Skill-abstracting continual reinforcement learning for safe, efficient, and comfortable autonomous driving through vehicle-cloud collaboration

Citation for the original published paper (version of record):

Chen, J., Zhao, C., Gao, K. et al (2025). Skill-abstracting continual reinforcement learning for safe, efficient, and comfortable
autonomous driving through vehicle-cloud collaboration. Computer-Aided Civil and Infrastructure Engineering, In Press. http://dx.doi.org/10.1111/mice.13503

Received: 4 January 2025 | Accepted: 22 April 2025

DOI: 10.1111/mice.13503

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

RESEARCH ARTICLE

Skill-abstracting continual reinforcement learning for safe, efficient, and comfortable autonomous driving through vehicle–cloud collaboration

Jing Chen[1] | **Cong Zhao**[1] | **Kun Gao**[2] | **Yuxiong Ji**[1] | **Yuchuan Du**[1]

[1]Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai, China

[2]Department of Architecture and Civil Engineering, Chalmers University of Technology, Gothenburg, Sweden

Correspondence
Cong Zhao, Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai 201804, China.
Email: zhc@tongji.edu.cn

Kun Gao, Department of Architecture and Civil Engineering, Chalmers University of Technology, Gothenburg SE-412 96, Sweden.
Email: gkun@chalmers.se

Funding information
National Natural Science Foundation of China, Grant/Award Number: 52472352; Shanghai Rising-Star Program, Grant/Award Number: 24QA2709600; Shanghai Municipal Science and Technology Major Project, Grant/Award Number: 2021SHZDZX0100; Fundamental Research Funds for the Central Universities, Grant/Award Number: 22120230311

Abstract

Safe, efficient, and comfortable autonomous driving is essential for high-quality transport service in an open road environment. However, most existing driving strategy learning approaches for autonomous driving struggle with varying driving environments, only working properly under certain scenarios. Therefore, this study proposes a novel hierarchical continual reinforcement learning (RL) framework to abstract various driving patterns as skills and support driving strategy adaptation based on vehicle-cloud collaboration. The proposed framework leverages skill abstracting in the cloud to learn driving skills from massive demonstrations and store them as deep RL models, mitigating catastrophic forgetting and data imbalance for driving strategy adaptation. Connected autonomous vehicles' (CAVs) driving strategies are sent to the cloud and continually updated by integrating abstracted driving skills and interactions with parallel environments in the cloud. Then, CAVs receive updated driving strategies from the cloud to interact with the real-time environment. In the experiment, high-fidelity and stochastic environments are created using real-world pavement and traffic data. Experimental results showcase the proposed hierarchical continual RL framework exhibits a 34.04% reduction in potentially hazardous events and a 9.04% improvement in vertical comfort, compared to a classical RL baseline, demonstrating superior driving performance and strong generalization capabilities in varying driving environments. Overall, the proposed framework reinvigorates streaming driving data, prevailing motion planning models, and cloud computation resources for life-long driving strategy learning.

1 | INTRODUCTION

The introduction of autonomous driving into road transportation holds great promise in the near future for safety and efficiency improvements of road traffic. However, the application of autonomous driving is grappling with challenges in the real world where traffic and road conditions are time-varying. The driving decision-making of autonomous vehicles (AVs) is implemented mainly with artificial intelligence technology (Wei et al., 2023). Even a well-trained autonomous driving model often struggles to tackle unfamiliar circumstances (Zhuang et al., 2025). When traffic and road conditions alter, it is necessary to make a rapid adaptation based on previous knowledge. Meanwhile, AVs currently begin to serve as taxis in cities. In this regard, achieving superior driving performance is imperative to deliver high-quality service. However, passenger requirements change on diverse routes due to passengers' age, gender, health, travel purposes, and so forth. Though main objectives (e.g., safety, efficiency, and comfort) may remain unchanged, it requires high-frequency adaptation of autonomous driving models for different preferences. Thus, how to update driving strategy according to streaming data is an urgent problem for large-scale applications of AVs.

Referring to human driving modes, the efficient and dynamic learning of autonomous driving strategy can be formulated as a continual learning problem. In real-world driving, human drivers can adjust driving strategies in new scenarios based on observed information and learned skills across their lifetime. Similarly, driving strategy learning in the real world is also a continuous evolution process. From this perspective, continual reinforcement learning (RL), which can act like humans with the capability of learning and adapting to new scenarios, is suitable for continually driving strategy learning. However, continual RL has two major challenges: catastrophic forgetting and data imbalance (Zhuang et al., 2025). Since continual RL models are updated using streaming driving data, this inevitably leads to the loss of previously learned knowledge after new information is obtained. Moreover, online driving information in different categories has varying amounts. For example, some long-tail scenarios seldom occur in training data, which may be easily covered by the data of other general scenarios (Du et al., 2023; Feng et al., 2023). The imbalanced data further exacerbate catastrophic forgetting in continual RL.

In continual RL, there are three main clusters to retain previous knowledge: explicit knowledge retention, leveraging shared structure, and learning to learn (Khetarpal et al., 2022). The main shortcoming of applying explicit knowledge retention in autonomous driving is the need for large storage memory because the driving decision data are generated every 1 s (Sun et al., 2024) or even 0.1 s (Zhu et al., 2020) in real-world driving. Leveraging shared structure represents that continual RL agents reuse the solutions of previously solved subproblems via function composition, meaningful information abstraction, or skills. That is continual RL agents acting like humans to plan, learn, and reason via automatically breaking a complex task into small subtasks they are proficient in (Khetarpal et al., 2022). However, driving tasks may change over time and vary in state observations, driving objectives, or action spaces. It is difficult to distinguish the boundaries of different tasks. Learning to learn, also known as meta-learning, has received significant attention in recent years (Mao et al., 2024). Meta-learning effectively improves sample efficiency and decomposes learning into two separate processes: meta-training and meta-testing. However, meta-learning has several drawbacks, including high computational costs due to the need for training on multiple tasks, strong dependence on large diverse datasets for effective learning, and the complexity of model design with sophisticated architectures and optimization strategies. These make meta-learning unsuitable for efficient model adaptation in autonomous driving. Thus, it is necessary to modify the existing conventional continual RL approach when applied to autonomous driving. Though hierarchical continual RL is proposed to solve the insufficient knowledge transfer issue (Nayyar & Srivastava, 2024; Pan et al., 2024), work that considers an active and agent-driven setting for exploring tasks in continual RL remains scarce to date (Khetarpal et al., 2022).

To tackle catastrophic forgetting and data imbalance in autonomous driving strategy adaptation, a novel continual RL approach for safe, efficient, and comfortable autonomous driving is proposed based on the development of vehicle–road–cloud integration systems. In the systems, AVs are connected to roadside units and the cloud. Any connected vehicle can upload driving trajectories to the cloud for storage and analysis. The cloud has strong computational resources and rapid responses to update driving strategy remotely based on driving trajectory data to help connected AVs (CAVs) adapt to real-time environments (Gao et al., 2024). Since CAVs encounter non-stationary environments, the goals and settings of driving tasks may vary all the time. Thus, a hierarchical continual RL framework is devised to extract low-level driving policies from imbalanced driving data while progressively optimizing high-level motion planning strategies for CAVs. Since car following is a popular autonomous driving function applied in real-world driving, we use car following as a primary traffic scenario to investigate. Specifically, the detailed contributions are summarized in the following.

1. A hierarchical continual RL framework for safe, efficient, and comfortable autonomous driving is designed. The framework solves a complex motion planning problem with a hierarchical structure by incorporating prior human knowledge and deep RL (DRL). At the low level, distinct driving skills are extracted. At a high level, a DRL agent further integrates multiple skills into the driving strategy adaptation according to varying environments and requirements.
2. A skill-abstracting continual RL approach is proposed. The skill extractors are represented by the DRL agents trained in parallel artificial environments that imitate the human life-long learning process by handling a task with relative skills. Parallel DRL agents automatically learn the boundaries of driving skills based on imbalanced streaming data. The parallel Markov decision process (PMDP) is modified, and a staged training strategy is proposed for parallel DRL agents. The shared skill extractors can handle data imbalance by automatically distinguishing driving skills and avoid catastrophic forgetting by storing the learned skills in DRL models.
3. A life-long driving strategy learning mode applied with vehicle–cloud collaboration is developed. Driving data and traffic and pavement conditions detected by connected vehicles are sent to the cloud and used to establish parallel environments. Prevailing motion planning models are used for abstracting driving skills. The could updates the learning-based motion planning models sent by CAVs via the proposed continual RL approach. Finally, the updated models are sent back to the CAVs.

The remainder of the paper is organized as follows. Section 2 introduces works related to this study. Section 3 proposes a hierarchical continual RL framework for safe, efficient, and comfortable autonomous driving. Section 4 gives details about continual RL-based driving strategy learning for CAVs. Section 5 shows the applications of the proposed approaches in real-world driving cases. Finally, Section 6 concludes this work and gives directions for future research.

2 | RELATED WORKS

Car following and lane change are common scenarios for applying autonomous driving. Since lane change consists of lane selection, trajectory planning, and speed planning, which is challenging to complete using end-to-end approaches, this study regards speed planning as a basic continual learning task, investigates speed planning in car-following scenarios, and further discusses the generalization and scalability capabilities. Conventionally, car-following models have two categories: rule-based and learning-based. Rule-based car-following models are generally developed based on limited professional knowledge. Rule-based car-following models offer efficient and interpretable motion planning but struggle with dynamic driving environments (X. Chen et al., 2024). Learning-based approaches can learn from historical and simulated driving data from dynamic environments to obtain human-like driving behavior or develop a superior driving strategy (Zhu et al., 2018).

In prevailing studies, supervised learning has verified that its testing performance is quite limited by the quality of historical driving data. The DRL algorithms have received much attention in autonomous driving in recent years (S. Chen et al., 2021; Kang et al., 2024; Shi et al., 2022). Specifically, DRL models are designed to train car-following behavior in various driving scenarios with multiple objectives. Zhu et al. (2020) proposed a DRL-based car-following model for safe, efficient, and comfortable speed planning of AVs. The reward function related to efficiency was designed based on the Next Generation SIMulation (NGSIM) dataset, which may not be appropriate for driving on other roads. However, the ride comfort improvement only aimed to restrain longitudinal acceleration with the assumption of even pavements in that study. J. Chen et al. (2023) then presented a comfortable and energy-efficient car-following model via DRL for autonomous driving on rough pavements. Although a comprehensive driving scenario with pavement conditions is considered in the investigation of the learning-based car-following model, the driving scenario merely comprised a rough pavement, a leading vehicle, and a CAV. Subsequently, Huang et al. (2024) proposed an enhanced human-in-the-loop RL approach for safe and efficient autonomous driving to deal with more complex driving scenarios, such as intersections, curves, ramps, and so forth. Sheng et al. (2024) introduced the intelligent driver model (IDM) as an expert model for CAV control strategy learning. Introducing human driving knowledge in DRL-based driving strategy learning can increase learning efficiency and improve driving safety and traffic flow efficiency. However, human knowledge is imbalanced and has distinct preferred driving objectives. Thus, how to efficiently extract diverse skills from human knowledge for specific scenarios remains to be investigated.

In an open driving environment, driving scenarios can differ greatly from those trained. One predictable challenging situation is that traffic and pavement conditions on distinct roads are very different from each other (Du et al., 2023), not to mention unpredictable situations. Meanwhile, both traffic and pavement conditions are crucial for driving safety and ride comfort (Weng et al., 2024; Yi

TABLE 1 Summarization of related works in deep reinforcement learning (DRL)-based autonomous driving.

Main research direction	Representative works	Main contributions
Safe and efficient driving	Huang et al. (2024)	Human mentor-guided RL framework for autonomous driving
Safe and efficient driving	Sheng et al. (2024)	Intelligent driver model (IDM) expert-enhanced vehicle control strategy learning
Safe, efficient, and comfortable driving	J. Chen et al. (2023)	DRL-based car-following model on rough pavements
Comfortable and energy-efficient driving	Du et al. (2022)	DRL-based free-driving model on rough pavements
Generalization in environments	Wei et al. (2023)	Continual RL using shared feature extractor with an elastic weight consolidation (EWC) loss
Generalization in tasks	X. Chen et al. (2024)	Continual RL using EWC and memory-aware synapses

et al., 2023). The continuous evolution of trained models on untrained driving scenarios is an urgent problem.

Recently, researchers noticed that continual learning allows models to adapt to new scenarios continuously. They have tried to introduce continuous learning into the training of DRL-based car-following models. Wei et al. (2023) proposed a continual RL approach for velocity control in autonomous driving to handle various environments using a shared feature extractor with an elastic weight consolidation (EWC) loss. Subsequently, X. Chen et al. (2024) further leveraged continual learning in car-following behavior learning against catastrophic forgetting using the EWC and memory-aware synapses. Though related knowledge can be recalled when dealing with new driving scenarios, existing continual RL-based car-following models cannot explain what has been learned and reused during the whole training process. The whole continual learning process is unexplainable, leading to trustworthiness and safety issues. Thus, the prevailing DRL-based and continual RL-based autonomous driving studies in Table 1 provide limited knowledge for driving strategy adaptation. It is necessary to explore an explainable and trustworthy way for knowledge reuse and continual learning in autonomous driving.

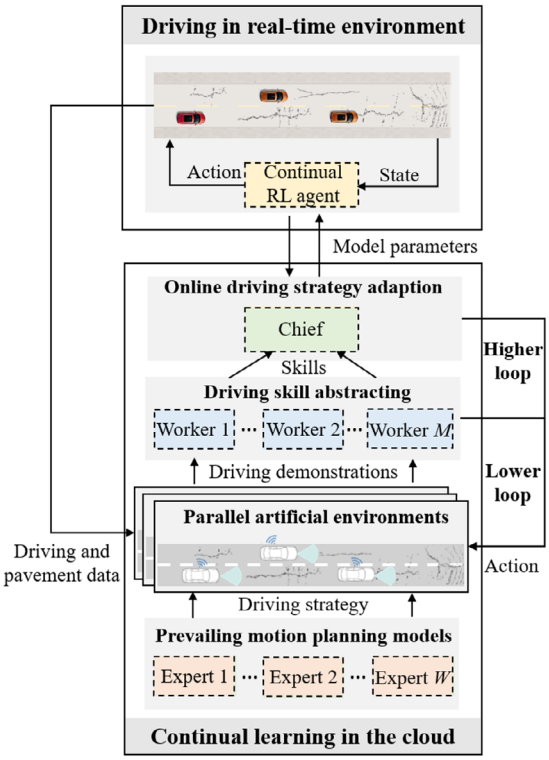

FIGURE 1 Hierarchical continual reinforcement learning (RL) framework structure.

3 | A HIERARCHICAL CONTINUAL RL FRAMEWORK FOR AUTONOMOUS DRIVING

With the development of vehicle–road–cloud integration systems (Gao et al., 2024), connected vehicles can upload data and model (e.g., real-time driving data, detected environment data, and motion planning models) to the roadside units and edge clouds in their driving areas, then each edge cloud, called cloud briefly below, can integrate data and conduct remote model update in its managed area. In the proposed framework shown in Figure 1, the real-time environment is the world CAVs drive in. The driving data from connected human-driven vehicles and detected pavement data from CAVs are uploaded to the roadside units and the cloud. The driving data is the driving trajectory information including vehicle location, speed, acceleration, and detection time, and time headway. The pavement data can include road name, district, road profiles, road roughness, distress type, and detection time. Since pavements are transient static elements in traffic systems, the changes in road profiles within days will not cause significant differences in passenger sensations. Thus, sufficient traffic data on each pavement can be obtained based on the collaborative awareness of onboard sensors (Du et al., 2023). Based on this crowd-sourced data, integrated driving trajectory and road profiles are obtained using data fusion in the cloud.

In the cloud, prevailing motion planning models designed by humans are revived as experts to generate action selections and demonstrations in the cloud; meanwhile, the cloud receives vehicle driving data and pavement data in areas it administers to establish parallel artificial environments. Then, parallel DRL agents are leveraged as workers in the cloud to learn driving skills from messy and imbalanced demonstrations, aiming to circumvent training large models and effectively reduce computational burdens. Further, a collaborative training strategy is proposed to maximize the policy distribution learned by parallel DRL agents to guarantee distinct driving skills are obtained. The collaborative training strategy contributes to fully leveraging existing driving knowledge and exploiting the learning capabilities of DRL agents. Finally, the cloud receives the parameters of learning-based motion planning models from CAVs as the chief to rapidly update driving strategies with consideration of driving skills and sends the updated chief model to the CAVs.

It is noteworthy that the framework in Figure 1 consists of two training loops. One is the lower loop, where parallel DRL agents (workers) learn to distinguish and summarize different driving skills from massive demonstrations in the cloud. The other one is the higher loop, in which the chief continuously adjusts the driving strategy to handle changing traffic and pavement conditions simulated in the parallel environment based on the learned skills. Experience pools are established to store demonstrations generated by parallel DRL agent training and the chief. In practice, the driving objectives of CAVs change according to passenger preferences and the driving styles of surrounding vehicles.

Specifically, a driving skill is defined as a driving strategy for solving a vehicle motion planning problem with a specific objective preference. The main reason is that safe, efficient, and comfortable autonomous driving is a complex motion planning problem for human drivers. In this complex problem, a human driver should consider traffic and pavement conditions (e.g., driving behavior of surrounding vehicles and road profiles) simultaneously, imagine the relationship between vehicle-road and vehicle-vehicle interactions, and balance multiple driving objectives. Indeed, human drivers may have more experience in some simple scenarios, such as safe and efficient car following on even pavements or efficient and comfortable driving on rough pavements. The learned knowledge is stored in their brain as driving skills that can be recalled and further processed when new driving scenarios are encountered. That is an ability of skill abstracting and continual evolution. The hierarchical framework tries to perform continual learning of driving strategies like human drivers.

In the proposed framework, the assumed level of autonomy for the CAV is Level 3. The CAV can operate fully autonomously and interact with human-driven vehicles under expected conditions, but the driver must take over when the system requests it. Meanwhile, CAVs should distinguish whether there is a package loss or communication error in the model parameter transmission from the cloud. If there is a packet loss or the vehicle–road–cloud integration system fails, the CAV will use the original driving strategy. Only when the system is operational and there is no packet loss, the learning-based motion planning model on the CAV will be updated. For the communication delay, considering that the distribution of time headway and pavement conditions are distinct in different areas (e.g., city districts; Du et al., 2023; Zhu et al., 2016, 2020), the maximum acceptable delay time is the shortest time a vehicle that driving through the area managed by the corresponding edge cloud.

The proposed framework has the following advantages. First, the hierarchical structure imitates the learning process of human drivers that abstract knowledge as different skills and then integrate them for specific application scenarios. Second, continual RL helps learning-based motion planning models continuously update during driving. Third, vehicle–cloud collaboration can fully leverage computation resources in the cloud to support the adaptation for CAVs. Fourth, the proposed hierarchical continual learning framework provides a better driving strategy that can deal with more real-world driving scenarios than those designed in simulated environments as a supplement or an additional option for CAVs' driving strategies. If the vehicle–cloud collaboration fails, CAVs can still use their driving strategies in designed driving scenarios.

4 | CONTINUAL RL-BASED DRIVING STRATEGY LEARNING

To deal with catastrophic forgetting and data imbalance, a PMDP is proposed to conduct exploration and exploitation for skill abstracting and model adaptation. Then, a staged training strategy is further proposed to balance exploration and exploitation within a limited training time for driving skills. Finally, the state, action, and reward functions of DRL-based car-following agents are defined for continual learning.

4.1 | PMDPs with skill abstracting

MDPs have been used as a standard method for motion planning of CAVs (Wang et al., 2021). However, representing a complex autonomous driving task in a single MDP is found to be data-inefficient. Thus, PMDPs were proposed to execute MDPs in parallel (Sucar, 2007). At each timestep, the action conducted in the environment is selected from

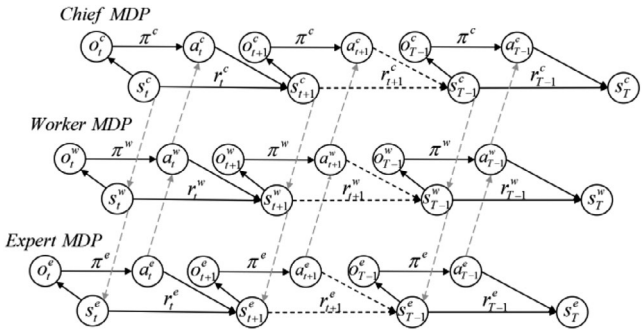

FIGURE 2 Schematic diagram of parallel Markov decision process (PMDP).

each MDP. Specifically, each process shares the same state and action space with different reward functions. However, in prevailing studies, the motion planning models are generally designed for specific autonomous driving scenarios. The state definitions are similar but not identical, while the action space and transition function can be the same. Therefore, the definitions of previous PMDPs are modified in this study, and novel PMDPs are proposed to formulate the core concept of the proposed hierarchical continual RL framework.

The proposed PMDPs contain a chief MDP P^c, worker MDPs P^s, and expert MDPs P^e for strategy adaptation, skill abstracting, and demonstration collection, respectively, as shown in Figure 2. The gray arrows mean the information transfer between the chief, worker, and expert MDPs. Their shared state space, action space, and specific state transition function are defined as S, A, and F, respectively. Since a driving skill is defined as a driving strategy for solving a vehicle motion planning problem with a specific objective preference, the driving skill learned by the m th worker is defined as K_m^w. In these PMDPs, a chief is responsible for interacting with a parallel environment E^c in the cloud, acting like a CAV. Workers are also applied in parallel environments E^w and used as skill extractors. Experts are applied in workers' parallel environment E^w using prevailing motion planning models. Thus, the parallel environment states for chief, workers, and experts are s^c, s^w, and s^e, respectively. The states in parallel environments are initialized with the states in the real-time environment. The action, reward, and transition function for the chief, workers, and experts are $[a^c, r^c, \phi^c(s_t^c, a_t^c)]$, $[a^w, r^w, \phi^w(s_t^w, a_t^w)]$, and $[a^e, r^e, \phi^w(s_t^w, a_t^w)]$, respectively. In the cloud, the workers learn driving skills from different perspectives to provide action selection suggestions. Thus, the chief and workers share the same state definition, reward function, and transition function, while the state definition and reward function for DRL-based experts can be changed according to driving scenarios and passenger requirements based on prevailing studies. The transition functions in parallel environments are set using vehicle dynamics models and detected driving data and pavement conditions, referring to model-based DRL approaches (Sheng et al., 2024).

The expert MDP represents the driving decision-making process of prevailing motion planning models in different scenarios. Worker MDP represents the driving decision-making process for specific driving tasks and objectives (driving skills). Specifically, driving skills are learned from the demonstrations, which is an offline learning process. As shown in Figure 1, the chief and workers are trained in parallel artificial environments of the cloud, called the chief environment and worker environment in the following. Chief MDP represents the intelligent driving decision-making process of a CAV in the chief environment with model parameters sent from the CAV in the real-time environment. When the chief MDP receives a state s_t^c from the chief environment, the state information will first pass to workers and then arrive at experts. Experts then produce action a_t^e based on their observations o_t^e and policies π^e. Then, the expert action a_t^e is transferred to workers to guide training. Meanwhile, workers also generate action a_t^w by copying expert action a_t^e to collect training data before the training begins or using their own policies π^w after the training begins based on worker observation o_t^w. Furthermore, worker actions a_t^w are sent to the chief and these actions are further integrated with the chief action selection a_t^c based on chief observations o_t^c. Subsequently, the chief action a_t^c conducts in the chief environment and generates state s_{t+1}^c, and the above process proceeds further. Since this study focuses on continually driving performance improvements in changing environments, it is assumed that there is no bias between state and observation for simplicity, and the states are directly used as observations in the following. For a certain driving event, there are T timesteps in an episode when terminal conditions are satisfied, and then the environments in the next episode can be regarded as being reset for a new episode. In practice, there can be several worker and expert MDPs, and the number of chief, worker, and expert MDPs is addable.

For classical actor–critic DRL algorithms, the objective of actor-network training is to maximize the expected accumulated rewards of a DRL agent. In contrast, workers collaborate for higher expected accumulated rewards of both the chief and workers in the proposed framework. Meanwhile, workers are asked to learn different driving skills of the car following by maximizing the distances between policies. In this way, workers abstract expert demonstrations as skills saved as DRL models against catastrophic forgetting. Specifically, skill abstracting is not influenced by the distribution of expert demonstrations because workers continuously explore and exploit distinct

policies. Thus, the objective function of workers' actor networks is modified with a chief Q value and a divergence measurement between parallel workers in the exploration stage:

$$\max_{\theta^{\mu_m^w}} \mathrm{E}_{s_t\sim E^c}\left[Q_m^w\left(s_t,\mu_m^w\left(s_t\left|\theta^{\mu_m^w}\right.\right)\left|\theta^{Q_m^w}\right.\right) + \lambda_1 Q^c\left(s_t\left|\mu^c\left(s_t\left|\theta^{\mu^c}\right.\right)\right|\theta^{Q^c}\right.\right) + \lambda_2\sum_{i\in I} D\left[\mu_m^w,\mu_i^w\right]\Bigg] \tag{1}$$

where the state s_t is given by the chief environment E^c at timestep t; $Q(s,\mu|\theta^Q)$ is the critic network with parameter θ^Q and the input of state s and policy μ; Q_m^w and Q^c are the main critic network of the m th workers and chief, respectively; μ_m^w and μ^c are the policies in the m th workers and chief, respectively; $\theta^{\mu_m^w}$ and θ^{μ^c} are the parameters in the actor networks of the m th worker and the chief, respectively; $\theta^{Q_m^w}$ and θ^{Q^c} are the parameters in the critic networks of the m th worker and the chief, respectively; λ_1 and λ_2 are penalty coefficients of the Q value and divergence, respectively; I is the set of other workers, and $I = [1,m)\cup(m,M]$; D is the divergence (e.g., KL divergence); $D[\mu_m^w,\mu_i^w]$ represents the distance between μ_m^w and μ_i^w. In Equation (1), the first Q maximizes the reward of the worker, the second Q maximizes the reward of the chief, and the third term maximizes the divergence between the workers' driving skills distributions.

4.2 | Training strategy for efficient convergence

To avoid being trapped in local optimum solutions, a noise for action selection is added in both the chief and workers. For convenience, we divide the training process into two main parts, naming them the exploration and convergence processes according to noise. Indeed, the whole training process of DRL models balances exploration and exploitation. Specifically, the exploration stage means noise exists to compel agents to explore unacquainted areas. Conversely, the convergence stage indicates that agents almost directly use the output actions to conduct with minimal noise. The action selection with noise is described as

$$a_t^x = \min\left(\max\left(a_{\min}^x,\mu^x\left(s_t\left|\theta^{\mu^x}\right.\right)+N_t\right),a_{\max}^x\right) \tag{2}$$

where x represents the chief c and workers w; a_t is the action selection at timestep t; $a_{\min}$ and $a_{\max}$ are minimum and maximum values of action selection, respectively; N_t is noise, randomly sampled from normal distribution data $N(0,\gamma_t^2)$, $\sigma_t = \gamma_n\sigma_{t-1}$; σ_t is the discount factor of noise.

In the cloud, workers are trained in parallel environments. Since there are several parallel DRL agents training for driving skill abstracting simultaneously, the convergence efficiency of DRL models is significant. For the workers, demonstrations from expert models provided acceptable actions. Therefore, workers are designed to learn expert action selection to begin with relatively good policies in the early training stage when the workers in the cloud are not trained before. Thus, the worker action selection $a_{m,t}^w$ in the early training stage can be described as

$$a_{m,t}^w = a_{j,t}^e \tag{3}$$

where $a_{m,t}^w$ is the action of the m th worker; and $a_{j,t}^e$ is the action of the j th expert model which provides demonstrations.

Based on the above actor-mimic, workers can explore and exploit more efficiently in new driving scenarios with relatively good beginnings. Subsequently, since there are many action selections, how to determine the action selections of the chief and workers based on parallel DRL and expert models is an issue to be solved. There are two main methods: higher-value action selection and average action leverage. Generally, actions with higher values represent higher expected accumulated rewards and better action selection. However, agents estimate the Q function with neural networks based on experiences. The distinct Q functions may lead to confusion in the Q value comparison. Thus, the same estimated Q function is used for comparison at each time, and actions with higher Q values in training are selected to conduct. The higher-value action selection of the workers and chief can be described as

$$a_{m,t}^w = \arg\max\left(Q_m^w\left(s_t,a_t^e\right),Q_m^w\left(s_t,a_{m,t}^w\right)\right) \tag{4}$$

$$a_t^c = \arg\max\left(Q^c\left(s_t,a_{m,t}^w\right),Q_m^c\left(s_t,a_t^c\right)\right) \tag{5}$$

When the noise N_t reduces, the linear weighted action between the chief and workers is used to interact with the chief environment as the chief's driving strategy. Specifically, workers have received some knowledge to abstract driving skills in handling complex motion planning problems from different perspectives. These driving skills are stored as parallel DRL models (workers) in the cloud, continuously evolving with the increment of driving data and providing valuable action selections. Thus, the final chief's action a_t can be determined as

$$a_t = \sum_m w_m a_{m,t}^w + w_c a_t^c \tag{6}$$

where n is the total number of workers; w_m and w_c are weightings of the worker m and chief. In this study, the weightings are set as $w_m = w_c = 1/(1+M)$ to treat each action of the workers and the chief equally as an example.

For a driving task with a discrete action space, a weighted action can be the nearest number in the action space.

Thus, the training process is staged based on action selection and policy training to ensure the chief can get a better driving strategy by comparing it with workers, and then the chief has an efficient and stable convergence in the later training process:

1. During the early training stage (action noise is set as its maximum value), the chief conducts a full exploration, while workers are trained using expert models. At each timestep, the state in the chief environment E^c is copied to parallel worker environment E^w. Experts output actions to conduct in parallel worker environments. Furthermore, workers receive states and rewards in the next timestep. These experiences $(s_t^e, a_t^e, r_t^e, s_{t+1}^e)$ from experts are used to train workers with relatively good beginnings.
2. In the middle training stage (action noise begins to decrease, and learning begins), for workers, actions with higher Q values are chosen by comparing the Q value of actions selected by experts. The chief's actions are compared with those generated by workers' driving skills. This can ensure the best action is selected based on current knowledge. Workers collaborate to maximize the Q value of the chief and workers; meanwhile, the policies of workers should maintain a certain distance to ensure workers can learn different policies.
3. In the later training stage (the chief and workers gradually converge), the average value of actions from the chief and workers is used as an example to interact with the chief environment to make the policy more robust. In this stage, the collaboration between workers continues.
4. In the last training stage (after convergence), the workers are further trained with their own experiences. Specifically, the chief's action still uses the average one as an example. In this stage, the collaboration stops, and workers train policies with their characteristics.

For DRL, if higher Q values can be obtained during the training process, the MDP can converge. In the early and middle training stages of the PMDP, the actions with higher Q values are selected and conducted for the workers and chief, contributing to high-quality exploration. Though the average action selection strategy may impact convergence, the chief has learned driving skills from workers, and the workers may have better action selection in some scenarios. Thus, the average action can help the chief to obtain more reliable actions in potentially hazardous scenarios, following the main idea of ensemble learning (Alam et al., 2020). Finally, the chief can be trained to converge.

4.3 | DRL agent definitions

Since the motion planning of autonomous driving involving speed and direction control is complicated, car following is considered as a primary driving scenario to continuously learn safe, efficient, and comfortable driving based on interactions with road infrastructure and vehicles. However, safe, efficient, and comfortable driving skill abstracting is challenging. For safe and efficient driving, there are a large number of studies providing paradigms of DRL agent settings (Du et al., 2022; Ye et al., 2019; Zhu et al., 2020). For comfortable driving, motion planning should avoid abrupt acceleration and deceleration (Genser et al., 2022). Meanwhile, driving on rough pavements should further consider how to relate passenger feelings with random road profiles and time-varying vehicle body vibrations. Thus, state, action, and reward function settings are summarized based on existing studies and our numerous trials to provide parallel DRL agents general assessments. Parallel DRL agents (workers) can distinguish different skills by maximizing the distance between learned policies.

Since car following is related to the driving behavior of a CAV and its leading vehicle, the transition function ϕ in parallel artificial environments can be described as simple kinematic models. The speed and position of a CAV and its leading vehicle in each timestep follow these formulations:

$$V_{t+1} = V_t + a_t \Delta T \tag{7}$$

$$\Delta V_t = V_{l,t} - V_t \tag{8}$$

$$\Delta S_{t+1} = \Delta S_t + \frac{(\Delta V_t + \Delta V_{t+1})\,\Delta T}{2} \tag{9}$$

where ΔT is the time resolution, generally set as 0.1 s; ΔV_t is the relative speed between the leading vehicle and following CAV at timestep t; V_t and $V_{l,t}$ are the speeds of the CAV and its leading vehicle at timestep t, respectively; ΔS_t is the clearance space between the CAV and its leading vehicle at timestep t.

For the chief and workers, the action is longitudinal acceleration within the bounds of $[-3, 3]$. Training the DRL models can be regarded as a process of balancing exploration and exploitation. Although the safety feature will be considered in the reward function, hazardous actions may still occur. A safeguard is required to avoid potentially hazardous scenarios. Thus, a rule-based safeguard is set to prevent CAVs from rear-end collisions:

$$a_t = \begin{cases} -3, & \Delta S_t < d_t \\ DRL\ \text{modeloutput}, & \text{otherwise} \end{cases} \tag{10}$$

$$d_t = V_t t_r + \frac{V_t^2}{2a_{\min}} - \frac{V_{l,t}^2}{2a_{\min}} \tag{11}$$

where d_t is the safe distance at timestep t; t_r is the reaction time, set as 1 s in this study; $a_{\min}$ is the minimum acceleration, which also is the maximum deceleration.

The motion planning of CAVs generally aims to achieve safe, efficient, and comfortable driving when following a leading vehicle, regardless of pavement conditions. Thus, a comprehensive motion planning model that mainly includes the states, reward functions, and actions of the chief and workers is designed. To receive enough environmental information, the state definition contains observations in both car-following and free-driving behaviors, defined as

$$s_t = [a_{t-1}, V_t, \Delta V_t, \Delta S_t, P_t] \tag{12}$$

where a_{t-1} is the longitudinal acceleration at the last timestep $t-1$, while the current timestep is t; P_t is the road pavement condition information of oncoming roads. Indeed, pavement information is high-dimensional and random. To understand the relationship between road profiles, speed, and ride comfort efficiently, the "maximum comfortable speed (MCS)" $V_{p,t}$ is used to represent the road profiles of oncoming roads as the road pavement condition information P_t (Du et al., 2022). In this way, DRL agents directly get the information of a speed threshold for comfortable driving, circumventing learning transition functions of non-linear and time-varying vehicle suspension systems.

Since the motion planning of CAVs should satisfy various passenger requirements, a reward function involves as many requirements as possible to provide a high-quality driving service. Generally, the total reward is the summation of features related to different objectives. Since passengers have diverse preferences, weights are used to adjust the importance of objectives. In this case, the weights are used to adjust each feature to a similar magnitude. Notably, weightings are responsible for showing differences in preference. The total reward is described as

$$r = w_1 R_{st} + w_2 R_{sd} + w_3 R_{eh} + w_4 R_{es} + w_5 R_{lj} + w_6 R_{la} + w_7 R_v \tag{13}$$

where r is the total reward; w_1, w_2, w_3, w_4, w_5, w_6, and w_7 are weightings that can be adjusted according to passengers' preferences; R_{st} and R_{sd} are the safety features related to time-to-collision (TTC) and safe distance; R_{eh} and R_{es} are the efficiency features associated with headway and speed; R_{lj} and R_{la} are the longitudinal comfort features evaluated by longitudinal jerk and acceleration, respectively; R_v is the vertical comfort feature related to passenger feelings.

For safety, TTC is a widely used metric to evaluate the potential hazard when following a vehicle in front (Li et al., 2022). A TTC value below the threshold indicates that a rear-end collision is probably occurring. Thus, if a TTC is below the threshold, the agents will be punished:

$$TTC_t = \begin{cases} -\frac{\Delta S_t}{\Delta V_t}, & V_t > V_{l,t} \\ \infty, & V_t \le V_{l,t} \end{cases} \tag{14}$$

$$R_{st} = \begin{cases} \log\left(\frac{TTC_t}{TTC_{thr}}\right), & 0 \le TTC_t \le TTC_{thr} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where TTC_t is the TTC at timestep t. TTC_{thr} is the threshold of the TTC values to classify safe and hazardous behaviors. Meanwhile, according to Equation (10), the rule-based safeguard can prevent rear-end collisions, but it is an emergency braking strategy that probably causes dangers for following vehicles. If the clearance space is smaller than the safe distance, the agents will be punished:

$$R_{sd} = \begin{cases} -10, & \Delta S_t < d_t \\ 0, & \Delta S_t \ge d_t \end{cases} \tag{16}$$

For efficiency, time headway works as a positive reward if the CAV can follow its leading vehicle while maintaining an expected headway. The reward function related to headway is expressed as a logarithmic normal probability distribution function fitted using human driving data. Since CAVs must interact with surrounding vehicles in an open road environment (Ji et al., 2023), especially with human drivers, human-like car-following strategies are encouraged. The headway also suggests the driving style of drivers in different places. Specifically, regardless of how fast a leading vehicle drives, the following CAV should learn how to mimic the human driving style and maintain an expected headway for driving efficiency and safety. Thus, the design of the headway feature is

$$R_{eh} = \frac{1}{h\sigma\sqrt{2\pi}} e^{\frac{-(\ln h - \mu)}{2\sigma^2}} \tag{17}$$

where h is the headway; μ and σ are the parameters to describe headway distributions using real-time traffic data. For example, R_{eh} can be set with $\mu = 0.49$ and $\sigma = 0.42$ in Shanghai, China (Zhu et al., 2016), while it can be set with $\mu = 0.4226$ and $\sigma = 0.4365$ in California (Zhu et al., 2020). In practice, the headway feature can be designed differently based on traffic data obtained from corresponding pavements. Notably, leading vehicle data in different places can be used to test the scalability of the proposed framework in various driving environments. Then, passengers may hope CAVs can drive as fast as possible within speed limits in free driving. According to the study of Du et al. (2023), a minimum speed ensures driving efficiency and avoids unexpected vehicle vibration. Thus, a

driving efficiency feature is designed as

$$R_{es} = \begin{cases} -\frac{(V_t - V_{\max})^2}{V_{\max}^2}, & V_t \geq V_{\min} \\ -\rho\frac{(V_t - V_{\max})^2}{V_{\max}^2}, & V_t < V_{\min} \end{cases} \quad (18)$$

where $V_{\max}$ is the maximum speed; and $V_{\min}$ is the minimum speed; ρ is the penalty coefficient, set as 11.

For ride comfort, most studies aimed to mitigate the absolute value of longitudinal acceleration and jerk for longitudinal ride comfort improvements. Since the maximum longitudinal acceleration and jerk are 3 and 60 m/s^3, respectively, they are divided by different base values. The rewards related to longitudinal ride comfort can be described as

$$j_t = \frac{a_t - a_{t-1}}{\Delta T} \quad (19)$$

$$R_{lj} = -\frac{j_t^2}{j_{\max}^2} \quad (20)$$

$$R_{la} = -\frac{a_t^2}{\alpha^2} \quad (21)$$

where j_t is the longitudinal jerk (change rate of longitudinal acceleration) at timestep t; $j_{\max}$ is the maximum longitudinal jerk, which is 60; α is the base value of longitudinal acceleration, set as 90. Thus, the values of rewards can be bound to a reasonable level and provide room for acceleration adjustment.

In real-world applications, CAVs may be dispatched for driving tasks in different areas with various pavement conditions. At this time, unsuitable motion planning strategies may produce dramatic vibration of vehicle bodies on rough pavements (Du et al., 2023). Therefore, vehicle body vibration should be considered to improve vertical ride comfort. However, pavement–vehicle–passenger systems are high-dimensional and difficult to formulate accurately in motion planning. Thus, the MCS is used to provide information on the maximum values of comfortable speeds on segments, which is calculated using vehicle body vibration simulation. In the simulation, the CAV is asked to drive on the pavement at different speeds. A full-car model then inputs road profiles of the right and left wheels in the time domain. Furthermore, vertical acceleration on the seat is recorded for ride comfort evaluation. The maximum speeds that can satisfy ride comfort standards from the MCS. The detailed simulation process was introduced in our previous study (Du et al., 2022). The MCS is only related to the CAV's position. When the speed is below the MCS, it is acceptable for most passengers, and agents will not be punished. Conversely, if the speed exceeds the MCS, the deviation guides training. Thus, the reward-related to vertical ride comfort can be designed as

$$R_v = \begin{cases} -\frac{|V_{p,t}^0 - V_t|}{V_{p,t}^0}, & V_t > V_{p,t}^0 \\ 0, & V_t \leq V_{p,t}^0 \end{cases} \quad (22)$$

where $V_{p,t}^0$ is the MCS at the current position.

5 | EXPERIMENTS AND RESULTS

To verify the effectiveness of the proposed hierarchical continual RL framework and skill-abstracting continual RL approach, this section first elaborates on the settings of the simulation environment, network structure, and parameters. Then, real-world pavement and leading vehicle data are used to conduct iterative training for sampling efficiency and training performance comparison. Further, the scalability of the proposed hierarchical continual RL framework is tested using a reward function and an expert different from the training ones. Finally, the generalization of a car-following model continuously trained based on the proposed framework is tested on untrained leading vehicle data.

5.1 | Simulation settings

To establish a more realistic simulation environment, road pavement and traffic data obtained in Shanghai, China, are used for pavement condition and reward function settings. The details about road pavement and traffic data in Shanghai were introduced in the works of Du et al. (2023) and Zhu et al. (2016). Since a CAV's driving style is determined by its reward function, the driving behavior of leading vehicles minimally influences the experimental results. Meanwhile, the environment should contain some randomness. Thus, traffic data are extracted from the NGSIM dataset as leading vehicles to provide a random driving environment and show the scalability of the proposed framework. The leveraged NGSIM dataset is described in the work of Zhu et al. (2020).

In this case study, we show the driving performances of different car-following models when the pavement quality and leading vehicle change rapidly. That is, we design sharp-change experiment scenarios to show the adaptation ability during life-long driving. In practice, CAVs may not drive on rough pavements in most driving time because there is a small proportion of rough pavements with an International Roughness Index value larger than 6.1 in the pavement dataset (Du et al., 2022). Meanwhile, CAVs may maintain smooth driving trajectories without

interference from cut-in and cut-out human-driven vehicles. In experiments, the life-long simulation process is compressed, and the adaptation process is shown.

Since CAVs have lane-keeping assistance, the CAVs are assumed to drive along centerlines when driving in lanes. Road profiles are sampled according to the left and right wheel locations. Fluctuating road profiles are processed as the MCS on road segments using the method proposed in our previous study (Du et al., 2022). In the parallel artificial environments, pavement and leading vehicle data are chosen stochastically from the datasets. Specifically, pavement data is selected first, and then vehicle data is randomly selected to set as a leading vehicle. In each timestep, the position and speed of the leading vehicle are set according to timestep, while the MCS of the pavement is set according to CAV's position in the chief environment. If the end of pavements and leading vehicle data arrives or rear-end collision occurs, the episode terminates and a new episode begins. To show the life-long learning process of the workers and chief in the cloud, we design a case in which a CAV drives into an area where the number of experiences in the cloud is lower than the size of workers' experience pools, meaning that the workers cannot start driving skill abstracting at the beginning. The driving skills in the cloud are modeled as workers and trained using demonstrations from prevailing free-driving and car-following models (experts) with random initialization. In practice, the workers can be continuously trained using streaming data from the real-time environment and experts.

Since prevailing rule-based and learning-based motion planning models can represent human knowledge for superior driving performances in safety, efficiency, or ride comfort, we leverage a rule-based car-following model and conventional DRL-based car-following models for safe, efficient, and comfortable driving as experts under different traffic and pavement conditions. The training of DRL-based free-driving and car-following models maximizes their Q values with specific rewards designed in the work of Du et al. (2022) and Zhu et al. (2020). The training of experts did not consider policy distributions. Specifically, the DRL-based free-driving model is trained on rough pavements, while the DRL-based car-following model is trained on even pavements and follows random leading vehicles.

In this case, two workers and two expert models are used as an example. The expert models directly use the DRL-based free-driving and car-following models in prevailing studies (Du et al., 2022; Zhu et al., 2020). Then, a rule-based car-following model, the IDM, is also used as an expert, and another worker is added to the framework as another example. Notably, these experts solve car-following problems with different emphases on objectives in parallel environments, but state transitions and action definitions are the same. Since the safe, efficient, and comfortable car following of CAVs has an intricate relationship between state, action, and reward function, all the actor–critic-based DRL algorithms suffer from balancing exploration and exploitation. Thus, deep deterministic policy gradient (DDPG) is used as a simple baseline to show the effectiveness of the proposed framework and approach (Lillicrap et al., 2015). The one trained with the hierarchical continual RL framework is called the continual DDPG (CDDPG) in the following.

TABLE 2 Parameter and weight values of the models.

Parameter	Value	Reward feature	Weight
Learning rate actor	0.0001	Safety R_{st}	1
Learning rate critic	0.001	Safety R_{sd}	10
Gamma	0.9	Efficiency R_{eh}	2
Replace iteration actor	500	Efficiency R_{es}	1
Replace iteration critic	300	Comfort R_{lj}	1
Memory capacity	15,000	Comfort R_{la}	0.1
Batch size	1024	Comfort R_{v}	5
Noise decay rate	0.9995		

In the proposed hierarchical continual RL framework, the chief and workers learn car-following strategies with a larger state space and a more complex reward function than experts. According to the experiments conducted in our previous study (Du et al., 2022), more hidden layers and neurons benefit stabler convergence and higher rewards. Thus, four hidden layers with 100-50-30-20 neurons are leveraged to learn policy and estimate the Q value in the chief and workers, while one hidden layer with 30 neurons and three hidden layers with 50-30-20 neurons are used for free-driving and car-following experts, following the settings in previous studies (Du et al., 2022; Zhu et al., 2020). In training, the workers are trained in parallel artificial environments to abstract driving skills. The parameter and weight values of the CDDPG, ensemble DDPG (EDDPG), and DDPG models are listed in Table 2.

5.2 | Training performance comparison

In this subsection, the training performance of the CDDPG is compared with a classical DDPG model and an EDDPG model in stationary and changing environments. The EDDPG model is trained by combining ensemble learning and the classical DDPG algorithm; that is, the motion planning can consider action selections from other motion planning models to improve robustness by using the average value of all acceptable actions (Dong et al., 2020). The CDDPG models consist of the chief and workers. Since

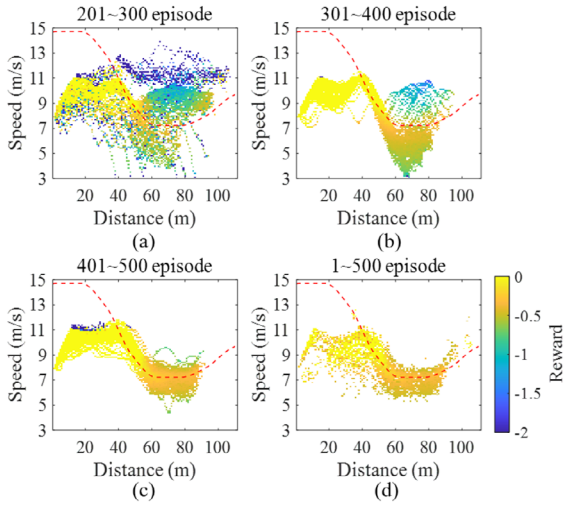

FIGURE 3 Exploration and exploitation of the deep deterministic policy gradient (DDPG) model.

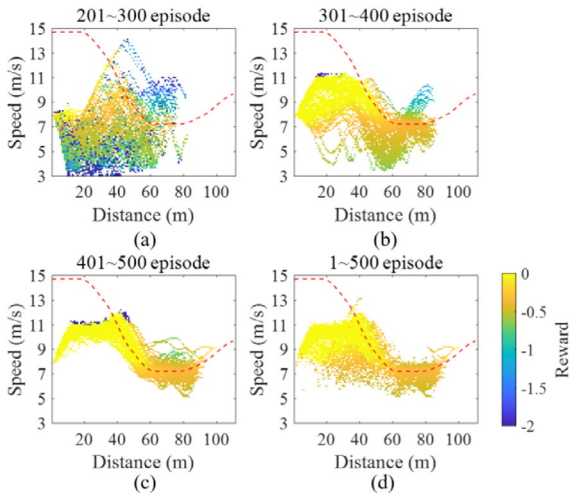

FIGURE 4 Exploration and exploitation of the continual DDPG (CDDPG) model.

the chief parameters are sent to the CAVs for update, the chief represents the training and driving performance of the CDDPG model. Thus, the chief in the CDDPG model is briefly called the CDDPG model when compared to the EDDPG and DDPG models in the following sections.

5.2.1 | Training in a stationary environment

To set a stationary environment, a rough pavement and a leading vehicle driving trajectory are sampled from the datasets. At each episode, a CAV starts at the beginning of the pavement, and the original speed is set as the leading vehicle's speed. The CAV then adjusts its speed to maintain a suitable headway and improve ride comfort by selecting suitable acceleration. Since the state is high-dimensional, the training process cannot be depicted with two- and three-dimensional pictures. Thus, the reward distribution on speed and position is shown in Figures 3 and 4 to illustrate the contributions of the proposed hierarchical continual RL framework. An average reward value is used to represent rewards at the same speed and position. Specifically, each subfigure depicts the reward distribution in 100 episodes (from 200 to 500 episodes). The final subfigure shows the average reward value during the whole training process (within 500 episodes). As shown in Figures 3 and 4, the red dotted line is the MCS at the corresponding position. During driving, a 60-meter MCS for the oncoming road is provided to potentially learn predictive car-following strategies (Du et al., 2022).

In this case, the expected driving strategy is to follow the leading vehicle when the MCS is much higher than the speed of the leading vehicle, then reduce the speed to close to or below the MCS when the MCS is lower than the current driving speed, and increase speed to chase the leading vehicle when the CAV drives on a relatively even segment. Compared to the DDPG model, the CDDPG model can explore in more extensive state space during 301 to 400 episodes (see Figures 3b and 4b). Finally, the CDDPG model obtains more knowledge about good action selection after 500-episode training (see Figures 3d and 4d). Thus, the proposed hierarchical continual RL framework can effectively improve the sampling efficiency with the sophisticated staged training strategy. That means the proposed hierarchical continual RL framework can effectively leverage distinct driving skills to improve exploration efficiency and obtain more high-quality driving strategies within limited learning time, contributing computational efficiency for driving strategy adaptation.

5.2.2 | Training in a changing environment

The training on changing pavement and traffic conditions is more difficult due to randomness, compared to a stationary environment. Thus, the training performance comparison on the training set is significant. Figure 5a,b demonstrates the training trajectories of the CDDPG, EDDPG, and DDPG models with changing pavement and traffic conditions. The transparent line represents the average reward in each episode, while the solid line is the scrolling reward with a window of 100 timesteps. In Figure 5a, the traditional policy ensemble contributes to a wider exploration space but also extends the time to convergence because some episodes are terminated early for hazardous actions. In the exploration process, the proposed staged training strategy helps agents receive high rewards and abundant experiences for learning. Compared

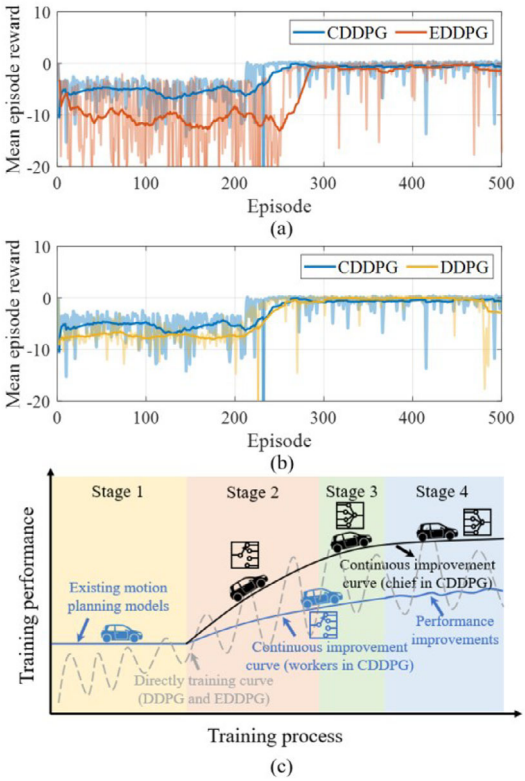

FIGURE 5 Training performances of the CDDPG, ensemble DDPG (EDDPG), and DDPG models in 500 episodes.

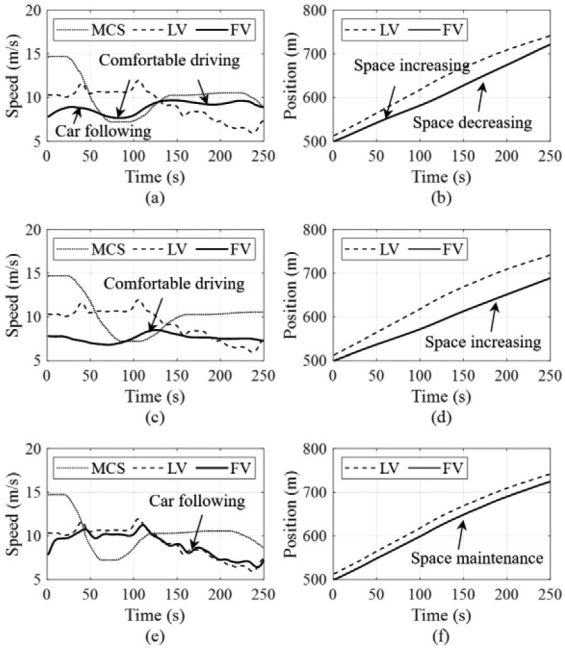

FIGURE 6 Speed profiles and movements of the best (a) and (b) CDDPG; (c) and (d) EDDPG; and (e) and (f) DDPG models at 370, 400, and 600 episodes.

to the EDDPG model, it is obvious that the training trajectory is much smoother, and the difficulty of training several models is reduced in the CDDPG model. In Figure 5b, the CDDPG model has higher rewards during the exploration process and has a more stable convergence curve, compared to the DDPG model. In summary, the proposed hierarchical continual RL framework contributes to efficient sampling and rapid convergence.

Figure 5c demonstrates the driving performance improvement process of the CDDPG model in different stages. The four stages correspond to the stages mentioned in Section 4.2. The black and blue curves represent the training trajectories of the chief and worker in the CDDPG model, respectively. The case study begins with the random initialization of the chief and workers. Prevailing motion planning models in existing studies (experts) are used to generate action selection of the workers in Stage 1, while the chief can be set as a learning-based motion planning model of the CAVs. Then, workers begin to collaborate for abstracting distinct driving skills, and the chief chooses the best driving strategies among its policy and workers' driving skills in Stage 2, ensuring the chief can conduct a good action based on available knowledge. Further, workers still collaborate, and the chief adjusts the driving strategy based on its policy and workers' driving skills in Stage 3. Finally, the workers further adjust model parameters using interactions with parallel environments in Stage 4, contributing to stable improvements of the chief driving performance.

For DRL, even if training converges, an ideal driving performance may not be obtained. Since a DRL model often gets into sub-optimal solutions, some unexpected situations still happen. To show the necessity of the proposed hierarchical continual RL framework, the models with the best driving performance in the training process are sampled to show the learning contents. Specifically, the best CDDPG, EDDPG, and DDPG models are sampled from 370, 400, and 600 episodes, respectively. As shown in Figure 6a,b, the CDDPG model can learn the expected speed profile efficiently with the least iterations. Compared to the EDDPG and DDPG models, which can only learn comfortable driving (see Figure 6c,d) and efficient car-following behavior (see Figure 6e,f), while the CDDPG model can adjust speed effectively according to the leading vehicle and pavement conditions.

In this case, the experiments are executed on a computer with Intel Core i9-12900H at 2.50 GHz and 16 GB RAM. The training times for the convergence of the CDDPG, EDDPG, and DDPG models in Figure 5 are 89.11, 317.12, and 26.38 s, respectively. The times used for 500-episode training of the CDDPG, EDDPG, and DDPG models are 1842.71, 3481.52, and 2035.42 s, respectively. The main reason is that the CDDPG model leverages the continual RL framework

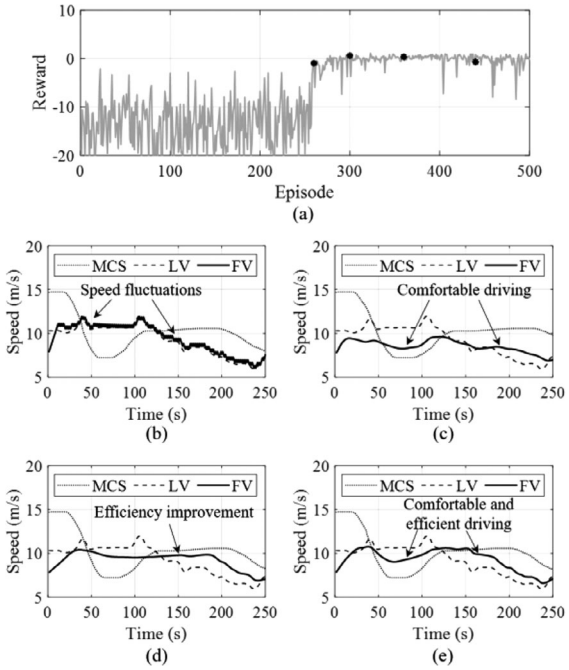

FIGURE 7 Training process of the chief in the CDDPG model with an untrained reward function and demonstrations from DRL experts. Subfigures (b), (c), (d), and (e) are the speed trajectories generated by the chief at 260, 300, 360, and 440 episodes, respectively.

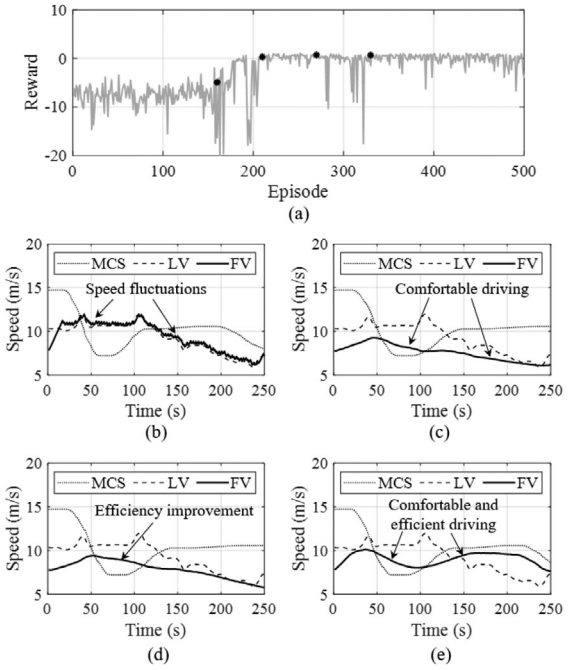

FIGURE 8 Training process of the chief in the CDDPG model with an untrained reward function and demonstrations from DRL and IDM experts. Subfigures (b), (c), (d), and (e) are the speed trajectories generated by the chief at 160, 210, 270, and 330 episodes, respectively.

to obtain more high-quality data, which can effectively save exploration time and contribute to fast convergence. However, since more than one model is running with the CDDPG model, the training time of the CDDPG model is a little more than that of the DDPG model.

5.3 | Scalability of the proposed framework

To show the scalability of the proposed hierarchical continual RL framework, the traffic data obtained in Shanghai is used, which differs from the expert DRL-based car-following model, to design an untrained efficiency feature R_{eh}. Since the design of R_{eh} depends on the characteristics of traffic flow data, the changes of R_{eh} also represents autonomous driving on different roads and efficiency requirements. The models at different episodes are sampled to show the training process at 260, 300, 360, and 440 episodes (see Figure 7a). As shown in Figure 7b–e, the chief first learns to reduce speed for comfortable driving, then aims to increase speed on even segments to maintain the expected headway, and finally obtains a comfortable and efficient driving performance. Since the headway in Shanghai is smaller than that in California, and headway adjustment strategies are lacking among experts, emergency braking often occurs in the exploration stage. In this case, comfortable driving is easier to learn. Thus, the chief first learns comfortable driving and adjusts speed to chase the leading vehicle. Particularly, the final driving speed on the rough segment is higher, compared to that in Figure 6a, due to the smaller headways in Shanghai. The training results indicate that the proposed hierarchical continual RL framework can be applied in changing environments where the reward function is changed according to traffic conditions and passenger preferences.

In the proposed continual RL framework, some rule-based car-following models can also provide experiences to simulate driving data collection in the cloud for driving skill abstracting and car-following model adaptation. Thus, the IDM is further introduced into the framework as an expert. Three skill extractors train in parallel with knowledge from the IDM and DRL-based car-following and free-driving models. The whole training process is shown in Figure 8a. Both the IDM and the DRL-based car-following model can provide driving data for car-following skill learning. Particularly, the IDM is collision-free with consideration of safe distance (Hoel et al., 2019). Thus, emergency braking seldom occurs, and the convergence accelerates, compared to Figure 7a. As shown in Figure 8b,c, the chief aims to reduce speed on rough segments and produce a smooth

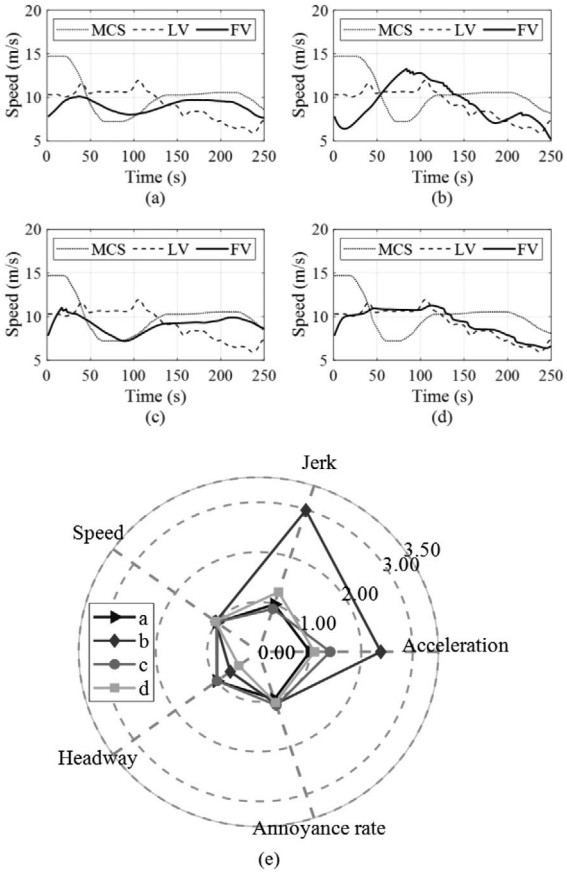

FIGURE 9 Speed profiles and driving performance of the (a) chief, (b) intelligent driver model (IDM) expert, (c) DRL-based free driving expert, and (d) DRL-based car-following expert in the CDDPG model at 330 episodes.

speed profile for ride comfort. In Figure 8c,d, the speed is initially below the MCS, and the chief accelerates to reduce headway for efficiency. Then, with preview information of the oncoming road in the state, the chief adjusts the speed in advance for comfortable driving on the rough segment. Finally, the chief learns to accelerate again, and the speed is lower than the MCS. The chief can obtain comprehensive knowledge with more workers and experts. Meanwhile, this suggests that the proposed framework can optimize the driving comfort of CAVs for passengers.

The superior training and driving performances of the chief, shown in Figures 7 and 8, benefit from the utilization of prevailing free-driving and car-following models (experts) and skill extractors (workers). To show the detailed improvements of the chief, speed profiles and driving performances of the chief and the experts are compared in the scenario with the sampled road and leading vehicles in Figure 9. The CDDPG model was trained with an untrained reward function and demonstrations from the DRL and IDM experts. In Figure 9a, the chief can generate a smooth speed profile, reduce speed on the rough segment, and increase speed on even segments. Specifically, most speed values of the chief are below the MCS to improve ride comfort. The IDM and the DRL-based car-following model prefer to drive with high speeds and small headways on the rough segment in Figure 9b,d, while the DRL-based free-driving model performs comfortable driving behavior in Figure 9c.

In Figure 9e, the absolute values of longitudinal jerk and acceleration are used to represent longitudinal comfort, the annoyance rate (AR) represents the proportion of passengers who cannot bear vehicle vibration, and speed and headway are used to evaluate driving efficiency. Generally, weighted root-mean-square acceleration is a metric for objective ride comfort evaluation (ISO, 1997). However, even the same vibration can cause different feelings in passengers due to their diverse requirements and physical qualities. Thus, the AR, which considers uncertainties of distinct feelings, modifies the objective evaluation results. The detailed calculation method was introduced in our previous work (J. Chen et al., 2023). The average value of each metric during this 250-s driving is further used to evaluate driving performance at a global level. To show all the metrics at a similar magnitude, we regard the driving performance of the chief as a baseline, which is set as 1, and the metrics of other expert models are divided by the chief's for comparison. The experimental results show that the chief can obtain a relatively small longitudinal jerk and the smallest longitudinal acceleration and AR among these models; meanwhile, the headway is relatively large for speed adjustment and driving safety. The results mean that the CDDPG-based car-following model can outperform conventional DDPG-based car-following models and the IDM in safety and ride comfort.

In the proposed hierarchical continual RL framework, the workers' policies can reflect the learned driving skills during the update process, where these driving skills are defined as motion planning strategies for specific driving objective preferences. In this study, a flexible boundary is designed for driving skills, meaning that the skill boundary is learned by cooperative workers. The main reason is that the proposed hierarchical continual RL framework may be used to learn various driving tasks and satisfy different passenger requirements. If the skill boundary is determined anthropically, the scalability of the proposed framework will be limited.

Figure 10 shows the testing speed profiles and movements of workers in Figure 8 at 330 episodes. As shown in Figure 10, the three workers have different driving performances. Specifically, Worker 1 learns to follow the leading vehicle closely but increases clearance space on rough segments. Worker 2 learns car following on even segments and comfortable driving on rough segments. Compared

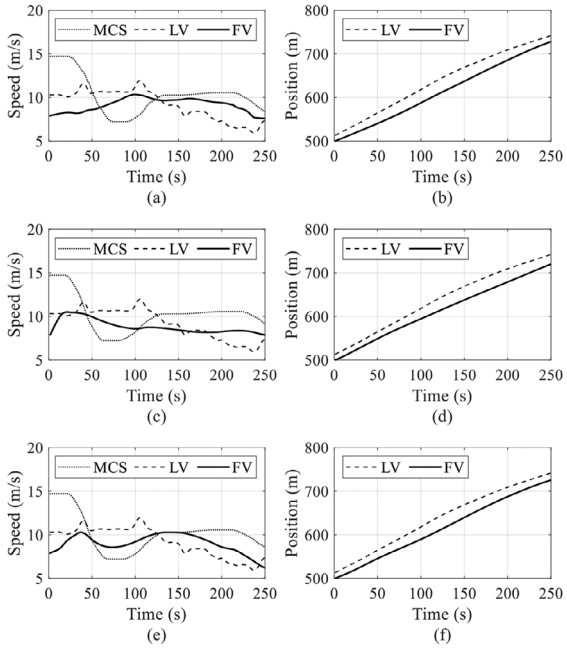

FIGURE 10 Speed profiles and movements of the (a) and (b) Worker 1, (c) and (d) Worker 2, and (e) and (f) Worker 3 at 330 episodes.

to Worker 2, Worker 3 further learns to increase driving speed to chase the leading vehicle when the pavement quality improves. Thus, the three workers have different driving preferences on rough and even segments when following a leading vehicle. However, most of them choose to increase speed to follow the leading vehicle, reduce speed to ensure ride comfort on rough segments, and then increase speeds to chase the leading vehicle when the pavement quality becomes better. In this way, the average action of the worker and chief used in this study can consider the weights and priorities of efficiency and ride comfort under different situations. Since the workers' driving skills are tested separately in Figure 10 to show driving performance clearly, their observations are distinct to the chief in Figure 8e at each timestep, which uses the average action of the chief and workers to conduct. They still can show what driving skills have been learned by the workers.

To evaluate performance degradation after model adaptation, a metric called degradation rate *DR* is devised based on the forgetting measures used in continual learning studies (Benkő, 2024) which is calculated using the scores of driving performances in the expert $Socre_{expert}$ and the models $Socre_m$ as follows. Since the CDDPG and EDDPG models are trained with experts, the degradation rates are calculated for the CDDPG and EDDPG models. To add more comparisons, the degradation rates of the DDPG model are also calculated to show the reward feature deviations from experts. The scores are the average reward features obtained in testing, which can be either postive

TABLE 3 Degradation rates (DRs) of models under different pavement conditions.

Pavement	Model	Score	DR
Even segments	Continual deep deterministic policy gradient (CDDPG)	−0.1975	0.17
	Ensemble DDPG (EDDPG)	−0.3000	0.78
	DDPG	−0.1807	0.07
	DDPG CF	−0.1690	–
Rough segments	CDDPG	−0.1618	1.32
	EDDPG	−0.0395	−0.43
	DDPG	−0.3963	4.69
	DDPG FD	−0.0696	–

or negative. Thus, the degradation rate is definded as

$$DR = \frac{Socre_{expert} - Socre_m}{\left|Socre_{expert}\right|} \tag{23}$$

The degradation rates of the models in the testing case are listed in Table 3. The DDPG-based free-driving model and car-following model are abbreviated as DDPG FD and DDPG CF, respectively. In the calculation of degradation rates, the efficiency feature of the DDPG CF expert and the comfort feature of the DDPG FD expert are used as baselines for car following on even and rough segments, respectively. The score is calculated using the efficiency and comfort features in the manuscript. Since a safeguard is set to avoid rear-end collision, the main objective of driving on even and rough segments is efficiency and comfort, respectively. As shown in Table 3, the CDDPG model has better knowledge inheritance capability in efficient and comfortable driving, compared to the conventional EDDPG and DDPG models. On even segments, the main reason for little degradation in efficient driving is that the CDDPG model needs to reduce speeds in advance to prepare for driving on rough segments, which is acceptable at the global level. On rough segments, the negative value of degradation rate represents that the EDDPG model learns better comfortable driving behavior than the DDPG FD expert as shown in Figure 6. Since the CDDPG model also considers the driving efficiency from the global perspective, the driving speeds on rough segments are higher than those of the EDDPG model and DDPG FD expert and lower than the DDPG model. Thus, the degradation rate of CDDPG on rough segments is between those of the EDDPG and DDPG models.

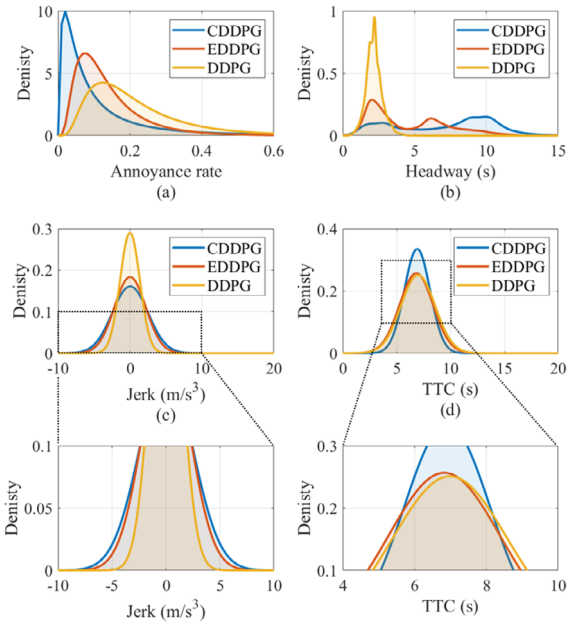

FIGURE 11 Driving performance on the testing dataset. TTC, time-to-collision.

5.4 | Generalization of the proposed framework

5.4.1 | Generalization to untrained leading vehicles

To show the driving performance after adaptation in untrained driving scenarios, we first use a fixed rough pavement and changing leading vehicles to establish an autonomous driving testing environment. The AR, longitudinal jerk, headway, and TTC are used to evaluate driving performances of different models in vertical comfort, longitudinal comfort, efficiency, and safety, respectively. Figure 11 uses lognormal, normal, and multi-kernel distribution functions to fit the distribution of each metric. As shown in Figure 11a, the CDDPG model has a smaller AR, meaning that the CDDPG model provides more comfortable autonomous driving that more passengers can accept. Figure 11b illustrates that the CDDPG and EDDPG models have two kernels in the headway distribution, while the DDPG model only has one kernel. This suggests that the CDDPG and EDDPG models can perform car following and comfortable free-driving, while the CDDPG model can adjust speed more effectively for ride comfort improvements according to changing pavement and traffic conditions.

Figure 11c demonstrates that the CDDPG model has larger absolute longitudinal jerks to adjust speed in time. Figure 11d shows the distribution of the TTC values from 0 to 9 s. Different studies proposed distinct TTC thresholds, such as 1 to 9 s, 0 to 4 s (Zhu et al., 2020), and 1.5 to 5 s (Vogel, 2003). Thus, we choose the TTC values in the range from 0 to 9 s to analyze. The CDDPG model evidently has the fewest TTC values below 4 s among the three models, indicating that the CDDPG model is relatively safe.

TABLE 4 Evaluation of driving performance with metrics.

Model	TIT (s^2)	AR (%)	Jerk (m/s^3)	Headway (s)
CDDPG	3.72	12.54	0.74	7.04
EDDPG	4.78	14.24	0.93	4.49
DDPG	5.64	21.58	0.40	2.18

Abbreviations: AR, annoyance rate; TIT, time-integrated time-to-collision.

It is reasonable that there are similar distributions among the models. For longitudinal jerk, the CDDPG model adjusts speed according to leading vehicle movements and pavement conditions with a relatively large absolute jerk value. However, the absolute jerk values of the CDDPG model are not much larger than those of the EDDPG model because the CDDPG model has learned speed planning on different pavement conditions. In contrast, the DDPG model, which learns car-following behavior, only uses small absolute jerk values to keep a suitable headway and clearance space.

For the TTC, the CDDPG model can effectively reduce unsafe actions (the number of the TTCs smaller than 4 s) based on expert knowledge and worker collaborative learning. However, without the staged training strategy, it is challenging for several agents to learn the expected driving performance simultaneously. The sharp changes of longitudinal acceleration in the EDDPG model make it difficult to keep safe when approaching its leading vehicles. As a result, the small TTC values in the EDDPG model are the most among these models.

It is noteworthy that the TTC is an instantaneous metric, which is calculated at each timestep and is unable to showcase safety performance during the whole driving process. Thus, we further leverage time-integrated TTC (TIT) to evaluate driving safety with the duration of TTC values below a certain threshold (Xu et al., 2021). The TIT can be calculated with the equations in the following when $\forall 0 \le TTC_{i,t} \le TTC^*$.

$$\sum_i TIT_i = \sum_i \sum_t \left[TTC^* - TTC_{i,t}\right] * \tau_{sc} \qquad (24)$$

where TTC_i is the TTC value of the ith event using the i^{th} leading vehicle data in the testing set; TTC^* is the TTC threshold value used for TIT caculation, set as 4 s (Xu et al., 2021); $TTC_{i,t}$ is the TTC value at timestep t in event i; and τ_{sc} is the time interval, equal to ΔT and set as 0.1 s.

Table 4 shows the average TIT, AR, jerk, and headway values of the models in testing. Compared to the DDPG model, both the CDDPG and EDDPG models can

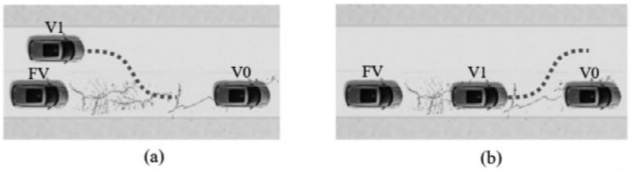

FIGURE 12 Schematic diagrams of (a) cut-in and (b) cut-out scenarios.

effectively decrease the duration time of unsafe driving behavior. Specifically, the CDDPG model performs better in safety improvements. The CDDPG model can decrease the TIT by 22.18 % and 34.04 %, compared to the EDDPG and the DDPG models, respectively. Since the CDDPG model instructs CAVs to slow down on rough segments, CAVs will keep larger clearance space, compared to the EDDPG and the DDPG models, and CAVs' driving speeds may be lower than their leading vehicles. Though CAVs will increase speed to catch up with their leading vehicles, there are a great number of large TTC values of the CDDPG model caused by deceleration for comfortable driving on rough segments. Thus, there is an obvious improvement in driving safety for the CDDPG model, but this also causes relatively large headways and longitudinal jerks. For vertical ride comfort, the CDDPG model can reduce the AR by 9.04%, compared to the DDPG model, meaning that the number of passengers feeling uncomfortable can be significantly decreased by 9.04%.

5.4.2 | Generalization to untrained cut-in and cut-out scenarios

In training, the CDDPG-based car-following model focuses on following a certain vehicle. Intuitively, the model can learn car-following behavior after massive iterations. However, when a vehicle cuts in or cuts out, the change of the leading vehicle may lead to an uncomfortable or unsafe acceleration selection with a sharp change in acceleration or a small TTC value. Thus, to show the generalization of the CDDPG-based car-following model, driving performances in cut-in and cut-out scenarios are tested.

The testing includes the following scenarios: (1) a vehicle cutting in to be a new leading vehicle of the CAV and (2) a leading vehicle cutting out to make a farther vehicle in front of the CAV to be a new leading vehicle. To establish the cut-in and cut-out scenarios, three vehicles are used herein. As shown in Figure 12, the V0 represents the vehicle that is farther from the CAV, the V1 stands for the vehicle to perform cut-in or cut-out operation, the LV is the leading vehicle that switches between V0 and V1, and the FV is the CAV, meaning a following vehicle. In the cut-in scenario, to provide a space for V1 to cut in, the V0 accelerates to the maximum speed at 65 timesteps. The leading vehicle changes from V0 to V1 when V1 changes to the lane on the right in the cut-in scenario of Figure 12a. In the cut-out scenario of Figure 12b, the leading vehicle changes from V1 to V0 when V1 changes to the left lane.

In this way, cut-in and cut-out scenarios are regarded as the changes in leading vehicles. The simulation experiment is conducted in Simulation of Urban MObility (SUMO). In this study, intelligent speed planning in an open road environment is investigated, while the lane change models in SUMO are used. In the simulation, the width of a lane is set as 3.5 m, and the maximum speed is 60 km/h. It is assumed that all the vehicles drive along the road centerline, and the positions are recorded at every 20 timesteps. It is noteworthy that even if the lane-changing vehicle is the CAV, the speed planning process is similar because the CDDPG model observes the leading vehicle and pavement conditions to select longitudinal acceleration. Since the vehicles are assumed to drive along the centerline of roads, the vehicles in a lane have the same lateral location value, which generates overlap areas in the figure. Thus, in Figure 13, we set the lane width as 3.5 m and biased values of 0.7 m for the locations of V0 and V1, while the lateral location of FV is the road centerline to show the change of locations clearly.

In the cut-in scenario, vehicle V0 drives in front of the CAV, and V1 should find a chance to cut in without collision. As shown in Figure 13a–d, for the CDDPG and EDDPG models, the speed is relatively low for ride comfort, and it is easy for a vehicle to cut in. In contrast, a vehicle needs to wait a long time for a cut-in chance in the DDPG model due to the small clearance space between the CAV and V0. Figure 14 demonstrates the absolute jerk and the numbers of the TTC values of the models. Figure 14a illustrates that the CDDPG and EDDPG models have lower maximum absolute jerk, meaning that the proposed hierarchical continual RL framework can effectively improve ride comfort and generalize to untrained scenarios. In Figure 13a-1,b-1, the driving speed of the CDDPG model is higher than the EDDPG model after 100 s, indicating that the CDDPG model can adjust speed in a changing environment and learn an efficient driving mode. In Figure 14b, the numbers of the TTC values higher than 9 s are used to evaluate driving safety. Specifically, the actions with a TTC value higher than 9 s suggest that the driving behavior is relatively or absolutely safe, which includes the number of infinite TTC values. It is obvious that the CDDPG model can improve ride comfort and efficiency without sacrificing driving safety.

In the cut-out scenario, it is easy for V1 to cut out and drive into the left lane. As shown in Figure 13e–h, the CDDPG, EDDPG, and DDPG models can follow the

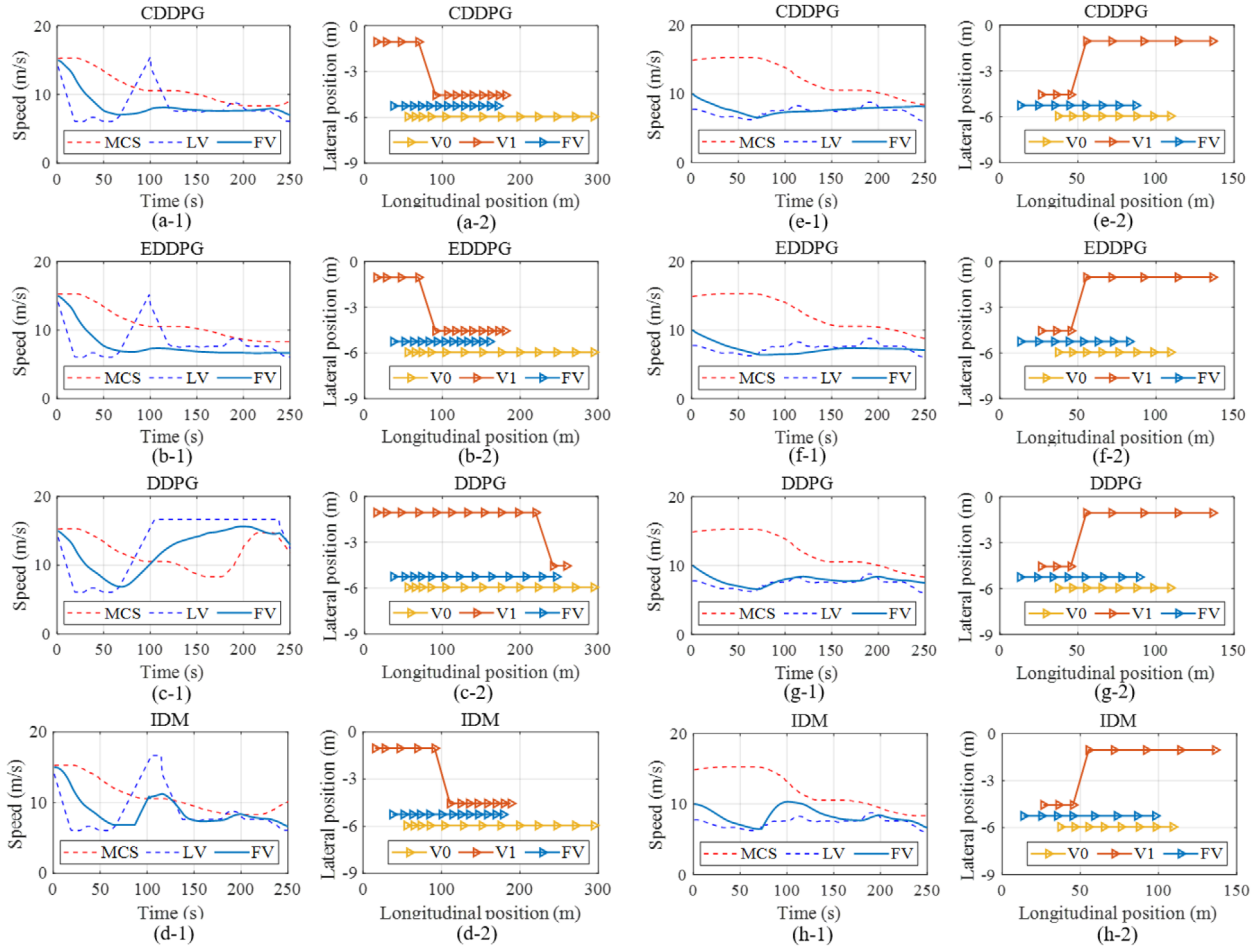

FIGURE 13 Speed profiles and movements of vehicles in (a), (b), (c), and (d) cut-in and (e), (f), (g), and (h) cut-out scenarios.

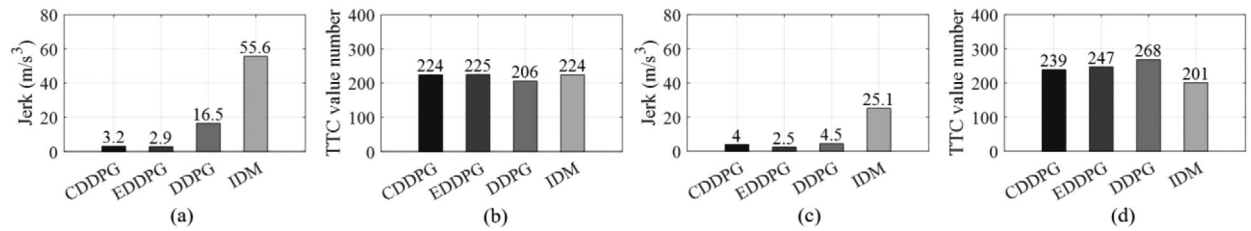

FIGURE 14 Jerk and time-to-collision (TTC) values of vehicles in (a) and (b) cut-in and (c) and (d) cut-out scenarios.

leading vehicle smoothly during the whole driving process, while there is a sharp acceleration when the leading vehicle changes in the IDM. The maximum absolute longitudinal jerks of the CDDPG, EDDPG, and DDPG models are smaller, compared to the IDM, as shown in Figure 14c. Meanwhile, in Figure 14d, the smooth speed profiles of the CDDPG, EDDPG, and DDPG models also result in higher TTC values, compared to the IDM, meaning that they are safer than the IDM.

Based on these cases, the proposed hierarchical continual RL framework and skill-abstracting continual RL approach can thus be demonstrated. In this framework, the prevailing motion planning models can be used to provide references and relatively good beginnings for driving skill abstracting. Parallel skill extractors in the cloud collaborate to explore and exploit distinct areas of the action space to learn solutions from different perspectives. With comprehensive knowledge, a CAV can continuously update its learning-based car-following model to obtain strong generalization capability and solve complex driving tasks successfully. By applying the proposed framework in the DDPG algorithm, the CDDPG-based car-following model

can learn expected driving behavior within limited iterations and adjust longitudinal acceleration according to traffic and pavement conditions. Compared to the baseline models, the CDDPG model has superiorities in ride comfort and driving safety. Even in untrained scenarios, such as cut-in and cut-out scenarios, the CDDPG model can adjust speed smoothly without sacrificing driving safety.

6 | CONCLUSION

This study proposes a hierarchical continual RL framework for safe, efficient, and comfortable autonomous driving that leverages a novel continual RL approach and vehicle-cloud collaboration. The proposed continual RL apporach integrates prevailing motion planning models with parallel RL agents. Parallel DRL agents engage in collaborative learning to abstract driving skills from massive demonstrations and contribute to rapid and stable remote update of CAVs' motion planning models in the cloud. The efficacy of the proposed framework is evaluated based on real-world pavement and traffic data. Experimental testing demonstrates that the proposed framework exhibits an expanded exploration space within limited iterations and scalability across varying numbers of driving skills. Results from the experimentation on untrained driving scenarios reveal noteworthy improvements facilitated by the CDDPG model. Specifically, the CDDPG model showcases a commendable 34.04% reduction in the duration of potentially hazardous driving behavior and a notable enhancement of vertical comfort by 9.04%, when compared to a classical DDPG baseline. These results underscore the robust generalization capabilities of the proposed framework and highlight the superiority of continual RL in the motion planning of CAVs.

In future research, the intelligent coordination of CAVs and traffic infrastructure will be investigated. In a traffic system, driving safety, efficiency, and comfort in road segments and intersections are both important. Thus, based on existing traffic signal control (Kim et al., 2023; Zhang et al., 2022), how to design intelligent traffic signal control to coordinate with CAVs in a vehicle–road–cloud integration system will be considered. Moreover, other algorithms and approaches contribute to rapid adaptation, such as the neural dynamic classification algorithm (Rafiei & Adeli, 2017), dynamic ensemble learning algorithm (Alam et al., 2020), finite element machine (Pereira et al., 2020), and self-supervised learning (Rafiei et al., 2022, 2024) will be further investigated.

ACKNOWLEDGMENTS

This work was sponsored by the National Natural Science Foundation of China under Grant 52472352, in part by the Shanghai Rising-Star Program under Grant 24QA2709600, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100, and in part by the Fundamental Research Funds for the Central Universities under Grant 22120230311.

REFERENCES

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, *32*, 8675–8690.

Benkő, B. (2024). Example forgetting and rehearsal in continual learning. *Pattern Recognition Letters*, *179*, 65–72.

Chen, J., Zhao, C., Jiang, S., Zhang, X., Li, Z., & Du, Y. (2023). Safe, efficient, and comfortable autonomous driving based on cooperative vehicle infrastructure system. *International Journal of Environmental Research and Public Health*, *20*, 893.

Chen, S., Dong, J., Ha, P., Li, Y., & Labi, S. (2021). Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles. *Computer-Aided Civil and Infrastructure Engineering*, *36*, 838–857.

Chen, X., Tiu, P., Han, X., Chen, J., Wu, Y., Zheng, X., & Zhu, M. (2024). *Continual learning for adaptable car-following in dynamic traffic environments*. arXiv preprint arXiv:2407.14247. https://arxiv.org/abs/2407.14247

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, *14*, 241–258.

Du, Y., Chen, J., Zhao, C., Liao, F., & Zhu, M. (2023). A hierarchical framework for improving ride comfort of autonomous vehicles via deep reinforcement learning with external knowledge. *Computer-Aided Civil and Infrastructure Engineering*, *38*(8), 1059–1078.

Du, Y., Chen, J., Zhao, C., Liu, C., Liao, F., & Chan, C.-Y. (2022). Comfortable and energy-efficient speed control of autonomous vehicles on rough pavements using deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, *134*, 103489.

Feng, S., Sun, H., Yan, X., Zhu, H., Zou, Z., Shen, S., & Liu, H. X. (2023). Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, *615*, 620–627.

Gao, B., Liu, J., Zou, H., Chen, J., He, L., & Li, K. (2024). Vehicle-road-cloud collaborative perception framework and key technologies: A review. *IEEE Transactions on Intelligent Transportation Systems*, *25*, 19295–19318.

Genser, A., Spielhofer, R., Nitsche, P., & Kouvelas, A. (2022). Ride comfort assessment for automated vehicles utilizing a road surface model and Monte Carlo simulations. *Computer-Aided Civil and Infrastructure Engineering*, *37*, 1316–1334.

Hoel, C.-J., Driggs-Campbell, K., Wolff, K., Laine, L., & Kochenderfer, M. J. (2019). Combining planning and deep reinforcement learning in tactical decision making for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, *5*, 294–305.

Huang, Z., Sheng, Z., Ma, C., & Chen, S. (2024). Human as AI mentor: Enhanced human-in-the-loop reinforcement learning for safe and efficient autonomous driving. *Communications in Transportation Research*, *4*, 100127.

International Standards Organization. (1997). *2631-1: Mechanical vibration and shock-evaluation of human exposure to whole-body vibration-Part 1: General requirements*. ISO 2631-1:1997(en), ISO.

Ji, Y., Ni, L., Zhao, C., Lei, C., Du, Y., & Wang, W. (2023). TriPField: A 3D potential field model and its applications to local path planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *24*(3), 3541–3554.

Kang, K., Park, N., Park, J., & Abdel-Aty, M. (2024). Deep Q-network learning-based active speed management under autonomous driving environments. *Computer-Aided Civil and Infrastructure Engineering*, *39*(21), 225–3242.

Khetarpal, K., Riemer, M., Rish, I., & Precup, D. (2022). Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, *75*, 1401–1476.

Kim, G., Kang, J., & Sohn, K. (2023). A meta–reinforcement learning algorithm for traffic signal control to automatically switch different reward functions according to the saturation level of traffic flows. *Computer-Aided Civil and Infrastructure Engineering*, *38*, 779–798.

Li, Q., Peng, Z., Feng, L., Zhang, Q., Xue, Z., & Zhou, B. (2022). Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*, 3461–3475.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). *Continuous control with deep reinforcement learning*. arXiv preprint arXiv:1509.02971. https://arxiv.org/abs/1509.02971

Mao, Z., Liu, Y., & Qu, X. (2024). Integrating big data analytics in autonomous driving: An unsupervised hierarchical reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, *162*, 104606.

Nayyar, R. K., & Srivastava, S. (2024). *Autonomous option invention for continual hierarchical reinforcement learning and planning*. arXiv preprint arXiv:2412.16395. https://arxiv.org/abs/2412.16395

Pan, C., Yang, X., Wang, H., Wei, W., & Li, T. (2024). *Hierarchical continual reinforcement learning via large language model*. arXiv preprint arXiv:2401.15098. https://arxiv.org/abs/2401.15098

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J. P., & Adeli, H. (2020). FEMa: A finite element machine for fast learning. *Neural Computing and Applications*, *32*, 6393–6404.

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*, 3074–3083.

Rafiei, M. H., Gauthier, L. V., Adeli, H., & Takabi, D. (2022). Self-supervised learning for electroencephalography. *IEEE Transactions on Neural Networks and Learning Systems*, *35*, 1457–1471.

Rafiei, M. H., Gauthier, L. V., Adeli, H., & Takabi, D. (2024). Self-supervised learning for near-wild cognitive workload estimation. *Journal of Medical Systems*, *48*, 107.

Sheng, Z., Huang, Z., & Chen, S. (2024). Traffic expertise meets residual RL: Knowledge-informed model-based residual reinforcement learning for CAV trajectory control. *Communications in Transportation Research*, *4*, 100142.

Shi, H., Zhou, Y., Wang, X., Fu, S., Gong, S., & Ran, B. (2022). A deep reinforcement learning-based distributed connected automated vehicle control under communication failure. *Computer-Aided Civil and Infrastructure Engineering*, *37*, 2033–2051.

Sucar, L. E. (2007). Parallel Markov decision processes. *Advances in probabilistic graphical models*, Vol. 213, 295–309.

Sun, W., Zhang, F., Liu, W., & He, Q. (2024). Optimal control of connected autonomous vehicles in a mixed traffic corridor. *IEEE Transactions on Intelligent Transportation Systems*, *25*(5), 4206–4218.

Vogel, K. (2003). A comparison of headway and time to collision as safety indicators. *Accident Analysis & Prevention*, *35*, 427–433.

Wang, Y., Hou, S., & Wang, X. (2021). Reinforcement learning-based bird-view automated vehicle control to avoid crossing traffic. *Computer-Aided Civil and Infrastructure Engineering*, *36*, 890–901.

Wei, D., Xing, J., Yang, S., Lu, Y., & Huang, Y. (2023). Continual reinforcement learning for autonomous driving with application on velocity control under various environment. *2023 7th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, Changsha, China (pp. 1–8).

Weng, Z., Liu, C., Du, Y., Wu, D., & Leng, Z. (2024). Integrating spatial and channel attention mechanisms with domain knowledge in convolutional neural networks for friction coefficient prediction. *Computer-Aided Civil and Infrastructure Engineering*. Advance online publication. https://doi.org/10.1111/mice.13391

Xu, X., Wang, X., Wu, X., Hassanin, O., & Chai, C. (2021). Calibration and evaluation of the responsibility-sensitive safety model of autonomous car-following maneuvers using naturalistic driving study data. *Transportation Research Part C: Emerging Technologies*, *123*, 102988.

Ye, Y., Zhang, X., & Sun, J. (2019). Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment. *Transportation Research Part C: Emerging Technologies*, *107*, 155–170.

Yi, R., Zhou, Y., Ou, J., Wang, X., Ding, F., & Nie, Q. (2023). A 2D-connected automated vehicle car-following control algorithm. *Computer-Aided Civil and Infrastructure Engineering*, *38*, 2560–2575.

Zhang, Z., Guo, M., Fu, D., Mo, L., & Zhang, S. (2022). Traffic signal optimization for partially observable traffic system and low penetration rate of connected vehicles. *Computer-Aided Civil and Infrastructure Engineering*, *37*, 2070–2092.

Zhu, M., Wang, X., & Wang, X. (2016). Car-following headways in different driving situations: A naturalistic driving study. *CICTP 2016*, Shanghai, China (pp. 1419–1428).

Zhu, M., Wang, X., & Wang, Y. (2018). Human-like autonomous car-following model with deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, *97*, 348–368.

Zhu, M., Wang, Y., Pu, Z., Hu, J., Wang, X., & Ke, R. (2020). Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. *Transportation Research Part C: Emerging Technologies*, *117*, 102662.

Zhuang, H., Fang, D., Tong, K., Liu, Y., Zeng, Z., Zhou, X., & Chen, C. (2025). Online analytic exemplar-free continual learning with large models for imbalanced autonomous driving task. *IEEE Transactions on Vehicular Technology*, *74*(2), 1949–1958.

How to cite this article: Chen, J., Zhao, C., Gao, K., Ji, Y., & Du, Y. (2025). Skill-abstracting continual reinforcement learning for safe, efficient, and comfortable autonomous driving through vehicle–cloud collaboration. *Computer-Aided Civil and Infrastructure Engineering*, 1–21. https://doi.org/10.1111/mice.13503