Multi-Timescale Hierarchical Reinforcement Learning for Unified Behavior and Control of Autonomous Driving

Guizhe Jin, Zhuoren Li, Bo Leng, Ran Yu, Lu Xiong and Chen Sun

***Abstract*—Reinforcement Learning (RL) is increasingly used in autonomous driving (AD) and shows clear advantages. However, most RL-based AD methods overlook policy structure design. An RL policy that only outputs short-timescale vehicle control commands results in fluctuating driving behavior due to fluctuations in network outputs, while one that only outputs long-timescale driving goals cannot achieve unified optimality of driving behavior and control. Therefore, we propose a multi-timescale hierarchical reinforcement learning approach. Our approach adopts a hierarchical policy structure, where high- and low-level RL policies are unified-trained to produce long-timescale motion guidance and short-timescale control commands, respectively. Therein, motion guidance is explicitly represented by hybrid actions to capture multimodal driving behaviors on structured road and support incremental low-level extend-state updates. Additionally, a hierarchical safety mechanism is designed to ensure multi-timescale safety. Evaluation in simulator-based and HighD dataset-based highway multi-lane scenarios demonstrates that our approach significantly improves AD performance, effectively increasing driving efficiency, action consistency and safety.**

***Index Terms*—Reinforcement learning, motion and path planning, autonomous driving, multiple timescale.**

I. Introduction

REINFORCEMENT learning (RL) has demonstrated strong capabilities in solving sequential decision-making problems, making it a promising paradigm for autonomous driving (AD) applications [1], [2]. However, current RL-based AD approaches often suffer from inappropriate policy output structures, resulting in weak correlations between agent outputs and actual driving behavior. Typically, the RL agent directly outputs vehicle control commands, such as steering angle and acceleration [3], [4]. Fluctuations in policy network's outputs that arise from its sensitivity to small state variations can cause inconsistent control sequences [5]. This makes it difficult to achieve stable and coherent driving, especially in lane-structured scenarios, thereby increasing risks [6], [7].

Hierarchical policy output structures are better suited to AD tasks than directly outputting control commands, as they more closely resemble human driving [8]. Behavioral science indicates that human driving behavior is inherently hierarchical in nature, involving both conscious trajectory planning and subconscious action control [9]. Based on this, a common RL approach is to use a hierarchical structure where the high-level policy outputs discrete semantic decisions or trajectory goals, while the low-level rule-based policy generates control commands [10]. However, this design limits the flexibility of RL in vehicle control and makes it difficult to produce optimal control commands that adapt to high-level outputs [11].

In contrast, a hierarchical structure in which both high- and low-level actions are generated by RL policies better leverages RL's flexibility, enabling unified learning of driving behaviors and control commands for complex tasks [12]. Some studies implement this approach using either a single RL agent with a parameterized actor-critic architecture or two independently trained agents [13], [14]. However, these designs often impose timescale consistency constraints on both levels, leading to either fluctuating high-level behaviors or slow low-level control responses [12]. In practice, high-level policy require long-timescale behavioral goals, while low-level policy need short-timescale immediate control [15]. Moreover, given the safety-critical nature of driving, capturing the safer hierarchical policy outputs is also essential.

Therefore, this paper proposes a Multi-Timescale Hierarchical RL approach for autonomous driving. Specifically, we design a hierarchical RL policy structure with high- and low-level components operating at two different timescales: the high-level policy generates long-timescale guidance-actions (i.e., motion guidance), while the low-level policy produces short-timescale execution-actions (i.e., control commands). Both policies are jointly trained to achieve unified optimal performance. Furthermore, we construct a hybrid continuous-discrete action-based motion guidance, enabling multimodal driving behaviors consistent with structured road constraints—discrete laterally and continuous longitudinally. To further enhance safety, we develop a hierarchical safety mechanism that operates in parallel with the policy structure. The main contributions are as follows:

1) A multi-timescale hierarchical RL framework is proposed, in which unified-trained policies generate high-level motion guidance and low-level control commands at different timescales, improving driving efficiency while reducing behavior fluctuations.
2) A novel motion guidance, explicitly represented by hybrid action, is designed to capture multimodal driving behaviors on structured road. It provides more comprehensive guidance into low-level extend-state through incremental update, enhancing driving performance.
3) A supporting hierarchical safety mechanism is proposed,

Manuscript received: 26 June 2025; Revised 10 September 2025; Accepted 9 October 2025. This paper was recommended for publication by Editor Aniket Bera upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the National Science Fund for Distinguished Young Scholars of China under Grant No. 52325212, in part by the National Natural Science Foundation of China under Grant No. 52372394, and in part by the Fundamental Research Funds for the Central Universities (22120230311).

Guizhe Jin, Zhuoren Li, Bo Leng, Ran Yu and Lu Xiong are with the School of Automotive Studies, Tongji University, Shanghai 201804, China. (Email: jgz13573016892@163.com, 1911055@tongji.edu.cn, lengbo@tongji.edu.cn, 2433113@tongji.edu.cn, xiong_lu@tongji.edu.cn).

Chen Sun is with the Department of Data and Systems Engineering, University of Hong Kong, Hong Kong. (Email: c87sun@hku.hk).

Digital Object Identifier (DOI): see top of this page.

comprising safety evaluation and correction modules, along with a safety-aware termination function. It evaluates motion guidance risks and generates safer high- and low-level actions, enhancing driving safety across different timescales.

II. Related Works

A general RL-based AD approach with a hierarchical architecture combines RL with a rule-based method for vehicle control. Specifically, RL is used for the high-level policy, producing outputs that can either be semantic decisions (e.g., lane changes) [7], [10], [16], [17], motion primitives from a discrete space [8], [18], [19], or target points in a continuous space [6], [20]–[22]. Based on these behavior goals, the low-level rule-based controller generates actual vehicle control commands (e.g., steering angle, acceleration). However, this structure limits the flexibility of the RL policy due to indirect vehicle control. Additionally, the low-level controller may fail to respond effectively to dynamic environmental changes, or its response may deviate from the intended high-level behavior, preventing unified optimization across both levels [11].

In contrast, using RL policies to simultaneously generate both abstract driving behaviors and concrete control commands is more advanced. Some classical studies construct implicit hierarchical policy structures [5], [13], [23]–[26], or train independent RL agents for high- and low-level policies [11], [12], [14], [27], to enhance unified optimization between the two levels. However, these methods face timescale consistency constraints, making it difficult to set appropriate timescales for both levels. Specifically, a too-short timescale leads to driving behavior fluctuations, while a too-long timescale slows responses to dynamic environment changes.

Further, some studies attempt to use RL with different timescales to construct hierarchical policy structures, commonly adopting a skill-based approach [9], [28], [29]. For such an approach, low-level RL agents are pre-trained to output sequences of control commands over short timescales, known as motion skills. A high-level RL agent is then trained to select the optimal motion skill from this skill space. This approach breaks the timescale consistency constraint between different levels, thereby better leveraging the flexibility of RL. However, the skill space is typically fixed, and the high-level RL policy essentially learns to combine these time-extended control commands. This limits the potential of RL to explore optimal actions at each driving step [30].

To address the limitations of previous works, we propose a multi-timescale hierarchical RL approach. Two joint-trained hierarchical RL policies output long-timescale abstract motion guidance and short-timescale concrete control commands, respectively. Few studies have explored similar approaches in AD [15], [30], and those that do have notable shortcomings: (1) high-level outputs are restricted to either purely discrete or continuous action spaces, failing to match structured road constraints; and (2) hierarchical policies lack safety considerations. In contrast, we integrate the parameterized actor-critic (P-AC) technique into the hierarchical structure, explicitly representing high-level motion guidance with discrete-continuous hybrid actions. Additionally, a hierarchical safety mechanism is designed to support the policy structure. The framework of our approach is illustrated in Fig. 1.

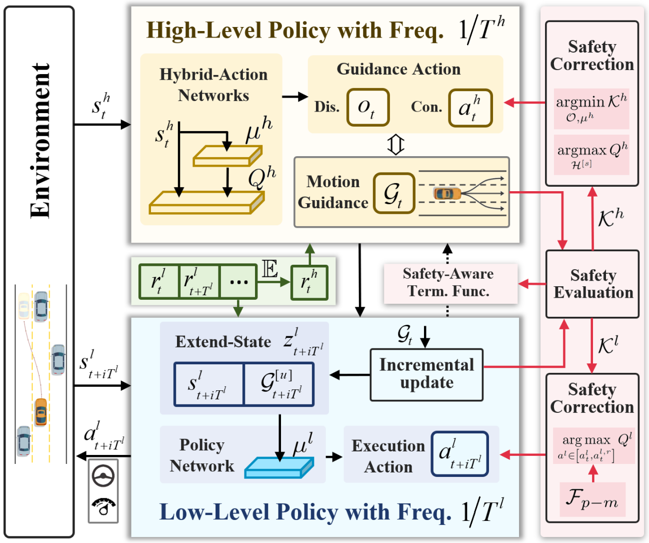


Fig. 1. The framework of multi-timescale hierarchical RL approach. It consists of two unified-trained policies at different levels, with supporting safety mechanisms. High-level motion guidance that, combined with environment features, forms the extended input of the low-level policy, while low-level rewards are fed back in expectation to the high-level for joint optimization.

III. Methodology

A. Framework Construction

1) MDP Re-Formulation: Inspired by DAC [31], we reformulate the training as two augmented Markov Decision Processes (MDPs): high-MDP $\mathcal{M}^h$ and low-MDP $\mathcal{M}^l$. The high-level policy π^h and low-level policy π^l make decisions within their respective MDPs and are optimized jointly.

The $\mathcal{M}^h$ can be defined by a tuple $< \mathcal{S}^h, \mathcal{H}^h, \mathcal{R}^h, \mathcal{T}^h, \gamma >$, where: 1) $\mathcal{S}^h$ is the high-level state space, derived from the environment; 2) $\mathcal{H}^h$ is the hybrid action space, composed of discrete and continuous subspaces, $\mathcal{O}$ and $\mathcal{A}^h$; 3) $\mathcal{R}^h$ is the high-level reward function, determined by low-level rewards and agent violations; and 4) $\mathcal{T}^h$ and γ are the high-level transition function and discount factor.

Similarly, the $\mathcal{M}^l$ is defined by $< \mathcal{Z}^l, \mathcal{A}^l, \mathcal{R}^l, \mathcal{T}^l, \gamma >$, where: 1) $\mathcal{Z}^l$ is the low-level state space, combining the original low-level state space $\mathcal{S}^l$ and all motion guidance mapping to $\mathcal{H}^l$; 2) $\mathcal{A}^l$ is the low-level continuous action space; 3) $\mathcal{R}^l$ is the low-level reward function derived from environmental feedback; and 4) $\mathcal{T}^l$ is the transition function.

At each long timestep T^h, $\pi^h(o, a^h \mid s^h)$ outputs guidance-action $(o, a^h) \in \mathcal{H}^h$, with $s^h \in \mathcal{S}^h$. Each guidance-action explicitly represents a motion guidance $\mathcal{G}$ via a bi-directional mapping: $\mathcal{G} \leftrightarrow (o, a^h)$. Hereafter, $\mathcal{G}(o, a^h)$ denotes the motion guidance corresponding to (o, a^h). Then, at each short timestep T^l, $\pi^l(a^l \mid z^l)$ receives extend-state $z^l = (s^l, \mathcal{G}(o, a^h))$ from the environment and the high-level policy π^h, where $z^l \in \mathcal{Z}^l$ and $s^l \in \mathcal{S}^l$, to output a execution-action $a^l \in \mathcal{A}^l$. The timesteps are related by $T^h = nT^l$, where n is determined by the termination function β, i.e., $\arg_i \left[\beta(z_{t+iT^l}^l) = 1\right]$. Here, $z_{t+iT^l}^l$ is required for the i-th output of π^l after receiving $\mathcal{G}$. Thus, each high-level motion guidance corresponds to a variable-length sequence of low-level control commands.

2) Safety Mechanism Definition: The safety mechanism for the hierarchical policy includes: 1) safety evaluation module,

2) high- and low-level independent safety correction module, and 3) the safety-aware termination function. The safety evaluation module crosses both levels and generates risk severity of motion guidance $\mathcal{K}(\mathcal{G}(o, a^h), s)$ at different timescales. The safety correction module fuses $\mathcal{K}$ with action values to produce safer actions $o^{[s]}$, $a^{h,[s]}$, and $a^{l,[s]}$. The safety-aware termination function β combines the safety evaluation results from both levels, prioritizing high-level correction to further enhance safety.

B. Multi-Timescale Hierarchical Policy Design

The policies π^h and π^l operate in two parallel augmented MDPs, which is trained simultaneously under same sampling conditions [31]. Optimizing π^h requires the parameterized actor-critic algorithm [32], while any policy optimization algorithm can be applied to π^l. To achieve joint optimality of π^h and π^l, a strong coupling between the two policies is essential: 1) The guidance-action from π^h is incorporated into the extend-state as input to π^l, and 2) the rewards obtained by π^l within the timestep T^l are also used to update π^h.

1) High-Level Policy: In high-MDP, the state-action value function for the optimal high-level policy is defined by the following Bellman optimality equation:

$$
\begin{aligned}
&Q^h\left(s_t^h, o_t, a_t^h\right) = \\
&\quad \mathbb{E}\left[r_{t+T^h}^h + \gamma \max_{o\in\mathcal{O}} \sup_{a^h\in\mathcal{A}^h} Q^h\left(s_{t+T^h}^h, o, a^h\right)\right],
\end{aligned}
\tag{1}
$$

where $r_{t+T^h}^h \in \mathcal{R}^h$ is given by:

$$
r_{t+T^h}^h = \left(1 - f_v\left(s^l\right)\right)\mathbb{E}_{i=1\sim n}\left[r_{t+iT^l}^l\right] + f_v\left(s^l\right)\mathcal{R}_{vio}, \tag{2}
$$

where $r_{t+iT^l}^l \in \mathcal{R}^l$ is the feedback reward from the environment for the i-th action of π^l. The violation flag function f_v is set to 1 in case of agent violations (e.g., vehicle collisions) and 0 otherwise. Introducing f_v prevents the dilution of violation-related rewards $\mathcal{R}_{vio}$ by expectation-seeking operations, ensuring that the high-level policy maintains a strong emphasis on violations. The definition of $\mathcal{R}_{vio}$ is provided in Sec. IV-A.

However, finding the optimal a^h in a hybrid action space is challenging. Following the idea of parameterized actor-critic, the high-level policy π^h outputs a^h through the cooperation between a deterministic policy network $\mu^h(s^h;\theta^h)$ and a value network $Q^h(s^h, o, a^h;\omega^h)$. Details of this cooperation can be found in our previous work [13]. Thus, with $\pi^h = (\cdot\,|\mu^h(\cdot;\theta^h), Q^h(\cdot;\omega^h), s^h)$, the optimal state-action value function can be rewritten as:

$$
\begin{aligned}
&Q^h\left(s_t^h, o_t, a_t^h\right) = \\
&\mathbb{E}\left[r_{t+T^h}^h + \gamma \max_{o\in\mathcal{O}} Q^h\left(s_{t+T^h}^h, o, \mu^h\left(s_{t+T^h}^h;\theta^h\right);\omega^h\right)\right].
\end{aligned}
\tag{3}
$$

This function's solution is the optimal guidance-action (o, a^h).

The (o, a^h) explicitly represents the motion guidance $\mathcal{G}$ through a bi-directional mapping:

$$
\mathcal{G} \leftarrow \Psi\left(o, a^h\right),\ \left(o, a^h\right) \leftarrow \Psi^{-1}\left(\mathcal{G}\right). \tag{4}
$$

where Ψ is an explicit representation function, depending on the practical significance of (o, a^h) and $\mathcal{G}$. A generalized example for AD is provided in Sec. IV-A.

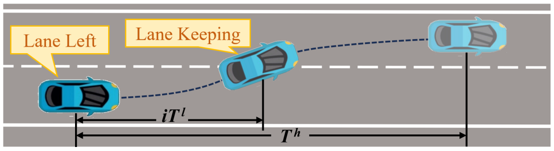


Fig. 2. Illustration of the low-level extend-state transition. Assume that π^h generates a guidance-action of 'Lane Left' for T^h. When the vehicle crosses the lane divider at iT^l, the guidance-action observed from the agent's viewpoint becomes 'Lane Keeping', even though the T^h has not yet ended.

2) Low-Level Policy: In low-MDP, $\mathcal{G}(o, a^h)$ is provided every T^h, while the low-level policy π^l acquires the observed extend-state $z_{t+iT^l}^l$ at much shorter timestep T^l. As a result, the high-level output observed from the low-level perspective may change, as illustrated in Figure 2 with a lane-change scenario. Directly using the fixed $\mathcal{G}(o, a^h)$ in constructing $z_{t+iT^l}^l$ would introduce state inconsistencies, hindering stable training of π^l. To address this, motion guidance is incrementally updated at each short timestep using physical information, allowing the guidance-action to be naturally updated:

$$
\left(o_{t+iT^l}^{[u]}, a_{t+iT^l}^{h,[u]}\right) \overset{\Phi}{\leftrightarrow} \mathcal{G}_{t+iT^l}^{[u]} \leftarrow \mathcal{G}_t\left(o_t, a_t^h\right), \forall i \in [1,\cdots,n], \tag{5}
$$

where the superscript u denotes the variable has undergone incremental updating. Accordingly, the actual low-level extend-state becomes $z_{t+iT^l}^l = (s_{t+iT^l}^l, \mathcal{G}_{t+iT^l}^{[u]})$. Since motion guidance is environment-specific, incremental updating using physical information is straightforward to implement. An example is provided in Sec. IV-A.

Therefore, in low-MDP, the state-action value function for the optimal low-level policy is given by the following Bellman optimality equation:

$$
Q^l\left(z_t^l, a_t^l\right) = \mathbb{E}\left[r_{t+T^l}^l + \gamma U\left(z_{t+T^l}^l\right)\right], \tag{6}
$$

$$
\begin{aligned}
U\left(z_{t+T^l}^l\right) = &\left(1 - \beta\left(s_{t+T^l}^l\right)\right) \sup_{a^l\in\mathcal{A}^l} Q^l\left(z_{t+T^l}^l, a^l\right) + \\
&\beta\left(s_{t+T^l}^l\right) \sup_{a^l\in\mathcal{A}^l} Q^l\left(\pi^h\left(s^h\right), s_{t+T^l}^l, a^l\right),
\end{aligned}
\tag{7}
$$

where $U(z_{t+T^l}^l)$ is the optimal state value function. Since π^l outputs continuous actions and is compatible with any optimization algorithm, it can be directly approximated using a policy network $\mu^l(z^l;\theta^l)$. Meanwhile, a value network $Q^l(z_t^l, a_t^l;\omega^l)$ is introduced to estimate the state-action value.

C. Hierarchical Safety Mechanism

The original motion guidance from π^h may pose safety risks. To quantify these risks, a safety evaluation module is used to assess risk severity: $\mathcal{K}_t^h(\mathcal{G}_t(o_t, a_t^h), s_t^h)$. Since the parameterized actor-critic provides values for all alternative actions, when risk exceeds a threshold, i.e., $\eta\mathcal{K}_t^h \geq \mathcal{K}_{th}$, the safety correction module generates safer guidance-actions guided by $\mathcal{K}^h$ and Q^h:

$$
\left(o_t^{[s]}, a_t^{h,[s]}\right) = \begin{cases} \underset{\mathcal{H}^{[s]}}{\operatorname{argmax}}\, Q^h\left(s_t^h, o, a^h;\omega^h\right) & \mathcal{H}^{[s]} \neq \emptyset \\ \underset{\mathcal{O},\mu^h}{\operatorname{argmin}}\, \mathcal{K}^h\left(\mathcal{G}\left(o, a^h\right), s_t^h\right) & \mathcal{H}^{[s]} = \emptyset \end{cases} \tag{8}
$$

$$
\mathcal{H}^{[s]} = \underset{o,a^h}{\arg}\left[\eta\mathcal{K}^h\left(\mathcal{G}\left(o, a^h\right), s_t^h\right) < \mathcal{K}_{th}\right]\Big|_{\forall o\in\mathcal{O}, a^h=\mu^h} \tag{9}
$$

where the superscript $[s]$ indicates processing by the safety mechanism. The variable η is an attention weight for $\mathcal{K}_t^h$, gradually increased during training to avoid early convergence to a conservative policy. The $\mathcal{H}^{[s]}$ represents a safe guidance-action space comprising alternative actions with risk severity below $\mathcal{K}_{th}$. Then, safer motion guidance $\mathcal{G}_t^{[s]}$ is reconstructed based on $(o_t^{[s]}, a_t^{h,[s]})$, and they are updated to $\mathcal{G}_t^{[s,u]}$ and $(o_t^{[s,u]}, a_t^{h,[s,u]})$ according to Eq. 5. Additionally, $(o_t^{[s]}, a_t^{h,[s]})$ forms a tuple with s_t^h, r_t^h, and $s_{t+T^h}^h$, which is stored in the high-level replay buffer $\mathcal{D}^h$ for updating π^h.

The safety mechanism for π^h cannot always ensure low risk severity, as execution-actions from π^l or dynamic environmental changes may lead to sudden risk increases. To address this, the safety evaluation module operating on a shorter timescale is introduced to assess risk severity: $\mathcal{K}_t^l(\mathcal{G}_t^{[s,u]}(o_t^{[s,u]}, a_t^{h,[s,u]}), s_t^l)$. Since π^l outputs a deterministic action, an alternative action $a_t^{l,r}$ is obtained from a priori conservative control model $\mathcal{F}_{p-m}$, to generate a safer action:

$$a_t^{l,[s]} = \begin{cases} a_t^l & \eta\mathcal{K}_t^l < \mathcal{K}_{th} \\ \underset{a^l \in \left[a_t^l, a_t^{l,r}\right]}{\arg\max}\ Q^l\left(z_t^l, a^l; \omega^l\right) & \eta\mathcal{K}_t^l \geq \mathcal{K}_{th} \end{cases} \tag{10}$$

where $\mathcal{F}_{p-m}$ may vary depending on the specific implementation, with details and an example provided in Sec. IV-B. The $a_t^{l,[s]}$ is used to control the agent's interaction with the environment and, together with z_t^l, r_t^l, and $z_{t+T^l}^l$, is stored in the low-level replay buffer $\mathcal{D}^l$ for updating π^l.

In fact, the high- and low-level safety mechanisms are not independent; they cooperate within a unified framework to safeguard both long-term behaviour and short-term control. When the low-level mechanism detects imminent risk of motion guidance, the high-level mechanism should be prioritized over a longer timescale to ensure longer-term safety. Accordingly, the safety evaluations results of π^h and π^l are integrated, resulting in a safety-aware termination function β:

$$\beta\left(z_{t+iT^l}^l\right) = f_v\left(s_{t+iT^l}^l\right) \vee [i = n_{\max}] \vee \mathcal{C}\left(z_{t+iT^l}^l\right) \tag{11}$$

$$\mathcal{C}\left(z_{t+iT^l}^l\right) = \left[\left(\eta\mathcal{K}_{t+iT^l}^l \geq \mathcal{K}_{th}\right) \wedge \left(\eta\mathcal{K}_t^h < \mathcal{K}_{th}\right)\right] \tag{12}$$

Three conditions trigger $\beta(z_{t+iT^l}^l) = 1$: 1) the agent is in violation, i.e., $f_v = 1$; 2) the cumulative number of low-level decisions reaches the limit $n_{\max}$, which can be fixed or task-dependent; and 3) $\mathcal{C}(z_{t+iT^l}^l) = 1$, indicating that, during the operation of π^l, the risk severity exceeds the threshold while the safety correction of π^h is inactive.

D. Policy Optimization Training

The update of π^h involves two networks: $Q^h(\cdot\,;\omega^h)$ and $\mu^h(\cdot\,;\theta^h)$. Corresponding target networks, $Q_*^h(\cdot\,;\omega^h)$ and $\mu_*^h(\cdot\,;\theta^h)$, are introduced, being updated using a soft update parameter τ. The gradient for updating $Q^h(\cdot\,;\omega^h)$ is computed from randomly sampled transitions $< s_t^h, o_t^{[s]}, a_t^{h,[s]}, r_{t+T^h}^h, s_{t+T^h}^h >$ from $\mathcal{D}^h$, and is given by:

$$\begin{aligned} \nabla\mathcal{L}_t\left(\omega^h\right) &= -\left[y\left(Q^h\right) - Q^h\left(s_t^h, o_t^{[s]}, a_t^{h,[s]}; \omega^h\right)\right]\nabla_{\omega^h} Q^h \\ y\left(Q^h\right) &= r_{t+T^h}^h + \gamma\max_{o\in\mathcal{O}} Q_*^h\left(s_{t+T^h}^h, o, \mu_*^h\left(s_{t+T^h}^h; \theta_*^h\right); \omega_*^h\right). \end{aligned} \tag{13}$$

For $\mu^h(\cdot;\theta^h)$, the update objective is to maximize the value function over all discrete actions, so its policy gradient is:

$$\nabla\mathcal{J}_t\left(\theta^h\right) = \sum_{o\in\mathcal{O}} \nabla_{\theta^h}\mu^h\left(s_t^h; \theta^h\right)\nabla_{a^h} Q^h\left(s_t^h, o, a^h; \omega^h\right)\Big|_{a^h=\mu^h} \tag{14}$$

Algorithm 1 Training process of our method

```
Require: total training steps $T$, $T^h$, $T^l$, $\{\tau\}$, $\mathcal{K}_{th}$, $n_{\max}$.
Initialize: replay buffer $\{\mathcal{D}^h, \mathcal{D}^l\}$, high- and low-level
    networks $\{Q^h, \mu^h, Q_*^h, \mu_*^h, Q^l, \mu^l, Q_*^l, \mu_*^l\}$.
for $t = 0$ to $T$ do
    Get $s_t^h$ from environment.
    Select $a_t^h \sim \mu^h\left(s_t^h; \theta^h\right)$ for $\forall o \in \mathcal{O}$.
    Select $\left(o_t, a_t^h\right) \sim Q^h\left(s_t^h, o, a_t^h; \omega^h\right)|_{\forall o\in\mathcal{O}}$.
    Construct motion guidance $\mathcal{G}_t$ according to $\Psi\left(o_t, a_t^h\right)$.
    Get safer $\left(o_t^{[s]}, a_t^{h,[s]}\right)$ and $\mathcal{G}_t^{[s]}$ according to Eq. 8.
    while not $\beta$ do
        Select $a_t^l \sim \mu^l\left(z_t^l; \theta^l\right)$.
        Get safer $a_t^{l,[s]}$ according to Eq. 10.
        Get $s_{t+T^l}^l$ and $r_{t+T^l}^l$ from environment.
        Incremental update for $\mathcal{G}_{t+T^l}^{[s,u]}\left(o_{t+T^l}^{[s,u]}, a_{t+T^l}^{h,[s,u]}\right)$.
        Get $\beta\left(z_{t+T^l}^l\right)$ according to Eq. 11.
        Update $\omega_{t+T^l}^l$, $\theta_{t+T^l}^l$, $\omega_*^l$, $\theta_*^l$.
        $z_t^l \leftarrow z_{t+T^l}^l$, $i \leftarrow i+1$, $t \leftarrow t+T^l$.
    end while
    $n \leftarrow i$, $T^h = nT^l$, $s_{t+T^h}^h \leftarrow s_{t+T^l}^l$.
    Get $r_{t+T^h}^h$ according to Eq. 2.
    Update $\omega_{t+T^h}^h$, $\theta_{t+T^h}^h$, $\omega_*^h$, $\theta_*^h$.
    $s_t^h \leftarrow s_{t+T^h}^h$, $i \leftarrow 0$, $\beta \leftarrow 0$.
end for
```

Similarly, updating the low-level $Q^l(\cdot;\omega^l)$ and $\mu^l(\cdot;\theta^l)$ relies on target networks $Q_*^l(\cdot;\omega_*^l)$ and $\mu_*^l(\cdot;\theta_*^l)$. The gradient of $Q^l(\cdot;\omega^l)$ is computed based on $< z_t^l, a_t^{l,[s]}, r_{t+T^l}^l, z_{t+T^l}^l >$, which is randomly sampled from $\mathcal{D}^l$:

$$\begin{aligned} \nabla\mathcal{L}_t\left(\omega^l\right) &= -\left[y\left(Q^l\right) - Q^l\left(z_t^l, a_t^{l,[s]}; \omega^l\right)\right]\nabla_{\omega^l} Q^l, \\ y\left(Q^l\right) &= r_{t+T^l}^l + \gamma U\left(z_{t+T^l}^l\right)\Big|_{\pi_*^h, s^h = s_{t+T^l}^l}, \end{aligned} \tag{15}$$

where π_*^h denotes the high-level target policy. The gradient for updating $\mu^l(\cdot;\theta^l)$ is:

$$\nabla\mathcal{J}_t\left(\theta^l\right) = \nabla_{\theta^l}\mu^l\left(z_t^l; \theta^l\right)\nabla_{a^l} Q^l\left(z_t^l, a^l; \omega^l\right)\Big|_{a^l=\mu^l}. \tag{16}$$

The training procedure of our multi-timescale hierarchical RL, with safety mechanisms, is presented in Algorithm 1.

IV. Implementation

The highway multi-lane scenario is a common yet challenging environment, requiring the ego vehicle (EV) to dynamically adjust its position and speed within defined lanes to ensure efficiency, action consistency, and safety. Therefore, we implement our approach in this setting.

A. Two Augmented MDPs Formulation

Both high-MDP and low-MDP involve three key elements: 1) Action space: The high-MDP adopts a hybrid action space for motion guidance under road constraints, focusing on long-term target position planning. In contrast, low-MDP uses a continuous action space to generate control commands, focusing on short-term speed adjustment. 2) State space: The two MDPs share a same original state space, i.e., $\mathcal{S}^h = \mathcal{S}^l$. 3) Reward function: The high-MDP's reward is implicitly defined by that of the low-MDP and requires no separate design.

1) High-MDP Action Space: In lanes that are discrete laterally but continuous longitudinally, the hybrid action space $\mathcal{H}^h$ of the high-level policy is defined as follows:

$$\left\{\begin{array}{l}\mathcal{O}:\{-w_r,0,w_r\}\\ \mathcal{A}^h:\left[\min\left(\sqrt{4R_0w_r-w_r^2},\frac{v_e^2}{2a_{\max}^-}\right),e^{|v_e|+w_r}\right]\end{array}\right. \tag{17}$$

where w_r is the lane width, R_0 is the minimum turning radius, $2a_{\max}^-$ is the maximum braking acceleration, and v_e is the EV's speed. The $\mathcal{O}$ allows o to represent lane selection, restricting the target to the current or adjacent lane. Given the EV's state and kinematics, the $\mathcal{A}^h$ allows a^h to represent the selection of a feasible target location within the chosen lane. Together, (o,a^h) specifies a target point without further processing, which can be considered as a coarse-grained motion guidance.

To provide finer-grained guidance for the low-level policy, the motion guidance is represented as a series of path points, which are generated from a polynomial curve: $\mathcal{G}=\arg_{(x_j,y_j)}[y_j=\sum_{m=0}^{5}c_mx_j^m]$, where $j\in[1,\cdots,g]$. Here, $\mathcal{G}$ is a set of g target points lying on a fifth-degree polynomial parameterized by coefficients c_m. In the Frenet frame, with the EV at the origin, x_j and y_j are the longitudinal and lateral coordinates of the j-th target point. The start point is set by the EV's current state: $(x_1,y_1)=(x_e,y_e)$, with heading angle $\varphi_1=\varphi_e$. The end point is given by π^h: $(x_g,y_g)=(o,a^h)$, with φ_g from lane information. Solving the resulting linear system yields the coefficients c_m [13]. The mapping $(o,a^h)\equiv(x_g,y_g)\in\mathcal{G}$ defines an explicit representation function Ψ, which satisfies Eq. 4 given the current EV and road states.

In timestep T^h, the frenet frame moves with EV. Since the points in $\mathcal{G}$ are defined relative to previous frenet frame, their coordinates must be updated. A simple coordinate transformation yields the updated set $\mathcal{G}^{[u]}$, corresponding to Eq. 5.

2) Low-MDP Action Space: The control commands required for EV driving are acceleration and steering angle. Thus, the execution action output by π^l consists of two items: $a^l=(\delta_e,a_e)$. According to general vehicle kinematics, the low-level action space is defined as:

$$\mathcal{A}^l=\left\{\left[-\pi/6\text{ rad},\pi/6\text{ rad}\right],\left[-3\text{ m/s}^2,3\text{ m/s}^2\right]\right\}. \tag{18}$$

3) State Space: Both π^h and π^l should account for the states of the EV and surrounding vehicles (SVs) in adjacent lanes. Thus, the state space is defined as:

$$\mathcal{S}^h\equiv\mathcal{S}^l=\left\{\begin{array}{c}[ID_{lane},x_e,y_e,\varphi_e,v_e^x,v_e^y]^{\text{EV}},\\ [p_k,\Delta x_k,\Delta y_k,\varphi_k,\Delta v_k^x,\Delta v_k^y]_{k\in[1\cdots6]}^{\text{SVs}}\end{array}\right\}, \tag{19}$$

The EV state includes lane ID, longitudinal and lateral positions, heading angle, and longitudinal and lateral speeds. SVs ahead of and behind the EV in the current and adjacent lanes are considered, with up to six SVs in total. Each SV state includes a presence flag, relative longitudinal and lateral positions, relative heading angle, and relative longitudinal and lateral speeds. The EV only considers SVs within the observation range $\Delta x\in[-80\,\text{m},\,160\,\text{m}]$.

4) Reward Function: The reward function is designed to consider driving safety, efficiency, and action consistency:

$$\begin{aligned}&\mathcal{R}^l=\mathcal{R}_s+\mathcal{R}_e+\mathcal{R}_c,\mathcal{R}_s=-10f_v-5\left(\mathcal{K}^h+\mathcal{K}^l\right),\\&\mathcal{R}_e=|v_e-v_*|/v_*-\max\left(0,v_p-v_e/v_p\right),\\&\mathcal{R}_c=-(0.5|\delta_e|+0.2|\Delta\delta_e|)-(0.5|a_e|+0.2|\Delta a_e|)\end{aligned} \tag{20}$$

The weights for each term reflect the intended driving behavior: safety first, efficiency, and action consistency as a secondary objective. In safety reward $\mathcal{R}_s$, $f_v=1$ indicates an EV violation, such as road departure or collision with SVs. The results from the safety evaluation module are also included. In efficiency reward $\mathcal{R}_e$, the EV is encouraged to reach the target speed v_*, while speeds below the threshold v_p are penalized. We set v_* and v_p to $18m/s$ and $5m/s$, respectively. In action consistency reward $\mathcal{R}_c$, fluctuations in control commands are penalized to promote smoothness.

B. Safety Mechanism Formulation

1) Risk Severity Evaluation: Artificial Potential Fields (APF) integrate discrete events into a unified field over high-dimensional observations [26], providing reliable and expressive measures of driving risk severity. Based on APF, we propose a risk severity evaluation model for $\mathcal{G}$:

$$\mathcal{K}=\frac{1}{g}\sum_{j=1}^{g}\left[\mathcal{I}_j\cdot\max_{k\in[1,\cdots,6]}\left\{\rho_j^k\right\}\right], \tag{21}$$

where ρ_j^k is the risk potential field at the j-th point relative to the k-th SV. Since points closer to the EV are more critical, the importance at the i-th point is: $\mathcal{I}_j=1-e^{K_r(j-g)}$, where K_r is a decay rate coefficient. Additionally, ρ_j^k is defined as:

$$\begin{aligned}&\rho_j^k=w_1e^{\left(-\frac{1}{2}\mathrm{P}_1\mathrm{B}^{-1}\mathrm{P}_1^{\mathrm{T}}\right)}+w_2e^{\left(-\frac{1}{2}\mathrm{P}_2\mathrm{B}^{-1}\mathrm{P}_2^{\mathrm{T}}\right)}\\&\mathrm{B}=\left[\begin{array}{cc}X_s^2&0\\0&Y_s^2\end{array}\right],\mathrm{P}_1=\left[\begin{array}{cc}\Delta x_j^k&\Delta y_j^k\end{array}\right],\\&\mathrm{P}_2=\left[\begin{array}{cc}{}_{\Delta a_x^k<0}\Delta x_j^k&{}_{\Delta a_y^k<0}\Delta y_j^k\end{array}\right]\end{aligned} \tag{22}$$

where $w_1\in[0.5,1]$ and $w_1+w_2=1$. In addition, X_s and Y_s are the minimum safe distances in the longitudinal and lateral directions, respectively. The Δx_j^k and Δy_j^k are the longitudinal and lateral distances of the j-th point relative to the k-th SV.

2) Prior Control Model Design: To obtain $a^{l,r}$ and simplify the design, we combine the widely used Intelligent Driver Model (IDM) and Stanley path-tracking algorithm as a conservative prior control model. Specifically, IDM determines the acceleration based on environment, while Stanley algorithm computes the steering angle according to motion guidance.

V. Experiments

A. General Settings

1) Environment: The training scenario is built in Highway-Env with three defined lanes [33]. At the start of each episode, the EV and SVs are randomly placed in any lane with random initial speeds. SVs follow the IDM and MOBIL models, allowing lane changes to reach their target speeds, which may interfere with the EV. All vehicles are modeled using the Kinematic Bicycle Model [34]. The vehicle capacity (V/C) ratio, representing traffic density.

With this setup, training is conducted for 2,000 episodes using five seeds. Testing is then performed over 100 episodes on both Highway-Env and the HighD dataset [35], with each episode limited to 100s. Notably, the traffic density in Highway-Env is set to 0.3, presenting a challenging scenario.

TABLE I
COMPARISON OF ALL METHODS

Method	High-Level		Low-Level	Is MT
	Model	Output		
PPO [4]	N/A	N/A	PPO	No
Gen-H-RL [7]	VB	Dis-Beh.	PID	Yes
Class-HRL [14]	VB	Dis-Beh.	AC	No
RL-PTA [13]	P-AC	Path Points	P-AC	No
MHRL-I [15]	VB	Dis-Beh.	AC	Yes
Skill-Critic [30]	AC	Path Points	AC	Yes
MTHRL-H (Ours)	P-AC	Path Points	AC	Yes
MTHRL-HS (Ours)	P-AC	Path Points	AC	Yes

2) Comparison Methods: We select several popular RL-based AD methods as baselines, which have different policy structures. **PPO**, serving as a low-level-only baseline, directly outputs control commands without any hierarchical guidance. The General Hierarchical method (**Gen-H-RL**, used as a high-level-only baseline) employs a high-level RL policy to select discrete behaviors for a rule-based low-level controller. Classical Hierarchical RL (**Class-HRL**) combines a high-level value-based policy with a low-level actor-critic structure. **RL-PTA** has an implicit hierarchical policy structure with a fixed timescale. **MHRL-I** and **Skill-Critic** are preliminary methods exploring multi-timescale RL policy training, with discrete and continuous high-level outputs, respectively. In contrast, our method features a more advanced hierarchical policy structure, in which P-AC is used to generate **H**ybrid action-based motion guidance (path points), with a supporting **S**afety mechanism included. The details of all methods are shown in Table I. For all MT-based methods, the high- and low-level outputs operate at 1s and 0.1s, respectively, as this has proven effective [6].

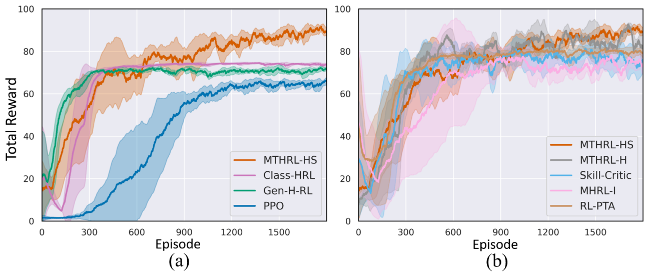
3) Evaluation Metrics: To comprehensively evaluate the performance of each driving policy, we select key metrics across four aspects:

- Overall Performance: Total Reward (**TR**) per episode.
- Efficiency: 1) Driving Speed (**DS**); 2) Total Lane Changes (**TLC**) per episode.
- Action Consistency: 1) Absolute Steering angle (**AS**); 2) Absolute Acceleration (**AA**); 3) Centerline Departure Distance (**CDD**).
- Safety: 1) Collision Rate (**CR**); 2) TTC in EV's Current lane (**TTC-C**); 3) TTC in EV's Target lane (**TTC-T**). TTC-C and TTC-T are identical if no lane change occurs.

B. Performance Comparison

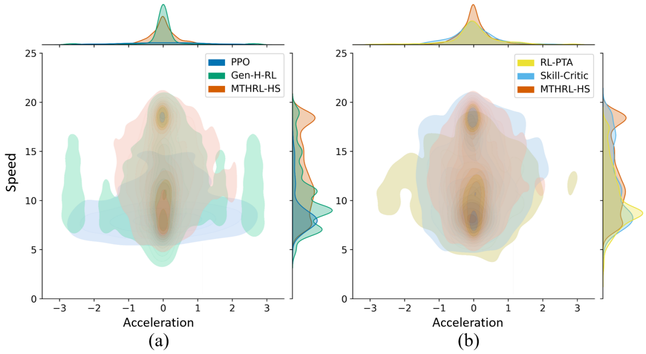
1) Training: The total reward curves for all methods during training are shown in Fig. 3. All methods converge after 1,800 episodes. Our proposed MTHRL-HS achieves the highest reward, indicating superior driving performance.

In Fig. 3(a), PPO converges more slowly and achieve significantly lower rewards than methods using hierarchical techniques, suggesting that direct control output hinders effective policy learning. Gen-H-RL, which applies RL only at the high level, converges fastest but yields relatively lower rewards. In contrast, Class-HRL uses RL at both levels, leading to marginally improved performance yet slightly slower convergence. The methods in Fig. 3(b) generally benefit from more advanced hierarchical structures, resulting in better policies. RL-PTA, while implicitly hierarchical with a fixed timescale, achieves strong performance with less fluctuation across different seeds. MHRL-I/Skill-Critic perform significantly worse than MTHRL-H, demonstrating that hybrid action-based motion guidance, which better aligns with road structure, leads to superior policies compared to purely continuous or discrete actions. Furthermore, MTHRL-HS, with its safety mechanism, further enhances policy performance.



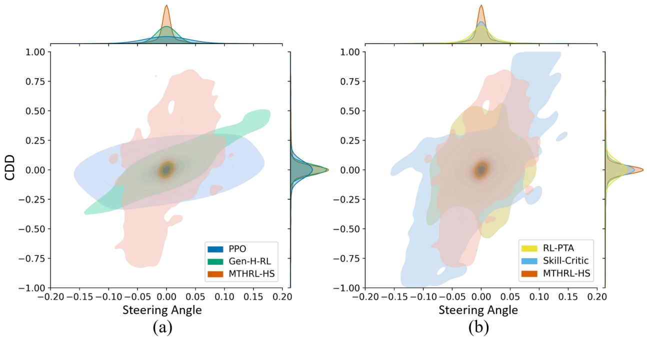
(a) (b)

Fig. 3. The training process of our method with comparison methods.



(a) (b)

Fig. 4. The joint distribution of acceleration and speed in testing.



(a) (b)

Fig. 5. The joint distribution of steering angle and CDD in testing.

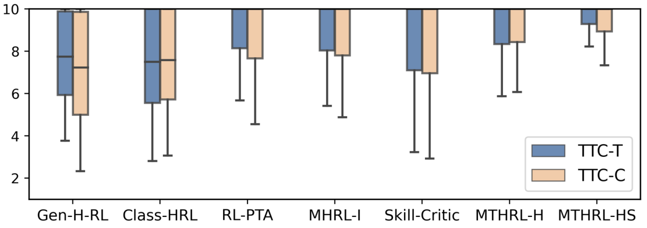


Fig. 6. The distributions of TTC-C and TTC-T in testing.

2) Testing: The testing results further show that different methods produce distinct driving behaviors, as detailed in Table II. Notably, our method achieves the highest TR, consistent with the training results. Compared to RL-PTA, which has the second-highest TR, MTHRL-H improves TR by 4.4%, and with the safety mechanism 'S', this improvement increases to 9.5%. After analyzing all the metrics, we provide additional details for some key metrics of the well-performing methods. The joint distribution of acceleration and speed is shown in Fig. 4, while the joint distribution of steering angle and CDD is shown in Fig. 5. The Fig. 6 further presents the distributions

TABLE II
KEY INDICATORS OF TEST RESULTS IN THE HIGHWAY-ENV SIMULATION ENVIRONMENT

Method	**Overall Perf.**	**Efficiency**		**Action Consistency**			**Safety**		
	TR	**DS** $[m/s]$	**TLC**	**AS** $[rad]$	**AA** $[m/s^2]$	**CDD** $[m]$	**CR**	**TTC-T** $[s]$	**TTC-C** $[s]$
PPO	68.22(2.10)	8.87(2.20)	1.11(0.93)	0.086(0.184)	0.98(0.68)	0.22(0.35)	0.01%	9.14(1.85)	9.19(2.03)
Gen-H-RL	72.65(1.94)	10.43(3.41)	6.22(1.98)	0.022(0.090)	0.42(0.63)	0.09(0.24)	0.33%	7.67(1.93)	7.13(2.38)
Class-HRL	73.20(0.58)	9.99(3.16)	5.68(1.85)	0.044(0.133)	0.59(0.55)	0.12(0.29)	0.29%	7.42(2.18)	7.51(2.10)
RL-PTA	81.37(2.70)	11.21(3.02)	7.89(2.55)	0.024(0.052)	0.54(0.51)	0.13(0.25)	0.10%	9.10(1.19)	8.87(1.50)
MHRL-I	78.91(4.99)	11.03(3.60)	7.62(3.10)	0.039(0.069)	0.49(0.48)	0.12(0.19)	0.09%	9.05(1.26)	8.93(1.41)
Skill-Critic	79.26(4.01)	11.15(3.79)	7.15(2.54)	0.023(0.042)	0.50(0.44)	0.10(0.20)	0.11%	8.89(1.87)	8.73(1.95)
MTHRL-H(ours)	84.99(3.04)	**12.68(2.87)**	**8.19(2.69)**	**0.012(0.040)**	**0.31(0.37)**	**0.07(0.14)**	0.07%	9.14(1.36)	**9.15(2.05)**
MTHRL-HS(ours)	**89.14(2.85)**	12.61(3.81)	8.03(2.60)	0.013(0.034)	0.33(0.36)	0.07(0.16)	**0.03%**	**9.36(1.26)**	9.03(1.89)

TABLE III
KEY INDICATORS OF TEST RESULTS IN THE HIGHD DATASET

Method	**Overall Perf.**	**Efficiency**		**Action Consistency**			**Safety**		
	TR	**DS** $[m/s]$	**TLC**	**AS** $[rad]$	**AA** $[m/s^2]$	**CDD** $[m]$	**CR**	**TTC-T** $[s]$	**TTC-C** $[s]$
PPO	81.62(3.54)	10.22(1.92)	0.68(1.20)	0.081(0.188)	1.08(0.65)	0.18(0.23)	0.02%	9.23(1.88)	9.12(1.68)
Gen-H-RL	87.40(5.33)	13.08(2.40)	5.73(2.14)	0.030(0.073)	0.36(0.51)	0.07(0.18)	0.20%	8.06(1.85)	7.66(2.00)
Class-HRL	89.25(3.11)	12.82(3.15)	5.55(2.16)	0.041(0.121)	0.74(0.58)	0.11(0.20)	0.21%	7.84(1.99)	7.89(2.07)
RL-PTA	95.57(7.04)	15.01(2.98)	6.96(3.84)	0.022(0.050)	0.60(0.52)	0.12(0.23)	0.05%	9.21(1.15)	8.99(1.43)
MHRL-I	94.91(8.95)	14.13(3.63)	7.42(4.23)	0.028(0.062)	0.53(0.50)	0.12(0.18)	0.08%	9.10(1.21)	9.04(1.28)
Skill-Critic	95.30(8.01)	14.80(4.04)	7.08(3.80)	0.021(0.039)	0.59(0.39)	0.10(0.21)	0.05%	8.96(1.77)	8.97(1.81)
MTHRL-H(ours)	96.62(6.30)	**16.18(3.00)**	**7.39(3.52)**	0.013(0.041)	0.37(0.35)	**0.06(0.15)**	0.05%	9.15(1.76)	**9.10(1.88)**
MTHRL-HS(ours)	**96.77(5.51)**	16.09(3.44)	7.07(3.27)	**0.013(0.037)**	**0.32(0.30)**	0.07(0.17)	**0.02%**	**9.30(1.33)**	9.09(1.72)

[*] Boldface indicates the best result among all methods; underline indicates the best result among baseline methods.

of TTC-C and TTC-T as boxplots.

For driving efficiency, MTHRL-H achieves the highest DS and TLC, with DS improved by 13.1% over the suboptimal method. This indicates that the multi-timescale hierarchical policy enables more flexible lane-changing behaviors to enhance driving efficiency, while the introduction of safety mechanism does not significantly compromise this performance. More specifically, under the same testing environment, Fig. 4 shows that the peak speed distributions for each method correspond to two potential cases: one near 20 m/s, representing opportunities for overtaking to reach the target speed; and another near 10 m/s, indicating the EV must follow SVs due to traffic congestion. Notably, MTHRL-HS achieves better lane-change timing in the former case, yielding the highest peak speed distribution. In the latter case, it effectively chooses to follow a faster SV, shifting the peak speed distribution upward. In contrast, other methods show lower driving efficiency, with worse DS, TLC, and speed distributions. In particular, PPO, lacking hierarchical structures, tend to adopt overly conservative following policies.

For driving action consistency, MTHRL-H shows the lowest means and standard deviations for AS, AA, and CDD, which are 50.0%, 26.2%, and 22.2% lower than those of the suboptimal method, respectively. The safety mechanism does not adversely affect these metrics. As shown in Fig. 4 and Fig. 5, our method produces acceleration, steering angle, and CDD values more tightly centered around 0. Even during lane changes, while CDD increases, steering angles remain smaller than with other methods. This suggests that high-level hybrid action-based motion guidance improves action consistency. It reduces fluctuations in control commands, resulting in smoother longitudinal and lateral driving behaviors. In contrast, PPO, which directly outputs control commands, lead to more fluctuating behaviors and make it harder to keep the EV on the centerline. Among hierarchical methods, generating finer-grained path points at the high level yields greater action consistency than discrete behavior generation.

For driving safety, excluding the over-conservative PPO, MTHRL-H reduces the CR by 22.2% compared to the suboptimal method. With the safety mechanism, the reduction in CR increases to 66.7%, clearly demonstrating its effectiveness in improving safety. Additionally, TTC-C and TTC-T reflect the policy's ability in ensuring safer driving during interactions with SVs. As shown in Table II and Fig. 6, MTHRL-HS achieves the highest TTC-C and TTC-T, indicating a more cautious driving style. Meanwhile, the safety mechanism results in a significantly higher TTC-T than TTC-C, highlighting its ability to guide the EV into safer lanes.

3) Validation on HighD Dataset: The metrics in Table III show the driving performance of each method on the HighD dataset. Compared to the Highway-Env scenario, all methods perform better due to lower traffic density in HighD. MTHRL-H and MTHRL-HS remain the top performers overall, despite a smaller TR gap with other methods, demonstrating strong adaptability and robust policy performance in real traffic. Benefiting from the multi-timescale hierarchical architecture and hybrid-action-based motion guidance, MTHRL-HS and MTHRL-H maintain more efficient and stable driving policies. Unlike testing in Highway-Env, MTHRL-HS performs nearly similar to MTHRL-H in action consistency, even slightly outperforming it. For safety, excluding the over-conservative PPO, MTHRL-HS further reduces CR to 0.02% through hierarchical safety mechanism. This indicates that the safety mechanism not only enhances safety but also enables RL policy to maintain more consistent actions when encountering unfamiliar SVs. Therefore, above results confirm the superiority of our method across all driving metrics and its strong potential for real-world deployment. Notably, since real deployment is safety-critical, MTHRL-HS is the preferred choice, as it significantly enhances safety with only minor efficiency loss.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2025.3623016

VI. Conclusion and Future Work

This paper proposes a Multi-Timescale Hierarchical RL approach for AD. The approach features a hierarchical policy structure: a high-level RL policy generates long-timescale motion guidance, while a low-level RL policy produces short-timescale vehicle control commands. Therein, a hybrid action-based explicit representation is designed for motion guidance to better adapt to structured roads and to facilitate addressing low-level state inconsistencies. In addition, supporting hierarchical safety mechanisms are introduced to enhance the safety of both high- and low-level outputs. We evaluate our approach against advanced baselines in both simulator-based and HighD data-based highway multi-lane scenarios, and conduct a comprehensive analysis of various driving behavior metrics. Results demonstrate that our approach effectively improves driving efficiency, action consistency, and safety.

Future work aims to: i) extend our approach to more complex scenarios such as bidirectional roads and intersections, introducing generalization techniques as necessary; ii) design advanced safety mechanisms with comparisons to safety-oriented baselines; and iii) incorporate higher-fidelity vehicle dynamics models toward deployment in real vehicular systems.

References

[1] D. Hu, L. Mo, J. Wu, and C. Huang, ""feariosity"-guided reinforcement learning for safe and efficient autonomous end-to-end navigation," *IEEE Robot. Autom. Lett.*, 2025.

[2] H. Deng, Y. Zhao, Q. Wang, and A.-T. Nguyen, "Deep reinforcement learning based decision-making strategy of autonomous vehicle in highway uncertain driving environments," *Automotive Innovation*, vol. 6, no. 3, pp. 438–452, 2023.

[3] X. Tang, B. Huang, T. Liu, and X. Lin, "Highway decision-making and motion planning for autonomous driving via soft actor-critic," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4706–4717, 2022.

[4] T. Agarwal, H. Arora, and J. Schneider, "Learning urban driving policies using deep reinforcement learning," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, pp. 607–614, 2021.

[5] L. Chen, Y. He, Q. Wang, W. Pan, and Z. Ming, "Joint optimization of sensing, decision-making and motion-controlling for autonomous vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4642–4654, 2022.

[6] L. Wang, J. Liu, H. Shao, W. Wang, R. Chen, Y. Liu, and S. L. Waslander, "Efficient reinforcement learning for autonomous driving with parameterized skills and priors," *arXiv preprint arXiv:2305.04412*, 2023.

[7] Y. Xia, S. Liu, Q. Yu, L. Deng, Y. Zhang, H. Su, and K. Zheng, "Parameterized decision-making with multi-modality perception for autonomous driving," in *IEEE Int. Conf. on Data Eng. (ICDE)*, pp. 4463–4476, IEEE, 2024.

[8] C. Lu, H. Lu, D. Chen, H. Wang, P. Li, and J. Gong, "Human-like decision making for lane change based on the cognitive map and hierarchical reinforcement learning," *Transp. Res. Part C Emerg. Technol.*, vol. 156, p. 104328, 2023.

[9] Y. Gurses, K. Buyukdemirci, and Y. Yildiz, "Developing driving strategies efficiently: A skill-based hierarchical reinforcement learning approach," *IEEE Control Syst. Lett.*, 2024.

[10] Y. Liao, G. Yu, P. Chen, B. Zhou, and H. Li, "Integration of decision-making and motion planning for autonomous driving based on double-layer reinforcement learning framework," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 3142–3158, 2023.

[11] R. Zhao, Z. Sun, and A. Ji, "A deep reinforcement learning approach for automated on-ramp merging," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, pp. 3800–3806, IEEE, 2022.

[12] Z. Zhang, E. Yurtsever, and K. A. Redmill, "Extensive exploration in complex traffic scenarios using hierarchical reinforcement learning," *arXiv preprint arXiv:2501.14992*, 2025.

[13] G. Jin, Z. Li, B. Leng, W. Han, and L. Xiong, "Stability enhanced hierarchical reinforcement learning for autonomous driving with parameterized trajectory action," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, 2024.

[14] J. Peng, S. Zhang, Y. Zhou, and Z. Li, "An integrated model for autonomous speed and lane change decision-making based on deep reinforcement learning," *IEEE Trans. Intell.Transp. Syst.*, vol. 23, no. 11, pp. 21848–21860, 2022.

[15] L. Chen, Y. He, W. Pan, F. R. Yu, and Z. Ming, "A novel generalized meta hierarchical reinforcement learning method for autonomous vehicles," *IEEE Network*, vol. 37, no. 4, pp. 230–236, 2023.

[16] S.-H. Lee, Y. Jung, and S.-W. Seo, "Imagination-augmented hierarchical reinforcement learning for safe and interactive autonomous driving in urban environments," *IEEE Trans on Intell. Transp. Syst.*, 2024.

[17] J. Wang, H. Sun, and C. Zhu, "Vision-based autonomous driving: A hierarchical reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 11213–11226, 2023.

[18] G. Jin, Z. Li, B. Leng, and M. Shao, "Deep reinforcement learning lane-change decision-making for autonomous vehicles based on motion primitives library in hierarchical action space," *Artif. Intell. Auton. Syst.*, vol. 1, no. 2, pp. 1–2, 2024.

[19] H. Lu, C. Lu, Y. Yu, G. Xiong, and J. Gong, "Autonomous overtaking for intelligent vehicles considering social preference based on hierarchical reinforcement learning," *Automotive Innovation*, vol. 5, no. 2, pp. 195–208, 2022.

[20] T. Zhou, L. Wang, R. Chen, W. Wang, and Y. Liu, "Accelerating reinforcement learning for autonomous driving using task-agnostic and ego-centric motion skills," in *Proc. IEEE/RSJ Int. Conf. Intell. Rob. Syst (IROS)*, pp. 11289–11296, 2023.

[21] M. Dalal, D. Pathak, and R. R. Salakhutdinov, "Accelerating robotic reinforcement learning via parameterized action primitives," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, pp. 21847–21859, 2021.

[22] S. Chen, L. Yang, Z. Mao, M. Hou, L. He, and W. Song, "Unified planning framework with drivable area attention extraction for autonomous driving in urban scenarios," *IEEE Robot. Autom. Lett.*, 2025.

[23] Y. Lin, X. Liu, and Z. Zheng, "Discretionary lane-change decision and control via parameterized soft actor–critic for hybrid action space," *Machines*, vol. 12, no. 4, p. 213, 2024.

[24] G. Jin, Z. Li, B. Leng, W. Han, L. Xiong, and C. Sun, "Hybrid action based reinforcement learning for multi-objective compatible autonomous driving," *arXiv preprint arXiv:2501.08096*, 2025.

[25] Z. Li, G. Jin, R. Yu, B. Leng, and L. Xiong, "Interaction-aware deep reinforcement learning approach based on hybrid parameterized action space for autonomous driving," in *Proc. SAE Intell. Connected Veh. Symposium (SAE ICVS)*, 2024.

[26] D. Chen, H. Li, Z. Jin, H. Tu, and M. Zhu, "Risk-anticipatory autonomous driving strategies considering vehicles' weights based on hierarchical deep reinforcement learning," *IEEE Trans on Intell. Transp. Syst.*, 2024.

[27] Q. Guo, O. Angah, Z. Liu, and X. J. Ban, "Hybrid deep reinforcement learning based eco-driving for low-level connected and automated vehicles along signalized corridors," *Transp. Res. Part C Emerg. Technol.*, vol. 124, p. 102980, 2021.

[28] Z. Mao, Y. Liu, and X. Qu, "Integrating big data analytics in autonomous driving: An unsupervised hierarchical reinforcement learning approach," *Transp. Res. Part C Emerg. Technol.*, vol. 162, p. 104606, 2024.

[29] K. Lee, S. Kim, and J. Choi, "Adaptive and explainable deployment of navigation skills via hierarchical deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 1673–1679, IEEE, 2023.

[30] C. Hao, C. Weaver, C. Tang, K. Kawamoto, M. Tomizuka, and W. Zhan, "Skill-critic: Refining learned skills for hierarchical reinforcement learning," *IEEE Robot. Autom. Lett.*, 2024.

[31] S. Zhang and S. Whiteson, "Dac: The double actor-critic architecture for learning options," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019.

[32] J. Xiong, Q. Wang, Z. Yang, P. Sun, L. Han, Y. Zheng, H. Fu, T. Zhang, J. Liu, and H. Liu, "Parametrized deep q-networks learning: Reinforcement learning with discrete-continuous hybrid action space," *arXiv preprint arXiv:1810.06394*, 2018.

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

[34] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?," in *2017 IEEE intell. veh. symp. (IV)*, pp. 812–818, IEEE, 2017.

[35] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, pp. 2118–2125, 2018.