Reinforcement Learning for Traffic Signal Control in Hybrid Action Space

Haoqing Luo, Yiming Bie, and Sheng Jin

***Abstract*— The prevailing reinforcement-learning-based traffic signal control methods are typically staging-optimizable or duration-optimizable, depending on the action spaces. In this paper, we use hybrid proximal policy optimization to synchronously optimize the stage specification and green interval duration. Under reformulated traffic demands, the intrinsic imperfections of (implementing optimization in) discrete or continuous action spaces are revealed. By comparison, hybrid action space offers a unified search space, in which our proposed method is able to better balance the trade-off between frequent switching and unsaturated release. Experiments in both single-agent and multi-agent scenarios are given to demonstrate that the proposed method reduces queue length and delay by an average of 12.72% and 11.89%, compared to the state-of-the-art RL methods. Furthermore, by calculating the Gini coefficients of right-of-way, we reveal that the proposed method does not harm fairness while improving efficiency.**

***Index Terms*— Traffic signal control, deep reinforcement learning, proximal policy optimization, hybrid action space.**

I. Introduction

TRAFFIC signal control coordinates the temporal-spatial resource allocation among different traffic movements and signalized junctions, to maintain a traffic supply-demand balance. Efficient right-of-way distribution helps alleviate the daunting congestion and economize on social expenditures, which has attracted extensive research interest.

Over the past decades, a variety of methods have been proposed to deal with the Gordian knot mentioned above. Based on transportation theories, model-based fixed-time control [1], [2], [3] and traffic-responsive control [4], [5], [6] are proposed, policies of which are blamed for being largely human-crafted heuristics reliant. On the other hand, benefitting from the emergence of deep reinforcement learning (DRL) [7], [8], [9], RL-based methods [10], [11] begin to flourish, as they offer the prospect of reaching beyond human intuition. For more details, please refer to Section. II.

Manuscript received 19 December 2022; revised 9 June 2023 and 26 September 2023; accepted 3 December 2023. This work was supported in part by the Natural Science Foundation of Zhejiang Province under Grant LR23E080002, in part by the National Natural Science Foundation of China under Grant 72361137006, and in part by the Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies. The Associate Editor for this article was Y. Wang. *(Corresponding authors: Yiming Bie; Sheng Jin.)*

Haoqing Luo and Sheng Jin are with the Institute of Intelligent Transportation Systems, College of Civil Engineering and Architecture, and the Center for Balance Architecture, Zhejiang University, Hangzhou 310058, China (e-mail: ludvikluo@zju.edu.cn; jinsheng@zju.edu.cn).

Yiming Bie is with the School of Transportation, Jilin University, Changchun 130022, China (e-mail: yimingbie@126.com).

Digital Object Identifier 10.1109/TITS.2023.3344585

Both RL algorithms and definitions of Markov decision process (MDP) [12] contribute to the variety of RL-based methods, many of which have stressed the importance of state and reward characterizations, to obtain a more precise representation of the manipulation. Action, on the other hand, is typically discrete or continuous, depending on the optimization objectives. In this paper, we specify that the staging is a discrete variable, whereas the green interval duration is a continuous variable. That is, people may optimize the green interval duration under a pre-specified staging or optimize the staging with a fixed decision interval. Experiments in Section. V-E will demonstrate that the compromise above will eventually lead to suboptimal policies.

Specific RL algorithms are proposed recently to learn policies directly over the original hybrid action space, rather than through discretization or continualization. Parameterized DQN (P-DQN) proposed by [13] is a joint architecture of DQN [7] and DDPG [14], where the actor and critic networks are modified to generate corresponding continuous and discrete parameters of the final parameterized action [15]. It is further improved by Multi-Pass DQN (MP-DQN) [16], which tackles the action value's reliance on joint action. Proximal policy optimization (PPO) [8] is a state-of-the-art policy-based algorithm, which is then extended (by making use of multiple policy heads) by Hybrid PPO (H-PPO) [17] to approximate the overall parameterized action. These pioneering techniques enable us to implement traffic signal control optimization directly in the hybrid action space.

The main contributions of this paper are summarized as follows.

1) By customizing H-PPO,[1] we develop an RL-based traffic signal control method to optimize both staging and timing simultaneously.
2) We evaluate the effectiveness of our proposed method in both single-agent and multi-agent scenarios, and the experimental results demonstrate that our approach outperforms the state-of-the-art methods.
3) We provide an explanation of how optimizing traffic signal control in hybrid action space leads to a performance boost.

The remainder of this paper is organized as follows. Section. II provides a review of existing research on model-based and RL-based methods. In Section. III, we define

[1]The source code could be directly accessed and referenced from https://github.com/Metro1998/hppo-in-traffic-signal-control.

the optimization problem, and Section. V-E reveals intrinsic imperfections of different optimization strategies, which serves as the motivation for this paper. The methodology, including the introduction of PPO and H-PPO, is discussed in Section. IV. Section. V presents simulation experiments that validate the superiority of hybrid action space over other approaches. The trade-off between frequent switching and unsaturated release, as well as the distribution equality of the right-of-way, are discussed in Section. VI. Finally, we draw conclusions in Section. VII.

II. Literature Review

A. Model-Based Traffic Signal Control

As reviewed in [18], model-based methods have advanced considerably over the past decades and can be roughly classified into two categories.

Fixed-time strategies regarding the optimal staging, splits, cycles, and even offsets are derived offline from historical statistics, so as to minimize the travel time or maximize the throughput. Well-known instances are stage-based SIGSET [19], phase-based approach [20] in isolated scenarios, and TRANSYT [3] in coordinated scenarios. The main drawback of fixed-time strategies is that their settings are based on historical rather than real-time data [18]. This might be a crude simplification considering both sporadic and long-term fluctuations in demands.

With the development of surveillance technologies, traffic-responsive control has gotten rid of the reliance on historical statistics. One simple example is the vehicle-interval method applicable to two-stage intersections proposed by [2]. SCOOT [4] is the traffic-responsive version of TANSYT, which runs repeatedly online to estimate the effect of a small incremental change in signal timings. Besides, OPAC [5], RHODES [21], and store-and-forward [22] based TUC [6] are also proposed, which are widely deployed in today's traffic signal control system.

B. RL-Based Traffic Signal Control

Generally, the methods mentioned above tend to cast traffic signal control into an optimization problem under certain rigorous assumptions about the traffic model. The human-designed heuristics do make the problem tractable, but unfortunately, also lead the problem to deviate from reality. On the other hand, model-free RL-based methods learn policies just from trial and error, rendering themselves more appropriate for today's urban traffic.

1) RL Algorithm: Value-based algorithms simply approximate the value function, the expected cumulative reward from a specific state or state-action pair, and policies are implicitly derived from the learned value functions. Tabular-based Q-learning [23] is the first to be introduced by [24], followed by DQN [7] in [25] and [26] and enhanced D3QN [27], [28] in [29]. Furthermore, utilizing the distributed architecture (i.e., Ape-X DQN [30]), [11] has improved the convergence speed and exploration capability dramatically. Impressing us with policy interpretability and data efficiency, the value-based algorithms still fail in continuous action space due to their dependence on the value function for decision-making. Even though the discretization trick [29] could bridge the gap, it breaks the natural structure of continuous policies (e.g., monotonicity) and suffers a lot when it comes to approximating the huge space of discrete actions.

Policy-based algorithms, on the other hand, directly update the policy in a direction that maximizes the generic objective. Specifically, [31] deploys DDPG [14] to learn a deterministic policy, whereas [32], [33], [34] utilize A2C [35] to learn a stochastic policy. In general, policies under policy-based algorithms will be explicitly derived from policy heads, for example, a discrete policy (generated) from the softmax layer or a continuous policy (generated) from the hyperbolic tangent activation, making it possible to approximate policy in continuous or even hybrid action space.

2) MDP Representation: A state is defined as the current situation of an environment. In the previous works, a variety of heuristic measurements are proposed, such as queue length [25], [26], [36], [37], [38], waiting time [39], [40], elapsed time [25], [34], [41], position, speed [29], [42], and current phase [10], [11]. In order to obtain information representation of different scales, these variables are defined in the form of scalar values, one-hot vectors, and image-like matrices.

A reward is defined as the goal of a reinforcement learning problem, which has been proven effective enough to derive behaviors studied in natural and artificial intelligence [43]. A reward signal should be properly designed to minimize the average travel time which is the most widely accepted measurement in transportation. Travel time, however, is demonstrated to be inefficient in practice and has been substituted with other rewards, such as queue length [11], [25], waiting time [29], delay [32], [44], speed [45], [46], throughput [47], and their surrogates [42], [48].

An action is the specific implementation of a policy, whose impact has been underestimated. In the previous works, different action definitions are involved. (1) Determine the (green interval) duration of the incoming stage [10]. (2) Determine the split over the whole cycle [29], [31]. (3) Decide whether to proceed to the next stage in a cyclic sequence [26], [49]. (4) Determine the incoming stage among the available stages [11], [32], [34], [50], [51], [52]. (5) Determine the incoming stage and its duration synchronously [53]. Nevertheless, limited by the algorithms, the optimization objective in most studies is either discrete or continuous, as discussed in Section. I. While it is not their intention to compromise on the above action definitions, they have to due to the absence of appropriate algorithms. Of course, we will detail the motivation and importance of a parameterized action defined in hybrid action space in Section. V-E.

There is a related work [53] that also delves deep into the research of hybrid action space. It is indeed a remarkable work, while it differs from ours in three significant aspects. Firstly, the validation of its hybrid action space's effectiveness is based on only single-agent scenarios with synthetic demand, whereas our study encompasses both single-agent scenarios with synthetic demand and multi-agent scenarios with real-world demand. Secondly, we also provide a clear explanation

TABLE I
EXISTING RL-BASED MODELS WITH DIFFERENT MDP DEFINITIONS, ALGORITHMS, AND ENVIRONMENTS

Reference	Environment & Demand	Algorithm	State	Action	Space	Reward
[25]	An 1×8 corridor; A 2×2 grid network; A 3×3 grid network Synthetic demand	DQN	Queue length; Elapsed time	– Determine the incoming stage among the available stages	Discrete	Sum of the queue length and elapsed time
[26]	An isolated intersection Synthetic demand	DQN but with a self-designed approximation network SAE	Queue length	2 phases Remain or change in a cyclic sequence	Discrete	Absolute difference between the queue length of two phases
[10]	A road network in Tehran Real-world demand	SARSA; Q-Learning; Actor-Critic	Current time; Current phase; Number of vehicles	– Determine the duration of the incoming stage	Continuous	Negative increment of the queue length
[11]	An isolated intersection Real-world demand	CNN; Ape-X DQN but with an interpretable approximation network FRAP	Queue length; Current phase	8 phases Determine the incoming stage among the available stages	Discrete	Average queue length of all movements
[50]	A 1×3 corridor; A 1×5 corridor; Four avenues Real-world demand	DQN	Pressure; Current phase	8 phases Determine the incoming stage among the available stage	Discrete	Negative pressure
[29]	An isolated intersection Synthetic demand	CNN; Dueling Double DQN	Position and speed	4 phases determine the phase split over the whole cycle	Continuous	Negative increment of the cumulative waiting time
[51]	An isolated intersection Synthetic and real-world demand	RNN; REINFORCE; Attention mechanism	General traffic characteristics	8 phases Determine the incoming stage among the available stages	Discrete	Negative pressure
[53]	An isolated intersection Synthetic demand	MP-DQN	Queue length; Current phase	4 phases Determine the incoming stage and its duration synchronously	Hybrid	Negative increment of the queue length
Ours	An isolated intersection; A 1×7 corridor; An 4×5 grid network Synthetic and real-world demand	Distributed H-PPO	Queue length	8 phases Determine the incoming stage and its duration synchronously	Hybrid	Negative increment of the queue length

in the traffic context for how optimizing traffic signal control in the hybrid space leads to a performance boost, which is not covered in [53]. Most importantly, the RL algorithms differ a lot, MP-DQN in [53] is a value-based algorithm incorporating DQN and DDPG, whereas our H-PPO, based on PPO, is a policy-based algorithm, and we will surely provide a detailed comparison between these two algorithms in Section. V-F.

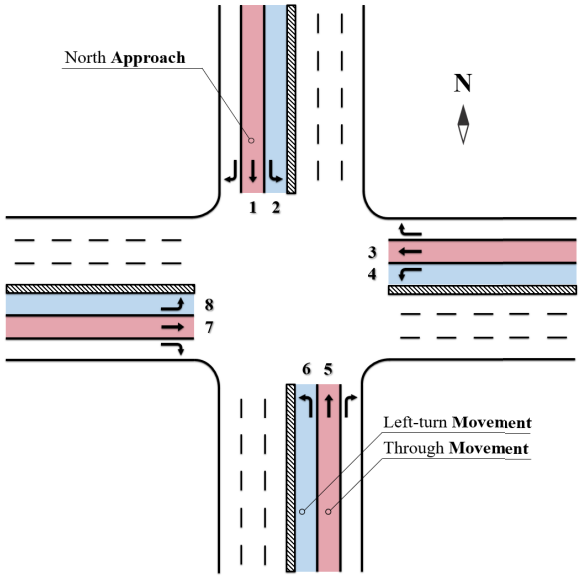
Overall, to better illustrate the differences between existing literature and ours, Table I offers a summary of recent studies on RL-based methods with different MDP definitions, algorithms, environments, and traffic demands.

III. PROBLEM DEFINITION

This paper defines the environment $\mathcal{E}$ as an isolated intersection or an urban section with multiple intersections, where the traffic demand $\mathcal{D}$ is generated utilizing either synthetic data or real-world data. On these intersections, intelligent agents $\mathcal{G}$ are deployed to optimize the traffic flow and reduce congestion by making decisions on the staging and timing of the traffic signals.

A. Traffic Environment Definition

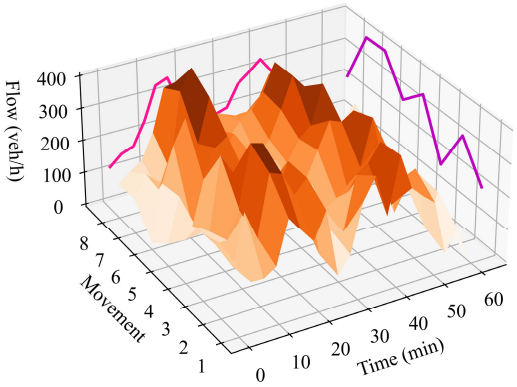
- **Organization**. The intersection has four entry approaches, named North, South, East, and West. In Figure. 1(a), the north entry approach is marked out, which has three lanes corresponding to left-turn, through, and right-turn, respectively.
- **Signal**. To deal with the passing decision dilemma, a three-second yellow signal is set at the end of each stage, corresponding to its pre-specified speed limit, which is 40km/h. A one-second red clearance is also adopted to avoid the potential collisions of different conflicting movements during the signal switch.
- **Movement**. Movements are fractions of traffic flowing in a certain direction. As shown in Figure. 1(a), there are eight traffic movements under the regulations and the right-turn movements are not taken into consideration. Generally, the right-turn traffic can pass under any circumstances, but it is lower prioritized and must yield to antagonistic movements with higher priorities.
- **Stage**. As shown in Figure. 1(b), in contrast to most existing works that considered only four or even two stages, we have adopted the maximum eight stages to approximate the overall staging space.
- **Demand**. Urban traffic flows, as an expression of human behavior, are variable in time and space [5]. Wild fluctuations in urban traffic flows are observable over a period of time, which is referred to as the time-of-day effect. On the other hand, urban traffic flows are typically truncated and discontinuous under signal regulations, resulting in instant disequilibrium across different movements. As shown in Figure. 1(c), to account for aforementioned phenomena, we introduce two notions in our traffic demand formulation, namely, **time disequilibrium** and



(a) Isolated intersection layout

Stage	1	2	3	4	5	6	7	8
Movement Groups								

(b) Eight available signal stages



(c) Traffic demand of different movements across a period of time

Fig. 1. Illustration of the traffic environment and demand.

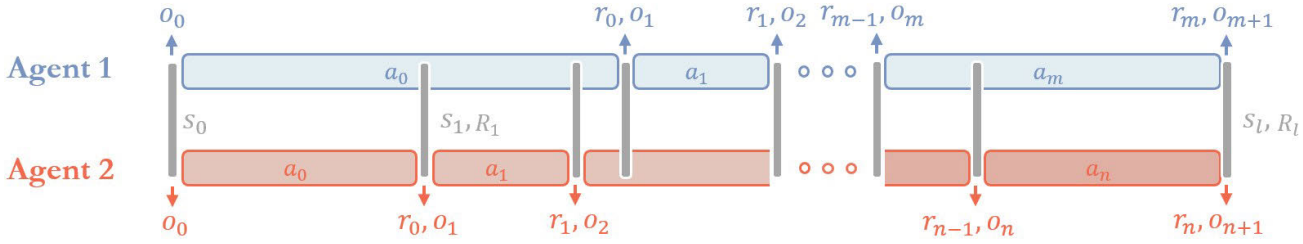


Fig. 2. MacDec-POMDP for asynchronous decision-making MARL.

movement disequilibrium. Time disequilibrium, marked with the pink line, refers to fluctuations in the average flow rate over time, while movement disequilibrium, marked with the purple line, indicates the imbalance between different movements at any given time.

B. Reinforcement Learning Formulation

The traffic signal control problem could be formulated by the decentralized partially observable Markov decision process (Dec-POMDP) $\langle \mathcal{N}, \mathcal{S}, \mathcal{O}, \mathcal{A}, R, P, \gamma \rangle$ [54]. Following its definition, $\mathcal{N} = \{1, \ldots, n\}$ is the set of agents, $\mathcal{S}$ is the state space, $\mathcal{O} = \prod_{i=1}^{n} o^i$ is the product of agents' primitive observation spaces, $\mathcal{A} = \prod_{i=1}^{n} a^i$ is the product of agents' primitive action spaces, namely joint action spaces, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the joint reward function, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function, and $\gamma \in [0, 1]$ is the discount factor. While using the synchronous decision-making architecture to model traffic signal control exposes three specific problems:

- **Low-level granularity**. The traffic signal control requires a precision of at least one second, so the Δ_t should be set to be equal to or less than one second, which many algorithms fail to meet.
- **High inference cost**. Intuitively, an agent only needs to make a decision when its previous action ends. However, synchronous decision-making forces all agents to make decisions at every time step, resulting in a higher cost of model inference.
- **Incomplete actions**. Agent in synchronous decision-making architecture usually does not know when its action ends. Practically, this means that synchronous decision-making may struggle with tasks such as countdowns or broadcasting current intersection information.

To enable asynchronous decision-making, as shown in Figure. 2, macro-action $\langle \{o^i\}, \{a^i\}, \{r^i\}, \{\zeta^i\} \rangle$ [55] is added to Dec-POMDP. $\{o^i\}$ is a finite set of macro-observations for each agent i, $\{a^i\}$ is a finite set of their corresponding macro-actions, $r^i : o^i \times a^i \to \mathbb{R}$ is the corresponding macro-reward, and $\zeta^i : o^i \times a^i \times \mathcal{S} \to [0, 1]$ is the macro-action transition probability function. Each agent i aims to learn a policy $\pi^i(a^i|o^i)$, which determines the best action to take given the observation o^i, so that the following total discounted return is maximized, where $\mathcal{T}^i$ represents the trajectory of agent i:

$$R_t^c = \sum_{i=1}^{n} \sum_{k=0}^{\infty} \gamma^k r^i_{t^i+k+1}. \quad t^i \sim \mathcal{T}^i \tag{1}$$

In the context of traffic signal control, the specific MDP representation takes the form of:

- **Observation**: the queue length q^m of each traffic movement. Note that the superscript m indicates the m^{th} traffic movement and a vehicle is defined as being queuing up when its speed falls below $0.1km/h$.
- **Action**: a parameterized action (u, x) in hybrid action space $a^i = \bigcup_{u\in\mathcal{U}_d}\{(u, x) \mid x \in \mathcal{X}_u\}$, following the notations in [15]. Specifically, u is the discrete stage indicator selected from a finite stage set $\mathcal{U}_d = \{u_1, u_2, \ldots, u_8\}$, and $\mathcal{X}_u \subseteq \mathbb{R}^u$ is the corresponding real-valued continuous parameter, in other words, available duration of the stage.
- **Reward**: the difference between total queue lengths $r_i = \sum_{m=1}^{8} q_{t-1}^m - \sum_{m=1}^{8} q_t^m$, which are yielded at two consecutive timesteps, $t-1$ and t respectively.

IV. Methodology

In this section, we commence with an overview of PPO and its integration with Generalized Advantage Estimation (GAE) [56]. Subsequently, we delve into the details of H-PPO's specialized neural network architecture and its corresponding implementation. Lastly, we explore the expansion of H-PPO to the realm of Multi-Agent Reinforcement Learning (MARL).

A. Proximal Policy Optimization

PPO is a new family of policy-based algorithms for DRL, which alternate between sampling data through interaction with the environment, and optimizing a surrogate objective function using stochastic gradient ascent [8].

$$L^{PPO}(\theta) = -\hat{\mathbb{E}}_t\Big[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\Big] \tag{2}$$

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\bar{\theta}}(a_t|s_t)} \tag{3}$$

where the expectation $\hat{\mathbb{E}}_t[\ldots]$ indicates the empirical average over a finite batch of samples. The first term inside the minimization, $r_t(\theta)\hat{A}_t$, measures the generalized advantage considering the importance sampling trick [57]. The second term, $clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t$, modifies the surrogate objective by clipping the probability ratio $r_t(\theta)$, which removes the incentive for $r_t(\theta)$ to stride outside of the interval $[1-\epsilon, 1+\epsilon]$. Here, ϵ is a hyperparameter to impose restrictions on gradient updates, say, $\epsilon = 0.2$. Note that the generalized advantage estimation $\hat{A}$ [56] is not exactly the difference between action value and state value, say, $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$. The exact generalized advantage estimation takes the form of:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \ldots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \tag{4}$$

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{5}$$

where parameters γ and λ are introduced to implement the bias-variance trade-off when using an approximate value function $V^{\pi,\gamma}$. Different from the one mentioned in Eq. 1, γ is denoted as a variance reduction parameter in an undiscounted problem. On the other hand, similar to the one applied in the eligibility trace [58], λ is essentially a credit assignment parameter of the general advantage estimator. For more details, please refer to [8] and [56].

In specific implementations, the aforementioned surrogate objective $L^{PPO}(\theta)$ will be computed and automatically differentiated by incorporating a policy gradient ascent algorithm (e.g., Adam [59]). Generally, PPO doesn't simply approximate the policy but also utilizes the learned state value function to derive a variance-reduced advantage estimation.

$$L^{VF}(\phi) = \hat{\mathbb{E}}_t\Big[\sum\nolimits_{t'>t}\gamma^{t'-t}r_{t'} - V_\phi(s_t)\Big]^2 \tag{6}$$

B. Hybrid Proximal Policy Optimization

PPO can handle discrete or continuous action space separately, while it fails confronted with hybrid action space. Accordingly, [17] has proposed hybrid proximal policy optimization (H-PPO) to implement reinforcement learning directly over the original hybrid action space.

H-PPO is an actor-critic architecture that consists of multiple parallel sub-actors to decompose the structured action space (into simpler ones), and a global critic to guide updates. As shown in Figure. 3, the critic remains unchanged compared to PPO and it approximates the expected discounted return G_t from the state s_t following the policy π_{θ_d} and π_{θ_c}. To generate the discrete stochastic policy π_{θ_d}, the discrete actor of H-PPO outputs k values $f_{a_1}, f_{a_2}, \ldots, f_{a_k}$ respectively, and the discrete action to take is randomly sampled from the softmax($\vec{f}$) distribution, as marked red on the right of Figure. 3. Similarly, the continuous actor will also output k values, which are the means of different Gaussian distributions, as marked blue on the right of Figure. 3. Variables sampled from Gaussian distributions are further input into a hyperbolic tangent to implement the normalization.[2] The continuous parameter exactly selected is indexed by the discrete action and will be remapped to its real range clipped by the minimum green and maximum green. The discrete action along with the corresponding continuous parameter altogether constitutes the final parameterized action. A pseudo-code of H-PPO is also involved for a better understanding of the manipulations and updates.

C. Asynchronous Decision-Making in MARL

In multi-agent scenarios, incorporating the green light duration into the action space necessitates asynchronous decision-making. This concept deviates from the commonly deployed synchronous decision-making framework and has ignited recent theoretical research [60] and applications in specific domains, such as robotics [61], [62]. The following discussion will investigate how H-PPO, coupled with asynchronous decision-making, could be deployed in the existing multi-agent paradigm (i.e., fully decentralized and centralized training decentralized execution frameworks).

1) Fully Decentralized: Agents within such a framework can only access their own information, and one simple implementation is independent PPO (IPPO). Although a completely distributed architecture is prone to environmental instability, its practical performance remains promising [63], [64].

[2]The normalization will clip the output into the range of $[-1, 1]$, rather than a very small or large one (depending on the scale), which helps to stabilize the gradient backpropagation.

Algorithm 1 Hybrid Proximal Policy Optimization in Actor-Critic Style

Initialize the critic V_ϕ and actor newtorks π_{θ_d}, π_{θ_c}, $\pi_{\bar{\theta}_d}$, $\pi_{\bar{\theta}_c}$ with orthogonal parameters ϕ, π_{θ_d}, π_{θ_c}, $\pi_{\bar{\theta}_d}$, $\pi_{\bar{\theta}_c}$
Initialize the rollout buffer $\mathcal{D}$
for $i \leftarrow 1$ **to** I **do**
  // rollout phase
  for *worker* $\leftarrow 1$ **to** N **do**
    Execute parameterized action $a \sim \pi_{\bar{\theta}_d}$ and $x \sim \pi_{\bar{\theta}_c}$ for T timesteps
    Push trajectories $\{s_t, (a_t, x_t), r_t, s_{t+1}\}$ into the buffer $\mathcal{D}$
  end
  // learning phase
  for $s \leftarrow 1$ **to** NT/M **do**
    Sample a minibatch from the trajectories
    for $j \leftarrow 1$ **to** J **do**
      Calculate the policy loss $L^P(\theta_d)$, $L^P(\theta_c)$ ▷ Eq. 2
      Update $L^P(\theta_d)$ and $L^P(\theta_c)$ with respect to θ_d and θ_d based on Adam
    end
    for $k \leftarrow 1$ **to** K **do**
      Calculate the value function loss $L^{VF}(\phi)$ ▷ Eq. 6
      Update $L^{VF}(\phi)$ with respect to ϕ based on Adam
    end
  end
  $\pi_{\bar{\theta}_d} \leftarrow \pi_{\theta_d}$, $\pi_{\bar{\theta}_c} \leftarrow \pi_{\theta_c}$
end
Return θ_d^*, θ_c^*, ϕ^*

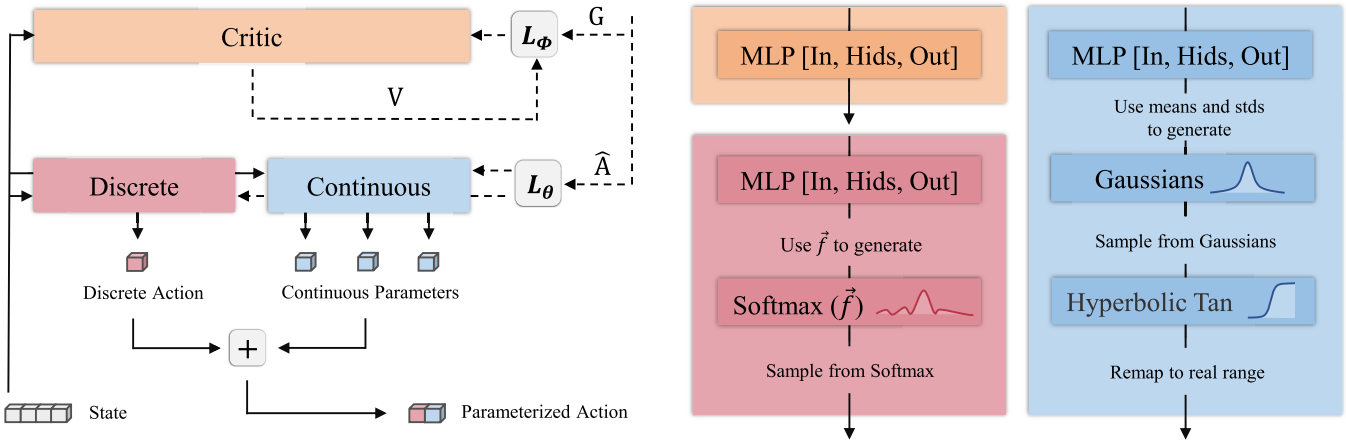


Fig. 3. Detailed architecture of H-PPO.

To facilitate the transition from synchronous decision-making to asynchronous one, it suffices to collect the trajectories of each agent and perform gradient descent accordingly:

$$L^{VF}(\phi_i) = \hat{\mathbb{E}}_{o^i \sim \rho_{\pi^i}, t \sim \mathcal{T}^i}\left[\sum_{t'>t} \gamma^{t'-t} r^i_{t'} - V_{\phi_i}(o^i_t)\right]^2 \tag{7}$$

$$L^P(\theta_i) = -\hat{\mathbb{E}}_{o^i \sim \rho_{\pi^i}, a^i \sim \pi^i, t \sim \mathcal{T}^i}\Big[min(r_t(\theta_i)\hat{A}^i_t, clip(r_t(\theta_i), 1-\epsilon, 1+\epsilon)\hat{A}^i_t)\Big] \tag{8}$$

$$r(\theta_i) = \frac{\pi_{\theta_i}(a^i|o^i)}{\pi_{\bar{\theta}_i}(a^i|o^i)} \tag{9}$$

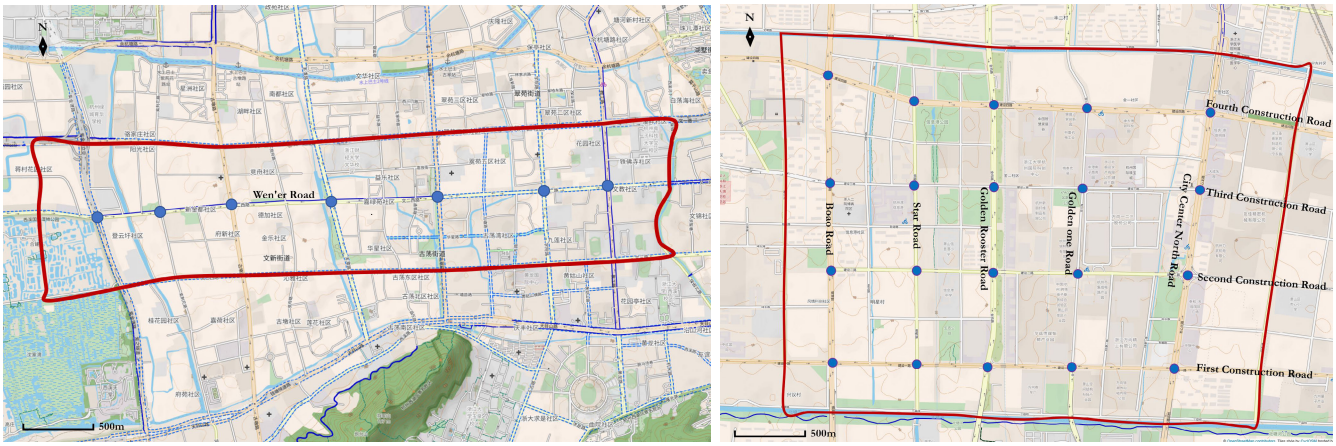
$$\delta^i_t = r^i_t + \gamma V_{\phi_i}(o^i_{t+1}) - V_{\phi_i}(o^i_t) \tag{10}$$

where $t \sim \mathcal{T}^i$ indicates the trajectories are sampled from timesteps at which the i^{th} agent makes a decision, and $a \sim \pi^i$ is an abbreviation for $a \sim \pi_{\theta_i}$.

2) Centralized Training Decentralized Execution: Agents within such framework assume access to complete information during the training. An example of this architecture is Multi-Agent PPO (MAPPO), which is similar to IPPO except for the input to the value networks [65]. Therefore, its actors' learning in an asynchronous manner can solely follow that of IPPO. As for its centralized critic, it can be updated as follows:

$$L^{VF}(\boldsymbol{\phi}) = \hat{\mathbb{E}}_{s \sim \rho_{\boldsymbol{\pi}}, t \sim \boldsymbol{\mathcal{T}}}\left[\sum\nolimits_{t'>t} \gamma^{t'-t} R_{t'} - V_{\boldsymbol{\phi}}(s)\right]^2 \tag{11}$$

where s is the agent-specific global state rather than the concatenated observation, as suggested in [65]. $t \sim \boldsymbol{\mathcal{T}}$ indicates the trajectories are sampled from timesteps where at least one agent makes a decision, and $\mathcal{R}$ represents the joint reward as defined in Section. III-B.



(a) Gudang Sub-district, Hangzhou, China (b) Shushan Sub-district, Hangzhou, China

Fig. 4. Road networks for real-world datasets.

V. Simulation Experiment

A. Simulation Setting

Our experiments are conducted in SUMO (Simulation of Urban MObility), an open-source, microscopic, time-discrete, and space-continuous traffic simulation platform designed to handle large-scale networks [66]. The simulation of our experiments is episode-based, which lasts 3600 seconds in this paper. Car-following is modeled using IDM [67] (please refer to Appendix I for more details), and lane-changing is not considered for simplicity.

B. Traffic Demand Datasets

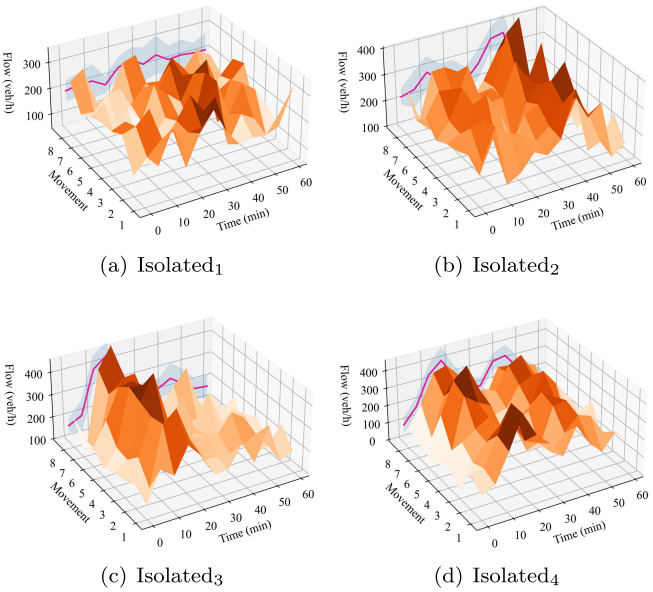
1) Real-World Datasets: we collect electronic police data from our collaborators in Hangzhou. This dataset comprises time-stamped records collected by surveillance cameras installed near intersections and each record in this dataset contains time, camera ID, and information about vehicles. Based on the coverage of electronic police data, as shown in Figure. 4, we have selected two road networks, namely, ***Corridor***$_{1\times7}$ and ***Grid***$_{4\times5}$, and the road structures of these networks are retrieved from OpenStreetMap.[3] By employing travel chain reconstruction techniques, we then generate origin-destination (OD) matrices for these test regions and feed them into the SUMO simulation platform. The detailed data statistics regarding these two regions are listed in Tables II. Note that to obtain a more accurate estimation of the traffic flow rate, it is calculated within a time window of 15 minutes.

2) Synthetic Datasets: we adopt the same isolated intersection layout as presented in Figure. 1. As demonstrated in Figure. 5, four sets of synthetic traffic demand are then generated, denoted as ***Isolated***$_{1-4}$, each lasting for one hour. To be specific, we have utilized the bimodal normal distribution and the uniform distribution to model the time disequilibrium and movement disequilibrium effects, respectively.

[3]https://www.openstreetmap.org

TABLE II
Data Statistics of Real-World Scenarios

Scenarios	Time period (min)	Arrival rate (veh/hour)		
		Mean	Std	Max
$Corridor_{1\times7}$	0 - 15	160.85	63.42	308
	15 - 30	180.12	58.34	376
	30 - 45	190.74	65.73	404
	45 - 60	178.26	54.26	352
$Grid_{4\times5}$	0 - 15	140.82	44.07	312
	15 - 30	135.47	38.12	296
	30 - 45	133.53	42.86	284
	45 - 60	142.74	46.73	304



(a) Isolated$_1$ (b) Isolated$_2$

(c) Isolated$_3$ (d) Isolated$_4$

Fig. 5. Illustration of synthetic traffic demand datasets.

C. Constraints

To ensure basic driving safety, the red clearance interval is considered at the end of each stage, which could be determined on the basis of traffic organization and speed limits. In practice,

as mentioned in Section. III-A, the red clearance interval is set to be one second for the sake of simplicity.

Both the minimum and maximum green times are taken into account when determining the duration of each stage. To be more specific, we have set a maximum green time of fifty seconds to impose an upper bound and a minimum green time of ten seconds to take care of driver reaction times and pedestrian crossings.

D. Evaluation Metrics

To measure the performance of a given model in one round of the simulation, we use the **average delay** per vehicle and **average queue length** over one episode as the evaluation metrics, formulated as below.

$$M_{delay} = \frac{1}{|T|}\sum_{t=0}^{|T|}\left[\frac{D_t}{\sum_{r\in R}\sum_{v\in V(r,t)}1}\right]$$

$$D_t = \sum_{r\in R}\sum_{v\in V(r,t)}\left[t - t_0\left(v,r\right) - t_{min}\left(r\right)\right] \tag{12}$$

$$M_{queue} = \frac{1}{|T|}\sum_{t=0}^{|T|}\sum_{l\in L_{in}} q\left(t,l\right) \tag{13}$$

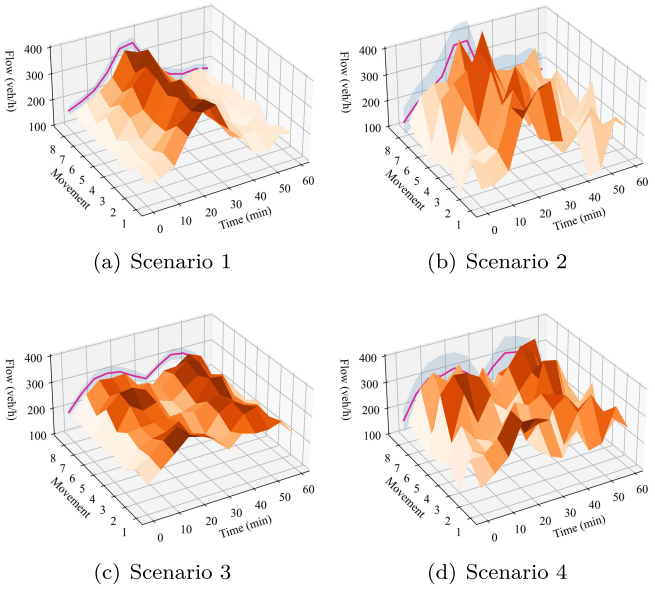
where - t is the timestep; - $|T|$ is the running time of one episode; - D_t is the cumulative travel delay of all vehicles finishing their routes at timestep t; - R is the set of routes at the intersection(s); - $V\left(r,t\right)$ is the set of vehicles finishing the route r at timestep t; - $t_0(v,r)$ is the timestep when vehicle v enters into route r; - L_{in} is the set of incoming lanes at the intersection(s); - $q\left(t,l\right)$ is the queue length of lane l at timestep t.

E. Intrinsic Imperfections of Different Optimization Strategies

Here, we associate action spaces with different optimization strategies. Note that the action space pattern in **italics** now represents the **alias** of a specific optimization strategy.

- The ***Continuous*** optimization strategy involves a pre-specified staging, while the (green interval) duration of each stage is optimizable. At each timestep t, the agent calculates the optimal duration of the incoming stage or the split over the entire cycle.
- The ***Discrete*** optimization strategy involves a fixed decision interval, while the staging is acyclic and optimizable. At each timestep t, the agent determines the incoming stage among the available stages.
- The ***Hybrid*** optimization strategy allows for the optimization of both staging and timing. At each timestep t, the agent synchronously determines the incoming stage and the corresponding duration.

1) Continuous Optimization Strategy: Determining a demand-oriented staging in advance is nearly impossible due to the lack of prior information. While some may attempt to substitute it with historical statistics, this crude simplification has been proven inefficient in implementation [18] because of the sporadic and long-term fluctuations in traffic demand. Previous works, as a result, did not show enough considerations for the staging design and have adopted relatively simple ones:



(a) Scenario 1 (b) Scenario 2

(c) Scenario 3 (d) Scenario 4

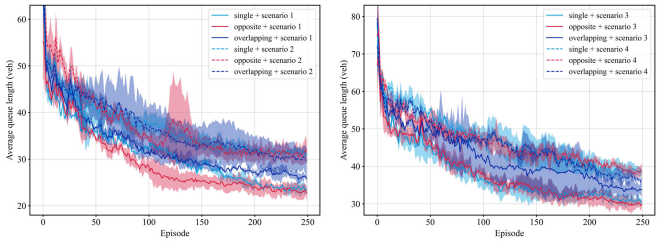
Fig. 6. Illustration of synthetic traffic demand regarding scenarios 1-4.

- **Single** [29]: $\{N_{TL} \rightarrow E_{TL} \rightarrow S_{TL} \rightarrow W_{TL}\}$. It's a four-phase staging and is used to denote that vehicles are released from a **single** approach.
- **Opposite** [10]: $\{N_{TL} \rightarrow E_{TL} \rightarrow S_{TL} \rightarrow W_{TL}\}$. It's a four-phase staging and is used to denote that vehicles are released in **opposite** directions.
- **Overlapping**: $\{NS_T \rightarrow N_{TL} \rightarrow NS_L \rightarrow EW_T \rightarrow E_{TL} \rightarrow EW_L\}$. It's an eight-phase staging following the ring-barrier structure, with **overlapping** phases included.

Intuitively, a poorly-designed traffic staging scheme limits the search space to a prior-biased subspace, potentially leading to suboptimal policies. Experiments are conducted based on scenarios 1-4 to validate this conjecture. To better understand the demand settings, scenarios depicted in Figure. 6 should be paired together, such as scenarios 1 and 2. In the former, the variances in traffic demand among different movements are relatively inconspicuous at any given time, as indicated by the pink shadow. As for the latter, it shares the same average traffic flow rate as the previous one but exhibits a more pronounced movement disequilibrium effect.

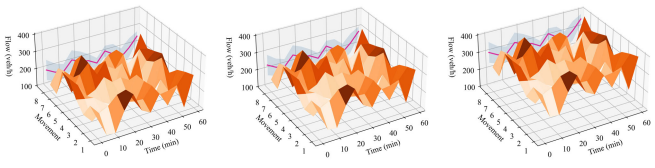
Utilizing the PPO with continuous action space, the *continuous* optimization strategy is implemented in the preceding scenarios. Figure. 7 illustrates the control performance of different stagings under different demand scenarios, where the average queue length serves as the performance metric.

In scenario 1 shown in Figure. 7(a), as indicated by solid lines, it is reasonable to observe that "single" consistently exhibits similar control performance to "opposite", as the movement disequilibrium in this scenario is almost negligible and these two stagings share the same demand-supply match. The control performance of "overlapping " has somewhat degraded compared to the previous two stagings. It is supposed that in situations where traffic flows are relatively symmetrical, the existence of overlapping phases has resulted in the non-saturation release of certain movements.



(a) Performance evaluation based on scenarios 1-2 (b) Performance evaluation based on scenarios 3-4

Fig. 7. Performance evaluation of different staging initialization based on scenarios 1-4.



(a) Scenario 5 (b) Scenario 6 (c) Scenario 7

Fig. 8. Illustration of synthetic traffic demand regarding scenarios 5-7.

When a more pronounced movement disequilibrium is imposed in scenario 2, as indicated by dashed lines, "single" and "opposite" finally fall behind in control performance compared to those in scenario 1. It is suggested that in the presence of movement disequilibrium, the agent has to ask for an excess of the green interval duration to mitigate the impact of demand-supply mismatch, which correspondingly leads to the non-saturation release of certain movements. As for "overlapping", its performance has also shown a certain degree of degradation, though not as severe as that of the "single" and "opposite" stagings. This finding suggests that the staging with overlapping phases indeed exhibits some degree of adaptability, though limited, when confronted with asymmetric traffic flows.

Intrinsic Imperfection: continuous control strategy with fixed staging cannot always provide a perfect counterbalance to the movement disequilibrium effects.

2) Discrete Optimization Strategy: Intuitively, *discrete* is able to tackle the movement disequilibrium effects, by dynamically allocating right-of-way among different movement groups. However, it fails when it comes to time disequilibrium. Under heavy traffic conditions, for example, it is recommended to compress the proportion of lost time by pursuing a longer green interval duration. Conversely, under light traffic conditions, it is recommended to avoid unsaturated release by pursuing a smaller green interval. These phenomena indicate that the ideal decision interval should be demand-oriented rather than pre-fixed. This conjecture is further validated based on synthetic scenarios 5-7 in Figure. 8. These scenarios share the same degree of movement disequilibrium but differ in their average flow rates. Specifically, the average flow rates of these three scenarios are set to 150 veh/h, 200 veh/h, and 250 veh/h, respectively, to simulate **light**, **medium**, and **heavy** traffic.

As shown in Figure. 9, a rough grid search with a five-second resolution is initially conducted to evaluate the control performance of different decision intervals. According to the Figure. 9(a-c), we find that different decision intervals indeed lead to different levels of control performance. Moreover, the corresponding scatter plots also indicate the existence of an optimal decision interval. Therefore, a more refined grid search with a one-second resolution within a reasonable range is then performed to determine the optimal decision interval.

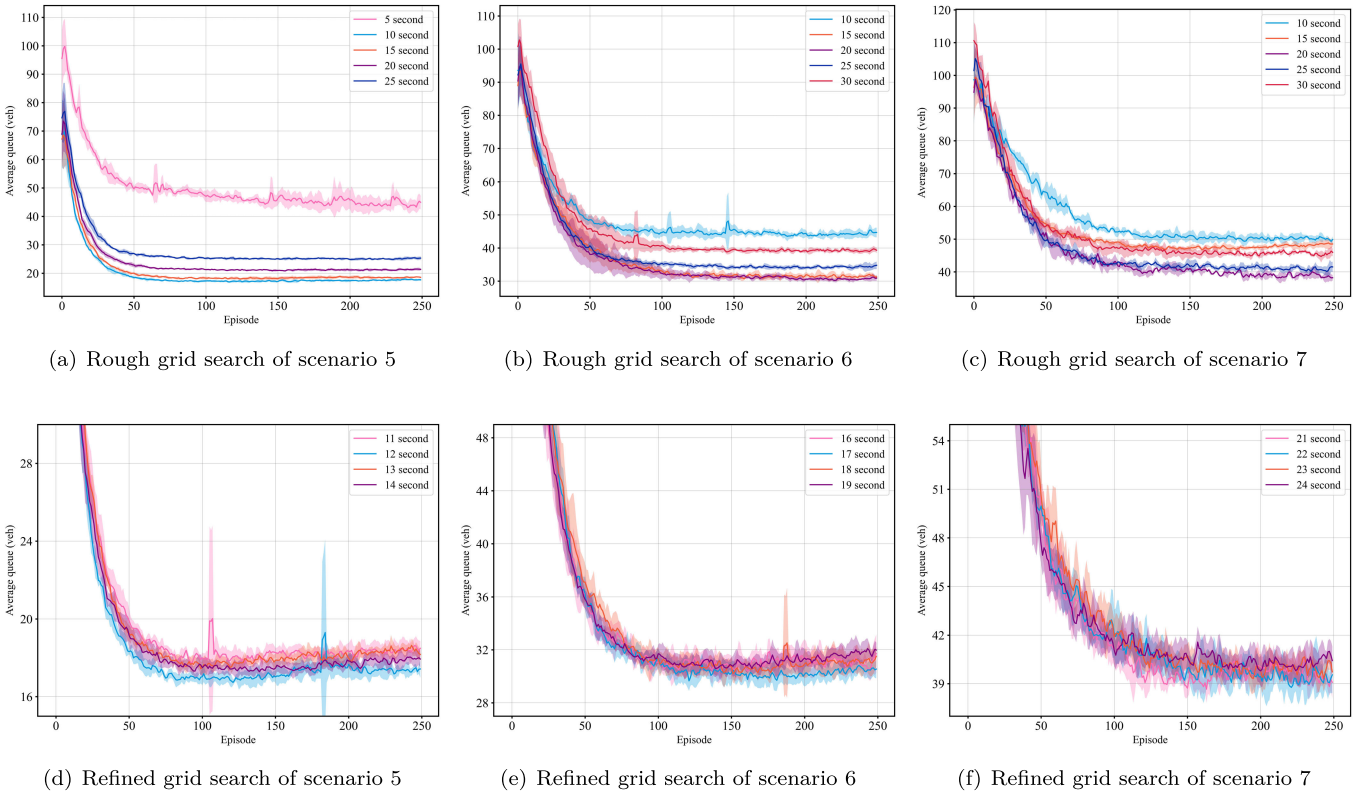
Figure. 10 presents the overall control performance of different decision intervals under different traffic demand scenarios. The scatter plots of different scenarios all appear in a 'V' shape. Specifically, when the decision interval is small, such as 5 seconds, the agent tends to switch its stage more frequently, leading to an increase in lost time and a consequent decrease in control performance. Conversely, when the decision interval is large, for example, 30 seconds, lost time is reduced, but the non-saturation release of vehicles occurs in certain directions. Being dependent upon the decision interval, therefore is not a universal solution to time disequilibrium. Even if the ideal decision interval could be determined by a grid search, it's just the so-called optimum for a whole episode but not the exact optimum for every timestep.

Intrinsic Imperfection: the decision interval of the *discrete* control strategy is a demand-oriented hyperparameter. Moreover, it should be set to less than or equal to one second to guarantee the minimum control accuracy, which almost all algorithms fail to meet.

F. Performance Evaluation

Our method H-PPO is compared with the following state-of-the-art model-based and RL-based benchmarks in terms of average queue length and average delay.

- **Fixed-Time** [1]. Implementing optimization based on historical statistics, Fixed-Time control is widely applied in steady and unsaturated traffic conditions. Moreover, it is one specific policy within the optimization strategies with pre-specified staging but optimizable timing.
- **Max-Pressure** [68]. Max-Pressure aims to reduce the risk of over-saturation by balancing the (queue length) pressure between different traffic movements. At each timestep t, it greedily selects movement groups with the maximum pressure. It's one specific policy within the optimization strategies with fixed decision intervals but the optimizable staging.
- **PPO-continuous** [8]. Utilizing the PPO with continuous action space, PPO-continuous could determine the duration of the incoming stage or the split over the whole cycle, under a pre-specified staging. Specifically, two policies are adopted, "opposite" and "single", which are defined in Section. V-E1.
- **PPO-discrete** [8]. Utilizing the PPO with discrete action space, at each timestep t, PPO-discrete selects the incoming stage among the available stages. Similar to Section. V-E2, the optimal decision interval for each environment is determined by a grid search.
- **MP-DQN** [53]. It is a state-of-the-art algorithm that implements optimization in hybrid action space. Moreover, it's



(a) Rough grid search of scenario 5 (b) Rough grid search of scenario 6 (c) Rough grid search of scenario 7

(d) Refined grid search of scenario 5 (e) Refined grid search of scenario 6 (f) Refined grid search of scenario 7

Fig. 9. Grid search of the optimal decision intervals regarding scenarios 5-7.

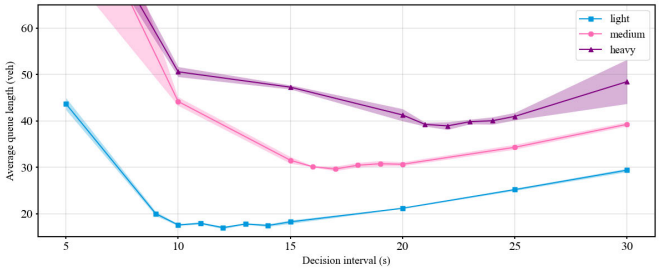


Fig. 10. Performance evaluation of different decision intervals based on scenarios 5-7.

a value-based algorithm that incorporates both DQN and DDPG networks, which generate discrete policies and continuous policies, respectively.

- **H-PPO**. Using H-PPO, it can simultaneously determine the upcoming stage and its duration. However, we didn't employ vanilla H-PPO. Instead, we used async-HPPO in single-agent scenarios, and async-IHPPO in multi-agent scenarios.

1) Benchmark Selection Logic: We initially evaluate the performance of different PPO variants, while the popular state-of-the-art RL-based algorithms defined in discrete or continuous action spaces, such as FRAP [11] and IntelliLight [48], are not included. Firstly, it's our primary intention to validate the superiority of the *hybrid* optimization strategy over others. Therefore, to ensure credibility, a training environment with the same MDP definition, the same RL algorithm, and similar neural networks is required, all of which are satisfied by the PPO variants. Secondly, algorithms used in previous works are actually covered by the PPO variants. For instance, FRAP is an algorithm based on distributed D3QN but with a self-designed and interpretable Q-Network. It implements the same optimization strategy as PPO-discrete.

Regarding MP-DQN [53], given that it also implements the *hybrid* optimization strategy, a performance comparison is inevitable. Similarly, we have made every effort to ensure implementation[4] consistency during the evaluation process.

2) Evaluation Result: The control performances of all methods in terms of average queue length and average delay are presented in Tables III and IV. Note that the **improvement** is defined as the percentage by which H-PPO surpasses the best benchmark. Both Fixed-Time and Max-Pressure are deterministic methods, whereas the PPO family and MP-DQN are all stochastic methods, of which variances processed under five random seeds are therefore involved.

As expected, RL methods tend to outperform conventional ones like Fixed-Time and Max-Pressure, benefitting from their abilities to capture real-time information and offer the prospect of reaching beyond human intuition.

Compared to PPO variants, H-PPO exhibits superior control performance. Specifically, H-PPO outperforms the best-performing PPO variants by 15.38% and 14.07% in terms of average queue length and delay, respectively. Additionally,

[4]The implementation code are mainly referenced from https://github.com/abderraouf2che/Hybrid-Deep-RL-Traffic-Signal-Control and https://github.com/cycraig/MP-DQN.

TABLE III
PERFORMANCE EVALUATION IN TERMS OF AVERAGE QUEUE LENGTH WITH **BEST** AND SECOND BEST HIGHLIGHTED

Algorithms	Scenarios					
	$Isolated_1$	$Isolated_2$	$Isolated_3$	$Isolated_4$	$Corridor_{1\times7}$	$Grid_{4\times5}$
Fixed-Time	41.82	64.69	43.35	44.76	452.19	1036.24
Max-Pressure	28.98	41.20	29.46	41.87	301.24	891.25
PPO-discrete	28.42±0.77	40.22±0.79	30.46±1.09	40.53±1.19	281.41±6.96	675.84±19.76
PPO-continuous single	29.25±2.16	37.50±4.13	21.27±1.45	37.15±2.50	282.15±6.26	984.51±42.14
PPO-continuous opposite	27.55±2.31	37.34±2.42	29.49±2.35	38.12±0.98	227.95±7.77	962.06±48.96
MP-DQN	24.31±2.78	36.52±3.82	23.16±2.96	39.58±3.32	277.08±37.14	664.56±107.21
H-PPO	**21.31±0.59**	**31.59±0.81**	**19.24±0.66**	**31.69±1.28**	**205.77±4.57**	**554.97±14.06**
Improvement	12.34%	13.50%	9.54%	14.70%	9.73%	16.49%

TABLE IV
PERFORMANCE EVALUATION IN TERMS OF AVERAGE DELAY WITH **BEST** AND SECOND BEST HIGHLIGHTED

Algorithms	Scenarios					
	$Isolated_1$	$Isolated_2$	$Isolated_3$	$Isolated_4$	$Corridor_{1\times7}$	$Grid_{4\times5}$
Fixed-Time	73.80	93.05	84.65	94.92	213.49	684.23
Max-Pressure	57.96	73.49	60.68	77.51	201.47	501.21
PPO-discrete	55.32±0.77	82.29±3.16	68.57±2.55	80.93±2.21	190.85±6.38	498.43±19.21
PPO-continuous single	60.56±5.08	77.55±5.23	44.94±2.57	74.96±4.92	191.53±5.87	456.40±47.28
PPO-continuous opposite	58.82±2.94	76.28±3.72	62.99±4.43	76.23±1.85	195.23±5.61	441.61±9.70
MP-DQN	51.20±7.09	70.75±6.41	48.30±3.82	74.44±3.64	225.13±31.48	473.56±55.43
H-PPO	**40.70±0.95**	**64.16±3.12**	**40.28±1.22**	**66.35±2.49**	**172.51±3.52**	**394.52±8.03**
Improvement	20.51%	9.31%	10.37%	10.87%	9.61%	10.66%

H-PPO demonstrates faster convergence speed (as evidenced by the slope of the convergence curves[5] in Appendix III) and slightly better robustness, with an average variance reduction of 14.25% and 8.32%. This result provides compelling evidence for the superiority of the hybrid action space over other spaces, and we will offer a more detailed analysis of the reasons for this outperformance in the subsequent section.

In comparison to the state-of-the-art algorithm in hybrid action space, MP-DQN, our algorithm H-PPO outperforms it with an average reduction of 17.58% and 16.23% in queue length and delay, as well as a substantial average reduction of 78.55% and 68.65% in their corresponding variances. This can be attributed to the distinct network structures utilized by these two algorithms. MP-DQN, which incorporates both DQN and DDPG, is essentially a nested network structure. Specifically, its DQN that outputs discrete policies, in addition to the queue length input consistent with H-PPO, also receives the optimal continuous policy inputs inferred by the DDPG network. It is supposed that using the output of one network as the input of another usually amplifies the approximation errors of the preceding network. In contrast to MP-DQN, H-PPO generates its discrete and continuous policies independently and utilizes the early stopping trick to prevent the over-updating of certain policy networks, which guarantees that the approximation error of a single network will not propagate to another.

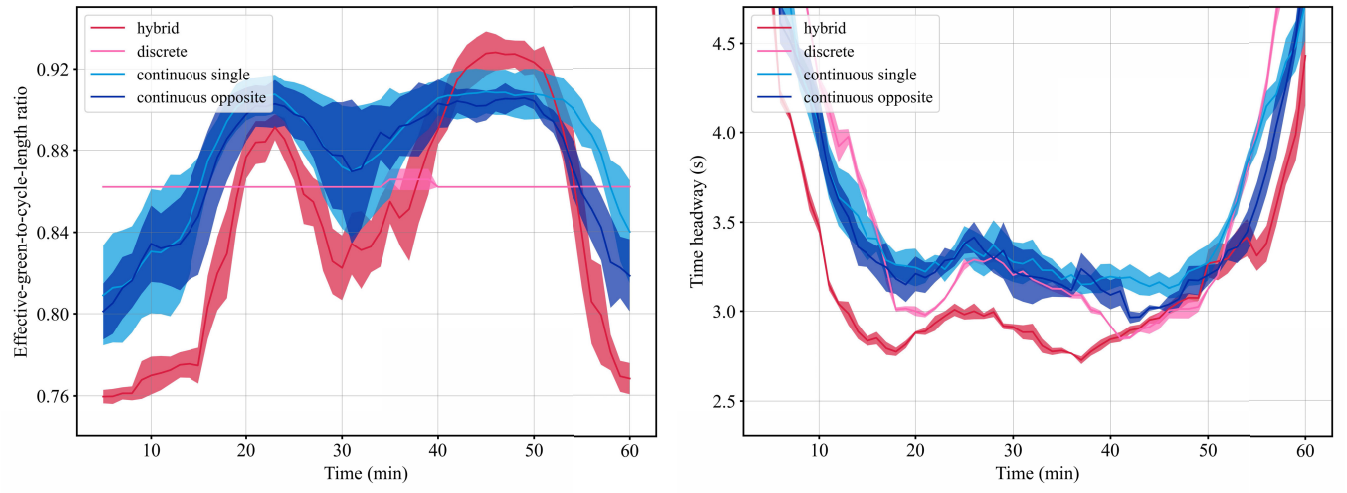
[5]The sudden drop in PPO-continuous is a result of the state normalization trick.

VI. DISCUSSION

A. Naïve Philosophy

Technically, there are two factors contributing to the service ceiling of an intersection. Lost time, including start-up and clearance lost time, is the explicit loss of the manipulation process. On the other hand, unsaturated release, resulting from the right-of-way redundancies of certain movements, inhibits manipulation implicitly. While you can't have your cake and eat it, too. Increasing the green interval duration (less frequent switching, in other words) will inevitably lead to unsaturated release, and vice versa, corresponding to the 'V' shape curve in Figure. 10. Therefore, it is of great importance to reach the equilibrium point at each timestep t, effectively balancing the trade-off between frequent switching and unsaturated release, which we refer to as the **naïve philosophy** of traffic signal control.

In Figure. 11, we present the **effective-green-to-cycle-length ratio** and **time headway** of different optimization strategies over one evaluation episode. Note that these metrics are calculated and averaged within a five-minute rolling



(a) Effective-green-to-cycle-length ratio (b) Time headway

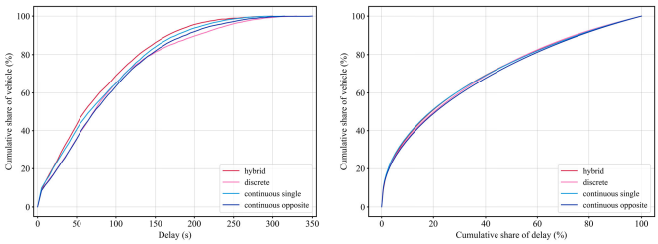
Fig. 11. Illustration of the naïve philosophy based on $Isolated_4$.

window. The effective-green-to-cycle-length ratio is defined as:

$$O = \frac{\sum_i g_i}{\sum_i [g_i + l_i]} = \frac{\sum_i g_i}{\sum_i [g_i + (l_s + (Y + R_c - e))_i]} \tag{14}$$

where - i is the stage within the rolling window $\mathcal{W}$; - g is the effective green time of the stage; - l is the sum of the start-up lost time and clearance lost time; - l_s is the start-up lost time, 2s in this paper; - Y is the yellow change interval, 3s in this paper; - R_c is the red clearance interval, 1s in this paper; - e is the extension of effective green, 2s in this paper. Due to its inherent greed, the effective-green-to-cycle-length ratio of *discrete* remains almost unchanged throughout the manipulation process. This kind of rigidity, as analyzed in Section. V-E2, deprives itself of the ability to adjust to time disequilibrium effects, resulting in the unsaturated release in the first ten minutes as well as the frequent switching during two traffic peaks. Although the optimal decision interval could be determined by a grid search, it's still a local optimum as the decision interval is optimized across the whole episode rather than on every timestep.

Both *hybrid* and *continuous* make use of the time-variant green interval duration to address the time disequilibrium effects. Specifically, the traffic demand during the first ten minutes is relatively low, and there is no risk of oversaturation. The green interval duration is therefore lowered in order to pursue a smaller time headway (i.e., more saturated release). At around thirty minutes, the traffic demand increases gradually and oversaturation becomes inevitable. *Hybrid* and *continuous* then extend the green interval duration to avoid frequent switching. However, the *continuous'* prior-biased search space again deprives itself of the direct satisfaction towards the real-time demand. As a result, *continuous* has to ask for an excess of the green interval duration to dilute the impact of supply-demand mismatch, which correspondingly results in the unsaturated release in Figure. 11(b). *Hybrid*, on the other hand, is stage-based and could meet traffic demand in a more direct manner, which allows it to adopt a more aggressive policy.



(a) cumul. delay distribution (b) Lorenz curve

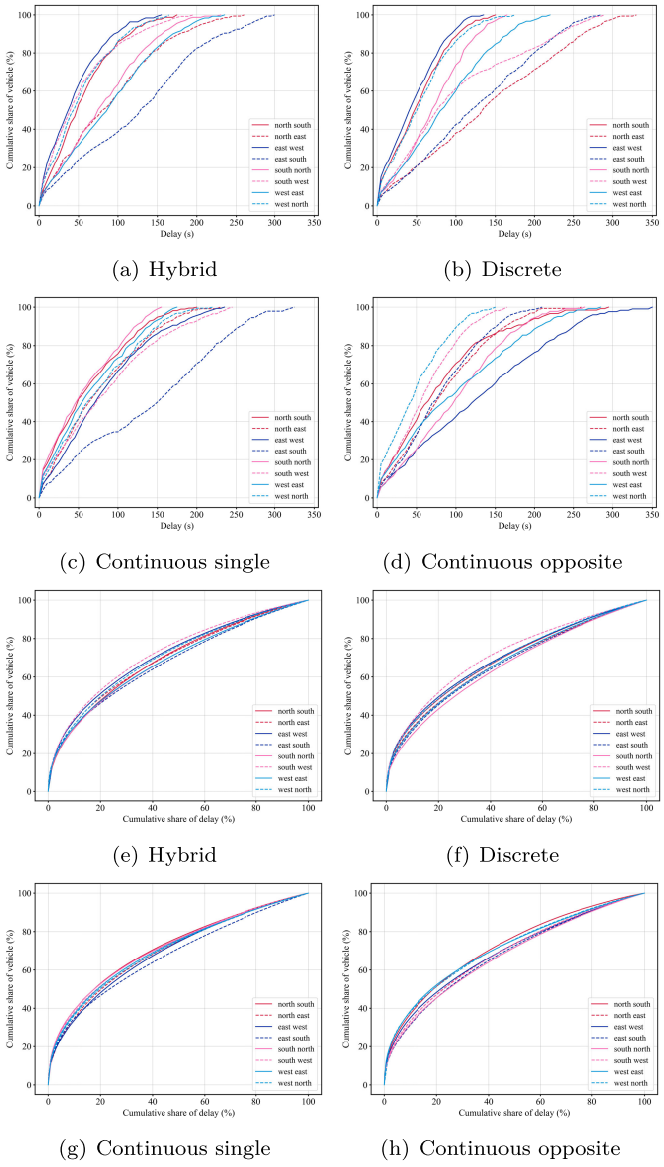
Fig. 12. Illustration of efficiency and fairness of all vehicles based on Isolated_4.

We are not suggesting that the optimal policy should be saturated-release-oriented or that *discrete* and *continuous* have given up implementing the trade-off. The crux of the matter is that according to the results of our experiments, both *discrete* and *continuous*, limited by their action spaces, do not perform well in reaching the equilibrium point on every timestep. *Hybrid*, leveraging hybrid action space, is more flexible and could better implement the naïve philosophy, which explains why H-PPO outperforms PPO-discrete or PPO-continuous in Section. V-F.

B. Efficiency and Fairness

There is a concern that the agent with an acyclic staging may maximize the aggregate interests at the expense of the minority. In other words, as the average delay converges to the optimum, some vehicles are likely to suffer severer delays.

The cumulative delay distribution and corresponding Lorenz curve for all vehicles under different optimization strategies are illustrated in Figure. 12. According to the cumulative delay distributions, *Hybrid* is demonstrated to be the most efficient



(a) Hybrid (b) Discrete

(c) Continuous single (d) Continuous opposite

(e) Hybrid (f) Discrete

(g) Continuous single (h) Continuous opposite

Fig. 13. Illustration of efficiency and fairness of different movements based on Isolated$_4$.

strategy, as the efficiency is defined as the expected delay of one vehicle (i.e., the area to the left of the cumulative delay distribution), which is consistent with the training results in Section. V-F. It is unexpectedly revealed that the Lorenz curves of different optimization strategies are found to be similar. With corresponding Gini coefficients ranging from 0.289 to 0.310, the deviation is only 7.27%, indicating that right-of-way under stage-based strategies is equally distributed as that under phase-based strategies.

We also analyze the cumulative delay distributions and corresponding Lorenz curves for different movements under different optimization strategies, as shown in Figure. 13. Similarly, the cumulative delay distributions indicate that some movements are indeed more efficient than others. This efficiency dispersion may be partly attributable to differences in demand and supply, as higher demand leads to less efficient release somehow, and different optimization strategies may affect supply differently. Through the Lorenz curves, we investigate the right-of-way distribution for each movement. As illustrated in Figure. 13(e-h), there is some dispersion among movements, but never does a Gini coefficient deviate from the average by more than ten percent, and the dispersion is similar under different optimization strategies. Moreover, fairness seems to be unrelated to efficiency, at least in this case, as there is no evidence that vehicles on more efficient movements will be released more equally or not.

Furthermore, it appears that fairness and efficiency are not necessarily correlated, at least in this case, as there is no evidence to suggest that vehicles on more efficient routes are released more equally or not.

In fact, the right-of-way distribution has always been a zero-sum game, that is, the (right-of-way distribution) inequality is inevitable and we should consider fairness when pursuing efficiency. Experiments demonstrate that our proposed method not only helps improve efficiency but also, though counterfactually, guarantees that its right-of-way is equally distributed as that under other policies.

VII. Conclusion

In this paper, we utilize H-PPO to optimize both the staging and duration of traffic signals in a synchronous manner. Based on single-agent and multi-agent scenarios, the intrinsic imperfections of discrete or continuous action spaces for traffic signal control have been revealed. By merging these action spaces into a hybrid one, our proposed method is able to better balance the trade-off between unsaturated release and frequent switching, therefore having shown superior performance compared to existing benchmarks. Furthermore, our analysis of Gini coefficients also demonstrates that our method does not compromise on fairness while improving efficiency.

Moving forward, our future work is threefold: (1) We plan to incorporate the proposed method into a state-of-the-art multi-agent reinforcement learning architecture, where the policy monotonic improvement guarantee and the asynchronous execution are available. (2) We are wondering whether it will facilitate richer cooperation behaviors in the traffic signal control context. (3) And if there is any, is it interpretable?

Appendix I
Car-Following Model (IDM) Set-up

See Table V.

Appendix II
PPO Family Set-up

See Table VI and Figure 14.

In order to investigate the impact of the two main parameters, clipping (ϵ) and GAE discount (λ), on the PPO family, we have conducted grid search experiments based on isolated$_1$. ϵ is the hyperparameter used to clip the policy objective and determine how much the new policy can deviate from the old policy while still improving the objective function. It is typically set to a small value between 0.1 and 0.3.

TABLE V
IDM SET-UP FOR ALL EXPERIMENTS

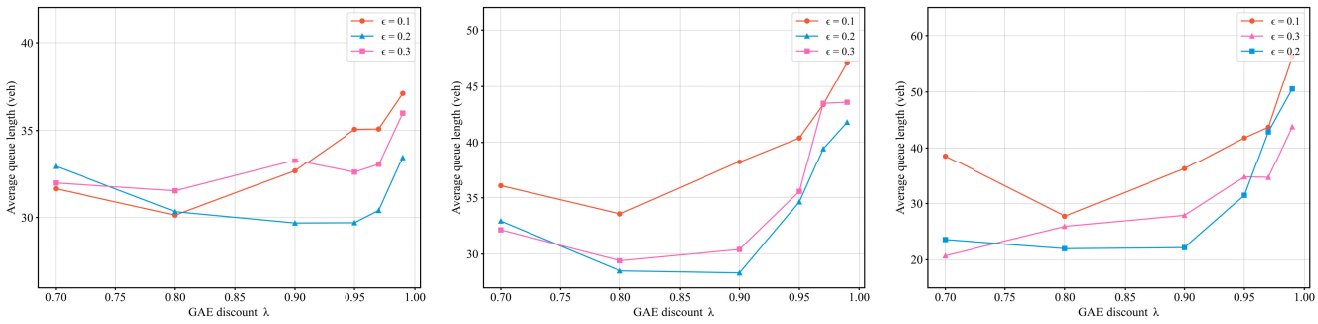
Attribute	Description	Value
Min gap	Minimum gap when standing (m).	2.5
Accel	The acceleration ability (m/s^2).	2.6
Decel	The deceleration ability (m/s^2).	4.5
Emerg decel	The maximum deceleration ability in case of emergency (m/s^2).	9
Startup delay	The extra delay before restartup (s).	1
Tau (τ)	The driver's desired time headway (s).	2
Delta (δ)	Acceleration exponent.	4
Stepping	The internal timestep length (s).	0.25
Speed limit	The maximum speed allowed on the lane (m/s).	15
Speed factor	The vehicle's expected multiplier for speed limit.	Norm(0.9, 0.15)
Depart speed	The speed of the vehicle at insertion.	"Random"[a]

[a] A random speed between 0 and max speed is used, where max speed = speed limit $*$ speed factor.

TABLE VI
PPO FAMILY SET-UP FOR ALL EXPERIMENTS

Hyperparameter	PPO-con	PPO-dis	H-PPO
Rollout buffer size	2000	2000	2000
Batch size	256	256	256
Critic learning rate	0.001	0.001	0.001
Actor learning rate	0.0003	0.0003	0.0003
Std learning rate	0.004	–	0.004
Learning rate decay	0.995	0.995	0.995
Initial log std	-0.4	–	-0.4
Critic hidden layers	[256, 128, 64]	[256, 128, 64]	[256, 128, 64]
Actor hidden layers	[256, 128, 64]	[256, 128, 64]	[256, 128, 64]
Discount factor (γ)	0.99	0.99	0.99
GAE discount (λ)	0.95	0.9	0.8
PPO clipping (ϵ)	0.2	0.2	0.2
Entropy coefficient(α)	–	0.005	0.005
Policy epochs	10	20	20
Target KL	0.05	0.025	0.025 (0.05)[a]
Orthogonal with gain	1.41	1.41	1.41

[a] The data in parentheses are parameters for continuous actors, the same as below.



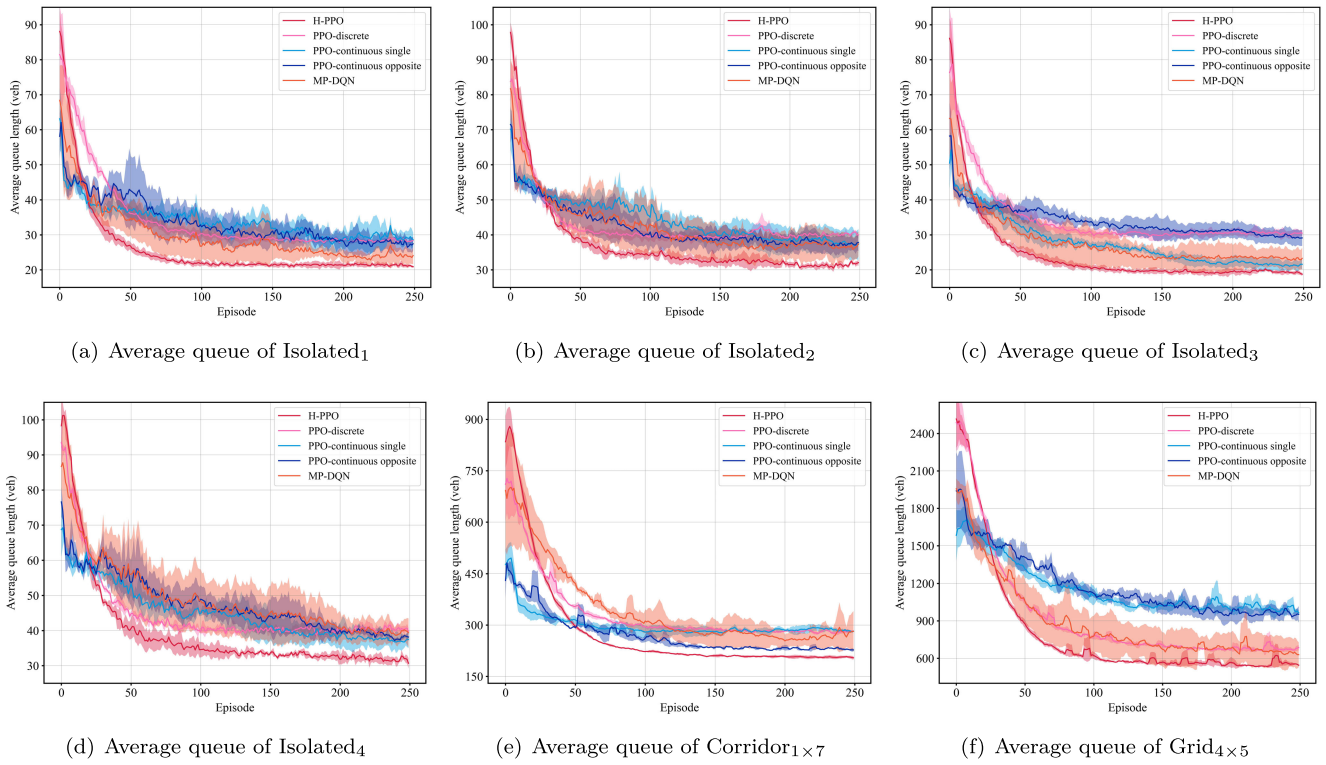
(a) Grid search of PPO-con (b) Grid search of PPO-dis (c) Grid search of H-PPO

Fig. 14. Grid search of PPO family regarding clipping (λ) and GAE discount (ϵ) based on $isolated_1$.

Through grid search, with a resolution of 0.1, we find that the optimal ϵ is 0.2, which is consistent with the results reported in [8]. This indicates that being too conservative or too aggressive in policy updates harms final performance. As for λ, it represents a trade-off between unbiasedness and low variance in estimating the advantage. Through a grid search, we have determined a suitable value of λ for H-PPO, PPO-discrete, and PPO-continuous to be 0.8, 0.9, and 0.95, respectively. Furthermore, we would like to emphasize that the parameters are not over-tuned during the evaluation process and are mainly based on isolated$_1$.

APPENDIX III
TRAINING RESULTS

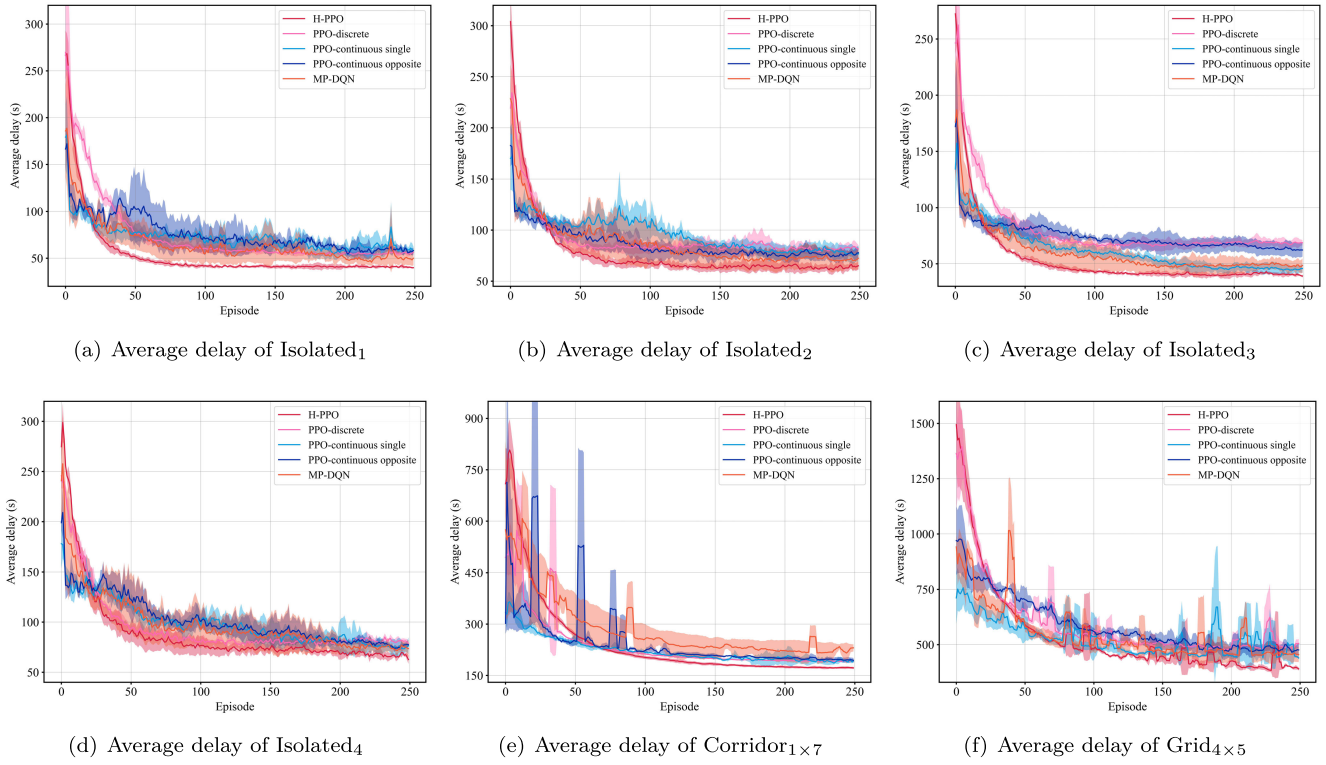
See Figures 15 and 16.



(a) Average queue of Isolated$_1$ (b) Average queue of Isolated$_2$ (c) Average queue of Isolated$_3$

(d) Average queue of Isolated$_4$ (e) Average queue of Corridor$_{1\times7}$ (f) Average queue of Grid$_{4\times5}$

Fig. 15. Training results of algorithms with different action spaces in terms of average queue length.



(a) Average delay of Isolated$_1$ (b) Average delay of Isolated$_2$ (c) Average delay of Isolated$_3$

(d) Average delay of Isolated$_4$ (e) Average delay of Corridor$_{1\times7}$ (f) Average delay of Grid$_{4\times5}$

Fig. 16. Training results of algorithms with different action spaces in terms of average delay.

ACKNOWLEDGMENT

The authors would like to thank Weiye Cang, who was the master's student from Peking University, for his patient guidance and companionship during the article writing.

REFERENCES

[1] F. V. Webster, "Traffic signal settings," Road Res. Lab., Her Majesty's Stationery Office, London, U.K., Tech. Paper 39, 1958.
[2] A. J. Miller, "A computer control system for traffic networks," in *Proc. 2nd Int. Symp. Theory Traffic Flow*, London, U.K., 1963, pp. 200–220.

[3] D. I. Robertson, "'Transyt' method for area traffic control," *Traffic Eng. control*, vol. 11, p. 8, Oct. 1969.

[4] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and M. C. Royle, "The SCOOT on-line traffic signal optimisation technique," *Traffic Eng. Control*, vol. 23, no. 4, pp. 190–192, 1982.

[5] N. H. Gartner, "OPAC: Strategy for demand-responsive decentralized traffic signal control," *IFAC Proc. Volumes*, vol. 23, no. 2, pp. 241–244, Sep. 1990.

[6] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to traffic-responsive network-wide signal control," *Control Eng. Pract.*, vol. 10, no. 2, pp. 183–195, Feb. 2002.

[7] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Jan. 2015.

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

[9] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. ICML*, in Proceedings of Machine Learning Research, vol. 80, Jul. 2018, pp. 1861–1870.

[10] M. Aslani, S. Seipel, M. S. Mesgari, and M. Wiering, "Traffic signal optimization through discrete and continuous reinforcement learning with robustness analysis in downtown Tehran," *Adv. Eng. Informat.*, vol. 38, pp. 639–655, Oct. 2018.

[11] G. Zheng et al., "Learning phase competition for traffic signal control," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1963–1972.

[12] R. Bellman, "A Markovian decision process," *Indiana Univ. Math. J.*, vol. 6, no. 4, pp. 679–684, 1957.

[13] J. Xiong et al., "Parametrized deep Q-networks learning: Reinforcement learning with discrete-continuous hybrid action space," 2018, *arXiv:1810.06394*.

[14] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Int. Conf. Mach. Learn.*, 2014, pp. I-387–I-395.

[15] W. Masson, P. Ranchod, and G. Konidaris, "Reinforcement learning with parameterized actions," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1934–1940.

[16] C. J. Bester, S. D. James, and G. D. Konidaris, "Multi-pass Q-networks for deep reinforcement learning with parameterised action spaces," 2019, *arXiv:1905.04388*.

[17] Z. Fan, R. Su, W. Zhang, and Y. Yu, "Hybrid actor-critic reinforcement learning in parameterized action space," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2279–2285.

[18] P. Diakaki and D. Kotsialos, "Review of road traffic control strategies," *Proc. IEEE*, vol. 91, no. 12, pp. 2041–2042, Dec. 2003.

[19] R. B. Allsop, "SIGSET: A computer program for calculating traffic capacity of signal-controlled road junctions," *Traffic Eng. Control*, vol. 12, pp. 58–60, Jan. 1971.

[20] G. Improta and G. E. Cantarella, "Control system design for an individual signalized junction," *Transp. Res. B, Methodol.*, vol. 18, no. 2, pp. 147–167, Apr. 1984.

[21] P. Mirchandani and F.-Y. Wang, "RHODES to intelligent transportation systems," *IEEE Intell. Syst.*, vol. 20, no. 1, pp. 10–15, Jan. 2005.

[22] D. C. Gazis and R. B. Potts, "The oversaturated intersection," in *Proc. 2nd Int. Symp. Theory Road Traffic Flow*. Paris: Organization for Economic Cooperation and Development, 1965, pp. 221–237.

[23] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. thesis, Dept. Comput. Sci., Kings College Univ. Cambridge, Cambridge, U.K., 1989.

[24] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *J. Transp. Eng.*, vol. 129, no. 3, pp. 278–285, May 2003.

[25] L. A. Prashanth and S. Bhatnagar, "Reinforcement learning with average cost for adaptive control of traffic lights at intersections," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 1640–1645.

[26] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 3, pp. 247–254, Jul. 2016.

[27] 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.

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

[29] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1243–1253, Feb. 2019.

[30] D. Horgan et al., "Distributed prioritized experience replay," 2018, *arXiv:1803.00933*.

[31] N. Casas, "Deep deterministic policy gradient for urban traffic light control," 2017, *arXiv:1703.09035*.

[32] S. S. Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function-based reinforcement learning," *IET Intell. Transp. Syst.*, vol. 11, no. 7, pp. 417–423, Sep. 2017.

[33] Y. Xiong, G. Zheng, K. Xu, and Z. Li, "Learning traffic signal control from demonstrations," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2289–2292.

[34] Z. Mo, W. Li, Y. Fu, K. Ruan, and X. Di, "CVLight: Decentralized learning for adaptive traffic signal control with connected vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 141, Aug. 2022, Art. no. 103728.

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

[36] H. Wei et al., "CoLight: Learning network-level cooperation for traffic signal control," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1913–1922.

[37] L. Xu, J. Xu, X. Qu, and S. Jin, "An origin-destination demands-based multipath-band approach to time-varying arterial coordination," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17784–17800, Oct. 2022.

[38] D. Ma, J. Xiao, X. Song, X. Ma, and S. Jin, "A back-pressure-based model with fixed phase sequences for traffic signal optimization under oversaturated networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5577–5588, Sep. 2021.

[39] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2020.

[40] W. Yao et al., "Understanding travel behavior adjustment under COVID-19," *Commun. Transp. Res.*, vol. 2, Dec. 2022, Art. no. 100068.

[41] L. Xu, S. Jin, B. Li, and J. Wu, "Traffic signal coordination control for arteriais with dedicated CAV lanes," *J. Intell. Connected Vehicles*, vol. 5, no. 2, pp. 72–87, 2022.

[42] E. Van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in *Proc. Learn., Inference Control Multi-Agent Syst.*, vol. 1, 2016, pp. 21–38.

[43] D. Silver, S. Singh, D. Precup, and R. S. Sutton, "Reward is enough," *Artif. Intell.*, vol. 299, Oct. 2021, Art. no. 103535.

[44] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140–1150, Sep. 2013.

[45] R. Zhang, A. Ishikawa, W. Wang, B. Striner, and O. K. Tonguz, "Using reinforcement learning with partial vehicle detection for intelligent traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 404–415, Jan. 2021.

[46] J. Wu and X. Qu, "Intersection control with connected and automated vehicles: A review," *J. Intell. Connected Vehicles*, vol. 5, no. 3, pp. 260–269, 2022.

[47] M. Aslani, M. S. Mesgari, and M. Wiering, "Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 732–752, Dec. 2017.

[48] H. Wei, G. Zheng, H. Yao, and Z. Li, "IntelliLight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2496–2505.

[49] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic Road Transport Support Systems*. Cham, Switzerland: Birkhauser, 2016, pp. 47–66.

[50] H. Wei et al., "PressLight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1290–1298.

[51] A. Oroojlooy, M. Nazari, D. Hajinezhad, and J. Silva, "AttendLight: Universal attention-based reinforcement learning model for traffic signal control," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 4079–4090.

[52] W. Zhao, Y. Ye, J. Ding, T. Wang, T. Wei, and M. Chen, "IPDALight: Intensity- and phase duration-aware traffic signal control based on reinforcement learning," *J. Syst. Archit.*, vol. 123, Feb. 2022, Art. no. 102374.

[53] S. Bouktif, A. Cheniki, and A. Ouni, "Traffic signal control using hybrid action space deep reinforcement learning," *Sensors*, vol. 21, no. 7, p. 2302, Mar. 2021.
[54] F. A. Oliehoek and C. Amato, *Concise Introduction to Decentralized POMDPs*, 1st ed. Cham, Switzerland: Springer, 2016.
[55] C. Amato, G. Konidaris, L. P. Kaelbling, and J. P. How, "Modeling and planning with macro-actions in decentralized POMDPs," *J. Artif. Intell. Res.*, vol. 64, pp. 817–859, Mar. 2019.
[56] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2015, *arXiv:1506.02438*.
[57] T. Kloek and H. K. van Dijk, "Bayesian estimates of equation system parameters: An application of integration by Monte Carlo," *Econometrica*, vol. 46, no. 1, p. 1, Jan. 1978.
[58] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[60] Y. Xiao, W. Tan, and C. Amato, "Asynchronous actor-critic for multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds. Red Hook, NY, USA: Curran Associates, 2022, pp. 4385–4400.
[61] Y. Yuan and A. R. Mahmood, "Asynchronous reinforcement learning for real-time control of physical robots," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 5546–5552.
[62] C. Yu et al., "Asynchronous multi-agent reinforcement learning for efficient real-time multi-robot cooperative exploration," in *Proc. Int. Conf. Auto. Agents Multiagent Syst.* Richland, SC, USA: Int. Found. Auton. Agents Multiagent Syst., 2023, pp. 1107–1115.
[63] C. S. de Witt et al., "Is independent learning all you need in the StarCraft multi-agent challenge?" 2020, *arXiv:2011.09533*.
[64] J. Hu, S. Hu, and S.-W. Liao, "Policy regularization via noisy advantage values for cooperative multi-agent actor-critic methods," 2021, *arXiv:2106.14334*.
[65] C. Yu et al., "The surprising effectiveness of PPO in cooperative multi-agent games," in *Advances in Neural Information Processing Systems*, vol. 35, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds. Red Hook, NY, USA: Curran Associates, 2022, pp. 24611–24624.
[66] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.
[67] M. Treiber and A. Kesting, "Traffic flow dynamics: Data, models and simulation," *Phys. Today*, vol. 67, no. 3, p. 54, 2014.
[68] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transp. Res. C, Emerg. Technol.*, vol. 36, pp. 177–195, Nov. 2013.

Haoqing Luo received the B.E. degree in transportation engineering from Zhejiang University, Hangzhou, China, in 2021, where he is currently pursuing the M.E. degree. His research interests include traffic signal control, reinforcement learning, and multi-agent systems.

Yiming Bie received the Ph.D. degree in traffic engineering from the School of Transportation, Jilin University, Changchun, China, in 2012. He is currently a Professor with the School of Transportation, Jilin University. His research interests include traffic signal control, public transport, and intelligent transport systems. In these areas, he has published over 50 journal articles.

Sheng Jin received the B.E., M.E., and Ph.D. degrees in transportation engineering from Jilin University, Changchun, China. He is currently a Professor with the Institute of Intelligent Transportation Systems, College of Civil Engineering and Architecture, Zhejiang University, Hangzhou, China. His research interests include connected and automated vehicles, driving behavior modeling, traffic signal control, AI for transportation, and intelligent transportation systems.