Event-Triggered Deep Reinforcement Learning Using Parallel Control: A Case Study in Autonomous Driving

Jingwei Lu, Liyuan Han, Qinglai Wei, *Senior Member, IEEE*, Xiao Wang, *Senior Member, IEEE*, Xingyuan Dai, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—This paper utilizes parallel control to investigate the problem of event-triggered deep reinforcement learning and develops an event-triggered deep Q-network (ETDQN) for decision-making of autonomous driving, *without training an explicit triggering condition*. Based on the framework of parallel control, the developed ETDQN incorporates information of actions into the feedback and constructs a dynamic control policy. First, in the realization of the dynamic control policy, we integrate the current state and the previous action to construct the augmented state as well as the augmented Markov decision process. Meanwhile, it is shown theoretically that the goal of the developed dynamic control policy is to learn the variation rate of the action. The augmented state contains information of the current state and the previous action, which enables the developed ETDQN to directly design the immediate reward considering communication loss. Then, based on dueling double deep Q-network (dueling DDQN), we establish the augmented action-value, value, and advantage functions to directly learn the optimal event-triggered decision-making policy of autonomous driving without an explicit triggering condition. It is worth noticing that the developed ETDQN applies to various deep Q-networks (DQNs). Empirical results demonstrate that, in event-triggered control, the developed ETDQN outperforms dueling DDQN and reduces communication loss effectively.**

***Index Terms*—Autonomous driving, deep reinforcement learning, deep Q-network, event-triggered control, parallel control.**

I. Introduction

Autonomous driving has been brought to the fore in recent years with the rapid rise of information technologies [1], [2], [3], [4], [5], [6]. In addition to urban scenarios, autonomous driving is also employed in several enclosed scenarios, such as area mining, airport logistics, and port logistics. Fig. 1 presents an example of autonomous driving in mining areas. In general, the following four modules make up the majority of an autonomous vehicle: perception and localization, high-level path planning, behavior arbitration (or low-level path planning), and motion control [1], where the path planning, including the low-level and high-level ones, can be considered as decision-making problems. The decision-making system of an autonomous vehicle takes actions based on the surroundings and the states of the autonomous vehicle [2]. Therefore, the decision-making of the autonomous vehicle directly impacts the performance of autonomous driving and is of great research significance. Deep reinforcement learning (DRL), as a showpiece technique of deep learning, has become an overarching technique for achieving human-level control and decision-making [7], [8], [9], [10] and has achieved abundant achievements in autonomous driving [11], [12], [13], [14].

Event-triggered control (ETC) has caught the attention of researchers in the past decade [15], [16], [17]. ETC aims to reduce the communication between agents (or controllers) and actuators by devising a time-varying sampling period mechanism. Different from the traditional time-triggered control (TTC) with sampling instants determined by a fixed sampling period, sampling instants in ETC are decided by elaborate events, that is, the action (or control) is updated only after a specific event occurs, while the action remains unchanged between two events. As a result, ETC exhibits great potential for copying with complex systems or limited bandwidth problems. From another point of view, ETC is also a class of anthropomorphic control mechanisms. Assuming that we drive a vehicle, we do not constantly change our driving speed or steering in most cases, but only take action when a specific event occurs. For example, if there is a slow-moving vehicle in front of our vehicle and we need to slow down

Manuscript received 1 March 2023; revised 20 March 2023; accepted 23 March 2023. Date of publication 27 March 2023; date of current version 19 May 2023. This work was supported in part by the Key Research and Development Program 2020 of Guangzhou under Grant 202007050002, in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B090921003, in part by the National Natural Science Foundation of China under Grant U1811463, and in part by the Motion G, Inc. Collaborative Research Project for Modeling, Decision and Control Algorithms of Servo Drive Systems. *(Corresponding author: Jingwei Lu.)*

Jingwei Lu is with the Parallel Intelligence Innovation Research Center, Qingdao Academy of Intelligent Industries, Qingdao 266109, China, and also with the State Key Laboratory for Management and Control of Complex Systems, Chinese Academy of Sciences, Beijing 100190, China (e-mail: lujingweihh@gmail.com).

Liyuan Han, Qinglai Wei, Xingyuan Dai, and Fei-Yue Wang are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: hanliyuan2019@ia.ac.cn; qinglai.wei@ia.ac.cn; xingyuan.dai@ia.ac.cn; feiyue@ieee.org).

Xiao Wang is with the School of Artificial Intelligence, Anhui University, Hefei 230031, China, and also with the Qingdao Academy of Intelligent Industries, Qingdao 266114, China (e-mail: xiao.wang@ahu.edu.cn).

Source code of this paper is available at: https://github.com/lujingweihh/Event-triggered-deep-Q-network.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2023.3262132.

Digital Object Identifier 10.1109/TIV.2023.3262132

Fig. 1. Autonomous driving in mining areas (Provided by Waytous).

or overtake it. In ETC, we usually devise a triggering condition to determine mathematically whether the event has occurred, and the triggering condition is usually devised as a function of the difference between the current state and the state of the last sampling instant [18]. Specifically, an event will be triggered if the current state and the state of the last sampling instant diverge beyond a predetermined threshold. Based on the above idea, through the stability analysis of the closed-loop system under the ETC mechanism, ETC has achieved a large number of results in the control field [16], [17]. Reinforcement learning (RL) and optimal control, especially discrete-time optimal control, are closely related and both are usually based on Markov decision processes (MDPs) and dynamic programming [9], so ETC and DRL have gradually integrated and produced a few event-triggered DRL (ETDRL) methods in recent years [19], [20]. These ETDRL methods eschew the stability analysis and utilize DRL to train triggering conditions.

As aforementioned, most of the existing ETC methods, including the methods based on the stability analysis of the closed-loop system and DRL, are implemented along the lines of designing triggering conditions. In addition, the current ETDRL works focus on training triggering conditions using DRL, and their control policy is irrelevant to the DRL used for training triggering conditions, or the control policy is designed without using DRL. Meanwhile, the triggering condition and the control policy are decoupled and do not share deep neural network (DNN) parameters. To further integrate DRL and ETC as well as to perform high-level ETC for autonomous driving, this paper develops a novel ETDRL method *without training an explicit triggering condition*, which is referred to as the event-triggered deep Q-network (ETDQN), and applies the developed ETDQN to decision-making of autonomous driving. The main novelties are as follows.

1) Different from traditional state-feedback control policies, the developed ETDQN incorporates information of actions into the feedback and constructs a dynamic control policy using parallel control.
2) To realize the dynamic control policy, the augmented state and the augmented MDP (AMDP) are developed by integrating the current state and the previous action. We theoretically show that the goal of the dynamic control policy is to learn the variation rate of the action.
3) The augmented action-value function, which contains information of the current and previous actions, is developed and enables the RL agent to learn the optimal ETC policy directly without training an explicit triggering condition. Since the augmented action-value function contains information of two actions, we refer to it as the *double action-value function*.
4) Instead of comparing the difference between states to implement ETC [19], [20], the developed ETDQN determines whether the previous action applies to the current state through the augmented action-value function and then realizes ETC, which means that the augmented action-value function contains an "implicit triggering condition". Meanwhile, the implicit triggering condition and the control policy share DNN parameters.
5) To the best of our knowledge, it is the first time that an ETC method without an explicit triggering condition is developed and an event-triggered decision-making policy is developed for autonomous driving. In the meantime, ETC is discussed primarily at the decision-making level.

Notations: $\mathbb{R}$ and $\mathbb{N}$ are the sets of real numbers and natural numbers, respectively. $\mathbb{R}^n$ is the space of real n-vectors. $\mathbb{R}^{n\times m}$ presents the space of real $n \times m$ matrices. $\mathbb{I}$ denotes the indicator function. $\mathbb{E}[G]$ is the expectation of a random variable G. The superscript T denotes the transposition symbol. For a matrix $A = (a_{ij})_{m\times n} \in \mathbb{R}^{m\times n}$, $\text{vec}(A) = [a_1^\mathsf{T}, a_2^\mathsf{T}, \ldots, a_m^\mathsf{T}]^\mathsf{T}$ with $a_i \in \mathbb{R}^n$ being the ith column of A, $i = 1, 2, \ldots, m$. For two sets X and Y, $X \oplus Y$ denotes their Cartesian product.

II. Related Work

This section briefly introduces the research progress of related techniques, including DRL, parallel control, and ETC.

As aforementioned, DRL has become a prominent technique for decision-making of autonomous driving. In DRL, deep Q-network (DQN) and its improved versions are key techniques to tackle decision-making problems based on MDP, where the DRL agent interacts with the environment and learns actions to maximize the reward [9]. In what follows, we give a brief introduction. In [7], DQN was proposed by introducing the experience replay buffer and the target network to stabilize the training. In [21], double DQN (DDQN) was developed to mitigate over-optimistic value estimates of the action-value function. In [22], dueling DDQN was proposed by estimating the value function and the state-dependent action advantage function separately, achieving better performance. After that, several improvements were proposed to enhance the performance of DQN, including prioritized experience replay [23], multi-step learning [24], distributional RL [25], and noisy nets [26], and constituted rainbow DQN [27]. In autonomous driving, DRL is a buzzword as well. In [28], a latent DRL method was developed for end-to-end autonomous driving, where the driving policy was learned jointly based on a sequential latent environment model. In [29], a general framework of tactical decision-making for autonomous driving was developed, which combines planning and learning, in the form of Monte Carlo tree search and DRL.

In addition to DRL, parallel control is a powerful tool for handling complex systems based on the ACP (artificial systems, computational experiments, and parallel execution) methodology [30], [31], [32], [33], [34]. On the basis of data produced by actual systems, artificial systems are used to restore actual systems. In the computational experiments, intelligent methods, including various DQNs, can be used to obtain control policies based on actual and artificial systems. Through the parallel execution, the output data of artificial systems and actual systems are utilized to optimize control policies. Therefore, parallel control can be seen as a control method of virtual-reality interaction. In [35], the problem of ETC for discrete-time multi-player nonzero-sum games was studied, and an event-triggered optimal parallel control method was proposed, which predicts the future state and implements ETC based on the ACP methodology. In [36], an event-triggered nearly optimal control method was developed for a boiler-turbine system based on parallel control, which achieves constrained nonlinear optimal control and saves communication resources simultaneously. In recent years, a new parallel control policy, which introduces the action into the feedback, was proposed [37], [38], [39]. The newly proposed parallel control transforms the control policy from an algebraic form to a difference form, and thus can be seen as a "dynamic control policy". Currently, the parallel control policy has shown the potential to accomplish special control tasks as well as to improve control performance. In [40], an event-triggered near-optimal control method was proposed for optimal control of unknown discrete-time systems without restoring unknown systems, and this method also theoretically established a link between ETC and impulsive control through the proposed dynamic parallel control policy.

ETC has been studied early in the control field [15], [16], [17]. However, early ETC studies focused only on control stability, and other performance indices were not considered [16], [17]. In recent years, with the rapid development of machine learning, learning-based control has been proposed to implement event-triggered optimal control, i.e., to guarantee system stability while optimizing a specific performance index, among which the representative technique is adaptive dynamic programming (ADP)-based ETC [41], [42], [43]. ADP is homologous to RL [9], which makes it possible to extend these ADP-based ETC methods to RL. In the meantime, as the rapid rise of deep learning, ETDRL was gradually formed. In [19], the ETC problem of discrete-time systems was studied using DRL, and two learning approaches were proposed, including learning the triggering condition only with policy gradients and learning the control policy and the triggering condition using deep deterministic policy gradient (DDPG) with two DNNs. In [20], an event-triggered model predictive control was developed for path following of autonomous driving, which utilizes DRL to train the triggering condition while using model predictive control to achieve motion control.

In summary, utilizing DRL to achieve decision-making for autonomous driving is not new in the literature. However, the event-triggered decision-making policy for autonomous driving has not been reported in the literature, which motivates our research.

III. Problem Statement and Preliminaries

In this paper, our goal is to design an RL agent to automatically learn an optimal event-triggered driving policy for an autonomous vehicle.

A. *Decision-Making of Autonomous Driving*

In autonomous driving, low-level path planning (path planning for short) is a key issue, which usually refers to finding a feasible path that avoids collisions and complies with several metrics [1], [2]. Path planning can be implemented by classical optimization methods, or it can be considered as a multi-stage decision process, such as MDP, and implemented using RL. In RL-based path planning, the autonomous vehicle obverses information about surroundings to make the appropriate action and does not present an explicitly planned path at first, which is more in line with the behavior of human drivers. In the meantime, path planning is usually coupled with real-time control of the vehicle and plays a crucial role in the safety of autonomous driving.

In this paper, we consider path planning to be a multi-stage decision issue based on MDP and utilize DRL to achieve decision-making for path planning. Fig. 2 briefly displays the schematic of decision-making for autonomous driving, where the orange vehicle is the ego-vehicle while others are nearby vehicles. As shown in Fig. 2, our goal is to find a feasible planned path for the ego-vehicle in the presence of other uncontrolled vehicles through DRL. Unlike the previous DRL-based driving policies, we additionally seek to integrate the ETC mechanism into the driving policy, making the learned driving policy more consistent with the human driving policy. It is worth noting that, in DRL-based path planning, driving policies involve both discrete actions, such as lane-changing, accelerating, and decelerating, as well as continuous actions, such as controlling the throttle and the steering wheel. In this paper, we discuss driving policies with discrete actions based on MDP. The description of MDP settings for decision-making for path planning used in this paper, including states and actions, will be detailed in Section V.

B. *Q-Learning and Event-Triggered Control*

Q-learning is a class of RL techniques based on MDP, which is also the research basis of this paper. In what follows, we give a brief introduction to Q-learning.

A typical MDP consists of the following 5-tuples: $< S, A, P, R, \gamma >$, where S and A denote the sets of states and actions, respectively; $P : S \times A \to S$ is the transition probability determines the transition of any state $s_t \in S$ to any state $s_{t+1} \in S$ with $t \in \mathbb{N}$ denoting the time step; $R : S \times A \to \mathbb{R}$ presents the immediate reward; and γ is the discount factor.

For an RL agent behaving according to a control policy π, the action-value function, also known as the Q-function, is defined as follows:

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}\left[G_t \middle| s_t, a_t\right] \tag{1}$$

where $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ is the cumulative reward.

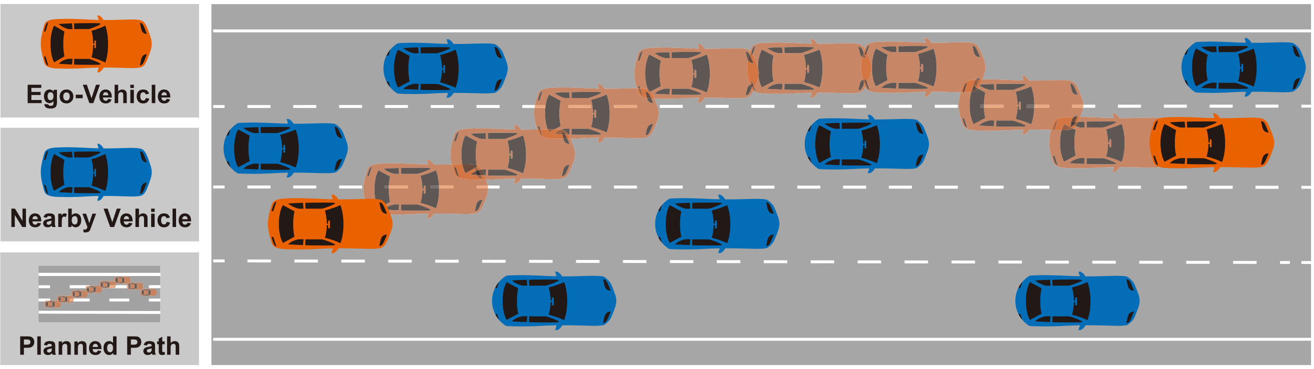


Fig. 2. Schematic of decision-making for autonomous driving.

Based on the Q-learning theory [9], the Q-function can be computed recursively using dynamic programming:

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}\left[r_t + \gamma Q^{\pi}(s_{t+1}, a_{t+1})\right]. \tag{2}$$

To maximize the cumulative reward, define the optimal Q-function as $Q^{*}(s_t, a_t) = \max_{\pi} Q^{\pi}(s_t, a_t)$, and then we have the following Bellman equation [9]:

$$Q^{*}(s_t, a_t) = \mathbb{E}\left[r_t + \gamma \max_{a_{t+1}} Q^{*}(s_{t+1}, a_{t+1})\right]. \tag{3}$$

Therefore, to obtain $Q^{*}(s_t, a_t)$, most Q-learning methods focus on efficiently solving the Bellman (3) in different tasks. To this end, several improved DQNs have been proposed, and an effective Q-learning method, known as dueling DDQN, was proposed in [22]. Dueling DDQN decomposes the Q-function into the value function and the advantage function, i.e.,

$$Q^{\pi}(s_t, a_t) = V^{\pi}(s_t) + A^{\pi}(s_t, a_t) \tag{4}$$

where $V^{\pi}(s_t)$ is the commonly used value function in RL and is as follows:

$$V^{\pi}(s_t) = \mathbb{E}_{a_t \sim \pi(s_t)}\left[Q^{\pi}(s_t, a_t)\right] \tag{5}$$

and thus the advantage function is given by

$$A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t). \tag{6}$$

By introducing the advantage function, dueling DDQN is capable of learning which states are valuable, without having to learn the effect of each action for each state, and achieve satisfactory results in some tasks [22].

As it is shown in the above introduction, Q-learning is implemented in the TTC mechanism, that is, the action is calculated and updated at every time instant t, which is a waste of communication resources in some cases. Different from the above TTC mechanism, the ETC mechanism only updates the action if a specific condition is met (or violated), and instants when the action is updated are called the triggering instants. In ETC, triggering instants, denoted by $\{t_k\}$ with $t_0 = 0$, $t_k < t_{k+1}$, $k \in \mathbb{N}$, are a monotone increasing sequence. The action remains constant using a zero-order holder (ZOH) during the time interval $[t_k, t_{k+1})$. Fig. 3 presents an intuitive comparison of TTC and ETC, where h represents the fixed sampling period in TTC. In the general MDP, the fixed sampling period is usually taken as $h = 1$.

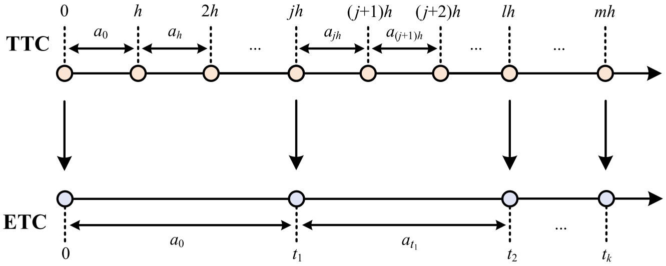


Fig. 3. Comparison between TTC and ETC.

In this paper, our objective is to design an RL agent to learn the event-triggered decision-making policy that determines the triggering instants $\{t_k\}$ and minimizes the cumulative reward in the scenario of autonomous driving simultaneously. It is worth noting that the developed ETDQN does not have an explicit triggering condition, but incorporates communication loss into the reward through parallel control and directly learns the event-triggered driving policy using a single DNN.

IV. Event-Triggered Deep Q-Learning for Autonomous Driving: A Parallel Control Approach

In this section, the developed ETDQN is presented based on parallel control, and its theoretical analysis is shown.

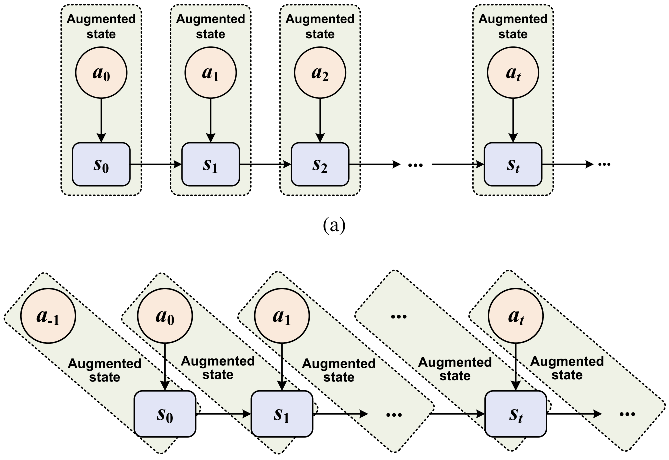
A. *Augmented Markov Decision Processes*

In the traditional RL, our goal is to obtain the following state feedback-based control policy:

$$a_t = \pi(s_t) \tag{7}$$

to accomplish a specific task, where $\pi(\cdot)$ is the control policy and can be derived based on the optimal control theory, i.e., linear quadratic regulator (LQR), or learned from interactions with the environment.

Parallel control is different from (7). Parallel control incorporates the action into the feedback and constructs a dynamic control policy [37]. In [40], a discrete-time parallel control



(a)

(b)

Fig. 4. Discrete-time parallel control for RL. (a) The method in [40]. (b) The method of this paper.

policy was developed as follows:

$$a_{t+1} = \varpi(s_t, a_t). \tag{8}$$

Inspired by [40], we propose a novel discrete-time parallel control policy as follows:

$$a_t = \varpi(s_t, a_{t-1}). \tag{9}$$

To achieve the parallel control policy (9), we need to construct the AMDP. First, assuming that the state and the action are in the column vector form, i.e., $s_t \in \mathbb{R}^n$ and $a_t \in \mathbb{R}^m$, and denoting the augmented state set as $\mathcal{S} \triangleq S \oplus A$, we can define the following augmented state:

$$\zeta_t = \left[s_t^{\mathsf{T}}, a_{t-1}^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathcal{S}. \tag{10}$$

Remark 1: It should be noted that, in DRL, the observation of the agent is not necessarily a vector, and the augmented state can not be designed according to (10). In this case, we can utilize DNNs to extract features in the vector form from the observation and then integrate it with action according to (10). In fact, the method of constructing the augmented state in parallel control is not unique and it should be customized according to the data type and task.

Then, based on the augmented state (10), we can derive the following AMDP: $< \mathcal{S}, A, \mathcal{P}, \mathcal{R}, \gamma >$, where $\mathcal{P} : \mathcal{S} \times A \to \mathcal{S}$ denotes the augmented transition probability, and $\mathcal{R} : \mathcal{S} \times A \to \mathbb{R}$ presents the augmented reward. Notice that the control policy (9) is designed by researchers, so no modification of the interaction approach with environment is required. Therefore, P for the state transition can remain unchanged, and $\mathcal{P}$ is the augmented transition probability additionally considers the artificial action transition. Fig. 4 further demonstrates two approaches for implementing discrete-time parallel control. In Fig. 4(b), $a_{-1} \in A$ is a virtual action and can be tailored according to specific needs.

Theorem 1: Take the parallel control policies (8) and (9) into consideration, and define the following variation rates of the action:

$$\mu_t = a_{t+1} - a_t \tag{11}$$

for (8), and

$$\mu_t = a_t - a_{t-1} \tag{12}$$

for (9). The goal of the parallel control policies is to learn the variation rate of the action μ_t.

Proof: First, we consider the case of the variation rate of the action (12). In the vast majority of cases, an MDP can be rewritten as the following discrete-time system:

$$s_{t+1} = F(s_t, a_t) \tag{13}$$

where $s_t \in S$ is the state, $a_t \in A$ is the action, $F(s_t, a_t)$ denotes the system function determined by interactive environment, and $t \in \mathbb{N}$ denotes the time step.

According to (12) and (13), we can derive

$$\begin{bmatrix} s_{t+1} \\ a_t \end{bmatrix} = \begin{bmatrix} F(s_t, a_t) \\ a_{t-1} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \mu_t \tag{14}$$

where $\mathbf{0}$ and $\mathbf{I}$ denote the zero and identity matrices with appropriate dimensions, respectively.

Based on the augmented state (10), we can obtain the following augmented discrete-time system:

$$\zeta_{t+1} = \mathcal{F}(\zeta_t, \mu_t) \tag{15}$$

where

$$\mathcal{F}(\zeta_t, \mu_t) = \begin{bmatrix} F(s_t, a_t) \\ a_{t-1} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \mu_t.$$

Therefore, designing the parallel control policy (9) for the discrete-time system (13) is equivalent to designing μ_t in (12) for the augmented discrete-time system (15).

The case of μ_t in (11) is similar to the above proof and is omitted here. ■

Remark 2: According to (11) and (12), the parallel control policy (8) can be viewed as the "forward difference control policy" while the parallel control policy (9) can be seen as the "backward difference control policy". It is worth noting that the augmented state ζ_t introduces information of the previous action a_{t-1} into the reward. Meanwhile, when the variation rate of the action $\mu_t = 0$, it means that there is no variation in the current action. Therefore, parallel control has a natural connection with ETC, a class of anthropomorphic control mechanisms.

B. Reward Design for Autonomous Driving

Without considering ETC, a reward function for decision-making of autonomous driving can be defined as follows:

$$r_t = c_v \tau(v_t) + c_i \mathbb{I}_i + c_c \mathbb{I}_c \tag{16}$$

where $c_v > 0$ and $c_i > 0$ are constants for the speed reward and the lane-keeping reward, respectively, $c_c < 0$ is a discount parameter for the vehicle collision reward, and $c_c < 0$ works at the step of the vehicle collision and the L steps before the vehicle collision, i.e.,

$$c_c = \alpha^{N_c - t} c_{c,\max} \tag{17}$$

with $c_{c,\max} < 0$ and $0 \leq \alpha \leq 1$ denoting a constant and a discount factor for vehicle collision, respectively, N_c representing the time step when the collision occurred, and $N_c - L \leq t \leq N_c$ and $L > 0$; $\tau(v_t)$ is the speed reward

$$\tau(v_t) = \frac{v_t - v_{\min}}{v_{\max} - v_{\min}} \tag{18}$$

with v_t denoting the current driving speed, and $v_{\max}$ and $v_{\min}$ representing the maximum and minimum driving speeds, respectively; and $\mathbb{I}_i$ and $\mathbb{I}_c$ are indicator functions for lane-keeping and vehicle collision, respectively, i.e.,

$$\mathbb{I}_i = \begin{cases} 1 & \text{if the ego-vehicle is in the middle of the lane} \\ 0 & \text{else} \end{cases}$$

and

$$\mathbb{I}_c = \begin{cases} 1 & N_c - L \leq t \leq N_c \\ 0 & \text{else.} \end{cases}$$

Based on the augmented state (10) and the reward (16), the reward considering communication loss is designed as

$$\rho_t = r_t + c_e \mathbb{I}_e \tag{19}$$

where $c_e < 0$ is the constant for ETC, and $\mathbb{I}_e$ is the indicator function for ETC, namely,

$$\mathbb{I}_e = \begin{cases} 1 & \text{if } a_t \neq a_{t-1} \\ 0 & \text{else.} \end{cases}$$

Remark 3: Recall that the reward in the traditional MDP does not contain information of the previous action, i.e., $R : S \times A \rightarrow \mathbb{R}$, so it is hard to directly design the reward (19) based on the traditional MDP. By constructing the augmented state, we can introduce information of the previous action a_{t-1} directly into the reward (19), which is the basis for achieving ETC without using an explicit triggering condition. Besides, the reward (19) also reflects the starting point of the developed ETDQN. We do not directly compare the difference between the current state and the state of the last triggering instant to implement ETC, which is the basis of most existing ETC methods [15], [16], [17], [18], [19], [20], but rather utilize the information of the current state and the previous action to determine whether the previous action still applies to the current state.

C. Parallel Control-Based Event-Triggered Deep Q-Network

In this section, based on the AMDP, we introduce the developed ETDQN based on dueling DDQN, and notice that the developed ETDQN is applicable to other DQNs as well.

Based on the developed augmented state, the following augmented Q-function is defined:

$$\mathcal{Q}^{\varpi}(\zeta_t, a_t) = \mathbb{E}_{\varpi}\left[\mathcal{G}_t \middle| \zeta_t, a_t\right] \tag{20}$$

where $\mathcal{G}_t = \sum_{k=0}^{\infty} \gamma^k \rho_{t+k}$ is the cumulative reward considering communication loss. It is worth noting that the augmented Q-function contains information about the previous action and can be rewritten as $\mathcal{Q}^{\varpi}(\zeta_t, a_t) \triangleq \mathcal{Q}^{\varpi}(s_t, a_t, a_{t-1})$, and is referred to as the *double action-value function*.

Remark 4: Since the augmented Q-function contains information about both the current action and the previous action, we can train our DRL agent directly according to the reward (19), which is different from other ETDRL methods and does not require additional training (or designing) of triggering conditions as in other ETC methods [19].

According to (4)–(6), we can obtain the augmented version of the value and advantage functions

$$\mathcal{V}^{\varpi}(\zeta_t) = \mathbb{E}_{a_t \sim \varpi(\zeta_t)}\left[\mathcal{Q}^{\varpi}(\zeta_t, a_t)\right] \tag{21}$$

and

$$\mathcal{A}^{\varpi}(\zeta_t, a_t) = \mathcal{Q}^{\varpi}(\zeta_t, a_t) - \mathcal{V}^{\varpi}(\zeta_t). \tag{22}$$

Similar to the optimal Q-function in (3), defining the optimal augmented Q-function as $\mathcal{Q}^*(\zeta_t, a_t) = \max_{\varpi} \mathcal{Q}^{\varpi}(\zeta_t, a_t)$, we can obtain

$$\mathcal{Q}^*(\zeta_t, a_t) = \mathbb{E}\left[\rho_t + \gamma \max_{a_{t+1}} \mathcal{Q}^*(\zeta_{t+1}, a_{t+1})\right]. \tag{23}$$

Meanwhile, we can also apply dynamic programming recursively to obtain $\mathcal{Q}^*(\zeta_t, a_t)$

$$\mathcal{Q}^{\varpi}(\zeta_t, a_t) = \mathbb{E}_{\pi}\left[\rho_t + \gamma \mathcal{Q}^{\varpi}(\zeta_{t+1}, a_{t+1})\right]. \tag{24}$$

Similar to other DQNs, we use a DNN to approximate the augmented Q-function (20), i.e., $\mathcal{Q}(\zeta_t, a_t|\theta)$ with the DNN parameters θ. To train the DNN, we minimize the following temporal difference (TD) error at iteration i:

$$\mathcal{L}_i(\theta_i) = E_{\mathcal{D}}\left[\mathcal{Y}_i^{\text{Target}} - \mathcal{Q}(\zeta_t, a_t|\theta_i)\right]^2 \tag{25}$$

where the target $\mathcal{Y}_i^{\text{Target}}$ varies according to different DQN methods. In our method, we utilize the target $\mathcal{Y}_i^{\text{Target}}$ similar to DDQN [21]

$$\mathcal{Y}_i^{\text{Target}} = \rho_t + \gamma \mathcal{Q}\left(\zeta_{t+1}, \arg\max_{a_{t+1}} \mathcal{Q}\left(\zeta_{t+1}, a_{t+1}|\theta_i\right) \middle| \theta^-\right) \tag{26}$$

where θ^- denotes the parameters of the target network. The target network $\mathcal{Q}(\zeta_t, a_t|\theta^-)$ is used to freeze the parameters of the online network $\mathcal{Q}(\zeta_t, a_t|\theta_i)$ for a fixed number of iterations while updating the online network parameters θ_i, which improves the stability of the method [7]. To mitigate overoptimistic value estimates of the Q-function, the target $\mathcal{Y}_i^{\text{Target}}$ additional choose the action a_{t+1} according to the online network $\mathcal{Q}(\zeta_t, a_t|\theta_i)$ and then compute the value of the Q-function according to the target network $\mathcal{Q}(\zeta_t, a_t|\theta^-)$ [21]. Meanwhile, it should be noted that some of the parameters in θ are shared by the value and advantage functions. The detailed illustration of dueling DDQN can be found in [22] and is omitted here.

Based on the augmented Q-function, the triggering instant is decided by

$$t_{k+1} = \inf\left\{t \middle| \underbrace{\arg\max_{a_t} \mathcal{Q}\left(\zeta_t, a_t|\theta\right)}_{a_t} \neq a_{t-1}, t > t_k\right\}. \tag{27}$$

By utilizing the developed augmented state, we can introduce communication loss into the reward. Thus, the well-trained augmented Q-function $\mathcal{Q}(\zeta_t, a_t|\theta)$ obtains information of the previous action a_{t-1} and can consider communication loss when computing the current action. In other words, when the augmented Q-function $\mathcal{Q}(\zeta_t, a_t|\theta)$ performs the current action, it additionally considers whether the previous action is still applicable. If the currently computed action is the same as the previous action, the action remains unchanged and is not transmitted to the actuator. Therefore, ETC can be realized. The developed ETDQN abandons the implementation of ETC by comparing the current state and the state of the last sampling instant [15], [16], [17], [18], [19], [20] and directly outputs the action considering communication loss through the augmented Q-function. Therefore, the instant that the current action is different from the previous action can be defined as the triggering instant, i.e., $a_t \neq a_{t-1}$, and the developed ETDQN does not necessarily record the state of the last triggering instant.

V. Experiments

To demonstrate the effectiveness of the developed ETDQN, we apply the developed ETDQN to an environment for decision-making of autonomous driving and present the analysis of experiment results in this section.

A. Experimental Setup

We perform a comprehensive evaluation of the developed ETDQN on Highway-Env [44], an environment for decision-making of autonomous driving similar to Fig. 2. There are six environments in Highway-Env, including highway, merge, roundabout, parking, intersection, and racetrack, and we conduct experiments on the highway environment. The baseline, dueling DDQN, and our ETDQN will be verified and compared on the highway environment with the same settings. In experiments, autonomous vehicle actions are discrete, including five types, namely, LANE LEFT, IDLE, LANE RIGHT, FASTER, and SLOWER. The observation of the ego-vehicle o_t is chosen as the form of kinematics. The observation (or state) is a $V \times F$ real matrix, i.e., $o_t \in \mathbb{R}^{V\times F}$, where V denotes the number of vehicles closest to the autonomous vehicle can be observed, and F is the size of features of each vehicle. In experiments, we select $V = 6$ and $F = 5$, and features of each vehicle include the presence of vehicles, vehicle velocities on the x and y axes, and offsets to the ego-vehicle on the x and y axes. To construct the augmented state (10), we convert the observation into a vector, that is, $s_t = \mathrm{vec}(o_t)$. More details about Highway-Env can be found in [44]. Meanwhile, in the augmented state (10), we utilize one-hot encoding for actions. According to the above description, the goal of the ego-vehicle is to learn an ETC driving policy to maximize the cumulative reward (19) based on the above settings.

B. Experiment Results and Analysis

In experiments, we train the developed ETDQN and dueling DDQN over 10^5 steps with the same setting. The parameters of MDP and DNN are as follows. The discount factor is chosen as $\gamma = 0.97$. The DNN is a fully connected neural network that contains 3 hidden layers with 1024 units in each layer. The learning rate is set to 5×10^{-5} and the dropout is set to 0.3. The batch size and the replay buffer size are 256 and 8192, respectively. Besides, ϵ-greedy is utilized during the training with ϵ linearly decaying from 1.0 to 0.05. The parameters of the reward (19) are selected as follows: $c_v = 0.5$, $c_i = 0.1$, $c_{c,\max} = -5$, $c_e = -1.5$, $\alpha = 0.8$, $L = 5$, $v_{\max} = 30$ m/s, and $v_{\min} = 20$ m/s.

To demonstrate the effectiveness of the developed ETDQN, the following performance indices are introduced. First, to evaluate communication loss, the following triggering frequency is defined in the test:

$$\mathcal{T} = \frac{\sum_{t=0}^{N} \mathbb{I}_{e,t}}{N} \tag{28}$$

where $\mathbb{I}_{e,t}$ denotes the indicator function for triggering events at the time step t, and $N \leq N_{\max}$ is the number of steps for an episode with $N_{\max}$ being the max steps in an episode. In the test, each model is tested $K = 5$ episodes with the maximum time step $N_{\max} = 100$. Then, the related performance indices, including the average cumulative reward $\overline{\mathcal{G}}$, the average steps $\overline{\mathcal{N}}$, the average speed $\overline{\mathcal{V}}$, and the average triggering frequency $\overline{\mathcal{T}}$, are defined as follows:

$$\overline{\mathcal{G}} = \sum_{k=1}^{K} \mathcal{G}_k / K \tag{29a}$$

$$\overline{\mathcal{N}} = \sum_{k=1}^{K} N_k / K \tag{29b}$$

$$\overline{\mathcal{V}} = \sum_{k=1}^{K} v_k / K \tag{29c}$$

$$\overline{\mathcal{T}} = \sum_{k=1}^{K} \mathcal{T}_k / K \tag{29d}$$

where $\mathcal{G}_k$, N_k, v_k, and $\mathcal{T}_k$ are the cumulative reward, the number of driving steps, the average speed, and the triggering frequency in the kth episode, respectively.

In what follows, we present and analyze experimental results. Fig. 5 plots the training curve of the developed ETDQN and the baseline, where the solid line shows the performance indices $\overline{\mathcal{G}}$, $\overline{\mathcal{N}}$, $\overline{\mathcal{V}}$, and $\overline{\mathcal{T}}$ per training episode and the shade shows their standard deviation. Fig. 5 shows that both the developed ETDQN and dueling DDQN enhance the cumulative reward and ensure that the DRL agent learns a better driving policy than the original one. Compared with dueling DDQN, the developed ETDQN achieves better performance in all performance indices. Table I numerically compares the developed ETDQN and dueling DDQN. The performance indices of the best driving policy with different methods, i.e., the driving policy with the highest average cumulative reward $\overline{\mathcal{G}}$, is presented in Table I. The developed ETDQN achieves significant advantages in the average cumulative reward and the average triggering frequency, with a 7.36% lower average triggering frequency.

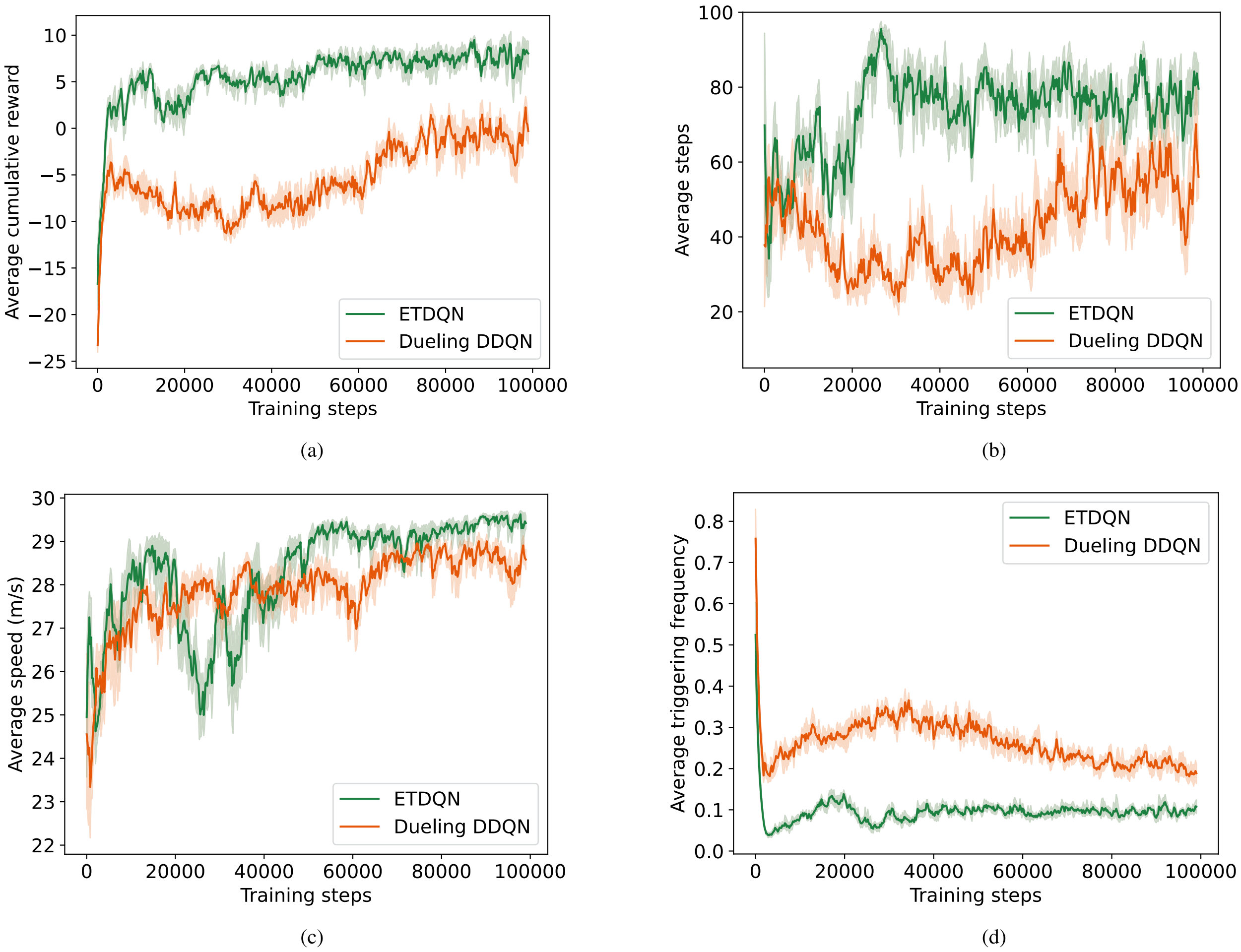


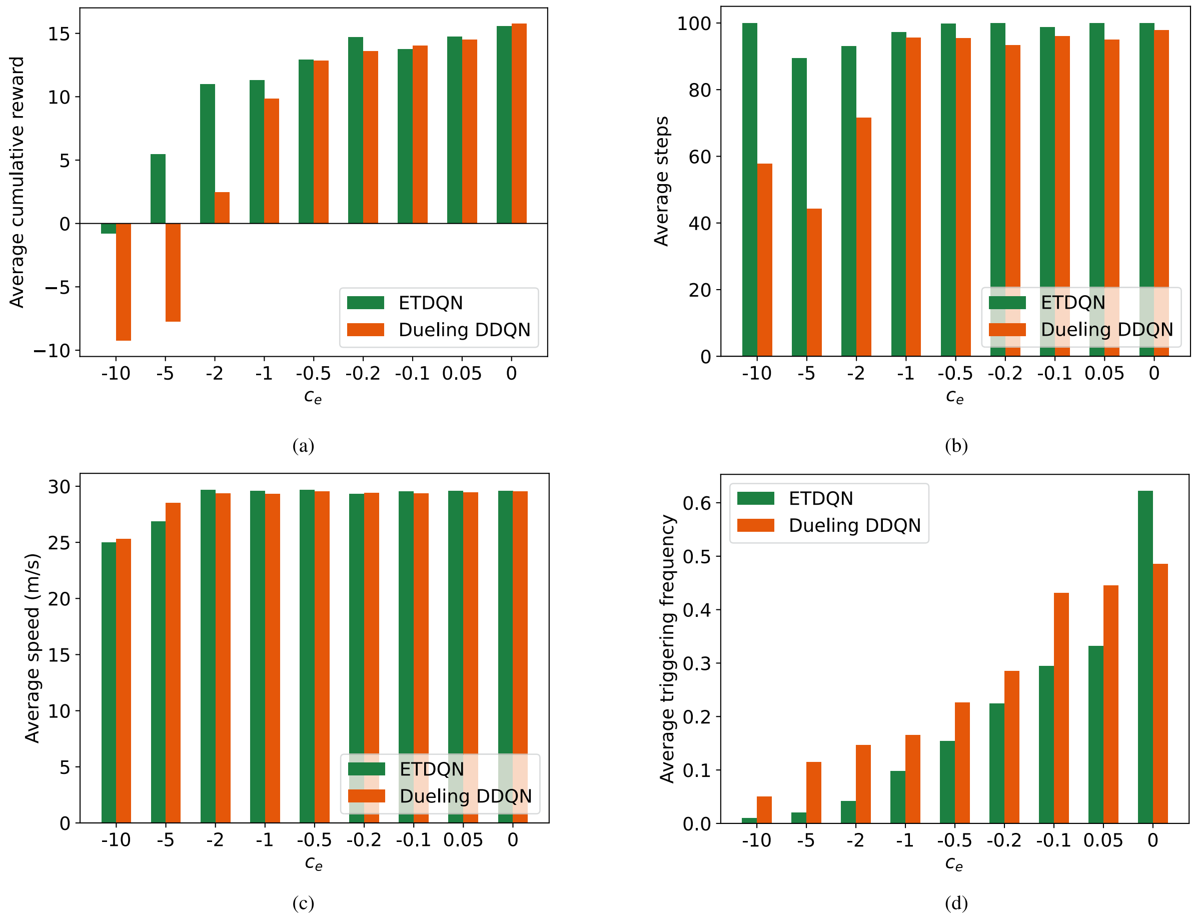
Fig. 5. Training curves of different performance indices. (a) $\overline{\mathcal{G}}$. (b) $\overline{\mathcal{N}}$. (c) $\overline{\mathcal{V}}$. (d) $\overline{\mathcal{T}}$.

TABLE I
PERFORMANCE COMPARISON WITH DIFFERENT METHODS

Performance	$\overline{\mathcal{G}}$	$\overline{\mathcal{N}}$	$\overline{\mathcal{V}}$	$\overline{\mathcal{T}}$
ETDQN	**11.4740**	**96.8**	**29.3331**	**6.55%**
Dueling DDQN	7.3774	77.6	29.3133	13.91%

This indicates that the developed ETDQN is quite effective in reducing communication loss. Also, the driving policy obtained by ETDQN allows the ego-vehicle to drive faster for a longer time, indicating that the developed ETDQN guarantees the ego-vehicle to learn a more generalized driving policy. By introducing information of the previous action into the augmented Q-function, the developed ETDQN demonstrates stronger learning ability for ETC, as its training curve steadily increases and then becomes stable with narrow shade in all performance indices.

Next, we further present an in-depth comparison with different ETC parameters. The performance indices of the best driving policy with different ETC parameters are shown in Fig. 6, and Table II presents the exact number of the performance indices. As shown in Fig. 6 and Table II, in most cases, the developed ETDQN outperforms dueling DDQN. When the ETC parameter $c_e = -2$, $c_e = -5$, or $c_e = -10$, the average steps of the driving policy obtained by dueling DDQN decreases dramatically, which means that dueling DDQN fails to learn a sustainable driving policy. The traditional Q-function does not provide information of the previous action, so dueling DDQN cannot guarantee that the traditional Q-function learns the connection between the variation rate of the action and communication loss. There is one exception, i.e., $c_e = 0$. Although the cumulative rewards are very close, the developed ETDQN has a triggering frequency that is 13.64% higher than dueling DDQN. This result may be caused by the augmented state, i.e., the augmented state introduces the previous action, but the reward with $c_e = 0$ does not instruct the DRL agent on how to apply information of the action. Therefore the augmented state imposes an additional burden on the DNN. It is worth noting that, as the value of the parameter c_e decreases, the triggering frequency also decreases. In the meantime, the average cumulative reward decreases as well. Thus, it is important to choose a reasonable ETC parameter to balance the driving performance against communication loss according to practical restrictions.



(a)

(b)

(c)

(d)

Fig. 6. Performance indices with different ETC parameters c_e. (a) $\overline{\mathcal{G}}$. (b) $\overline{\mathcal{N}}$. (c) $\overline{\mathcal{V}}$. (d) $\overline{\mathcal{T}}$.

TABLE II
PERFORMANCE COMPARISON WITH DIFFERENT ETC PARAMETERS c_e

	c_e	0	-0.05	-0.1	-0.2	-0.5	-1	-2	-5	-10
ETDQN	$\overline{\mathcal{G}}$	15.5713	**14.7587**	13.7590	**14.7013**	**12.9264**	**11.3213**	**10.9781**	**5.4857**	**-0.8025**
	$\overline{\mathcal{N}}$	**100.0**	**100.0**	**98.8**	**100.0**	**99.8**	**97.2**	**93.0**	**89.4**	**100.0**
	$\overline{\mathcal{V}}$	**29.5623**	**29.5690**	**29.5203**	29.3190	**29.6578**	**29.5679**	**29.6653**	26.8483	25.0000
	$\overline{\mathcal{T}}$	62.20%	**33.20%**	**29.46%**	**22.40%**	**15.43%**	**9.77%**	**4.14%**	**2.01%**	**1.00%**
Dueling DDQN	$\overline{\mathcal{G}}$	**15.7622**	14.5170	**14.0355**	13.6066	12.8474	9.8352	2.4772	-7.7504	-9.2450
	$\overline{\mathcal{N}}$	97.8	95.0	96.0	93.4	95.4	95.6	71.6	44.2	57.8
	$\overline{\mathcal{V}}$	29.5122	29.4238	29.3700	**29.3923**	29.5419	29.3232	29.3365	**28.5044**	**25.2982**
	$\overline{\mathcal{T}}$	**48.56%**	44.53%	43.10%	28.46%	22.60%	16.48%	14.65%	11.49%	4.98%

VI. CONCLUSION

This paper investigated the ETDRL problem for autonomous driving using parallel control and developed a triggering condition-free ETDQN. First, the augmented state and the AMDP were developed to construct the dynamic parallel control policy. Second, based on the AMDP, the reward considers communication loss can be devised directly, and augmented action-value, value, and advantage functions were given accordingly. The augmented action-value function contains information of the current state and the current and previous actions, which enables the developed ETDQN to implement ETC without training an explicit triggering condition. The developed ETDQN utilizes a DNN to implement two tasks simultaneously, namely, ETC and decision-making for autonomous driving, which can be regarded as multi-task learning. The empirical results show that the developed ETDQN is capable of reducing communication loss and outmatching dueling DDQN. Based on the developed ETDQN, the following topics will be studied in the future. First, explore ETDRL methods on the decision-making problem of high-dimensional action space. Second, perform our ETDQN in real-world vehicle testing.

References

[1] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. Field Robot.*, vol. 37, no. 3, pp. 362–386, 2020.

[2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.

[3] X. Tang et al., "Prediction-uncertainty-aware decision-making for autonomous vehicles," *IEEE Trans. Intell. Veh.*, vol. 7, no. 4, pp. 849–862, Dec. 2022.

[4] G. Sidorenko, A. Fedorov, J. Thunberg, and A. Vinel, "Towards a complete safety framework for longitudinal driving," *IEEE Trans. Intell. Veh.*, vol. 7, no. 4, pp. 809–814, Dec. 2022.

[5] M. Hasan, S. Mohan, T. Shimizu, and H. Lu, "Securing vehicle-to-everything (V2X) communication platforms," *IEEE Trans. Intell. Veh.*, vol. 5, no. 4, pp. 693–713, Dec. 2020.

[6] S. Teng, L. Chen, Y. Ai, Y. Zhou, Z. Xuanyuan, and X. Hu, "Hierarchical interpretable imitation learning for end-to-end autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 673–683, Jan. 2023.

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

[8] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[9] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[10] X. Dai, C. Zhao, X. Wang, Y. Lv, Y. Lin, and F.-Y. Wang, "Image-based traffic signal control via world models," *Front. Inf. Technol. Electron. Eng.*, vol. 23, no. 12, pp. 1795–1813, 2022.

[11] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 740–759, Feb. 2022.

[12] B. R. Kiran et al., "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[13] Y. Lin, J. McPhee, and N. L. Azad, "Comparison of deep reinforcement learning and model predictive control for adaptive cruise control," *IEEE Trans. Intell. Veh.*, vol. 6, no. 2, pp. 221–231, Jun. 2021.

[14] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 652–674, Sep. 2022.

[15] K.-E. Årzén, "A simple event-based PID controller," in *Proc. IFAC World Congr.*, 1999, pp. 423–428.

[16] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.

[17] P. Tallapragada and N. Chopra, "On event triggered tracking for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2343–2348, Sep. 2013.

[18] M. S. Mahmoud and Y. Xia, *Networked Control Systems: Cloud Control and Secure Control*. Oxford, U.K.: Butterworth-Heinemann Elsevier Ltd., 2019.

[19] D. Baumann, J.-J. Zhu, G. Martius, and S. Trimpe, "Deep reinforcement learning for event-triggered control," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 943–950.

[20] F. Dang, D. Chen, J. Chen, and Z. Li, "Event-triggered model predictive control with deep reinforcement learning for autonomous driving," 2022, *arXiv:2208.10302*. [Online]. Available: https://arxiv.org/abs/2208.10302

[21] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.

[22] Z. Wang, T. Schaul, M. Hessel, H. V. Hasselt, M. Lanctot, and N. D. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, vol. 48, pp. 1995–2003.

[23] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–21.

[24] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, vol. 48, pp. 1928–1937.

[25] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, vol. 70, pp. 449–458.

[26] M. Fortunato et al., "Noisy networks for exploration," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–21.

[27] M. Hessel et al., "Rainbow: Combining improvements in deep reinforcement learning," in *Proc. 32nd AAAI Conf. Artif. Intell. 30th Innov. Appl. Artif. Intell. Conf. 8th AAAI Symp. Educ. Adv. Artif. Intell.*, 2018, pp. 3215–3222.

[28] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5068–5078, Jun. 2022.

[29] C.-J. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine, and M. J. Kochenderfer, "Combining planning and deep reinforcement learning in tactical decision making for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 5, no. 2, pp. 294–305, Jun. 2020.

[30] F.-Y. Wang, "Parallel system methods for management and control of complex systems," *Control Decis.*, vol. 19, no. 5, pp. 485–489, May 2004.

[31] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.

[32] K. Wang, C. Gou, N. Zheng, J. M. Rehg, and F.-Y. Wang, "Parallel vision for perception and understanding of complex scenes: Methods, framework, and perspectives," *Artif. Intell. Rev.*, vol. 48, pp. 299–329, 2017.

[33] J. Lu, Q. Wei, Y. Liu, T. Zhou, and F.-Y. Wang, "Event-triggered optimal parallel tracking control for discrete-time nonlinear systems," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 52, no. 6, pp. 3772–3784, Jun. 2022.

[34] J. Lu, X. Wang, X. Cheng, J. Yang, O. Kwan, and X. Wang, "Parallel factories for smart industrial operations: From big AI models to field foundational models and scenarios engineering," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 12, pp. 2079–2086, Dec. 2022.

[35] J. Lu, Q. Wei, Z. Wang, T. Zhou, and F.-Y. Wang, "Event-triggered optimal control for discrete-time multi-player non-zero-sum games using parallel control," *Inf. Sci.*, vol. 584, pp. 519–535, Jan. 2022.

[36] Q. Wei, J. Lu, T. Zhou, X. Cheng, and F.-Y. Wang, "Event-triggered near-optimal control of discrete-time constrained nonlinear systems with application to a boiler-turbine system," *IEEE Trans. Ind. Inform.*, vol. 18, no. 6, pp. 3926–3935, Jun. 2022.

[37] F.-Y. Wang et al., "Where does AlphaGo go: From church-turing thesis to alphago thesis and beyond," *IEEE/CAA J. Automatica Sinica*, vol. 3, no. 2, pp. 113–120, Apr. 2016.

[38] J. Lu, Q. Wei, and F.-Y. Wang, "Parallel control for optimal tracking via adaptive dynamic programming," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 6, pp. 1662–1674, Nov. 2020.

[39] A. J. Muñoz-Vázquez, G. Fernández-Anaya, and J. D. Sánchez-Torres, "Adaptive parallel fractional sliding mode control," *Int. J. Adaptive Control Signal Process.*, vol. 36, no. 3, pp. 751–759, 2022.

[40] J. Lu, Q. Wei, T. Zhou, Z. Wang, and F.-Y. Wang, "Event-triggered near-optimal control for unknown discrete-time nonlinear systems using parallel control," *IEEE Trans. Cybern.*, vol. 53, no. 3, pp. 1890–1904, Mar. 2023.

[41] K. G. Vamvoudakis, A. Mojoodi, and H. Ferraz, "Event-triggered optimal tracking control of nonlinear systems," *Int. J. Robust Nonlinear Control*, vol. 47, no. 3, pp. 598–619, Mar. 2017.

[42] L. Dong, X. Zhong, C. Sun, and H. He, "Adaptive event-triggered control based on heuristic dynamic programming for nonlinear discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1594–1605, Jul. 2017.

[43] D. Wang, M. Ha, and J. Qiao, "Self-learning optimal regulation for discrete-time nonlinear systems under event-driven formulation," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 1272–1279, Mar. 2020.

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

Jingwei Lu received the Ph.D. degree in computer application technology from the University of Chinese Academy of Sciences, Beijing, China, in 2022. He is currently an Associate Professor with the Qingdao Academy of Intelligent Industries, Qingdao, China. He has authored or coauthored more than 20 journal and conference papers. His research interests include optimal control, adaptive dynamic programming, deep reinforcement learning, and autonomous driving. Dr. Lu was the Guest Editor of the IEEE Journal of Radio Frequency Identification. He was a Peer Reviewer for the IEEE Transactions on Cybernetics, IEEE Transactions on Industrial Electronics, IEEE Transactions on Intelligent Vehicles.

Liyuan Han received the bachelor's degree in information and computing science, and electrical engineering and automation from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2019. He is currently working toward the Ph.D. degree with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China, and the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing. His research interests include optimal control, adaptive dynamic programming, reinforcement learning, spiking neural network and their industrial applications.

Qinglai Wei (Senior Member, IEEE) received the B.S. degree in automation, and the Ph.D. degree in control theory and control engineering, from Northeastern University, Shenyang, China, in 2002 and 2009, respectively. From 2009 to 2011, he was a Postdoctoral Fellow with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is currently a Professor of the institute and the Associate Director of the laboratory. He has authored four books, and published over 80 international journal papers. His research interests include adaptive dynamic programming, neural-networks-based control, optimal control, nonlinear systems and their industrial applications. He is the Secretary of IEEE Computational Intelligence Society (CIS) Beijing Chapter since 2015. He was guest editors for several international journals. He was the recipient of IEEE/CAA Journal of Automatica Sinica Best Paper Award, IEEE System, Man, and Cybernetics Society Andrew P. Sage Best Transactions Paper Award, IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award, the Outstanding Paper Award of Acta Automatica Sinica, IEEE 6th Data Driven Control and Learning Systems Conference (DDCLS2017) Best Paper Award, and Zhang Siying Outstanding Paper Award of Chinese Control and Decision Conference (CCDC). He was the recipient of Shuang-Chuang Talents in Jiangsu Province, China, Young Researcher Award of Asia Pacific Neural Network Society (APNNS), Young Scientst Award and Yang Jiachi Tech Award of Chinese Association of Automation (CAA). He is a Board of Governors (BOG) member of the International Neural Network Society (INNS) and a Council Member of CAA.

Xiao Wang (Senior Member, IEEE) received the B.E. degree in network engineering from the Dalian University of Technology, Dalian, China, in 2011, and the M.E. and Ph.D. degrees in social computing from the University of Chinese Academy of Sciences, Beijing, China, in 2016. She is currently the President of Qingdao Academy of Intelligent Industries, and also a Professor of the School of Artificial Intelligence, Anhui University, Hefei, China. Her research interests include social network analysis, social transportation, cybermovement organizations, and multi-agent modeling. She is an Associate Editor ofIEEE Transactions on Computational Social Systems, IEEE Transactions on Intelligent Vehicles, ITSM, *Chinese Journal of Intelligent Science and Technology*.

Xingyuan Dai received the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2022. He is currently an Assistant Professor with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include intelligent transportation systems, machine learning, and deep learning.

Fei-Yue Wang (Fellow, IEEE) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined The University of Arizona, Tucson, AZ, USA, in 1990 and became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Chinese Talents Program from the State Planning Council, and in 2002, was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory for Management and Control of Complex Systems. His research interests include methods and applications for parallel intelligence, social computing, and knowledge automation. Prof. Wang was the recipient of the National Prize in Natural Sciences of China and became an Outstanding Scientist of ACM for his work in intelligent control and social computing in 2007, the IEEE ITS Outstanding Application and Research Awards in 2009 and 2011, respectively, and the IEEE Norbert Wiener Award in 2014. Since 1997, he has been serving as the General or Program Chair of over 30 IEEE, INFORMS, IFAC, ACM, and ASME conferences. He was the President of the IEEE ITS Society from 2005 to 2007, Chinese Association for Science and Technology, USA, in 2005, American Zhu Kezhen Education Foundation from 2007 to 2008, Vice President of the ACM China Council from 2010 to 2011, and the Vice President and the Secretary General of the Chinese Association of Automation from 2008 to 2018. He was the Founding Editor-in-Chief (EiC) of the *International Journal of Intelligent Control and Systems* from 1995 to 2000, the I*EEE Intelligent Transportation Systems Magazine* from 2006 to 2007, the IEEE/CAA Journal of Automatica Sinica from 2014 to 2017, and the *China's Journal of Command and Control* from 2015 to 2020. He was the EiC of the IEEE Intelligent Systems from 2009 to 2012, the IEEE Transactions on Intelligent Transportation Systems from 2009 to 2016, and the IEEE Transactions on Computational Social Systems from 2017 to 2020, and has been the Founding EiC of *Chinese Journal of Intelligent Science and Technology* since 2019 and the EiC of the IEEE Transactions on Intelligent Vechicles since 2022. He is currently the President of CAA's Supervision Council, IEEE Council on RFID, and the Vice President of the IEEE Systems, Man, and Cybernetics Society. He is a Fellow of INCOSE, IFAC, ASME, and AAAS.