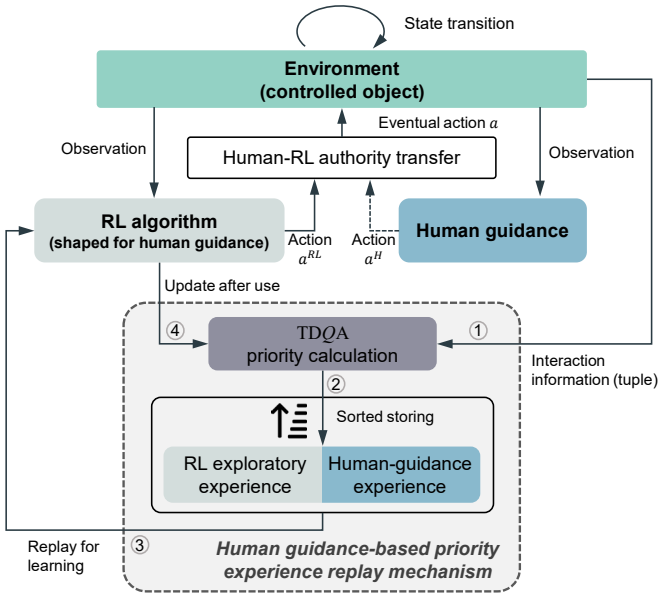

Fig. 1. The framework of the proposed human-guidance-based reinforcement learning algorithm. The RL algorithm in this paper is shaped in multiple aspects to adapt to human guidance. In the proposed human guidance-based priority experience replay mechanism, TDQA represents the proposed priority calculation scheme, and the number 1-4 indicates the flow sequence of data. The dotted line of the action signal represents that the framework allows intermittent human-in-the-loop guidance.

II. Related Work

Sample efficiency bottlenecks the training and performance of RL. Combining human guidance with RL is a promising way to mitigate the challenge. Three categories of human guidance have been integrated into RL. The first one is human feedback, where the human expert's prior knowledge about the task could be used to qualitatively or quantitatively score the RL behaviors [15]. In this manner, an RL-based unmanned ground vehicle was guided to run through a maze [16]. However, the feedback is high-demanding on human ability and thus is no longer popular in recent studies. The second branch is human intervention. Intervention is a more direct manifestation of human knowledge than giving feedback. RL agents are devised to reduce their confidence in adopted actions if intervention occurs[17]. [18] employed real humans to detect catastrophic actions of DQN in playing Atari games, where humans were required to intervene in the training process to block the risk. It punished the human intervened scenes through the reward-shaping technique to prevent RL from reaching the unfavorable situations again. With a similar idea, [19] devised a reward shaping-based PPO algorithm and made the RL agent complete the drone driving tasks under human interventions. In this paper, the above-mentioned reward shaping scheme is also adopted, and more importantly, we provide a theoretical derivation and related discussion on the optimality of the human intervention-based reward shaping method.

The human demonstration is the other way to enhance RL performance. For discrete-action RL, the DQfD algorithm [12] shaped the value function of DQN using human demonstration. [20] presented a double experience buffer setting to separately store the RL data and human demonstrations. For more complicated RL with actor-critic architecture, the policy function is usually modified to be compatible with learning from demonstration. The behavior cloning objective has been added to the objective of the policy function to greatly improve learning efficiency, which is a milestone in the field. In this way, dexterous manipulations of high degree-of-freedom robotic arms [21-23] and human-level game operation [17] were achieved based on the state-of-the-art RL algorithms. In this paper, the behavior cloning objective and its associated human guidance-based actor-critic framework is also integrated into our method. However, it is not reasonable for equal treatment on various demonstrations, which is adopted in existing methods. First, without optimizing the utilization, small-scale human demonstrations would be submerged in the numerous RL-generated data. Second, human guidance is variant due to the proficiency and status of participants, and some low-quality guidance can even impair RL performance. Noticeably, these drawbacks are to be overcome by the proposed prioritized experience utilization mechanism.

III. Preliminaries

In this section, we first introduce the notation and concept of off-policy actor-critic RL, then we illustrate the prioritized experience replay mechanism. All three parts in this section are the base for the proposed prioritized human guidance-based RL algorithm.

A. Notation

We consider a standard reinforcement learning setting where an RL agent interacts with the controlled environment. Such an interaction can be formulated as a discrete-time Markov decision process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, R, p)$. The state-space $\mathcal{S}$ consists of continuous state variables $\mathbf{s}$ and the action space constitutes continuous action variables $\mathbf{a}$. $R(\cdot\,|\mathbf{s},\mathbf{a}): \mathcal{S}\times\mathcal{A}\to r$ is a reward function mapping the state-action pair $(\mathbf{s},\mathbf{a})$ to a deterministic reward value r. The environment dynamics generates state transition probability $p(\cdot\,|\mathbf{s},\mathbf{a}): \mathcal{S}\times\mathcal{A}\to\mathcal{P}(\mathbf{s}')$ mapping the state-action pair $(\mathbf{s},\mathbf{a})$ to the probability distribution over the next state $\mathbf{s}'$.

At each time step t, the agent observes the state $\mathbf{s}_t\in\mathcal{S}$ and sends the action $\mathbf{a}_t\in\mathcal{A}$ to the environment, receiving the feedback of a scalar reward r_t and next state $\mathbf{s}_{t+1}$. The agent's behavior is determined by a policy $\pi(\mathbf{a}_t|\mathbf{s}_t): \mathcal{S}\to\mathcal{P}(\mathbf{a}_t)$, which maps a state to the probability distribution over candidate actions. We utilize ρ_π to represent the state-action distribution induced by the policy π.

B. Off-policy actor-critic architecture

The goal of RL is to optimize the policy which maximizes the expected value $\mathcal{V}$ over the environment dynamics. A Bellman value function (also called critic) is established to estimate $\mathcal{V}$ in a bootstrapping way. This value function is usually called Q. Under an arbitrary policy π, Q is defined as:

$$\begin{aligned} Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t) &= \int_{\mathbf{s}_{t+1}} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \int_{a_{t+1}} \pi(\mathbf{a}_{t+1}|\mathbf{s}_{t+1}) \\ &\quad \cdot [r_t + \gamma \cdot Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})] \cdot d\mathbf{a}_{t+1} \cdot d\mathbf{s}_{t+1} \\ &= r_t + \gamma \cdot \mathop{\mathbb{E}}_{(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) \sim \rho_{\pi}} [Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})] \end{aligned} \tag{1}$$

where $\gamma \in (0,1)$ is the discount factor. Then the policy function (also called actor) can be obtained concerning maximized Q, represented as:

$$\pi = \arg\max_{\pi} \mathop{\mathbb{E}}_{(\mathbf{s},\mathbf{a}) \sim \rho_{\pi}} [Q^{\pi}(\mathbf{s}, \mathbf{a})] \tag{2}$$

In practice, value function pursues the evaluation regarding only the optimal policy $\pi^{\star}$, regardless of the policy executing the interaction. Therefore, RL decouples the policy evaluation process and the policy's behavior, which makes the agent update in an off-policy manner.

We use neural networks as the function approximator to formulate the actor and critic, the objectives are then reached through the loss functions. Specifically, the loss function of the critic $\mathcal{L}^{Q}$, and the actor $\mathcal{L}^{\pi}$ can be expressed as:

$$\begin{aligned} \mathcal{L}^{Q}(\theta) = r_t + \gamma \cdot \mathop{\mathbb{E}}_{\mathbf{s}_{t+1} \sim p(\cdot|s_t, a_t)} [Q(\mathbf{s}_{t+1}, \pi(\cdot\,|\mathbf{s}_{t+1}; \phi); \theta)] - \\ Q(\mathbf{s}_t, \mathbf{a}_t; \theta) \end{aligned} \tag{3}$$

$$\mathcal{L}^{\pi}(\phi) = -Q(\mathbf{s}_t, \pi(\cdot\,|\mathbf{s}_t; \phi); \theta) \tag{4}$$

where $Q(\cdot;\theta)$ represents the parameterized critic function and θ represents the parameters of the critic network, $\pi(\cdot;\phi)$ represents the parameterized actor function and ϕ represents the parameters of the actor network.

C. Prioritized experience replay mechanism

The experience replay mechanism establishes an experience buffer to store the data at each interaction. Accordingly, the RL agent can retrieve data generated by previous policies from the buffer for policy evaluation and improvement.

Given an arbitrary time step t, the interaction between the RL agent and the environment generates a transition tuple, which is stored into the experience replay buffer as:

$$\mathcal{B} \leftarrow \zeta_t = (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1}) \tag{5}$$

Conventionally, the experience in the buffer is retrieved from the buffer using uniform random sampling. In a more efficient method, prioritized experience replay mechanism (PER) [24], the data sample is subjected to a nonuniform distribution $\mathcal{J}$, and its probability mass function $p_{\mathcal{J}} \sim \mathcal{J}$ can be expressed as:

$$p_{\mathcal{J}}(i) = \frac{\mathcal{p}_i^{\alpha}}{\sum_k \mathcal{p}_k^{\alpha}} \tag{6}$$

where $\alpha \in [0,1]$ is the scaling coefficient, $\mathcal{p}$ represents the priority of each tuple i, which is determined by the temporal difference (TD) error δ^{TD} and expressed as:

$$\begin{aligned} \mathcal{p}_i &= |\delta_i^{TD}| + \varepsilon \\ &= |r_i + \gamma \cdot Q(\mathbf{s}_{i+1}, \pi(\cdot\,|\mathbf{s}_{i+1}; \phi); \theta) - Q(\mathbf{s}_i, \mathbf{a}_i; \theta)| + \varepsilon \end{aligned} \tag{7}$$

where $\varepsilon \in \mathbb{R}^{+}$ is a small positive constant to guarantee the probability larger than zero. A larger TD error indicates an experience worth learning to a higher extent. Thus, the TD error-based prioritized experience replay mechanism can improve the RL training efficiency.

IV. Human-in-the-loop Reinforcement Learning

In this section, we first summarize the human behaviors in the RL training process which can be leveraged in the algorithm design. Based on that, we establish an actor-critic framework adapting to human guidance. Then, two modules are proposed to further improve RL in the context of human guidance: a novel prioritized experience replay mechanism concerning human demonstration, and a reward shaping technique concerning human intervention. Finally, a holistic human-in-the-loop RL algorithm is instantiated using the above components.

A. Human guidance behavior in the RL training process

We define two useful human guidance behaviors in the RL training process: intervention and demonstration.

Intervention. Human participants recognize RL interaction scenes and identify whether a guidance behavior should be conducted based on their prior knowledge and reasoning abilities. If human participants decide to intervene, they can manipulate the equipment to get the control authority (partially or totally) from the RL agent. The intervention generally happens when the RL agent conducts catastrophic actions or is stuck in local optima traps. Thus, RL could learn to avoid unfavorable situations from the intervention.

Demonstration. Human participants perform their actions when an intervention event happens, which generates the corresponded reward signal and next-step state. The generated transition tuple can be seen as a piece of demonstration data since it is induced by human policy instead of the RL's behavior policy. RL algorithm could learn human behavior from the demonstration.

State-of-the-art human-guidance-based RL algorithms have been integrating learning from intervention (LfI) [18], and learning from demonstration (LfD) [25]. In this paper, both LfI and LfD will be employed in the proposed architecture. Specifically, LfI based on the reward shaping technique is utilized in the reward function definition, while LfD plays its role in the underlying principles of the algorithm.

B. Human-guidance-based actor-critic framework

In this subsection, we elaborate on the interaction mechanism and learning objective of the proposed human-guidance-based actor-critic RL algorithm.

First, we focus on the interaction mechanism. In the standard interaction between RL and the environment, RL's behavior policy will output actions to explore the environment. Given an off-policy actor-critic RL, the above process is shown as:

$$\mathbf{a}_t^{RL} = \pi(\cdot\,|\mathbf{s}_t; \phi) + \xi_a \odot \mathbf{a}_t^{std} \tag{8}$$

where $\mathbf{a}_t^{std} \in \mathbb{R}^{\dim(\mathcal{A})}$ is a training-dependent variable that scales the exploration noise, $\odot$ represents the Hadamard product and $\xi_a \sim \mathcal{N}(0, \mathbf{I}^{\dim(\mathcal{A})})$.

We give full authority to human participants whenever they decided to take control in the training loop of RL. Thus, the eventual action is filtered by a mask as:

$$\mathbf{a}_t = \left(\mathbf{I}^{\dim(\mathcal{A})} - \mathbf{\Delta}_t\right) \cdot \mathbf{a}_t^{RL} + \mathbf{\Delta}_t \cdot \mathbf{a}_t^{H} \tag{9}$$

where $\mathbf{a}_t^H$ represents the action from the human participant's policy, $\mathbf{\Delta}_t \in \mathbb{R}^{\dim(\mathcal{A})}$ is a demonstration mask: it is an identity matrix when human demonstration happens and a zero matrix in the non-demonstrated step.

The interaction transition tuple ζ will be recorded and stored into the experience replay buffer once the action is sent to the environment. In particular, actions from the human policy and the RL policy are stored in the same buffer. For this context, the new transition tuple ζ is defined to discriminate human demonstrations from normal RL experiences as:

$$\zeta_i = (\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}_{i+1}, \mathbf{\Delta}_i) \tag{10}$$

Then, we focus on the learning objective. Given a batch of transition tuples with batch size N, there could exist data ζ_{N1} from the RL policy and $\zeta_{N2=N-N1}$ from the human policy. The critic network, based on the optimal value function, can learn from both policies. Thus, its loss function is calculated as:

$$\mathcal{L}^Q(\theta) = \frac{1}{N_1}\sum_{i}^{N_1}\left\|r_i + \gamma \cdot Q(\mathbf{s}_{i+1}, \pi(\cdot\,|\mathbf{s}_{i+1};\phi);\theta) - Q\left(\mathbf{s}_i, \mathbf{a}_i^{RL};\theta\right)\right\|^2 + \frac{1}{N_2}\sum_{j}^{N_2}\left\|r_j + \gamma \cdot Q\left(\mathbf{s}_{j+1}, \pi\left(\cdot\,|\mathbf{s}_{j+1};\phi\right);\theta\right) - Q\left(\mathbf{s}_j, \mathbf{a}_j^{H};\theta\right)\right\|^2 \tag{11}$$

Given the data from the human policy, the actor should learn from these demonstrations in addition to maximizing the critic's value. Hence, we devise the loss function of the actor network considering behavior cloning as:

$$\mathcal{L}^\pi(\phi) = \frac{1}{N_1}\sum_{i}^{N_1}[-Q(\mathbf{s}_i, \pi(\cdot\,|\mathbf{s}_i;\phi);\theta)] + \frac{1}{N_2}\sum_{j}^{N_2}[\omega \cdot \left\|\mathbf{a}_j^H - \pi\left(\cdot\,|\mathbf{s}_j;\phi\right)\right\|^2] \tag{12}$$

where ω is a manually determined constant that weighs the importance of behavior cloning.

It is noticeable that the mean squared error (MSE) losses involved in the above formulas are for exemplified calculation, meaning that they can be alternated by any loss functions.

C. Prioritized human-demonstration replay mechanism

In this subsection, we put forward a novel prioritized experience replay (PER) mechanism for human demonstration.

Human demonstrations are generally more critical than most exploration from RL's behavior policy due to prior knowledge and reasoning ability. Thus, a more effective method is needed to weigh human demonstrations among the buffer. We propose an advantage-based metric instead of TD-error of the normal PER to establish the prioritized replay mechanism.

First, we define an advantage measure regarding the human demonstration against the RL's behavior policy. Since the critic, i.e., value function, can evaluate the policy, we calculate the difference between the Q value of the human action and that of the RL action. Given a human-demonstration transition tuple $(\mathbf{s}_i, \mathbf{a}_i = \mathbf{a}_i^H, r_i, \mathbf{s}_{i+1})$, the priority level $\mathcal{p}$ is defined as:

$$\mathcal{p}_i \triangleq \left|\delta_i^{TD}\right| + \varepsilon + \exp[Q\left(\mathbf{s}_i, \mathbf{a}_i^H;\theta\right) - Q(\mathbf{s}_i, \pi(\cdot\,|\mathbf{s}_i;\phi);\theta)] \tag{13}$$

where exp is the exponential function to guarantee the non-negative advantage value.

We call the last term of the Eq. (13) the Q-advantage term, which evaluates to what extent should a specific human-demonstration tuple be retrieved except the TD-error metric. Through the RL training process, the RL agent's ability varies and the priority level of one human-demonstration tuple changes accordingly, which gives rise to a dynamic priority mechanism. We abbreviate Q-advantage as QA and call the above mechanism **TD*Q*A** to illustrate it combines two metrics as the measurement of human guidance. The QA term is removed for non-demonstration tuples when calculating the above equation, thus, the priority levels of non-demonstration data are aligned with those in the conventional PER.

In this manner, the experience in the buffer $\mathcal{B}$ subjects to a distribution $\mathcal{I}'$, and the probability mass function of the experience distribution $\mathcal{p}_{\mathcal{I}'} \sim \mathcal{I}'$ can be expressed as:

$$\mathcal{p}_{\mathcal{I}'}(i) = \frac{\mathcal{p}_i^\alpha}{\sum_k \mathcal{p}_k^\alpha} \tag{14}$$

We inherent the optimization trick of the conventional PER by using a sum-tree structure to store transition data, and the updating and sampling can be conducted with a complexity of $O(logN)$.

The priority mechanism introduces the bias to the estimation of the expectation of the value function since it changes the experience distribution in the buffer. Biased value network Q could have little impact on the RL asymptotic performance, yet it may affect the stability and robustness of the mature policy in some situations. As an optional operation, we can anneal the bias by introducing the importance-sampling weight to the loss function of the value network. The importance-sampling weight of a transition i is calculated as:

$$w_{IS}(i) = \left(\frac{1}{\mathcal{p}_{\mathcal{I}'}(i)}\right)^\beta \tag{15}$$

where $\beta \in [0,1]$ is a coefficient: the fully non-uniform sampling occurs if $\beta = 1$, and fully uniform sampling occurs if $\beta = 0$. β will gradually decrease to zero along with the training process.

The importance-sampling weight can be added to the loss function of the value network, expressed as:

$$\mathcal{L}^Q(\theta) = \mathop{\mathbb{E}}_{\zeta_{i\sim\mathcal{I}'}}\left[\mathcal{p}_{\mathcal{I}'}(i)^{-\beta} \cdot \left(r_i + \gamma \cdot Q(\mathbf{s}_{i+1}, \pi(\cdot\,|\mathbf{s}_{i+1};\phi);\theta) - Q(\mathbf{s}_i, \mathbf{a}_i;\theta)\right)\right] \tag{16}$$

Through the proposed PER, we prioritize human guidance over RL experiences. Moreover, high-quality demonstrations are prioritized to more extents, and the utilization efficiency of human demonstrations can be enhanced.

D. Human-intervention-based reward shaping mechanism

In this subsection, we introduce the human-intervention-based reward shaping technique. Naturally, there is no need for humans to provide guidance if the being-trained RL agent is executing a good policy. Therefore, to minimize the human workload, we assume human participants would intervene in the training process only when RL's behaviors are unfavorable. In this context, the intervention event can be seen as a negative signal and the corresponding state should be avoided by RL. This negative feedback can be realized by reward shaping, which will be detailed in this subsection.

We first identify the intervention event. Recall Eq. (9) defines a mask $\boldsymbol{\Delta}_t$, which is a time-sequential variable recording if the action $\mathbf{a}_t$ is conducted by human demonstration. Hence, the intervention time, i.e., the start time of a period of human demonstrations, can be represented by $(\boldsymbol{\Delta}_t = \mathbf{I})\&(\boldsymbol{\Delta}_{t-1} = \mathbf{0})$ in a time-sequential training process of RL, as illustrated in Fig. 2. It is noted that only the intervention time is to be punished by the reward shaping, since the states after humans intervention will be substituted by human demonstrations and cannot be seen as unfavorable. For instance, in Fig. 2, $\mathbf{s}_2$ is penalized while $\mathbf{s}_3$ and $\mathbf{s}_4$ are not.

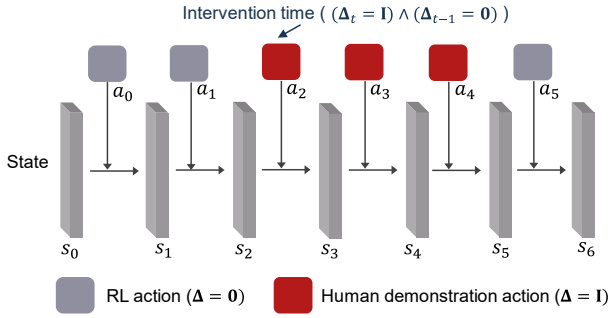

Fig. 2. The illustration of the intervention time step. In a time-sequential MDP, the first time step which is controlled by human demonstration is taken as the intervention time.

Then, we can shape the vanilla reward function with an additional penalized function:

$$r_t^{shape} = r_t + r_{pen} \cdot \left[\left(\boldsymbol{\Delta}_t = \mathbf{I}^{\dim(\mathcal{A})}\right) \wedge \left(\boldsymbol{\Delta}_{t-1} = \mathbf{0}^{\dim(\mathcal{A})}\right)\right] \quad (17)$$

where r_t^{shape} is the reward after shaping, r_{pen} is a scalar that weighs the intervention penalty.

The theoretical performance of this reward shaping scheme is discussed in the Appendix in detail.

E. Prioritized human-in-the-loop RL algorithm

In this subsection, we integrate all the above components and propose a holistic RL algorithm considering human guidance. It is noted that although the human guidance-based actor-critic framework in Section III-B and reward shaping in Section III-D are components of the algorithm, they are not the major novelty of this paper. To highlight our core idea of the prioritized human-demonstration replay mechanism of Section III-C, we name the proposed algorithm as **P**rioritized **H**uman-**I**n-the-**L**oop (**PHIL**) RL.

Specifically, we obtain the holistic human-in-the-loop RL configuration through equipping the human-guidance-based actor-critic framework with prioritized human-demonstration replay and intervention-based reward shaping mechanisms. We instantiate the PHIL algorithm based on one of the state-of-the-art off-policy RLs, i.e., twin delayed deep deterministic policy gradient (TD3) [26]. We also remind the above components are adaptive to various off-policy actor-critic RL algorithms.

In TD3, the target networks, namely, the target critic Q' with parameter θ' and target actor π' with parameter ϕ' are utilized to stabilize the algorithm update. And the actor's output becomes a deterministic value instead of a sample from the probability distribution.

Considering the role of human participants in the RL interaction process, the eventual action in the time step t can be expressed as:

$$\mathbf{a}_t = \left(\mathbf{I}^{\dim(\mathcal{A})} - \boldsymbol{\Delta}_t\right) \cdot \mathbf{a}_t^{RL} + \boldsymbol{\Delta}_t \cdot \mathbf{a}_t^{H} \quad (18a)$$

$$\mathbf{a}_t^{RL} = \pi(\cdot\,|\mathbf{s}_t; \phi) + \text{clip}(\boldsymbol{\epsilon}, -c, c).\ \boldsymbol{\epsilon} \sim \boldsymbol{\mathcal{N}}(\mathbf{0}^{\dim(\mathcal{A})}, \boldsymbol{\Sigma}^{\dim(\mathcal{A})}) \quad (18b)$$

where $\boldsymbol{\epsilon}$ is the noise coefficient vector dependent on the training proceed, c is the bounding of the exploratory action, $\boldsymbol{\Sigma}$ is the covariance matrix of the Gaussian distribution $\boldsymbol{\mathcal{N}}$.

A transition tuple is obtained through the above interaction step and stored into the proposed human-demonstration experience buffer as:

$$\mathcal{B} \leftarrow \zeta_t = (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1}, \boldsymbol{\Delta}_t) \quad (19)$$

Stored experience tuples will be retrieved for the training of the value and policy networks. An arbitrary transition tuple ζ with index i would be retrieved by the probability p, which is calculated by:

$$p(i) = \frac{p_i^{\alpha}}{\sum_k p_k^{\alpha}} \quad (20)$$

wherein the priority level p is:

$$p_t = |\delta_t^{TD}| + \varepsilon + \left(\boldsymbol{\Delta}_t = \mathbf{I}^{\dim(\mathcal{A})}\right) \cdot QA \quad (21a)$$

$$QA = \exp[Q'(\mathbf{s}_t, \mathbf{a}_t; \theta') - Q'(\mathbf{s}_t, \pi(\cdot\,|\mathbf{s}_t; \phi); \theta')] \quad (21b)$$

It is noticeable that Q-advantage is calculated by the target critic network Q' to avoid unstable updates.

Supposing the batched tuples with size N contains N_1 amount of non-demonstration transition tuples and $N_2 = N - N_1$ human-demonstration ones, the loss function of the critic can be expressed as:

$$\mathcal{L}^{Q_k}(\theta) = \frac{1}{N_1} \sum^{N_1} \| r_t + \gamma \cdot \min Q'(\mathbf{s}_{t+1}, \pi'(\cdot\,|\mathbf{s}_{t+1}; \phi'); \theta') \quad (22)$$

$$+\frac{1}{N_2}\sum_{j}^{N_2}\left\|r_j+\gamma\cdot\min_{l=1,2}Q_l^{'}\left(\mathbf{s}_{j+1},\pi^{'}\left(\cdot\left|\mathbf{s}_{j+1};\phi^{'}\right.\right);\theta^{'}\right)-Q_k\left(\mathbf{s}_j,\mathbf{a}_j^H;\theta\right)\right\|^2$$

where $k=1,2$ represents the index of two Q networks. Note the double Q network trick, which utilizes the smaller Q value of two networks ($l=1,2$), is introduced here to eliminate the value overestimation effect.

The loss function of the actor is calculated as:

$$\mathcal{L}^{\pi}(\phi)=\frac{1}{N_1}\sum_{i}^{N_1}[-Q_1(\mathbf{s}_i,\pi(\cdot\left|\mathbf{s}_i\right.;\phi);\theta)]+\frac{1}{N_2}\sum_{j}^{N_2}[\omega\cdot\left\|\mathbf{a}_j^H-\pi\left(\cdot\left|\mathbf{s}_j\right.;\phi\right)\right\|^2] \tag{23}$$

It is noticeable that the training of the policy network can be delayed stabilizing the algorithm, that is, the actor would be updated once given the critic updating d times.

Lumping all factors, the final algorithm is provided in Algorithm 1.

Algorithm 1 PHIL-TD3

Initialize the maximum training episode number E, and the episode length T.
Initialize the critic networks Q_1, Q_2 with parameter θ_1,θ_2 and the actor network π with parameter ϕ .
Initialize the target networks $\theta_{k=1,2}^{'}\leftarrow\theta_{k=1,2}$, $\phi^{'}\leftarrow\phi$.
Initialize the learning rate lr^Q, lr^{π}, priority coefficient α, and the soft update coefficient τ.
Initialize experience replay buffer $\mathcal{B}\leftarrow\varnothing$.
for $e=1$ **to** E **do**
  Observe the initial state $\mathbf{s}_1$
  for $t=1$ **to** T **do**
    if human intervened **then**
      Adopt human action $\mathbf{a}_t=\mathbf{a}_t^H$, set $\mathbf{\Delta}_t=\mathbf{I}$
    else
      Select RL action $\mathbf{a}_t=\mathbf{a}_t^{RL}=\pi(\cdot\left|\mathbf{s}_t\right.;\phi)+\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{\Sigma})$, set $\mathbf{\Delta}_t=\mathbf{0}$
    end if
    Observe reward r_t and new state $\mathbf{s}_{t+1}$
    Shape reward $r_t=r_t+r_{pen}\cdot[(\mathbf{\Delta}_t=\mathbf{I})\wedge(\mathbf{\Delta}_{t-1}=\mathbf{0})]$
    Store transition tuple $(\mathbf{s}_t,\mathbf{a}_t,r_t,\mathbf{s}_{t+1}^{'},\mathbf{\Delta}_t)$ into $\mathcal{B}$ with maximal priority $\mathcal{p}_t=max_{i<t}\,\mathcal{p}_i$
    Sample a batch of N transition tuples from $\mathcal{B}$, with the probability $p(i)=\mathcal{p}_i^{\alpha}/\sum_k\mathcal{p}_k^{\alpha}$
    Update priority: $\mathcal{p}_i=|\delta_i^{TD}|+\varepsilon+(\mathbf{\Delta}_i=\mathbf{I})\cdot\exp[Q^{'}(\mathbf{s}_i,\mathbf{a}_i;\theta)-Q^{'}(\mathbf{s}_i,\pi(\cdot\left|\mathbf{s}_i\right.;\phi);\theta)]$
    Update the critic networks using the gradient method: $\theta_{k=1,2}\leftarrow\theta_{k=1,2}-lr^{Q_{k=1,2}}\cdot\nabla_{\theta}\mathcal{L}^Q(\theta)$
    if t mod d **then**
      Update the actor network using the gradient method: $\phi\leftarrow\phi-lr^{\pi}\cdot\nabla_{\phi}\mathcal{L}^{\pi}(\phi)$
      Update the target networks: $\theta_{k=1,2}^{'}\leftarrow\tau\theta_{k=1,2}+(1-\tau)\theta_{k=1,2}^{'}$, $\phi^{'}\leftarrow\tau\phi+(1-\tau)\phi^{'}$
    end if
  end for
end for

V. Human Policy Model

In this section, a human policy model is established in conjunction with PHIL-RL. The model can relieve human workload in the human-in-the-loop RL process by imitating the behavior policy of actual human participants.

We train a regression model to imitate human policy simultaneously with RL, and this policy model can substitute humans when necessary. Consider human behaviors in the RL training process: the human participant is required to intervene in the control process when he/she believes the agent poorly behaves. Human interventions are usually imposed to the loop in an intermittent way and demonstrations are incrementally supplemented into the training set (buffer). Thus, we train the human policy model leveraging an online- and incremental-based imitation learning algorithm, i.e., the Data Aggregation (DAgger) [27], which is free from offline large-scale collection of the demonstration data.

It is noted that the human policy model does not aim to accurately mimic expert-level humans. In practice, the common situation is humans who cooperate with RL are non-proficient, and humans' performance can fluctuate with mental and physical status. Thus, we do not require the model to achieve expert-level performance. In essence, the human policy model is to provide roughly correct demonstrations for the RL agent.

Denoting the human policy model with $\mathcal{H}$, the objective is to find a policy $\pi^{\mathcal{H}}$ minimizing its difference $\mathcal{d}$ with the human policy π^H:

$$\pi^{\mathcal{H}}=\arg\min_{\pi}\mathbb{E}_{\mathbf{s}_i}[\mathcal{d}(\mathbf{s}_i,\pi^H)] \tag{24}$$

We initialize model $\mathcal{H}$ by replicating an untrained RL policy network. After the first human-intervention event, model $\mathcal{H}$ is established as:

$$\pi_0^{\mathcal{H}}(\varphi)\leftarrow\pi(\phi) \tag{25}$$

In subsequent episodes, we retrieve human demonstrations to conduct incremental learning with the loss function:

$$\mathcal{L}^{\mathcal{H}}(\varphi)=\mathbb{E}_{(\mathbf{s}_i,\mathbf{a}_i^H)}\left[\left\|\mathbf{a}_i^H-\pi^{\mathcal{H}}(\cdot\left|\mathbf{s}_i\right.;\varphi)\right\|^2\right] \tag{26}$$

and update the model with the gradient method as:

$$\pi_{e+1}^{\mathcal{H}}\leftarrow\pi_e^{\mathcal{H}}-lr^{\pi^{\mathcal{H}}}\cdot\nabla_{\varphi}\mathcal{L}^{\mathcal{H}}(\varphi) \tag{27}$$

where e is the episode number of the RL process.

Through the above update, model $\mathcal{H}$ would gradually be competent to accurately mimic human policy, and accordingly, substitute human participants to assist RL. It is noticeable that if using this human policy model to cooperate with PHIL, the activation conditions of model $\mathcal{H}$ shall be manually defined varying to specific environments.

VI. Problem Formulation

The proposed PHIL-TD3 algorithm, like most RLs, is universally adapted to any continuous-action decision and control tasks. Here we choose the end-to-end autonomous driving problem as the object, evaluating our algorithm in two challenging driving scenarios. It is noticeable that the RL-based autonomous driving problem can be solved by numerous reasonable settings, while the problem formulation in this section is to provide a fair environment for algorithm evaluation and comparison. In this section, two challenging autonomous driving scenarios are introduced to evaluate the control and

optimization performance of the proposed algorithm, then the standard optimization setting is established.

A. Autonomous Driving Scenarios

RL is better suited to the challenging driving tasks compared to rule-based or model optimization-based approaches due to its high representational and generalization capabilities. We choose two scenarios, shown in Fig. 3, to evaluate the RL performance. These scenarios are challenging to conventional autonomous driving strategies due to complex combinatorial relationships.

Unprotected left-turn: This scenario is illustrated in Figs. 3(a-b). The ego vehicle, i.e., the controlled vehicle, in the side road is trying to make a left turn and merge into the main road. No traffic signals guide the vehicles in the intersection. We assume the lateral path of the ego vehicle is planned by other techniques, while the longitudinal control is assigned to the RL agent. Surrounding vehicles are initialized with varying random velocities ranging from [4, 6] m/s and controlled by the intelligent driver model (IDM) [28] to execute lane-keeping behaviors. All surrounding drivers are set with aggressive characteristics, meaning that they would not yield to the ego vehicle. The control interval for all vehicles is set as 0.1 seconds.

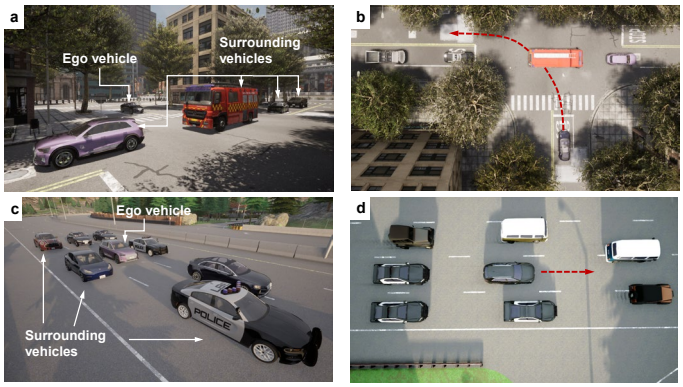

Fig. 3. Task environment configuration. a, the devised unprotected left-turn scenario in T-intersection, established in CARLA. b, the bird-view of the left-turn scenario, where the dotted line indicates a left-turn trajectory. c, the devised congestion scenario in the highway, established in CARLA. d, the bird-view of the congestion scenario, where the dotted line shows a car-following trajectory.

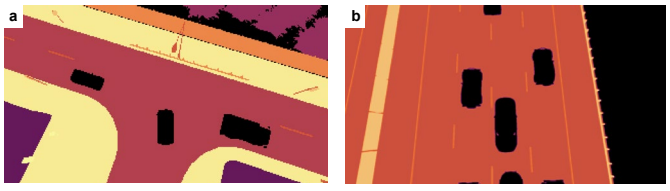

Fig. 4. Bird-view semantic graphs which constitute the state space. a, the left-turn scenario, b, the congestion scenario.

Highway congestion: This scenario is illustrated in Figs. 3(c-d). The ego vehicle is stuck in severe congestion and tightly surrounded by other vehicles; thus, it is trying to shrink the gap with its leading vehicles and conduct the car-following task with the target velocity. We assume the longitudinal control is completed by IDM with a target velocity of 6 m/s, while the lateral control is assigned to the RL agent. Surrounding vehicles are initialized with the velocity ranging from [4, 6] m/s and controlled by IDM to execute car-following behaviors. The control interval for all vehicles is set as 0.1 seconds. The crowded surrounding vehicles cover the lane markings and no specific one leading vehicle in the ego lane, which can lead the conventional lateral-planning approaches to be invalid in such a scenario.

B. RL-based autonomous driving problem definition

1) *State*: The bird-view semantic graphs are taken as the state information for the RL agent, shown in Fig. 4. Two consecutive frame images are used to constitute one state variable to enable temporal perception. We scale the camera-captured image to a smaller size to relieve the computational burden. The state variable can be expressed as:

$$\mathbf{s}_t = \{\mathbf{p}_{t-1}, \mathbf{p}_t | p \in [0,1]\} \tag{28}$$

where $\mathbf{p} \in \mathbb{R}^{45\times80}$ is a pixel matrix of which the elements are normalized.

2) *Action*: The action variable can be either lateral or longitudinal commands adaptive to different requirements. For the lateral control task in the congestion scenario, we choose the angle of the steering wheel as the action, expressed as:

$$\mathbf{a}_t = \{\delta_t | \delta \in [-5\lambda\pi, 5\lambda\pi]\} \tag{29}$$

where $\delta \in \mathbb{R}^1$ is the continuous steering command, of which the negative value indicates a left-turn command and the positive value corresponds to a right-turn command, and λ is the scaling factor that limits the steering range.

For the longitudinal control task in the left-turn scenario, we choose the accelerating/braking pedal aperture, expressed as:

$$\mathbf{a}_t = \{\eta_t | \eta \in [-1,1]\} \tag{30}$$

where $\eta \in \mathbb{R}^1$ is the continuous pedal aperture, of which the negative value indicates a braking command and the positive value corresponds to an accelerating command.

3) *Reward*: The goal of an autonomous vehicle is to rapidly complete traffic scenarios through safe and smooth driving behaviors. RL-based driving strategy achieves this by an appropriate reward function definition. The reward schemes of the two tasks in Fig.3 can be respectively defined as:

$$R^{leftturn}(\cdot\,|\mathbf{s}_t, \mathbf{a}_t) = r_{goal} \cdot \mathbf{1}(\mathbf{s}_t \in \mathcal{S}_{goal}) + r_{fail} \cdot \mathbf{1}(\mathbf{s}_t \in \mathcal{S}_{fail}) + r_{speed}(\mathbf{s}_t) \tag{31}$$

$$R^{congestion}(\cdot\,|\mathbf{s}_t, \mathbf{a}_t) = r_{goal} \cdot 1(\mathbf{s}_t \in \mathcal{S}_{goal}) + r_{fail} \cdot 1(\mathbf{s}_t \in \mathcal{S}_{fail}) + r_{steer}(\mathbf{s}_t) \tag{32}$$

where $r_{goal} = 10$ and $\mathcal{S}_{goal}$ is the set of goal states where the ego vehicle successfully completes the scenario; $r_{fail} = -10$ and $\mathcal{S}_{fail}$ is the set of failure states where the collision occurs; while $r_{speed} = -\|v_{ego} - v_{target}\|$ is the reward that encourages the target speed, i.e., 5m/s set in this paper; $r_{steer} = \|\delta_t - \delta_{t-1}\|$ is the reward that discourages frequent steering behaviors. It is noticeable that both r_{speed} and r_{steer} can implicitly play a role in promoting smooth driving. Additionally, we set the penalty term r_{pen} in Eq. (17) the

same as r_{fail} and incremented it to the above reward when human intervention occurs.

4) *Function approximator*: The function approximators of the value and policy functions are concrete by deep convolutional networks, as shown in Fig. 5.

5) *Auxiliary functions*: We define some auxiliary control functions independent of the RL action to achieve a complete control suit. When RL manipulates the steering wheel, the longitudinal control is achieved by an IDM. When RL manipulates the pedal aperture, the lateral motion target is to track the planned waypoints through a proportional-integral (PI) controller.

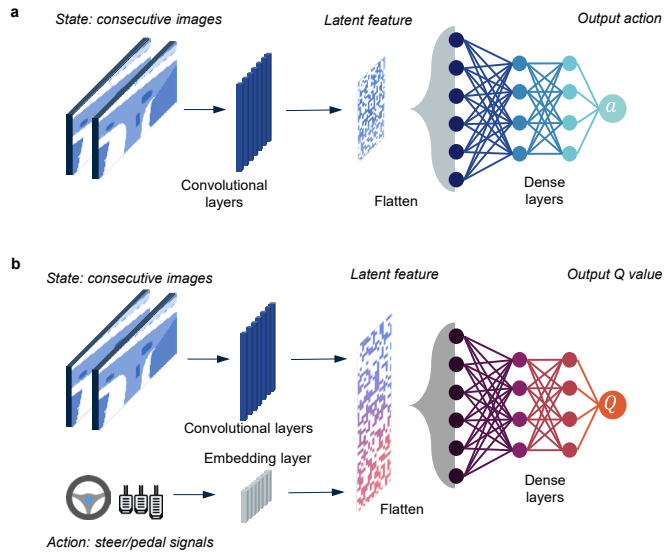

Fig. 5. a, the policy function architecture achieved by the neural network, where the target value network owning the same structure is omitted for brevity. b, the value network architecture achieved by the neural network, where the target value network owning the same structure is omitted for brevity.

VII. Experimental Validation

A. Baseline algorithms

We employ state-of-the-art in the domain of human-involved RL algorithms as baselines and compare their performance against the proposed algorithm.

IA-TD3. This baseline is derived from Intervention Aided Reinforcement Learning (IARL) [19], which is a representative combination of a continuous-action RL algorithm and human demonstration. The RL's policy network is modified to adapt to human demonstrated actions by introducing the behavior cloning objective. Once human intervention happens, the human demonstration will substitute the RL's exploratory action, and a penalty signal will impose on the reward value. In this study, we devise a modified IARL by replacing the on-policy base algorithm (PPO) with TD3, which essentially augmented the algorithm performance by improving the sample efficiency. We also implement the prioritized experience replay (PER) in this baseline for a fair comparison.

HI-TD3. This baseline is derived from Human Intervention Reinforcement Learning (HIRL) [18], which is a combination of a discrete-action RL algorithm and human demonstration. Once human intervention happens, the human demonstration will substitute the RL's exploratory action, and a penalty signal will take on the reward signal. In this study, we devise a modified HIRL by replacing the discrete-action base algorithm (DQN) with TD3, which augmented the algorithm performance by improving the representation and control precision. We also implement the PER in this baseline for a fair comparison.

RD2-TD3. This baseline is derived from Recurrent Replay Distributed Demonstration-based DQN (R2D3) [20], which is a representative combination of PER mechanism and human demonstration. In this study, we devise a modified algorithm by replacing DQN with TD3. The original R2D3 utilizes the recurrent neural network to augment performance, which is not the concerned technique in the context of this paper, thus, we remove the recurrent network structure and only focus on its replay distributed character regarding human demonstrations. Thus, we devise a Replay Distributed Demonstration-based (RD2) TD3 algorithm, which distributes human demonstration and RL exploratory experience into two experience buffers respectively and retrieves experiences by PER. The probability of utilizing human guidance instead of RL exploratory experience is aligned with the ratio of human guidance amount and total data amount.

Additionally, we employ the vanilla PER+TD3 that is shielded from human guidance as an ablated baseline.

B. Experiment setting

Multiple experiments are to evaluate the comprehensive performance of PHIL-TD3 against baselines. First, the training efforts of involved algorithms are comparatively evaluated in the two autonomous driving scenarios. Then the well-trained autonomous driving strategies are tested regarding control performance with several metrics. Last, a series of experiments involving both training and testing stages are conducted to analyze the mechanism of PHIL-TD3.

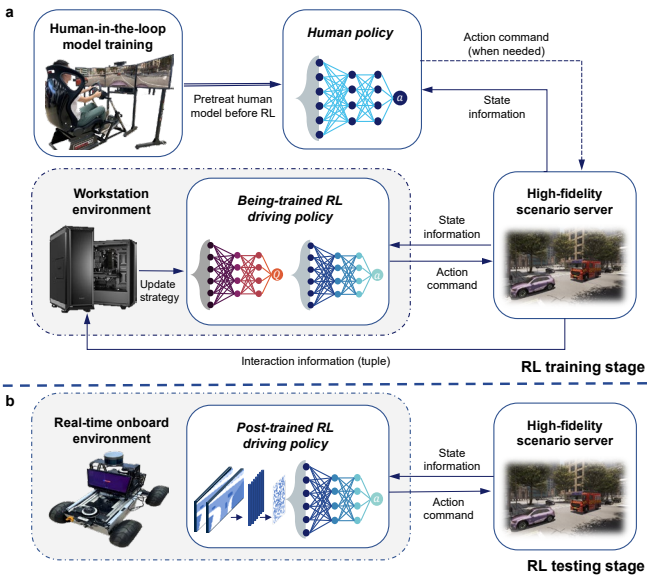

Fig. 6. The experimental workflows. a, the experimental workflow in the RL training stage. The dotted line represents the human policy model that is not

always sending commands. b, the experimental workflow in the RL testing stage.

The training hardware comprises a driving simulator and a high-performance workstation. The driving simulator is utilized to collect human data to train the human policy model complying with Section IV, and the workstation is dedicated to processing RL training. A high-fidelity autonomous driving simulation platform, CARLA [29], is employed to implement driving scenarios and generate RL-environment interaction information. The schematic diagram of the RL training stage is illustrated in Fig. 6(a).

The testing hardware is a robotic vehicle. The post-trained RL policy is implemented on the computation platform of the vehicle, which can communicate with the CARLA server through the wireless network. The on-board RL policy receives state information from CARLA and sends its control command back to remotely complete autonomous driving tasks. The robotic vehicle aims to test whether the RL policy is well-worked under the current onboard calculation and communication situations. The schematic diagram of the RL testing stage is demonstrated in Fig. 6(b).

The detailed configuration of the above experimental platform is provided in Appendix Table I. The algorithms are concreted based on neural networks, of which the architecture is illustrated in Appendix Table II. And the hyperparameters of the algorithms are given in Appendix Table III.

C. Evaluation of training efforts of RL algorithms

In this section, we explore whether human guidance can indeed improve the RL training, and further, which algorithm can achieve the best learning performance given the same human guidance. Additionally, we also investigate the effects of human guidance in dealing with RL tasks of different difficulties.

To eliminate the deviation brought by participant randomness and obtain repeatable results, we use the identical human model (see Section IV) to mimic human guidance behaviors in RL training processes. We fixate the sequence of random seeds and make the triggering conditions of human interventions invariant in all training attempts, which achieves a fair comparison across different algorithms. Two metrics are employed: the average reward of the training episode (excluding intervention-based shaping term), and the surviving distance of the ego vehicle in the training episode before a goal state or failure state in Eq. (31) occurs. A higher value of both metrics indicates a better learning performance.

Fig.7 visualizes the learning performance through curves, represented with a solid line of the mean value and an error band of the standard deviation. We run each algorithm five times in the unprotected left-turn scenario and demonstrate their learning processes in Figs. 7(a-b). The vanilla TD3 is struggling to improve its policy, while the other three algorithms achieve higher rewards and survive distances in a much shorter time, which indicates the effectiveness of human guidance. Among the human-involved algorithms, HI-TD3 performs the slowest learning process suggested by either reward or surviving distance, and IA-TD3 exhibits a faster convergence but with limited asymptotic performance. In opposite, PHIL rapidly seizes the opportunity of human guidance and learns the best asymptotic policy. It is noticeable that PHIL-TD3 achieves the best asymptotic average reward of the baselines in less than 50 episodes, improving the learning efficiency by over 700%. We also run the congestion scenario five times for each algorithm and plot the learning curves in Figs. 7(c-d). The comparable PHIL and IA-TD3 perform better than the other two baselines when considering the reward. While the metric of surviving distance further confirms this advantage and profitably differentiates the algorithm abilities. Specifically, PHIL wins the highest eventual score. IA-TD3 and HI-TD3 manifest comparable levels of asymptotic performance while IA-TD3 has an advantage in learning efficiency. In this scenario, PHIL-TD3 achieves the best asymptotic average surviving distance of the baselines in 220 episodes, improving the learning efficiency by over 120%. Overall, the results in this training session highlight the significant superiority of the proposed algorithm in learning performance.

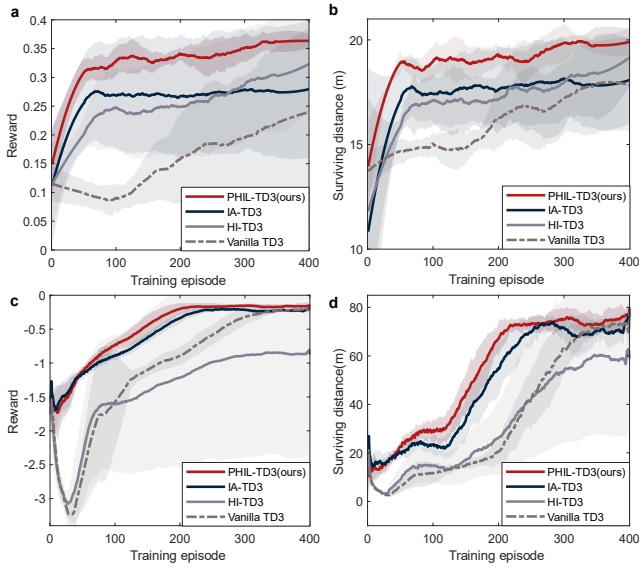

Fig.7. Learning efforts comparison. a-b, curves of training rewards and surviving distances in the left-turn scenario, respectively. c-d, curves of training rewards and surviving distances in the congestion scenario, respectively.

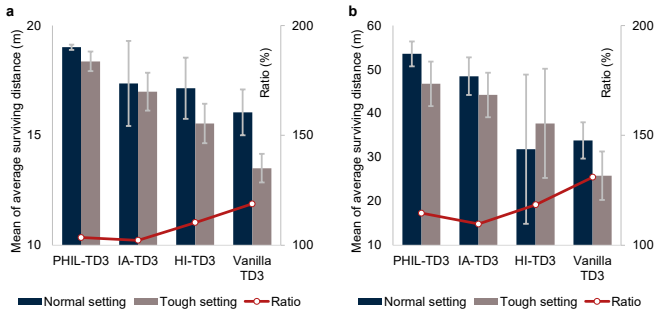

Fig. 8. a, results of the average surviving distance during an entire training session in the left-turn scenario, respectively. b, results of the average surviving distance during an entire training session in the congestion scenario. The ratio denotes the ratio of average surviving distance under normal and tough settings.

We further explore the learning performance of RLs with different task difficulties, which gives rise to Fig. 8. The normal

setting complies with the problem definition in Section V, which is adopted throughout the paper, while the tough setting changes consecutive-frame input of Eq. (28) into a single frame input, impairing the temporal perception ability of RL agents. At the high level, the statistical results of the normal setting are aligned with the trends of Figs. 7(a-d). And it is indicated that the tough setting does not change the performance ranking of algorithms despite the degradations in different degrees. At the detail level, the performance difference between the normal and the tough settings, i.e., the ratios in Fig. 8, can manifest more algorithmic characteristics. Specifically, PHIL-TD3 and IA-TD3, which own the behavior-cloning objective, are less affected by the incomplete problem definition of the tough setting, whereas HI-TD3, and vanilla TD3, which less or not rely on human guidance, are significantly degraded in the same condition. Despite the single-frame state in the autonomous driving task is not fairly reasonable, the findings through this comparison are useful. Since numerous complex real-world tasks are intractable to be well-defined or are only partially observable, the strong integration of human guidance into RL, e.g., behavior-cloning, can play a more remarkable role than pure RL algorithms.

Then, we investigate the contributions of different components in improving the performance of the proposed PHIL algorithm. The results are provided in Appendix Fig. 1. Three components, the behavior cloning objective of Eq. (12) of Section IV-B, the proposed prioritized experience replay mechanism of Section IV-C, and the intervention-based reward shaping mechanism of Section IV-D, are validated to be effective, respectively. The results show that the proposed prioritized human-demonstration replay mechanism plays a crucial role in improving the ultimate performance.

Last, we evaluate the computational efficiency. The CPU clock time of different algorithms is compared in Appendix Table IV. It is shown that the training time consumption of the proposed algorithm is similar to that of IA-TD3. This is because the proposed priority calculation scheme consumes very few computational resources. In all, the proposed PHIL-TD3 greatly improves the training efficiency and performance without requiring significantly higher computational resources.

D. Evaluation of testing performance of RL-based driving strategies

In this section, the post-trained driving strategies are tested in terms of autonomous driving performance, adaptiveness, and robustness, which can further evaluate the practicality of the above algorithms.

The zero-mean Gaussian noises, of which the standard deviation is 5% of the whole control domain, are injected to output commands of the driving strategies to test the robustness. More types and amounts of surrounding vehicles are added to construct variant scenarios to test the adaptiveness. We conduct 50 runs with the same sequence of random seeds for each post-trained strategy in each scenario. The success rate, which is defined as the number of completed runs divided by the total attempts in the same scenario, is taken as the metric for evaluating the safety performance in Fig. 9(a). Our PHIL-TD3 achieves the highest success rate in all scenarios, showing its superior task-completeness abilities. The vanilla TD3, albeit with its unstable training performance, performs competitively like IA-TD3 and HI-TD3 in the testing stage. Considering the two trained scenes (rows 1, 4) and noise-injected scenes (rows 2, 5), three baseline strategies behave acceptably, nevertheless, the scenario variants (rows 3, 6) significantly degrade their safety. Our PHIL, instead, maintains the highest ability regardless of varying testing conditions, manifesting itself with good robustness and adaptiveness. In Figs. 9(b-c), PHIL-TD3 once again shows its superiority in safety by the highest average surviving distance, and importantly, its performance stability is confirmed due to the lowest variance.

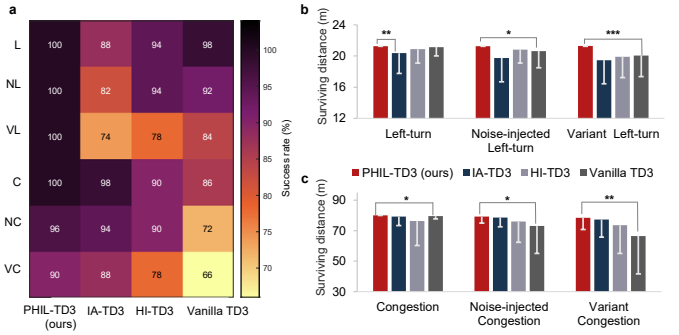

Fig. 9. The driving performance of different RL strategies under six autonomous driving scenarios. The two noise-injected scenarios and two variant scenarios are different with the two training scenarios, which can examine the robustness and adaptiveness, respectively. "C" and "L" refer to congestion scene and left-turn scene, respectively, while "N" and "V" denote noise-injected and variant scene, respectively. a, the heatmap of success rate. b, the barplot of surviving distance in the left-turn scenarios. The theoretical maximum surviving distance of the scenario is 21 meters. The error bar describes the standard deviation. c, the barplot of surviving distance in the congestion scenarios. The theoretical maximum surviving distance of the scenario is 80 meters. The error bar describes the standard deviation. The paired t-test is adopted for the statistical test.

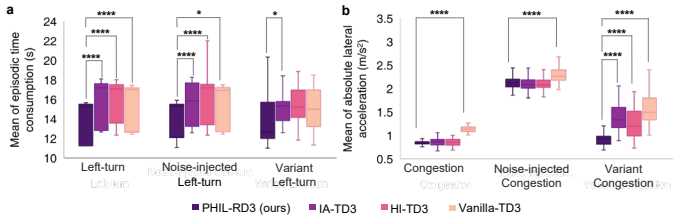

Fig. 10. a, the boxplot of time consumption of the episode without failure in the left-turn scenarios. b, the boxplot of average lateral acceleration of the episode in the congestion scenarios. The paired t-test is adopted for the statistical test.

Fig. 10 can further evaluate the detailed performance of driving strategies. Time consumption of the episode is the secondary target of RL optimization in the left-turn task, which is implied in the reward function of Eq. (31); thus, the related boxplot is illustrated in Fig. 10(a) to access this objective. It is found that the proposed strategy enjoys minimal time consumption, which is significantly different from other candidates. In congestion tasks, smoothness is the secondary target of the reward function of Eq. (32); thus, we choose the lateral acceleration as the smoothness measure and provide the associated boxplot in Fig. 10(b). The comparable human-

involved strategies show their superior smoothness to vanilla TD3 in the training and noise-injected scenes, while the variant congestion scenario profitably validates the advantage of PHIL-TD3.

Additionally, we compare the performance of three human guidance-based RL algorithms to the human guidance itself. Specifically, the surviving distance of these human-involved RLs are compared with the human policy model, and the results are provided in Table I. The results suggest the superiority of the proposed PHIL-TD3 over the human policy model.

Overall, our PHIL-TD3 perpetuates its predominance of training performance and takes the top spot in the testing stage.

Table I. Comparison of the surviving distance of human-related driving strategies. Mean and standard deviation are calculated by 50 evaluation seeds.

Surviving distance, meter, ↑	Left-turn	Noise-injected Left-turn	Variant Left-turn	Congestion	Noise-injected Congestion	Variant Congestion
PHIL-TD3 (ours)	**21.28±0.02**	**21.27±0.02**	**21.29±0.02**	**80.15±0.08**	**79.26±4.30**	**77.29±11.55**
IA-TD3	20.37±2.58	19.75±3.05	19.46±3.03	79.26±5.90	78.64±6.11	78.39±7.66
HI-TD3	20.87±1.76	20.63±1.74	19.90±2.65	76.27±16.00	76.02±13.72	73.57±18.49
Human policy model	20.70±1.82	20.88±1.32	20.90±1.21	80.11±0.07	77.66±12.27	75.15±15.20

E. Discussion on prioritized human experience utilization mechanism

In this subsection, we explore the effect of PHIL-RL from three aspects: the performance improvement by the TD*Q*A mechanism, the merit of the single-buffer experience replay structure, and the algorithmic robustness to bad demonstrations.

TD*Q*A, as the crucial innovation of PHIL-TD3, can improve learning performance in the context of human guidance-based RLs, as suggested in the above two sections. More specifically, it establishes a novel priority indicator to deal with various human guidance. Thus, we first evaluate TD*Q*A by comparing different priority schemes. "*Q*-adv" represents the scheme in which the priority of human guidance is calculated based only on *Q*-advantage. "TD", i.e., temporal difference, the scheme is inherited from the original PER method, but the TD weights of human demonstrations in it are doubled to highlight the human guidance in the replay buffer.

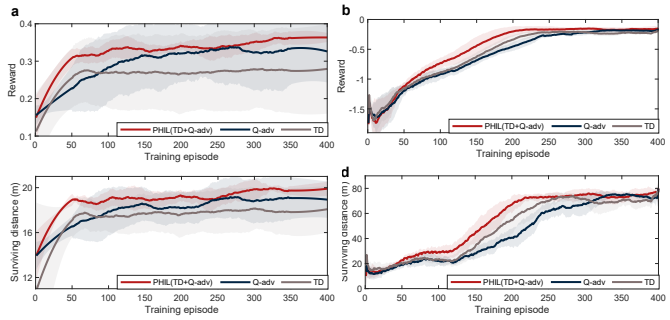

Fig. 11. Learning efforts of different experience replay mechanisms with different priority indicators. a-b, the training rewards algorithms in the left-turn and congestion scenario, respectively. c-d, the training surviving distances of algorithms in the left-turn and congestion scenario, respectively.

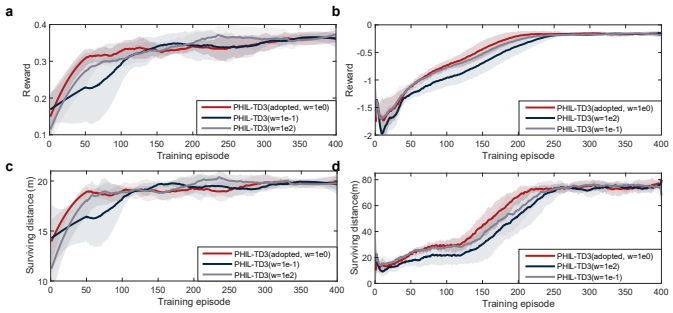

Fig. 12. Learning efforts of different experience replay mechanisms with different weighting schemes. a-b, the training rewards algorithms in the left-turn and congestion scenario, respectively. c-d, the training surviving distances of algorithms in the left-turn and congestion scenario, respectively.

Five learning attempts are conducted with the same sequence of random seeds for each candidate, and the corresponding learning curves are in Fig. 11. We find scheme comparison in two training scenarios shows similar trends when observing results in Figs. 11(a-b) and Figs. 11(c-d). The pure TD scheme learns faster than pure *Q*-advantage in both scenarios, yet its asymptotic scores (both reward and surviving distance) are significantly lower than those of the *Q*-advantage scheme. To be more specific, we evaluate different weigh of "TD" and "*Q*-adv" and provide the learning performance in Fig. 12. Under the same TD, *Q*-advantage is weighted with three importance levels. In particular, the equal weighting scheme, i.e., w=1e0, is the adopted default scheme in the paper, whereas the other two variants are for comparison. It is shown that a larger TD (w=1e-1) makes faster convergence but can lead to unfavorable asymptotic performance, while a larger *Q*-advantage (w=1e2) can achieve the same-level performance as the default setting, despite sometimes slower learning process. The above results, reveal the same performance trends as Fig. 11. That is, TD error accelerates the convergence speed and *Q*-advantage contributes to improving convergence performance.

Essentially, these two schemes score human guidance based on different indicators, and a better indicator can provide RL with more high-quality guidance to improve learning efficiency. Thus, we find TD indicator, as proved in conventional PER, is indeed beneficial to rapidly improve performance, nonetheless, the *Q*-advantage indicator is superior to the TD indicator in the later stage of the training process. The delayed superiority of *Q*-advantage complies with intuition since unlike the direct indicator as TD, the evaluation ability of the *Q* network, i.e., the source of *Q*-advantage, also needs to be trained. The proposed PHIL, which smartly combines both indicators, achieves the most favorable performance in the two scenarios, showing the effectiveness of the TD*Q*A mechanism.

PHIL puts the human guidance and exploratory experience of RL into the same experience replay buffer. This structure differs from the double distributed scheme which is represented by R2D3. To evaluate the performance of these two schemes under the devised autonomous driving tasks, RD2-TD3 is developed which utilizes TD as the indicator to respectively retrieve data from two buffers. Additionally, the TD*Q*A priority mechanism is ported to the RD2-TD3 setting forming the other

variant, RD2-PHIL. Five learning attempts with the same sequence of random seeds are conducted by RD2-TD3 and RD2-PHIL. Through learning curves in Figs. 13(a-d), it is found that the double distributed buffer scheme, i.e., RD2-PHIL, fails to achieve the same level of learning efficiency as the proposed PHIL. A possible reason behind this is that human guidance can only be utilized in a chunk way under the double-buffer setting, whereas the single buffer scheme of PHIL is more flexible and friendly to small-scale human guidance data. The conventional RD2-TD3 is least favorable, which is within expectation due to the lack of the TD*Q*A mechanism. To sum up, the results in Fig. 13 support the single-buffer structure utilized in the PHIL-TD3, and profitably suggest the effectiveness of the proposed TD*Q*A mechanism.

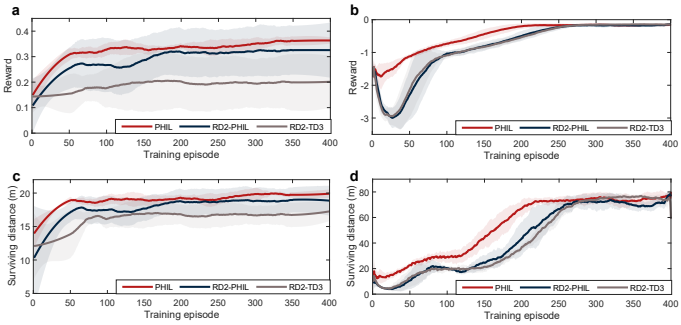

Fig. 13. a-b, the training rewards of algorithms with different experience replay structures in the left-turn scenario and congestion, respectively. c-d, the training surviving distances of algorithms with different experience replay structures in the left-turn and congestion scenario, respectively.

A general situation occurs that human guidance is not perfect, and thus an unqualified human participant can sometimes conduct actions that are harmful to the task. We test if the unfavorable guidance of the unqualified human would impair the learning process, that is, evaluating the robustness to harmful guidance. It is noticeable that the robustness discussed here is distinguished from that in Section VI-D: we discuss how the algorithms are affected by poor guidance instead of the anti-noise ability of post-trained driving strategies. The human intervention condition of the training stage keeps the same as foregoing experiments, while one-third of the demonstrations from the human model are replaced with random actions to simulate non-proficient human behaviors.

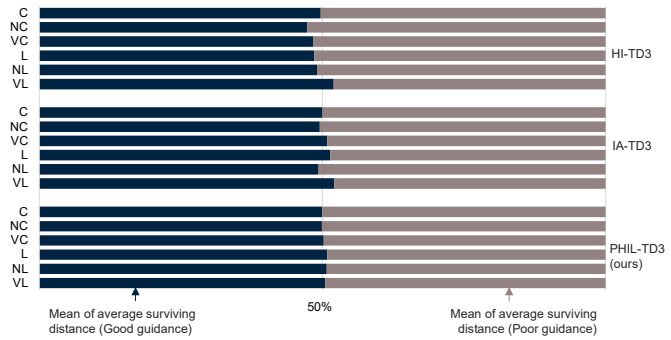

Fig. 14. The stacked barplot of the surviving distance of different human-guidance-based RL strategies under good/poor guidance in all scenarios.

Post-trained driving strategies under poor guidance are tested to conduct 50 runs in each scenario and are compared with those under the good guidance of Fig. 9. The stacked barplots in Fig. 14 provide the adversarial testing performance of three human-guidance-based RL algorithms under good and poor guidance. We take the average surviving distance as the metric and the less performance deterioration by poor guidance suggests better robustness. Our PHIL-TD3 exhibits good performance since a nearly 50:50 situation occurs in all six scenarios. IA-TD3 falls behind with a 2.1% degradation on average in poor guidance context, while HI-TD3 is even improved by an average of 3.6% extent given poor guidance. Intuitively, poor guidance would remarkably degrade PHIL and IA-TD3 since they utilize the behavior-cloning objective to learn from human guidance, while HI-TD3, which only substitutes partial RL explorations with human guidance, can be less affected. The not-degraded HI-TD3 and most-degraded IA-TD3 support the above idea. Our PHIL defeating IA-TD3 is attributed to the TD*Q*A mechanism: *Q*-advantage well access the quality of human demonstrations and feed more high-quality demonstrations to the RL agent; accordingly, the agent learns greater from good guidance than negative guidance. The secondary optimization target of RL, i.e., driving smoothness, is evaluated in Fig. 15 by acceleration distribution. The proposed PHIL-TD3 wins all scenarios by the most favorable smoothness which further confirms the abovementioned superiority.

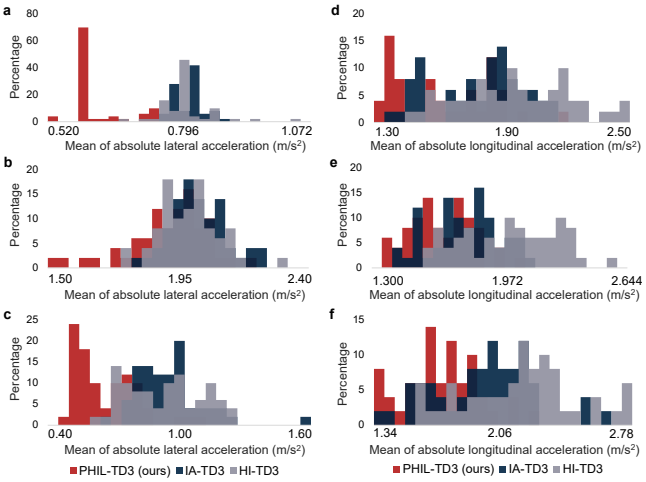

Fig. 15. a-c, the frequency distribution plot of the average absolute value of the longitudinal acceleration in the left-turn scene, noise-injected left-turn scene, and variant left-turn scene, respectively. The smaller acceleration indicates a better driving smoothness. d-f, t he frequency distribution plot of the average absolute value of the lateral acceleration in the congestion scene, noise-injected congestion scene, and variant congestion scene, respectively. The smaller acceleration indicates a better driving smoothness.

Overall, the TD*Q*A mechanism, as the core innovation of the PHIL-RL algorithm, contributes to the preponderant learning performance through its unique discriminatory power on the quality of human guidance. It also improves the robustness to poor guidance, which can relieve the requirement on the qualification of human guidance. Additionally, the single buffer setting is more favorable than the double distributed buffer scheme under autonomous driving tasks of this paper.

VIII. Conclusion

In this paper, we establish a human-guidance-based reinforcement learning framework and propose a novel experience utilization mechanism of human guidance. Based on that, we put forward an algorithm, PHIL-TD3, aiming at improving algorithmic abilities in the context of human-in-the-loop RL. We also introduce a human behavior modeling mechanism to relieve the human workload. PHIL-TD3 is employed to solve two challenging autonomous driving tasks, and its performance is comparatively evaluated against state-of-the-art human-guidance-based RLs as well as the non-guidance baseline. Three main points are obtained through experimental results:

1) The proposed PHIL-TD3 can improve the learning efficiency by over 700% and 120% under the adopted two situations, respectively, and achieve remarkably higher asymptotic performance compared to state-of-the-art human-guidance-based RLs.

2) The proposed PHIL-TD3 achieves the most favorable performance, robustness, and adaptiveness in a series of metrics under the adopted two challenging autonomous driving tasks.

3) The proposed TD*Q*A mechanism prominently contributes to the advance of PHIL-TD3 and can well discriminate the quality of various human guidance to relieve humans by less requiring on human proficiency.

Future work will focus on testing the performance of the proposed algorithm in more realistic scenarios, e.g., the real-world autonomous driving tasks.

References

[1] D. Silver *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.

[2] X. Gao, J. Si, Y. Wen, M. Li, and H. Huang, "Reinforcement Learning Control of Robotic Knee With Human-in-the-Loop by Flexible Policy Iteration," *IEEE Transactions on Neural Networks and Learning Systems,* 2021.

[3] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018: IEEE, pp. 2034-2039.

[4] J. Wu, Z. Wei, W. Li, Y. Wang, Y. Li, and D. U. Sauer, "Battery thermal-and health-constrained energy management for hybrid electric bus based on soft actor-critic DRL algorithm," *IEEE Transactions on Industrial Informatics,* vol. 17, no. 6, pp. 3751-3761, 2020.

[5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[6] M. Hessel *et al.*, "Rainbow: Combining improvements in deep reinforcement learning," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347,* 2017.

[8] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, 2018: PMLR, pp. 1861-1870.

[9] E. O. Neftci and B. B. Averbeck, "Reinforcement learning in artificial and biological systems," *Nature Machine Intelligence,* vol. 1, no. 3, pp. 133-143, 2019.

[10] M. L. Littman, "Reinforcement learning improves behaviour from evaluative feedback," *Nature,* vol. 521, no. 7553, pp. 445-451, 2015.

[11] M. Vecerik *et al.*, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *arXiv preprint arXiv:1707.08817,* 2017.

[12] T. Hester *et al.*, "Deep q-learning from demonstrations," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[13] G. Libardi, G. De Fabritiis, and S. Dittert, "Guided exploration with proximal policy optimization using a single demonstration," in *International Conference on Machine Learning*, 2021: PMLR, pp. 6611-6620.

[14] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *nature,* vol. 518, no. 7540, pp. 529-533, 2015.

[15] S. Krening, B. Harrison, K. M. Feigh, C. L. Isbell, M. Riedl, and A. Thomaz, "Learning From Explanations Using Sentiment and Advice in RL," *IEEE Transactions on Cognitive and Developmental Systems,* vol. 9, no. 1, pp. 44-55, 2017.

[16] J. MacGlashan *et al.*, "Interactive learning from policy-dependent human feedback," in *International Conference on Machine Learning*, 2017: PMLR, pp. 2285-2294.

[17] B. a. L. Ibarz, Jan and Pohlen, Tobias and Irving, Geoffrey and Legg, Shane and Amodei, Dario, "Reward learning from human preferences and demonstrations in Atari," *Advances in Neural Information Processing Systems,* vol. 31, 2018.

[18] W. Saunders, G. Sastry, A. Stuhlmüller, and O. Evans, "Trial without Error: Towards Safe Reinforcement Learning via Human Intervention," presented at the Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, Stockholm, Sweden, 2018.

[19] F. Wang *et al.*, "Intervention Aided Reinforcement Learning for Safe and Practical Policy Optimization in Navigation," *the Proceedings of The 2nd Conference on Robot Learning*, Proceedings of Machine Learning Research, 2018.

[20] C. Gulcehre *et al.*, "Making efficient use of demonstrations to solve hard exploration problems," in *International Conference on Learning Representations*, 2019.

[21] E. Senft, S. Lemaignan, P. E. Baxter, M. Bartlett, and T. Belpaeme, "Teaching robots social autonomy from in situ human guidance," *Science Robotics,* vol. 4, no. 35, 2019.

[22] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018: IEEE, pp. 6292-6299.

[23] A. Rajeswaran *et al.*, "Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[24] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952,* 2015.

[25] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems,* vol. 57, no. 5, pp. 469-483, 2009.

[26] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*, 2018: PMLR, pp. 1587-1596.

[27] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret

online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011: JMLR Workshop and Conference Proceedings, pp. 627-635.

[28] M. Liebner, M. Baumann, F. Klanner, and C. Stiller, "Driver intent inference at urban intersections using the intelligent driver model," in *2012 IEEE Intelligent Vehicles Symposium*, 2012: IEEE, pp. 1162-1167.

[29] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on robot learning*, 2017: PMLR, pp. 1-16.

[30] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Icml*, 1999, vol. 99, pp. 278-287.

APPENDIX

Theorem A1. ***Policy optimality invariance under the proposed human intervention-based reward shaping technique.*** *Let the intervention-based reward shaping function $F: S \times A \times S \to \mathbb{R}$ add a negative constant to the human intervened state as Eq. (17), if the human intervention will certainly occur at state $\mathbf{s}_t$ when the next state $\mathbf{s}_{t+1}$ is unacceptable, then the reward shaping function F does not change the policy optimality.*

Proof. According to [30], the potential-based reward shaping function $F: S \times A \times S$ is proven to be the only form that can preserve policy optimality. Specifically, F is represented as:

$$F(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) = \gamma\Phi(\mathbf{s}_{t+1}) - \Phi(\mathbf{s}_t) \quad \text{(A1)}$$

where $\Phi: S \to \mathbb{R}$ is called the potential function defined over the state space. Thus, the proof converts to construct the potential function Φ.
Define the potential function Φ as:

$$\Phi(\mathbf{s}_t) = \begin{cases} \frac{r_{pen}}{\gamma} & if\ \mathbf{s}_t\ is\ unacceptable \\ 0 & otherwise \end{cases} \quad \text{(A2)}$$

Then, when humans intervene in the state $\mathbf{s}_t$ (meaning $\mathbf{s}_{t+1}$ is unacceptable), F becomes:

$$F(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) = \gamma\Phi(\mathbf{s}_{t+1}) - \Phi(\mathbf{s}_t) = \frac{r_{pen}}{\gamma} \cdot \gamma - 0 = r_{pen} \quad \text{(A3)}$$

And when humans do not intervene the state, F becomes:

$$F(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) = \gamma\Phi(\mathbf{s}_{t+1}) - \Phi(\mathbf{s}_t) = 0 - 0 = 0 \quad \text{(A4)}$$

Lumping (A3) and (A4), F turns into the reward-shaping term of Eq. (17), shown as:

$$r_t^{shape} = r_t + F(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) = r_t + r_{pen} \cdot [(\Delta_t = \mathbf{1}) \wedge (\Delta_{t-1} = \mathbf{0})] \quad \text{(A5)}$$

where $[(\Delta_t = \mathbf{1}) \wedge (\Delta_{t-1} = \mathbf{0})]$ refers to the intervention event of the human. Hence, we complete the proof of the optimality of the proposed human intervention-based reward shaping, i.e., Eq. (17).

Remark 1: the theorem A1 is established on the below assumptions: humans are considered to owe invariant judgment on the environment state. In this manner, the Φ can be seen as a stable function defined in the state space.

Remark 2: The assumption of Remark 1 is hard to be maintained in practice. This is because 1) the varying mental and physical status of one specific human participant would affect its accurate judgment on the environment state; 2) the judgment on the environment will be varying across different human participants; 3) the state space in the context of deep networks (the image-based one in our manuscript) is intractable to be identified by humans accurately.

Table A-I. Configuration of the experimental platform

Type	Description	Details
Workstation	Operation system	*Ubuntu 20.04*
	CPU + RAM	*AMD Ryzen 3900X + 32GB*
	GPU	*NVIDIA RTX 2080S*
Driving simulator	Scenario software	*CARLA*
	Steering wheel suit	*Logitech G29*
	Displays	*Joint heads-up monitors×3*
	Other equipment	*Driver seat suit*
Robotic vehicle	Vehicle brand	*Wheeled UGV-Hunter*
	Size dimension	*1000mm × 740mm × 400mm*
	Communication type	*ROS publisher-subscriber*
	Calculation board	*Xavier NX Dev Kit*
Other	Programming	*Python*
	Neural network toolbox	*Pytorch*

Table A-II. The architecture details of the neural networks.

A. Architecture and details of the value neural network (critic), applied to all involved RL algorithms.

Parameter	Value
Input (state + action) shape	[80,45,2] + [1]
Network convolution Filter feature	[6,16] (kernel size 6 × 6)
Network pooling feature	Maxpooling (Stride 2)
Network fully connected layer feature	[256,128,64]

B. Architecture and details of the policy neural network (actor), applied to all involved RL algorithms.

Parameter	Value
Input (state) shape	[80,45,2]
Network convolution Filter feature	[6,16] (kernel size 6 × 6)
Network pooling feature	Maxpooling (Stride 2)
Network fully connected layer feature	[256,128,64]

c. Architecture and details of the DAgger-based human policy model

Parameter	Value
Input (state) shape	[80,45,1]
Network convolution Filter feature	[6,16] (kernel size 6 × 6)
Network pooling feature	Maxpooling (Stride 2)
Network fully connected layer feature	[256,128,64]

Table A-III. Hyperparameters settings of the algorithms.

a. Parameters for RL, which are universally used in all involved algorithms.

Parameter	Description	Value
Maximum episode	Cutoff episode number of the training process	400
Minibatch size (N)	Capacity of minibatch	128
Actor learning rate	Initial learning rate (policy/actor networks)	5e-4
Critic learning rate	Initial learning rate (value/critic networks)	2e-4
Learning rate decay	Delay of learning rate (per episode)	0.996
Activation function	Activation function of the networks	Relu
Initial exploration	Initial exploration rate of noise in ϵ greedy	1
Final exploration	Cutoff exploration rate of noise in ϵ greedy	0.05
Gamma (γ)	Discount factor of the Bellman equation	0.95
Soft updating factor	Parameter update frequency to target networks	1e-3
Noise scale (ϵ)	Noise amplitude of action in TD3	0.2
Bounding box (c)	Bounding of the exploratory action in TD3	1
Policy delay (d)	Update frequency of critic over actor	1

b. Hyperparameters for the PER mechanism, which are universally used in all involved algorithms.

Parameter	Description	Value
Replay buffer size	Capacity of PER buffer	1e5
Priority factor (α)	Priority scaling factor	0.6
Sample factor (β)	Importance sampling correlation	1
Offset factor (ε)	Tiny constant avoiding zero retrieving probability	1e-3

c. Hyperparameters for DAgger-based human policy model.

Parameter	Description	Value
Learning rate	Initial learning rate	1e-4
Activation function	Activation function of the network	Relu
Maximum episode	Cutoff episode number of the training process	50
Batch size	Capacity of minibatch	128

Table A-IV. The CPU clock time of 10000 training steps of algorithms.

Algorithm	Time consumption (s) per 10000 steps
PHIL-TD3	360.70
IA-TD3	348.71
HI-TD3	328.93
Vanilla-TD3	329.39

Note: The comparison only takes the time consumption of the algorithm (network) training and does not count the interaction stage with environment.

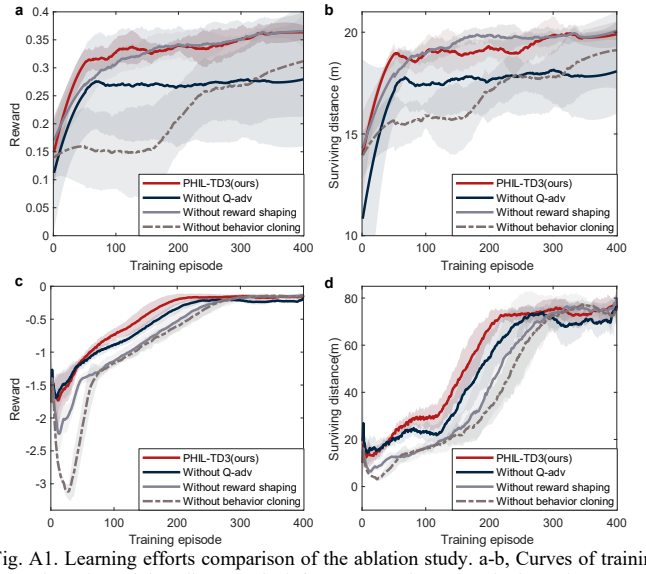

Fig. A1. Learning efforts comparison of the ablation study. a-b, Curves of training rewards and surviving distances in left-turn scenario, respectively. c-d, Curves of training rewards and surviving distances in congestion scenario, respectively.