nature machine intelligence

Article

https://doi.org/10.1038/s42256-023-00610-y

Continuous improvement of self-driving cars using dynamic confidence-aware reinforcement learning

Received: 14 May 2022

Accepted: 4 January 2023

Published online: 23 February 2023

Check for updates

Zhong Cao[1], Kun Jiang[1], Weitao Zhou[1], Shaobing Xu[1], Huei Peng[2] & Diange Yang[1]

Today's self-driving vehicles have achieved impressive driving capabilities, but still suffer from uncertain performance in long-tail cases. Training a reinforcement-learning-based self-driving algorithm with more data does not always lead to better performance, which is a safety concern. Here we present a dynamic confidence-aware reinforcement learning (DCARL) technology for guaranteed continuous improvement. Continuously improving means that more training always improves or maintains its current performance. Our technique enables performance improvement using the data collected during driving, and does not need a lengthy pre-training phase. We evaluate the proposed technology both using simulations and on an experimental vehicle. The results show that the proposed DCARL method enables continuous improvement in various cases, and, in the meantime, matches or outperforms the default self-driving policy at any stage. This technology was demonstrated and evaluated on the vehicle at the 2022 Beijing Winter Olympic Games.

Self-driving vehicles are being deployed in many parts of the world. However, their performance in unseen 'long-tail' cases is still a concern. Many data-driven methods such as reinforcement learning (RL) provide a potential way to learn from collected data and update the driving policy continuously[1]. RL has proven its worth in several application domains, for example, chess, Go[2,3] and video games[4]. Some recent research has started to train an RL-based self-driving policy in specific scenarios[5–8]. Nevertheless, few automotive companies are ready to deploy this technology on their production vehicles[9]. The two main concerns are: a data-driven agent usually requires a very long training time and the performance after training is not guaranteed[10]; and the trained agent may not outperform existing rule- or model-based policies[11] for a previously unseen scenario. There have been many efforts on improving training performance[12] and training efficiency[13], adding external safeguards[14,15] and collecting more data for training[16], but none fundamentally solved the crucial concerns of 'uncertain performance for long-tail cases', before the all-encompassing training dataset appeared.

In this Article, we call for a paradigm shift to train an RL agent that can be applied at any training stage with a guaranteed lower-bounded performance, and its performance is guaranteed to improve continuously from that lower bound with more data (Fig. 1a). The lower-bound performance comes from the default self-driving algorithms, which handle most driving cases, but should be further improved for the remaining long-tail cases[17–19]. This technique directly works with this given self-driving algorithm, and continuously receives data for improvement during driving (Fig. 1b). We refer to the proposed technique as the dynamic confidence-aware reinforcement learning (DCARL) agent, and the framework is shown in Fig. 1c. In this framework, the most important module is to train the policies' dynamic confidence value, which monotonically increases with more data. The dynamic confidence value is calculated based on not only the expected performance but also the confidence to achieve this performance. The DCARL agent has the following four key features, which may solve the previously mentioned concerns.

[1]School of Vehicle and Mobility, Tsinghua University, Beijing, China. [2]Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA.
e-mail: ydg@tsinghua.edu.cn

https://doi.org/10.1038/s42256-023-00610-y

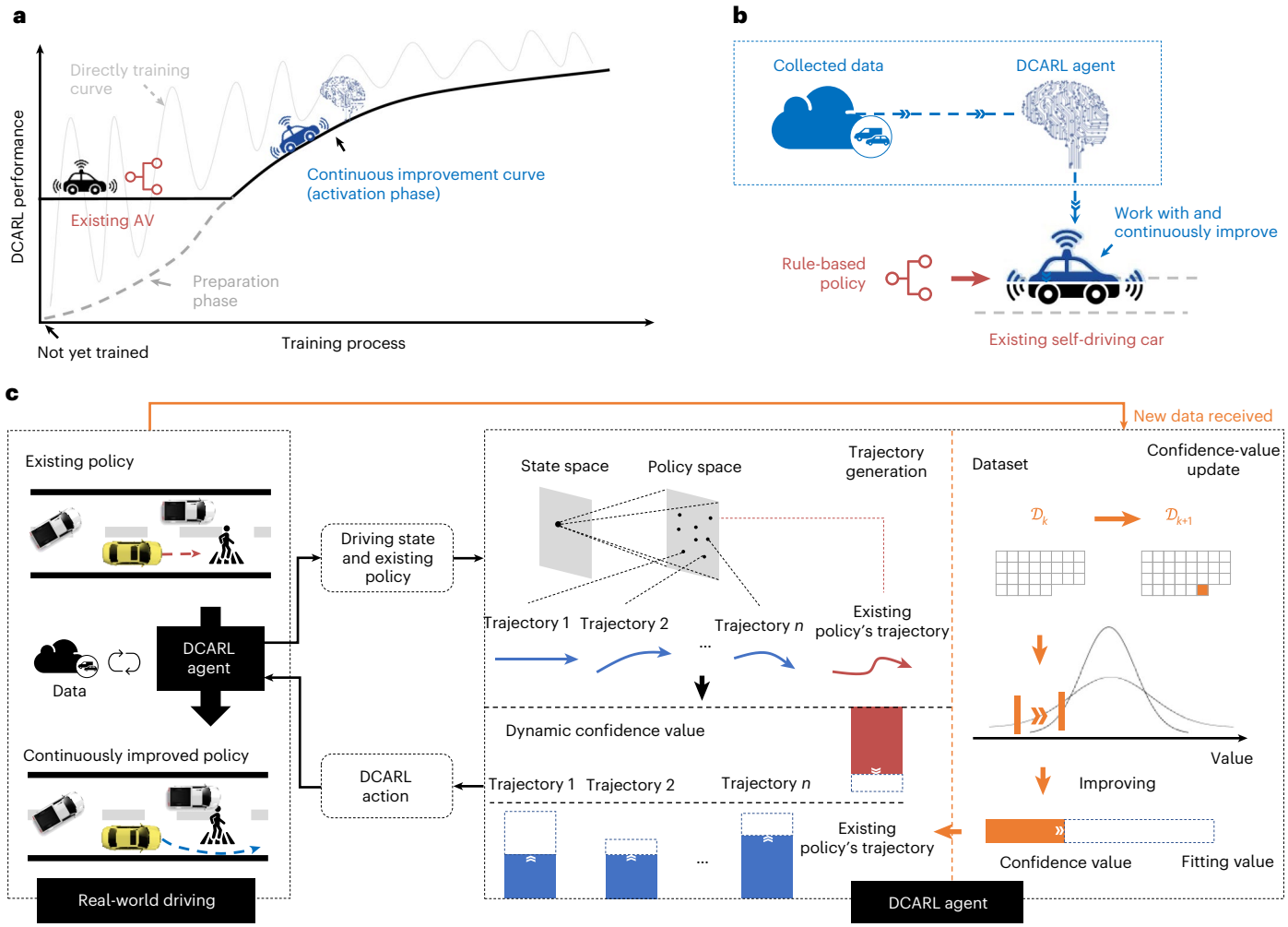

Fig. 1 | Continuous improvement concept and framework of the DCARL agent. a, Continuous improvement starting from the existing policy's performance and guaranteed improvement with more training. AV, autonomous vehicle. **b**, The DCARL agent. It works with an existing self-driving car and continuously collects data for policy improvement. **c**, The DCARL agent framework. The agent consists of three modules: trajectory generation, decision-making and confidence-value updating. In the confidence-value-updating module, the confidence value of the existing policy decreases, while that of other policies increases with more data. Credit: ego vehicle and social vehicle icons, macrovector / Freepik.com.

(1) No lengthy pre-training is required. Our DCARL agent can directly work with most existing self-driving algorithms that have the same state and action space as the baseline. For non-RL policies, a converter is required to align the state and action spaces. When training is insufficient, the agent can use the original driving policy (see the 'Not yet trained' point in Fig. 1a).
(2) Guaranteed improvement with more training data. Our DCARL agent can continuously improve with more training data under acceptable real-world uncertainty, including a fixed feedback error or a standard noise distribution (refer to the continuous improvement curve in Fig. 1a).
(3) Performance guaranteed by the existing self-driving algorithm. Many existing self-driving cars can drive safely in many scenarios but when newly collected data are used to train further, performance does not always improve[20]. When we combine the DCARL agent with these existing self-driving algorithms, the baseline performance is guaranteed by the existing driving policy (see the preparation phase in Fig. 1a).
(4) A purely data-driven self-driving algorithm eventually. With more training, our agent gains better performance than the original policy in more cases, and may eventually take over most scenarios to become a totally data-driven policy.

The classical RL framework in the literature also considers safety and aims at performance improvement, but in practice, more training does not always lead to better or safer performance. Furthermore, the assumptions and safety definitions may not target the proposed requirements for continuously improving the existing self-driving cars. For example, safe policy improvement[21] aims to avoid risky policy updating, but requires extensive data. Offline RL[22] can train the agent based on the limited dataset, but may not have a performance lower bound. In safe RL research[23], a widely used idea is to solve the constrained Markov decision process problem to make each updated policy satisfy the given constraints[24]. These methods usually need each policy's probability of violating constraints, but insufficient data or inaccurate models[25,26] may cause large errors, possibly leading to the failures in these methods. Introducing an expert or safeguard policy is another idea for safety. For example, apprenticeship learning[27,28] and DAgger[29] aim to train the policy to approach an expert policy. Safe DAgger[30] and Selective DAgger[31] will directly call the expert policy when the policy is very different from the expert. These methods aim to imitate an expert policy, but our agent can and is even encouraged to deviate from the baseline. Shield-RL[32] and legal safety policy[14] may use the manually designed principles to judge whether the current policy is safe, and call for the

https://doi.org/10.1038/s42256-023-00610-y

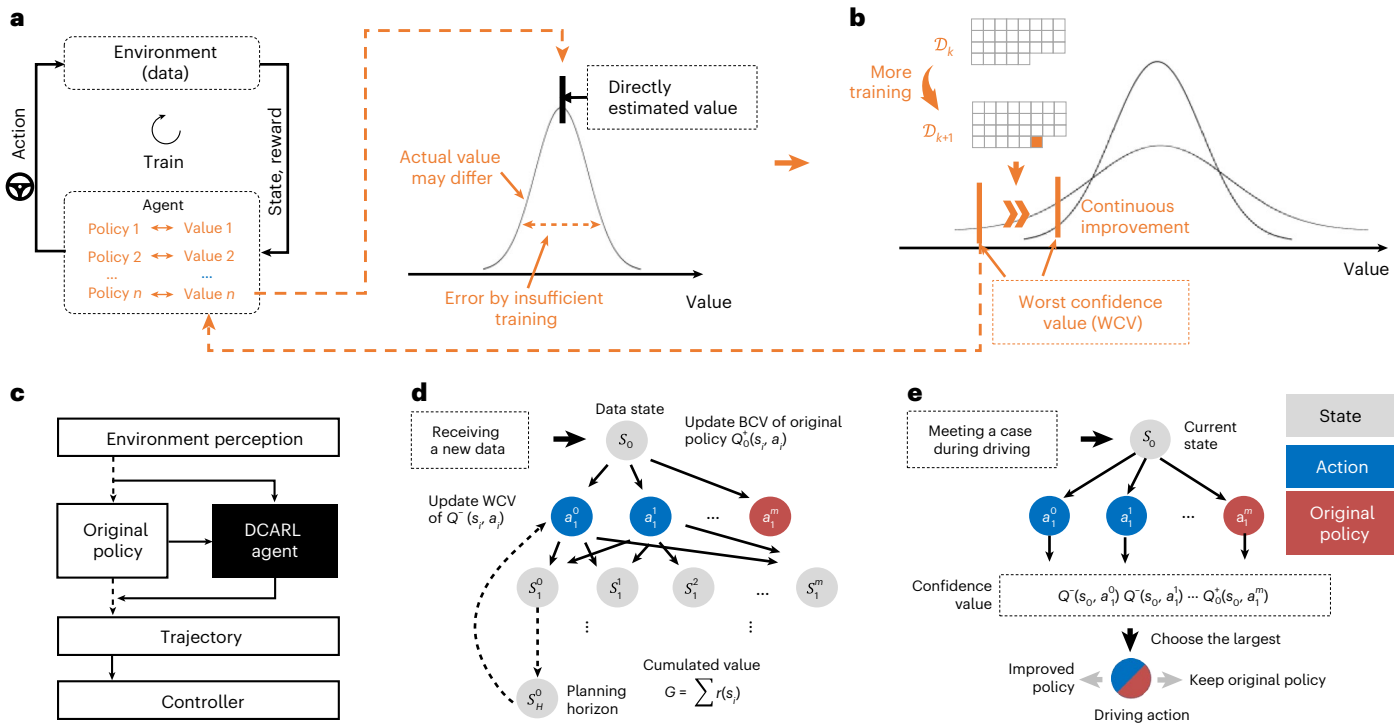

Fig. 2 | Continuous improvement process of the DCARL agent. a, Error of the classical value-based RL framework due to insufficient training. **b**, WCV used in the DCARL agent to represent the estimated performance and its confidence. More training continuously improves the WCV of the DCARL agent. **c**, DCARL agent workflow. **d**, Dynamic confidence-value-update process. With new data, the WCV of the candidate policies and the BCV of the original policy are updated. The superscript m is the symbol to indicate variables related to the original policy. **e**, Decision-making process of the DCARL agent. It chooses the action by comparing the BCV of the original policy and WCV of the RL policies.

safeguard policy when necessary, but these principles may not always match the real-driving conditions.

Our main concept for the continuous improvement is to estimate a dynamic confidence value to represent the expected performance of the policies as well as the confidence to achieve this performance. It can better adapt to the driving environment's uncertainties within the insufficient training data. This idea is inspired by the phenomenon that the epistemic uncertainty[33] during driving can be reduced as data are collected because of the observed model's confidence improving. Furthermore, some recent research seems to indicate that the human learning process is associated with improvement of the learning performance and confidence together[34]. Our previous work[20] proposed to use the fixed worst-possible driving performance as the confidence value. We build on this idea to further explore the dynamically updated confidence value for continuous improvement with more data and the real-vehicle application. The confidence value represents whether the expected performance can reflect its actual driving performance, considering the variance of the updated driving experience and the number of repetitions for a specific case. The worst confidence value (WCV) of a policy is defined as follows:

$$\begin{aligned}&\forall s \in \mathcal{S}, \forall \pi \in \mathbf{\Pi}, \quad Q_{\pi}^{-}(s, \pi(s), \mathcal{D}) \in \mathcal{Q}_{\delta}^{-}(s, \pi(s), \mathcal{D}),\\ &\text{where } \mathcal{Q}_{\delta}^{-}(s, \pi(s), \mathcal{D}) = \{q \in \mathbf{R} | P(\tilde{Q}_{\pi}(s, \pi(s)) \geq q | \mathcal{D}) \geq 1-\delta\}\end{aligned} \tag{1}$$

where $Q_{\pi}^{-}(s, \pi(s), \mathcal{D})$ denotes WCV when a policy π is used for a given state $s \in \mathcal{S}$. $\mathbf{\Pi}$ is a set containing the candidate policies. $\tilde{Q}_{\pi}(s, \pi(s))$ denotes the ideal value under the policy $\pi(s)$, namely, the actual expected cumulated rewards of the policy. This value only exists in concept and is not observed directly. P denotes the probability and q belongs to the real number field $\mathbf{R}$. $\mathcal{D}$ denotes the dataset and $\mathcal{D}_k \subset \mathcal{D}_{k+1}$, which denotes the received dataset at time t_k and t_{k+1}, respectively. $\delta \in (0,1)$ determines the required policy confidence. A lower δ value leads to a stricter confidence requirement but may also require more data for training. Equation (1) denotes that the probability of the actual value larger than the WCV is higher than $1-\delta$.

Newly received data are used to dynamically update the WCV to continuously improve the confidence value, shown in Fig. 2b, and the required property is as follows:

$$\forall s \in \mathcal{S}, \forall \pi, \forall \mathcal{D}_k, P\left(Q_{\pi}^{-}(s, \pi(s), \mathcal{D}_{k+1}) \geq Q_{\pi}^{-}(s, \pi(s), \mathcal{D}_k)\right) \geq 1-\delta_k \tag{2}$$

where equation (2) shows that it is a high probability that the WCV is improved with more training data.

Finally, the DCARL introduces an existing driving policy as the original policy and the performance baseline for improvement, denoted as π_0. We further calculate the best confidence value (BCV) $Q_{\pi_0}^{+}(s, \pi_0(s), \mathcal{D})$ of π_0 as the threshold to activate the trained RL policy. It follows the similar definitions of WCV, but converges from a very large value to its actual value as the data increase. When the data are insufficient, the large BCV can avoid the unreasonable activation of the RL agent. Thus, the DCARL can be no worse than the original policy at different training stages, shown as follows:

$$\begin{aligned}&\pi_{ci}(s, \mathcal{D}) : \arg\max_{\pi \in \Pi}\left(Q_{\pi}^{c}(s, \pi(s), \mathcal{D})\right),\\ &Q_{\pi}^{c}(s, \pi(s), \mathcal{D}) = \begin{cases} Q_{\pi}^{-}(s, \pi(s), \mathcal{D}), & \pi \neq \pi_0 \\ Q_{\pi_0}^{+}(s, \pi_0(s), \mathcal{D}), & \pi = \pi_0 \end{cases}\end{aligned} \tag{3}$$

where $\pi_{ci}(s, \mathcal{D})$ denotes the DCARL policy trained with data $\mathcal{D}$. Superscript c indicates the confidence value.

https://doi.org/10.1038/s42256-023-00610-y

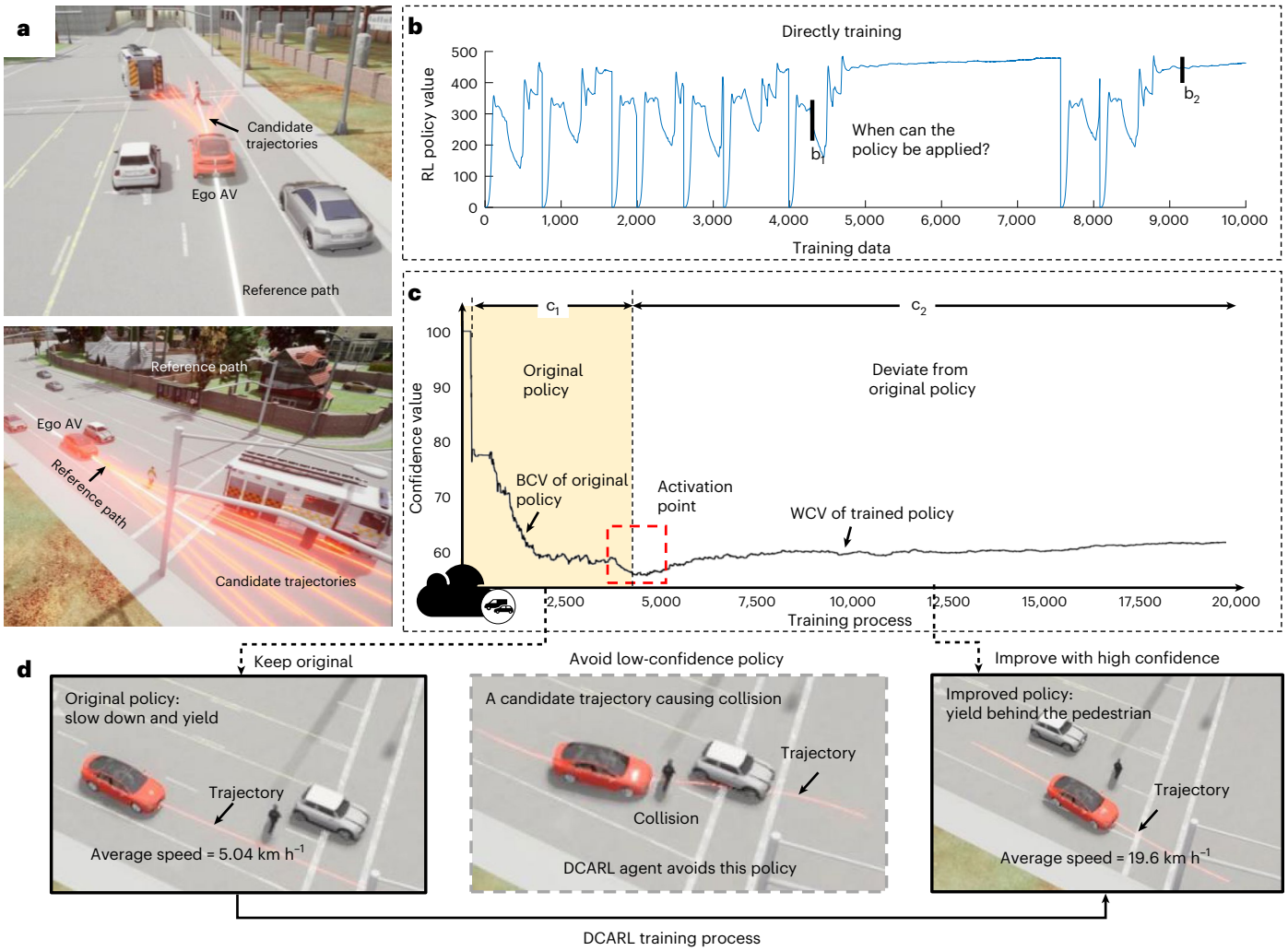

Fig. 3 | Continuously improved performance with more data. a, An example driving case in CARLA. CARLA is an open-source simulator [42]. The ego vehicle is driving on the road with three other vehicles and a pedestrian nearby. **b**, RL policy training value, which is the maximal state value of all candidate trajectories in the state–action space. Different training data may have different driving policies, e.g. the ego vehicle changes lanes at b_1 while keeps the lanes at b_2. Each datum is a collected driving trajectory. **c**, Confidence value of the proposed DCARL agent. There are two stages during the DCARL training. In the first stage (c_1), the DCARL agent uses the original policy to drive. In the second stage (c_2), the DCARL agent will take the maximal WCV of the candidate policies as the value function to generate the policy. **d**, Performance improvement in DCARL training process. The original policy in this scenario slows down and yields to the pedestrian. The final DCARL agent is still safe but not overly conservative.

The algorithms to achieve the DCARL agent consist of three parts: (1) an existing self-driving algorithm, (2) the dynamic confidence value updated through new training data and (3) the decision-making module.

(1) An existing self-driving algorithm (shown in Fig. 2c). The DCARL agent uses an existing self-driving algorithm as the original policy and the performance baseline. It keeps fixed, but its confidence value may change as training goes on, using the same dataset with the RL training process. For non-RL baseline policies, a converter is required to encode the original policy's input into a standard state vector. Furthermore, the confidence value of the original policy is initialized as positive infinity while other policies' values are negative infinity.

(2) The dynamic confidence-value update with new training data (shown in Fig. 2d). The function of this module is to estimate the WCV of the candidate policies and the BCV of the original policy, consisting of two steps. First, it estimates the maximum likelihood value (MLV) of the collected trajectories using the Monte Carlo principle with importance sampling rates. Then, it estimates the maximum possible error between the MLV and the actual value. The error due to insufficient training data can be estimated by the Lindeberg–Lévy theorem and bootstrapping principles. Namely, with more training data, training variance reduces, and there is lower error between the MLV and the actual value. With more training data, the WCV increases monotonically while the BCV of the original policy continuously decreases.

(3) A decision-making module (shown in Fig. 2e). When encountering a new driving case, the DCARL agent chooses the candidate driving trajectory that has the maximal confidence value. At the beginning of training, the original policy is used, as it is initialized with a large confidence value. Then, other policies' confidence values continue to increase and eventually some of them become higher than the original policy. That moment is called the activation time. After that, the agent deviates from the original policy. Achieving the activation level usually requires the agent's repeatedly good performance in a case to avoid mistaken activation due to only a few occasionally good performances.

https://doi.org/10.1038/s42256-023-00610-y

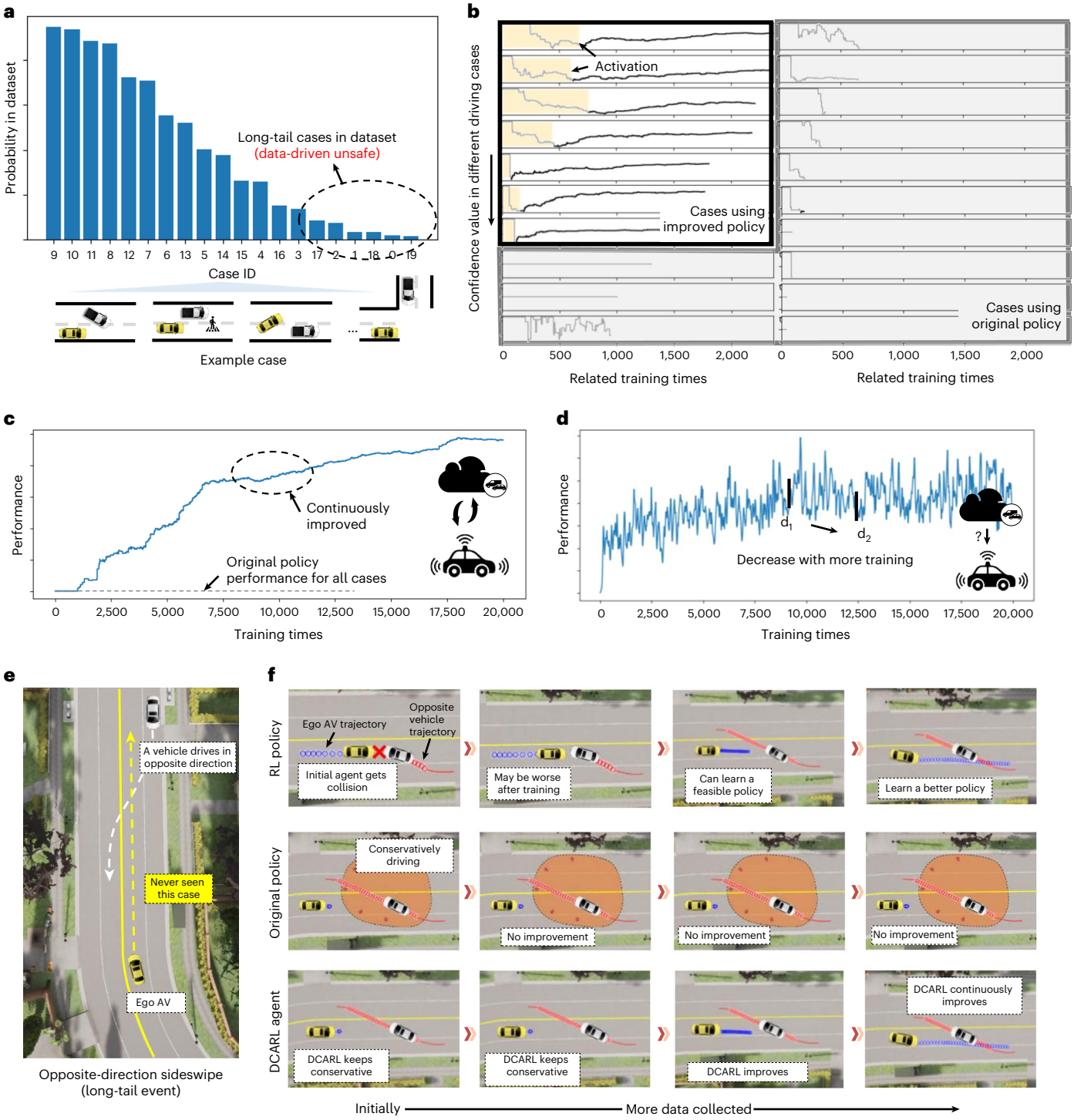

Fig. 4 | DCARL agent for long-tail cases. a, Collected data amount in testing cases. **b**, Confidence value of the DCARL agent in the 20 example cases. The policy in some cases has deviated from the original policy **c**, Overall performance of the 20 cases as the training data increase. **d**, Performance of the trained classic RL. It is clear that more data and more training do not always help, for example, performance deteriorates from d_1 to d_2. **e**, An example long-tail case, that is, opposite-direction sideswipe scenario. **f**, Performance comparison between DCARL, RL and original policy in a long-tail event. The trajectory is sampled within a fixed time interval that is, 0.3 s. Credit: ego vehicle and social vehicle icons, macrovector / Freepik.com.

The proposed technology mentioned above mainly targets the driving policy generation given the surrounding environment. This work assumes the perception module for detecting and tracking surrounding objects[35], and the control module to track the desired trajectories[36]. The vehicle configuration, algorithms and related source codes are open source for reproducing this work, available at https://github.com/zhcao92/DCARL (ref. [37]). In the following, we mainly evaluate the four mentioned features of DCARL.

Results

To validate the continuous improvement of the DCARL agent, we designed three experiments. In the first experiment, new training data

https://doi.org/10.1038/s42256-023-00610-y

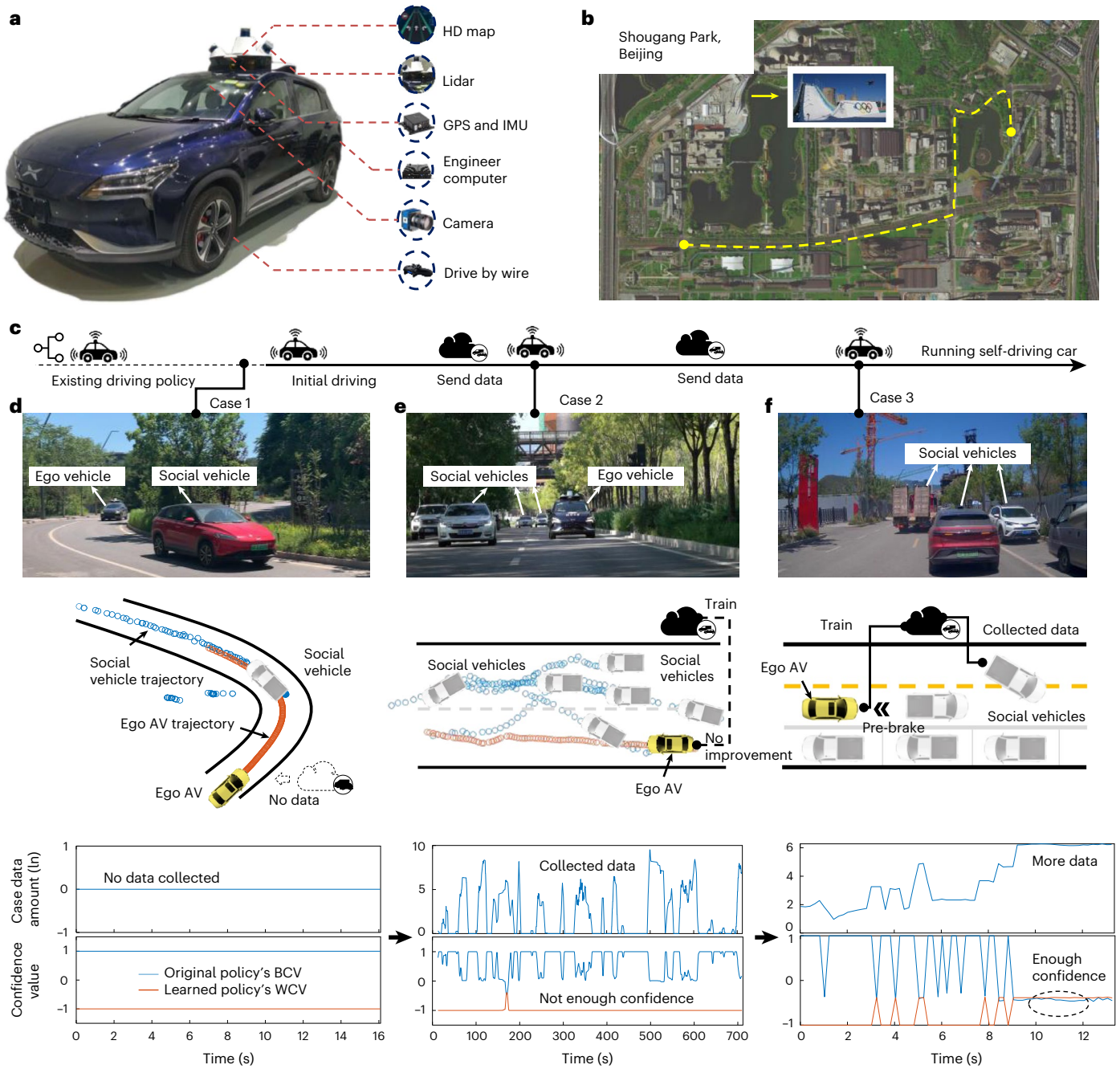

Fig. 5 | Experimental results. a, The experimental vehicle. HD map, high-definition map. GPS, global positioning system. IMU, inertial measurement unit. **b**, Reference route for testing in Shougang Park, Beijing. This is one of the parks used for the 2022 Beijing Winter Olympic Games. **c**, Evaluation process of the DCARL agent. **d**–**f**, Images, data amount and confidence value of driving case 1, which is a car-following scenario on a curved road, with no training (**d**), driving case 2, which is a two-road driving scenario (**e**), and driving case 3, where the DCARL agent uses a smooth pre-emptive brake rather than a late hard brake (from the original policy; **f**). Credit: ego vehicle and social vehicle icons, macrovector / Freepik.com.

continued to be obtained by simulations and fed to both the DCARL agent and a classic value-based RL agent, and their performance was recorded. Then, we manually set some of the test cases as long-tail (rare) cases by limiting the training data amount. Finally, the DCARL agent was deployed on an experimental vehicle driving on the open roads, which was demonstrated and evaluated at the 2022 Beijing Winter Olympic Games. These results can be rendered by the Supplementary Software.

Continuously improved performance with increasing data

We first study the performance of the DCARL agent with different volumes of data. The driving scenario is shown in Fig. 3a. The ego vehicle drives on a local road with some social vehicles. A pedestrian is also trying to cross the street. The training dataset contains various trajectories of this scenario and their previous driving performance. For comparison, we also use the classical value-based RL framework to train a pure data-driven agent, which maps the candidate trajectories to their expected performance from the dataset and chooses the trajectory with the highest value. Each training step of both agents operates after receiving a datum.

Classical RL agent driving performance. Figure 3b shows the training results of the classic RL agent using a widely used Deep Q-Learning

https://doi.org/10.1038/s42256-023-00610-y

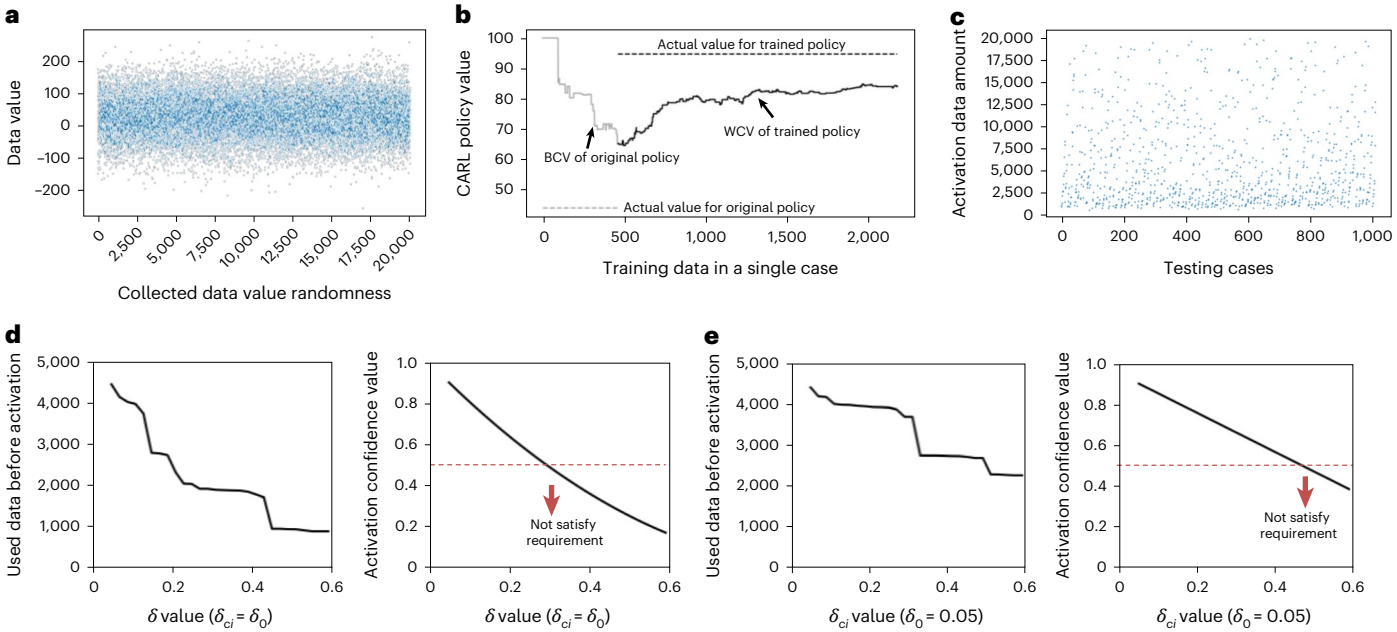

Fig. 6 | Data for evaluation and theoretical analysis. a, Cumulative reward value in the collected driving dataset. **b**, DCARL policy value for a selected driving case. **c**, Activation data amount for 1,000 testing cases. The data needed for activation vary widely for different driving cases. **d**, Activation required data and activation confidence with different δ_{ci} and δ_0. **e**, Activation required data and activation confidence with different δ_{ci}.

framework. It shows that performance improvement is not monotonic and the behaviours are quite different, for example, b_1 changes lanes, but b_2 does not. This figure exemplifies a crucial problem for RL agent training and application: the trained policy is not robust and not trustworthy. As a result, online learning and adaptation are risky, and it is unlikely RL agents will be deployed in the real world for mission-critical applications.

DCARL agent with continuous improvement. Figure 3c shows the performance of the proposed DCARL agent as training data increase. The process includes two stages. In the first stage (c_1), the BCV declines but the DCARL does not deviate from the baseline policy. This phenomenon is because the BCV is designed to converge from a very large value to its actual value as the data increase. When the data are insufficient, the large BCV can avoid the unreasonable activation of the RL agent. In this scenario, the original policy will conservatively slow down to yield to the crossing pedestrian, as shown in Fig. 3d.

With more training data, the WCV continues to increase, and the BCV of the original policy continues to decrease. When the DCARL agent has enough confidence that the RL agent will outperform the original policy, it enters the second stage (c_2) and the trained policy will be used. In this stage, the WCV improves continuously and the performance is guaranteed to be the same or better than the original policy. Compared with the original policy, the well-trained policy is still safe but increases the average driving speed from 5 km h^{-1} to 20 km h^{-1}.

This result indicates two benefits of the DCARL agent: its performance is guaranteed to be the same or better than the original policy, and with more data, it continues to improve.

DCARL agent for long-tail cases

In real-world driving, a self-driving vehicle can encounter some rarely seen cases. This long-tail case problem poses severe challenges to self-driving technology development. The collected dataset may have only few data on these long-tail cases, causing safety concerns for the data-driven agents. The proposed DCARL framework can use a conservative original policy[14], which always keeps a long distance to all the objects. This policy may be too conservative for normal driving, but the DCARL agent can fall back on this policy for safety in any driving case, especially when there is not enough confidence. As data are collected, the DCARL can continuously improve and deviate from the original policy in some cases for higher efficiency with high confidence. In this way, some long-tail cases can gradually turn into the common cases but other cases can still have the performance lower bound using the original conservative policy.

For evaluation, we first randomly sampled 20 driving cases, with a very different number of training data points, shown in Fig. 4a, to reflect the long-tail problem. Figure 4b shows the training stages in these cases, where the DCARL agent in some cases enters the second stage, while it keeps the original policy in other cases.

Figure 4c shows the overall performance of the 20 cases. The performance improves continuously with more training data. As the product original policy is conservative, the DCARL agent will start from a safe but conservative driving performance, but eventually achieve better driving performance. For comparison purposes, we also trained a classical value-based RL policy using the same training dataset, as shown in Fig. 4d. It shows that more data and more training do not always result in better performance. This is a crucial problem that the DCARL agent is designed to avoid.

For intuition, we further take a corner case, reported by Waymo, as an example[17], that is, opposite-direction sideswipe, shown in Fig. 4e. Note that (1) the tested agent has never seen the same or similar driving cases in the dataset, and (2) the initial RL policy failed at the beginning. Figure 4f shows the testing results at different training stages, compared with an RL agent and the product original policy. When the ego vehicle meets this long-tail case for the first time, the RL agent collides with the opposite vehicle, whereas the DCARL agent slows down at a very long distance. With more data collected in this case, the DCARL agent confirms that becoming closer to this opposite vehicle is safe in the following seconds, thus, it starts to deviate from the original policy. The RL agent also can improve finally, but the trained policy does not always improve the performance. These collisions cause strong safety concerns and hinder its application. If only using the original policy to drive, the performance will never change with more data, thus, it cannot turn a long-tail case into a common case.

In summary, the proposed DCARL agent enables continuous learning for self-driving algorithms working with a product conservative original policy. It is designed to eventually take over when it is adequately trained.

Field testing

We deployed the DCARL algorithm on our self-driving vehicle, as shown in Fig. 5a. The vehicle is equipped with sensors (Lidar and cameras), controllers and computing units. The detailed configurations and algorithms are included in Supplementary Section 1.

We deployed the vehicle at the 2022 Winter Olympic Games in Shougang Park, which is a geo-fenced urban driving environment, shown in Fig. 5b. The proposed DCARL agent directly works on an existing self-driving vehicle and starts with little knowledge of this operational domain. This existing driving policy can drive along the lanes considering the surrounding objects, but may not match real-world driving conditions and should be improved. This experiment will observe the driving performance changes with more data collected during the self-driving process.

The real-world learning process is shown in Fig. 5c. Initially, a manually designed driving policy achieves the basic driving function. Then, the vehicle starts to collect data and learn, that is, training while observing the baseline driving policy. All the driving cases randomly happen during real-world testing. We chose the following three cases as examples because they happen in three training phases, that is, initial training, preparation phase and improvement phase. Supplementary Section 2.3 introduces the details of these cases.

Initial performance without training. Figure 5d shows a car-following case on a curved section of the field. During this period, the DCARL agent has not collected any data for training yet, and the confidence value is the normalized initial value, that is, 1 for the original policy and −1 for other policies, as shown in Fig. 5d (bottom). According to the DCARL agent decision-making process, it uses the original policy, that is, a car-following model for driving.

Performance with inadequate training. Figure 5e shows a multi-lane driving case with multiple human-driven vehicles nearby. During this case, the agent has been trained for a short time, and it starts to estimate the confidence value of both the original policy and other candidate policies. However, the DCARL agent does not have enough confidence to outperform the original policy, as shown in Fig. 5e (bottom). Thus, the final selected policy for driving is still the original policy. In contrast, a self-driving car using a classical RL policy may have an updated policy with unknown performance.

Performance after adequate training. Figure 5f shows that a truck cuts-in in front of the leading vehicle, which is itself just metres ahead of the test vehicle. The original policy in this case would not have pre-emptively applied the brake because it does not well consider the two vehicles ahead. The DCARL agent, on the other hand, had collected enough driving data from similar cases and found a better policy that had enough confidence to outperform the original policy, as shown in Fig. 5f (bottom). In this way, the self-driving vehicle can pre-emptively brake and achieve a safer and smoother deceleration.

In summary, the proposed DCARL can directly work with an existing self-driving vehicle for road testing. It starts with a known level of driving performance, and with more data collected, the driving performance improves continuously. Namely, the DCARL can directly provide a continuous improvement ability to the existing road-testing self-driving cars without a lengthy pre-training.

Discussion

The results show that the proposed technology can work with most existing self-driving policies. The DCARL algorithm learns from driving data, and continues to monitor its worst confidence performance, and the RL agent will be activated only after it is guaranteed to be better than the original policy. This technology is implemented on a self-driving vehicle with a default policy, and continuous improvement was observed in the experiments.

This approach has three notable benefits. First, considering that many current self-driving algorithms work reasonably well in most cases but suffer from uncertain performance in long-tail cases, this technology provides a potential solution with continuous improvement. Second, an artificial-intelligence-based motion planner is mission-critical and cannot have unpredictable performance. The proposed technology ensures an understandable lower bound of performance provided by the original policy. Finally, this work integrates the RL and classical self-driving methods, and leverages the strength of the two types of technology.

A limitation of this work is that the DCARL agent may require more data than classical RL for the same performance because it requires high confidence. However, the lower data efficiency will not hinder the application of the DCARL policy, as the DCARL agent can be applied at any training stage due to the continuously improved performance from an existing policy. Figure 6 also discusses the required data under different settings. Furthermore, even with enough high confidence, the DCARL agent still may occasionally fail or be worse than the original policy, for example, when a surrounding vehicle is suddenly out of control. This work cannot totally avoid such conditions, but can guarantee that their probability of occurrence is very low during normal driving.

Methods

Problem definition

In this work, the self-driving planning problem is formulated as a Markov decision process[38], assuming that the conditional probability distribution of future states depends only on the present state, that is, the Markov property. The goal of classical RL is to find the optimal policy to maximize expected accumulated reward, that is, action–state–value function $\tilde{Q}_\pi(s,a)$, defined as:

$$\forall h \in \mathbf{N}, \tilde{Q}_\pi(s,a) := \mathbb{E}_\pi\left[\sum_{t=h}^{h+H-1} \gamma^{t-h} r_t | s_h = s, a_h = a\right] \tag{4}$$

where $\mathbb{E}[\cdot]$ denotes the expectation and r_t denotes the reward at time t. H denotes a finite planning horizon of the planner. h, a natural number $\mathbf{N}$, indicates that the planning starts from the time t_h. $\gamma \in (0,1]$ is a discount factor. The tildes indicate the actual value, distinguished from the estimated value $Q_\pi(s,a)$ during training.

The imagined problem is as follows: there is a self-driving vehicle with an original control policy π_0. The proposed DCARL agent is added to work with this original policy. The collected driving data are continuously fed into the DCARL agent. With more training data, the DCARL agent will gradually gain experience and confidence in some driving cases.

The goal of the DCARL agent can be formally defined as follows:

$$\forall s \in \mathcal{S}, \forall \mathcal{D}_k, \mathcal{D}_{k+1},$$
$$Q^-\left(\mathbb{E}_{\pi(\mathcal{D}_{k+1})}\left[\sum_{t=h}^{h+H-1} r_t(s_t,a_t)|s_h = s\right]\right) \geq Q^-\left(\mathbb{E}_{\pi(\mathcal{D}_k)}\left[\sum_{t=h}^{h+H-1} r_t(s_t,a_t)|s_h = s\right]\right) \tag{5}$$

$$\forall s \in \mathcal{S}, \mathcal{D}_k, \mathbb{E}_{\pi_{ci}(\mathcal{D}_k)}\left[\sum_{t=h}^{h+H-1} r_t(s_t,a_t)|s_h = s\right] \geq \mathbb{E}_{\pi_0}\left[\sum_{t=h}^{h+H-1} r_t(s_t,a_t)|s_h = s\right] \tag{6}$$

where π_0 denotes the original policy, mapping the state s to action a. $\mathcal{D}_k$ and $\mathcal{D}_{k+1}$ denote the received data at time t_k and t_{k+1}, respectively. $\mathcal{D}_{k+1}$ has more data than $\mathcal{D}_k$ and can lead to higher confidence and

better performance. $\pi_{ci}(\mathcal{D})$ denotes the generated policy after receiving dataset $\mathcal{D}$. $p(s)$ denotes the probability of the state s. $Q^{-}(\cdot)$ denotes the WCV. Note that the agent can dynamically update the trajectory at each time step, thus the danger will be responded if it is hitting the car within the horizon, or be punished when it is risky. A longer horizon may be safer, but may decrease the calculation efficiency. Furthermore, a very long future condition has a limited effect on the current state's value (and confidence). Thus, we use the finite horizon design in the DCARL agent. In general, equation (5), which describes the worst confidence lower bound, should improve or at least maintain from $\mathcal{D}_k$ to $\mathcal{D}_{k+n}$ and equation (6) means that the overall performance must be the same or better than the original policy.

This DCARL agent should update the WCV of the candidate policies and the BCV of the original policy. The WCV is defined in equations (1) and (2). Similarly, the BCV of the original policy, denoted by $Q^{+}_{\pi_0}(s,\pi_0(s),\mathcal{D})$ is defined as follows:

$$\begin{aligned}&\forall s\in\mathcal{S},\forall\pi\in\Pi,Q^{+}_{\pi}(s,\pi(s),\mathcal{D})\in\inf\mathcal{Q}^{+}_{\delta_0}(s,\pi(s),\mathcal{D}),\\&\text{where }\mathcal{Q}^{+}_{\delta_0}(s,\pi(s),\mathcal{D})=\{q\in\mathbf{R}|P(\tilde{Q}_{\pi}(s,\pi(s))\le q|\mathcal{D})\ge 1-\delta_0\}\end{aligned}\tag{7}$$

$$\forall s\in\mathcal{S},\forall\mathcal{D}_k,P\left(Q^{+}_{\pi_0}(s,\pi_0(s),\mathcal{D}_{k+1})\le Q^{+}_{\pi_0}(s,\pi_0(s),\mathcal{D}_k)\right)\ge 1-\delta_k\tag{8}$$

The final DCARL agent's policy $\pi_{ci}(s,\mathcal{D})$ follows equation (3), namely, the policy should have the maximal confidence value of all the candidate policies. With equations (1), (2), (7) and (8), the final policy $\pi_{ci}(s,\mathcal{D})$ can ensure continuous improvement, defined in equations (5) and (6) (see Theorem 1 for the proof). We further prove that when the information contains a fixed feedback error or a standard noise distribution[39,40], the proposed DCARL agent can still satisfy the requirements (see Theorems 4 and 5 for the proof). However, other kinds of uncertainties may not be well solved, for example, miss detected objects and so on.

The following sections introduce the approaches to estimate the WCV and its improvement property in equation (2) and the original policy's BCV in equation (8), as well as the DCARL policy generation in equation (3).

Dynamic update of confidence value

This section first introduces the dynamic dataset format for confidence-value updating. Then, the value-update process after receiving new data is described to meet the requirements in equations (1), (2), (7) and (8).

Dynamic dataset format and data collection. The driving data collected can be represented as a sequence:

$$\tau(s,a):=\{s,a,s_1,a_1,\ldots\}\tag{9}$$

where $\tau(s,a)$ denotes a trajectory starting from state s using action a. a belongs to the action space $\mathcal{A}$. The cumulative driving reward of this trajectory can be written as:

$$G(s,a)=\sum_{t=h}^{h+H-1}r_t(s_t,a_t)|s_h=s,a_h=a\tag{10}$$

where H denotes the horizon.

A data unit in dataset $\mathcal{D}$ can be defined as $d(\mathrm{s},\mathrm{a}):=(s,a,G(s,a))$, where $G(s,a)$ can be considered as an independent identically distributed sampling given the policy s, a, π, according to the Markov property.

The dataset contains many data units, defined as:

$$\begin{aligned}\mathcal{D}_k&=\{d_{i=1\ldots k}(s,a)\}\\\mathcal{D}_{k+1}&=\{\mathcal{D}_k,d_{k+1}(s,a)\}\end{aligned}\tag{11}$$

where $\mathcal{D}_k$ contains the collected data units. Specifically, $\mathcal{D}_k(s,a)\subseteq\mathcal{D}_k$ denotes all the data units starting from s,a.

WCV and BCV initialization. Initially, the DCARL agent does not have any data. The initialization of the confidence value WCV and BCV are as follows:

$$\forall s\in\mathcal{S},\forall a\in\mathcal{A},\quad Q^{-}_{\pi}(s,\pi(s)=a,\mathcal{D}_0)\to-\infty,Q^{+}_{\pi_0}(s,\pi_0(s)=a_0,\mathcal{D}_0)\to+\infty\tag{12}$$

Dynamic update of WCV and BCV when receiving new data. When new data $d_k=\{s,a,G(s,a)\}$ are received, the WCV $Q^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)$ and the BCV $Q^{+}_{\pi_0}(s,\pi_0(s),\mathcal{D}_k)$ are updated. The updating method is inspired by the sampling-based trajectory generation methods[7]. The basic idea is that the ego vehicle plans a trajectory for the whole planning horizon, and adjusts this trajectory when necessary. In each time step, it only considers the candidate trajectories. The self-driving vehicle recursively re-examines the situation and makes a new decision at each time step. This is necessary because the real world is full of uncertainties and cannot be accurately predicted. The average state–action value is then estimated by:

$$\forall s\in\mathcal{S},\bar{Q}_{\pi}(s,a)\leftarrow\bar{G}(s,a)=\frac{1}{n}\sum_{G(s,a)\in\mathcal{D}_k(s,a)}G(s,a)\tag{13}$$

where $\bar{G}(s,a)$ is a point estimation of $\bar{Q}_{\pi}(s,a)$. With a large number of samples, $\bar{G}(s,a)$ converges to $\bar{Q}_{\pi}(s,a)$. However, due to the environmental uncertainties and limited number of collected data for each driving scenario, the estimated value $\bar{G}$ may be different from the actual value $\bar{Q}_{\pi}(s,a)$.

The distribution can be estimated using the Lindeberg–Lévy theorem, namely, the probability $P(\bar{Q}_{\pi}(s,a),\mathcal{D}_k)$ becomes close to a Gaussian distribution when the data amount k is large enough, as follows:

$$\forall z\in\mathbf{R},s_i\in\mathcal{S}\lim_{n\to\infty}P\left[\sqrt{n}(\bar{G}_{\pi}(s_i)-Q_{\pi}(s_i,a_{\pi}(s_i)))\le z\right]=\Phi\left(\frac{z}{\sigma}\right)\tag{14}$$

where $\sigma^2=\mathrm{Var}(G_{\pi}(s_i))$. z represents any real number and n denotes the data number in the dataset. Φ denotes the cumulative distribution function of a standard normal distribution. On the basis of this assumption, the WCV $Q^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)$ is calculated using the cumulative distribution function of a Gaussian distribution:

$$Q^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)=\frac{\frac{1}{\mathrm{Var}(G_{\pi}(s_i))\sqrt{2\pi}}\int_{-\infty}^{Q^{-}_{\pi}}\exp\left(-\frac{(Q^{-}_{\pi}-\bar{G}_{\pi})^2}{2\mathrm{Var}(G_{\pi}(s_i))}\right)\mathrm{d}Q^{-}_{\pi}}{\frac{1}{\mathrm{Var}(G_{\pi}(s_i))\sqrt{2\pi}}\int_{-\infty}^{\infty}\exp\left(-\frac{(Q^{-}_{\pi}-\bar{G}_{\pi})^2}{2\mathrm{Var}(G_{\pi}(s_i))}\right)\mathrm{d}Q^{-}_{\pi}}=\delta\tag{15}$$

Equation (15) can satisfy the worst-performance lower-bound property in equation (1), but may not satisfy the continuous improvement property in equation (2). Thus, the WCV should be further adjusted to:

$$\begin{aligned}&Q^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)\in\mathcal{Q}^{-}_{\delta}(s,\pi(s),\mathcal{D}_k)\cap\mathcal{Q}^{-}_{ci}(s,\pi(s),\mathcal{D}_k)\\&\mathcal{Q}^{-}_{ci}(s,\pi(s),\mathcal{D}_k)=\{q\in\mathbf{R}|P(G_{k+1}\ge q|\mathcal{D}_k)\ge 1-\delta_k\}\end{aligned}\tag{16}$$

where $Q^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)$ should also be in the continuous improvement set $\mathcal{Q}^{-}_{ci}(s,\pi(s),\mathcal{D}_k)$, where the probability of the WCV lower than the next sampled data value is higher than $1-\delta_k$. $p(G_{k+1})$ is fitted by the previously collected data. In this way, the WCV can satisfy equation (1) as well as the continuous improvement property with more data (see Theorem 2 for proof).

Note that the Lindeberg–Lévy theorem needs a large amount of data (that is, $k>30$). When the data amount is small, $P(\bar{Q}_{\pi}(s,a),\mathcal{D}_k)$ is estimated using the bootstrapping method[41] from the dataset $\mathcal{D}_k(s,a)$. The main idea is to infer population properties from sampled data.

It does not make assumptions about the population distributions. Namely, the method offers information about $P(\tilde{Q}_\pi(s,a),\mathcal{D}_k)$ by inferring from D_π. To estimate $P(\tilde{Q}_\pi(s,a),\mathcal{D}_k)$ with the bootstrapping method, we first sample the subsets of $\mathcal{D}_k(s,a)$, denoted as $\mathcal{D}_k^1(s,a)$, $\mathcal{D}_k^2(s,a)\ldots\mathcal{D}_k^n(s,a)$. Each subset contains the same number of data units as the original $\mathcal{D}_k(s,a)$, but each data unit is uniformly sampled from the original dataset. In this way, we calculate $\bar{G}_k^1(s,a),\bar{G}_k^2(s,a),\ldots,\bar{G}_k^n(s,a)$ which are n independent and identically distributed samples from $P(\tilde{Q}_\pi(s,a),\mathcal{D}_k)$.

Both methods require a small number of data units (that is, 10). The WCV value remains at its initial value (near $-\infty$) when the amount of data is small. The WCV will continuously improve with more data. With sufficient data, the distribution will approach $\tilde{Q}_\pi(s,a)$. The way to update the BCV is similar to updating the WCV. The only difference is that the BCV requires the dataset $\mathcal{D}_k(s,\pi_0(s))$ and is obtained from the upper bound of the distribution $P(\tilde{Q}_\pi(s,a),\mathcal{D}_k)$.

Continuously improved policy generation

This section introduces the approach to generating the continuously improved policy in equation (3). The main idea is that the final policy should have a WCV that is not less than the original policy's BCV.

The policy is described as follows:

$$\begin{aligned}\pi_{ci}(s,\mathcal{D}) &= a_{ci}(s,\mathcal{D}) = \arg\max_{a\in\mathcal{A}}\left(Q^c_{\pi_a(\mathcal{D})}(s,a,\mathcal{D})\right)\\ Q^c_{\pi_a(\mathcal{D})}(s,a,\mathcal{D}) &= \begin{cases} Q^-_{\pi_a(\mathcal{D})}(s,a,\mathcal{D}), & a\neq a_0\\ Q^+_{\pi_0}(s,a_0,\mathcal{D}), & a=a_0\end{cases}\end{aligned} \tag{17}$$

where $Q^c_{\pi_a(\mathcal{D})}(s,a,\mathcal{D})$ denotes the confidence value of taking action a given data $\mathcal{D}$. The overall policy will be $\pi_{ci}(s,\mathcal{D})$. Theorem 3 proves that this policy can meet the requirement of equation (3).

As the action generation considers only the value for each case, the corresponding activation points may be different for each case. For example, if the data contain more driving cases about car-following but not so much about intersections, then learning progress for these two scenarios can vary widely. The existence of some long-tail cases will not affect the performance improvement property, which is the main contribution of this work.

The policy generation process for equation (17) can be described as follows. When a self-driving vehicle runs on the road and meets a case s, the DCARL agent will first calculate all the WCVs of all the candidate actions from the action space. Then, the DCARL agent will calculate the BCV of the original policy. When the maximal WCV of the candidate actions is larger than the original policy's BCV, then the RL agent is activated; otherwise, the original policy's action will still be used.

Data for evaluation and theoretical analysis

Data collection for evaluation. The training data for evaluation are collected by randomly sampling the state s, action a and cumulative rewards $G(s,a)$, to simulate the data collection process in real-world driving. For the evaluation, the environment will first randomly generate the truth value $\tilde{Q}_\pi(s,a)$ for $G(s,a)$. Then sampling $G(s,a)$ around the truth value following a normal distribution. Some sampled data values are shown in Fig. 6a.

Statistical results and analysis. Figure 6b shows the case shown in Fig. 3c but with the truth Q-value. In the first stage, the DCARL agent uses the original policy. The estimated BCV value approaches the actual value but is always larger than it. After about 500 data points, the WCV of one RL policy becomes larger than the BCV of the original policy. The DCARL then switches to using this RL policy.

The activation time for different cases varies widely. Figure 6c shows the activation time of 1,000 different driving cases. Activation time varies between 50 and 20,000 training steps.

Effects of δ value on the confidence and required data. The value of δ has two effects on the DCARL agent: decreasing the δ value will increase the required data amount when the DCARL agent deviates from the baseline policy (see Theorem 6); and making δ close to zero can ensure higher confidence for the continuous improvement performance.

This work set δ value at 0.05, which is fixed during the training and testing. The δ value can be set in the range of (0.0.29). The upper-bound value aims to ensure that the autonomous vehicle performs better than the original policy. Thus, the probability of outperforming the original policy should be at least larger than 50%, that is, $(1-\delta)(1-\delta_0)\geq 0.5$. If $\delta_0=\delta$, we can obtain $\delta<0.29$. The lower-bound value aims to ensure the autonomous vehicle can eventually deviate from the baseline policy. Namely, equation (16) should have a feasible solution, that is, $(1-\delta)(1-\delta_0)<1$. It can conclude that $\delta>0$.

Figure 6d,e shows the curve of the used data before activation and the activation confidence. It first adjusts δ_{ci} and δ_0 from 0.05 to 0.6. Then, it adjusts only δ_{ci} from 0.05 to 0.6, while δ_0 remains constant 0.05. We can conclude that a higher δ can decrease the required data for activation and also decrease the activation confidence.

Theorem 1

This theorem aims to prove that the generated DCARL policy $\pi_{ci}(s,\mathcal{D})$ can satisfy the continuous improvement requirements, which is the main conclusion of this work.

Proof of Theorem 1. There are two facts that need to be proved: (1) the performance of $\pi_{ci}(s,\mathcal{D})$ is always the same or better than the original policy π_0; and (2) more $\mathcal{D}$ results in a higher WCV.

(1) The performance of $\pi_{ci}(s,\mathcal{D})$ always improves from the original policy π_0, that is

$$\forall s\in\mathcal{S},\mathcal{D},\mathbb{E}_{\pi(\mathcal{D})}\left[\sum_{t=h}^{h+H-1}r_t(s_t,a_t)|s_h=s\right]\geq\mathbb{E}_{\pi_0}\left[\sum_{t=h}^{h+H-1}r_t(s_t,a_t)|s_h=s\right] \tag{18}$$

The policy $\pi_{ci}(s,\mathcal{D})$ is considered in two cases, that is, $\pi_{ci}(s,\mathcal{D})=\pi_0(s)$ and $\pi_{ci}(s,\mathcal{D})\neq\pi_0(s)$. When $\pi_{ci}(s,\mathcal{D})=\pi_0(s)$, $\tilde{Q}_{\pi_{ci}(\mathcal{D})}(s,\pi_{ci}(s,\mathcal{D}))=\tilde{Q}_{\pi_0}(s,\pi_0(s))$, satisfying equation (18). If $\pi_{ci}(s,\mathcal{D})\neq\pi_0(s)$, according to equation (3), we have

$$Q^-_{\pi_{ci}(\mathcal{D})}(s,\pi_{ci}(s,\mathcal{D}),\mathcal{D})\geq Q^+_{\pi_0}(s,\pi_0(s),\mathcal{D}) \tag{19}$$

Following equations (1) and (7)

$$\begin{aligned}&P(\tilde{Q}_\pi(s,\pi(s))\geq\tilde{Q}_{\pi_0}(s,\pi_0(s)))\geq\\ &P(\tilde{Q}_\pi(s,\pi(s))\geq Q^-_\pi(s,\pi(s),\mathcal{D}))*P\left(Q^+_{\pi_0}(s,\pi_0(s),\mathcal{D})\geq\tilde{Q}_{\pi_0}(s,\pi_0(s))\right)\\ &=(1-\delta)(1-\delta_0)\end{aligned} \tag{20}$$

Equation (20) can be understood as that for more than $(1-\delta)(1-\delta_0)$ probability, $\tilde{Q}_\pi(s,\pi(s))\geq\tilde{Q}_{\pi_0}(s,\pi_0(s))$. According to equation (4), we obtain, for more than $(1-\delta)(1-\delta_0)$ probability:

$$\forall s,\mathcal{D}_k,\mathbb{E}_{\pi(\mathcal{D}_k)}\left[\sum_{t=h}^{h+H-1}r_t(s_t,a_t)|s_h=s\right]\geq\mathbb{E}_{\pi_0}\left[\sum_{t=h}^{h+H-1}r_t(s_t,a_t)|s_h=s\right] \tag{21}$$

(2) More $\mathcal{D}$ results in a higher WCV.

When the WCV of all the candidate policies is improved with more data, then, according to the definition of $\pi_{ci}(s,\mathcal{D})$ in equation (3), it can be described as:

$$Q_\pi^-(s,\pi_{ci}(s,\mathcal{D}_{k+1}),\mathcal{D}_{k+1}) = \arg\max_{\pi\in\Pi}\left(Q_\pi^c(s,\pi(s),\mathcal{D}_{k+1})\right)$$
$$\geq Q_\pi^-(s,\pi_{ci}(s,\mathcal{D}_k),\mathcal{D}_{k+1}) \geq Q_\pi^-(s,\pi_{ci}(s,\mathcal{D}_k),\mathcal{D}_k) \geq Q_\pi^-(s,\pi_{ci}(s,\mathcal{D}_{k+1}),\mathcal{D}_k) \quad (22)$$

This theorem is proved.

Theorem 2

This theorem proves that the WCV $Q_\pi^-(s,\pi(s),\mathcal{D}_k,\delta)$ can be improved with more data, which supports the conclusion of continuous improvement.

Proof of Theorem 2. From the central limit theorem, $P(\tilde{Q}_\pi(s,a),\mathcal{D}_k)$ is close to a normal distribution. Thus, the WCV can be represented as:

$$Q_\pi^-(s,\pi(s),\mathcal{D}_k) = E(\mathcal{D}_k) - j(\delta)\sigma(\mathcal{D}_k) \quad (23)$$

where $j(\delta)$ denotes a constant, determined by δ according to the cumulative distribution function of a normal distribution.

To analyse the improvement property, we first assume a collected trajectory value G_{k+1} to form $\mathcal{D}_{k+1}$ and calculate the condition of G_{k+1} when it can satisfy the improvement requirement. The increased value is described as follows:

$$\begin{aligned} &Q_\pi^-(s,\pi(s),\mathcal{D}_{k+1}) - Q_\pi^-(s,\pi(s),\mathcal{D}_k) \\ &= E(\mathcal{D}_{k+1}) - E(\mathcal{D}_k) - j(\delta)(\sigma(\mathcal{D}_{k+1}) - \sigma(\mathcal{D}_k)) \end{aligned} \quad (24)$$

As the samples' variance is $1/k$ of the population variance:

$$\sigma(\mathcal{D}_{k+1}) - \sigma(\mathcal{D}_k) = \frac{\sigma_F}{\sqrt{k+1}} - \frac{\sigma_F}{\sqrt{k}} = -\sigma(\mathcal{D}_k)\left(\frac{\sqrt{k+1}-\sqrt{k}}{\sqrt{k+1}}\right) \quad (25)$$

where F is a symbol to indicate that σ_F denotes the standard deviation of the distribution G_k.

$$E(\mathcal{D}_{k+1}) - E(\mathcal{D}_k) = \frac{G_{k+1} - E(\mathcal{D}_k)}{k+1} \quad (26)$$

$$\begin{aligned} &Q_\pi^-(s,\pi(s),\mathcal{D}_{k+1}) - Q_\pi^-(s,\pi(s),\mathcal{D}_k) \\ &= \tfrac{G_{k+1}-E(\mathcal{D}_k)}{k+1} + j(\delta)\sigma(\mathcal{D}_k)\left(\tfrac{\sqrt{k+1}-\sqrt{k}}{\sqrt{k+1}}\right) \end{aligned} \quad (27)$$

Thus, condition $Q_\pi^-(s,\pi(s),\mathcal{D}_{k+1}) \geq Q_\pi^-(s,\pi(s),\mathcal{D}_k)$ holds if and only if:

$$G_{k+1} \geq E(\mathcal{D}_k) - j(\delta)\sigma(\mathcal{D}_k)(k+1-\sqrt{k(k+1)}) \quad (28)$$

Thus, if G_{k+1} satisfies that:

$$p\left(G_{k+1} \geq E(\mathcal{D}_k) - j(\delta)\sigma(\mathcal{D}_k)\left(k+1-\sqrt{k(k+1)}\right)\right) \geq 1-\delta_k \quad (29)$$

then, with a higher probability of $1-\delta_k$, the WCV can certainly be improved with newly received and satisfy the requirements in equation (2). The δ_k value is determined by estimation of G_{k+1} in advance, which can be obtained using the empirical distribution from the historical data:

$$\hat{F}_k(G) = \frac{\sum_{i=1}^k I(G_i \leq G)}{k} \quad (30)$$

where G_i denotes a historical collected value from the dataset $\mathcal{D}_k$. According to Glivenko–Cantelli theory, the distribution $\hat{F}_k(G)$ almost surely converges to true distribution that G_{k+1} sampled from:

$$\sup_G |\hat{F}_k(G) - F(G)| \to 0 \quad (31)$$

Thus, the $\hat{F}_k(G)$ is good estimation of $P(G_{k+1})$. In this way, the δ_k can be calculated using the δ value.

The theorem is proved.

Theorem 3

This theorem proves that using the action $a_{ci}(s,\mathcal{D})$ with the maximum confidence value can achieve the policy's continuous improvement. The action $a_{ci}(s,\mathcal{D})$ is used to drive the vehicle which is the output of the DCARL agent. Thus, this proof ensures that the final performance satisfies our claims.

Proof of Theorem 3. The action $a_{ci}(s,\mathcal{D})$ is considered in two cases, that is, $a_{ci}(s,\mathcal{D}) = a_0(s)$ and $a_{ci}(s,\mathcal{D}) \neq \pi_0(s)$.

If $a_{ci}(s,\mathcal{D}) = a_0(s)$:

$$\begin{aligned} &\pi_{ci}(s,\mathcal{D}) = a_{ci}(s,\mathcal{D}) = a_0(s) \Rightarrow \forall a \in \mathcal{A}, Q_{\pi_0}^+(s,a_0,\mathcal{D}) \geq Q_{\pi_{ci}(\mathcal{D})}^-(s,a,\mathcal{D}) \\ &\text{Considering that } \forall \pi,\mathcal{D},\pi(s,\mathcal{D}) \in \mathcal{A} \\ &\Rightarrow \forall \pi,\mathcal{D}, Q_{\pi_0}^+(s,a_0,\mathcal{D}) \geq Q_{\pi_{ci}(\mathcal{D})}^-(s,\pi(s,\mathcal{D}),\mathcal{D}) \geq Q_\pi^-(s,\pi,\mathcal{D}) \\ &\text{Thus, } \pi_{ci}(s,\mathcal{D}) = \arg\max_{\pi\in\Pi}\left(Q_\pi^c(s,\pi(s),\mathcal{D})\right) \end{aligned} \quad (32)$$

If $a_{ci}(s,\mathcal{D}) \neq a_0(s)$:

$$\begin{aligned} &\pi_{ci}(s,\mathcal{D}) = \arg\max_{a\in\mathcal{A}}\left(Q_{\pi_{ci}(\mathcal{D})}^-(s,a,\mathcal{D})\right) \\ &\Rightarrow \forall a \in \mathcal{A}, Q_{\pi_{ci}(\mathcal{D})}^-(s,a_{ci}(s,\mathcal{D}),\mathcal{D}) \geq Q_{\pi_{ci}(\mathcal{D})}^-(s,a,\mathcal{D}), \\ &\text{and } Q_{\pi_{ci}(\mathcal{D})}^-(s,a_{ci}(s,\mathcal{D}),\mathcal{D}) \geq Q_{\pi_0}^+(s,\pi_0(s),\mathcal{D}) \\ &\text{Considering that } \forall \pi,\mathcal{D},\pi(s,\mathcal{D}) \in \mathcal{A} \\ &\Rightarrow Q_{\pi_{ci}(\mathcal{D})}^-(s,a_{ci}(s,\mathcal{D}),\mathcal{D}) \geq Q_{\pi_{ci}(\mathcal{D})}^-(s,\pi(s,\mathcal{D}),\mathcal{D}) \geq Q_\pi^-(s,\pi,\mathcal{D}) \\ &\text{Thus, } \pi_{ci}(s,\mathcal{D}) = \arg\max_{\pi\in\Pi}\left(Q_\pi^c(s,\pi(s),\mathcal{D})\right) \end{aligned} \quad (33)$$

This theorem is proved.

Theorem 4

When the uncertainty brings a fixed feedback error, that is, $\hat{G}(s,a) = G(s,a) + \theta$, where $\hat{G}(s,a)$, $G(s,a)$ and $\theta \in \mathbf{R}$ denote the observed return value, actual value and the feedback error, respectively. Then, the proposed DCARL agent can still satisfy the outperforming and continuously improving requirements.

Proof of Theorem 4.

(1) With a fixed feedback error, the performance of $\pi_{ci}(s,\mathcal{D})$ always improves from the original policy π_0.

We first replace $G(s,a)$ in equation (13), to obtain

$$\begin{aligned} \bar{G}(s,a) &= \tfrac{1}{n}\sum G(s,a) = \tfrac{1}{n}\sum\left(\hat{G}(s,a) - \theta\right) = \tfrac{1}{n}\sum\hat{G}(s,a) - \theta \\ &\Rightarrow \bar{\hat{G}}(s,a) = \tfrac{1}{n}\sum G(s,a) + \theta \to \tilde{Q}_\pi(s,a) + \theta \end{aligned} \quad (34)$$

Following the definition of $Q_\pi^-(s,\pi(s),\mathcal{D}_k)$ and $Q_\pi^+(s,\pi(s),\mathcal{D}_k)$ in equation (16), it can be concluded that

$$\hat{Q}_\pi^-(s,\pi(s),\mathcal{D}_k) = Q_\pi^-(s,\pi(s),\mathcal{D}_k) + \theta; \hat{Q}_\pi^+(s,\pi(s),\mathcal{D}_k) = Q_\pi^+(s,\pi(s),\mathcal{D}_k) + \theta; \quad (35)$$

Then, if the outperforming condition in equation (19) satisfies, it will become

$$\begin{aligned}&\hat{Q}^{-}_{\pi_{ci(\mathcal{D})}}(s,\pi_{ci}(s,\mathcal{D}),\mathcal{D})-\hat{Q}^{+}_{\pi_0}(s,\pi_0(s),\mathcal{D})\geq 0\\&\Rightarrow Q^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)+\theta-Q^{+}_{\pi}(s,\pi_0(s),\mathcal{D}_k)-\theta\\&=Q^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)-Q^{+}_{\pi}(s,\pi_0(s),\mathcal{D}_k)\geq 0\end{aligned}\tag{36}$$

Thus, for more than $(1-\delta)(1-\delta_0)$ probability:

$$\forall s,\mathcal{D}_k,\mathbb{E}_{\pi(\mathcal{D}_k)}\left[\sum_{t=h}^{h+H-1}r_t(s_t,a_t)|s_h=\hat{s}\right]\geq\mathbb{E}_{\pi_0}\left[\sum_{t=h}^{h+H-1}r_t(s_t,a_t)|s_h=\hat{s}\right]\tag{37}$$

(2) More $\mathcal{D}$ results in a higher WCV.

Similarly, equation (22) can be rewritten as:

$$\begin{aligned}&\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_{k+1}),\hat{\mathcal{D}}_{k+1})=\arg\max_{\pi\in\Pi}\left(\hat{Q}^{c}_{\pi}(s,\pi(s),\hat{\mathcal{D}}_{k+1})\right)\\&\geq\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_k),\hat{\mathcal{D}}_{k+1})\geq\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_k),\hat{\mathcal{D}}_k)\geq\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_{k+1}),\hat{\mathcal{D}}_k)\end{aligned}\tag{38}$$

Then

$$\begin{aligned}&\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_{k+1}),\hat{\mathcal{D}}_{k+1})-\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_{k+1}),\hat{\mathcal{D}}_k)\geq 0\\&\Rightarrow Q^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_{k+1}),\hat{\mathcal{D}}_{k+1})+\theta-Q^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_{k+1}),\hat{\mathcal{D}}_k)-\theta\geq 0\\&\Rightarrow Q^{-}_{\pi}(s,\pi_{ci}(s,\mathcal{D}_{k+1}),\mathcal{D}_{k+1})\geq Q^{-}_{\pi}(s,\pi_{ci}(s,\mathcal{D}_{k+1}),\mathcal{D}_k)\end{aligned}\tag{39}$$

This theorem is proved.

Theorem 5

When the uncertainty brings a noise distribution on the data, that is, $\hat{G}(s,a)=G(s,a)+\theta$, where $\theta\sim\theta_0+p(\theta)$ denotes the distributed feedback error and $p(\theta)$ denotes the zero-mean distribution, the proposed DCARL agent can still satisfy the outperforming and continuously improving requirements.

Proof of Theorem 5.

(1) When $\theta_0=0$, with a zero-mean noise distribution, the performance of $\pi_{ci}(s,\mathcal{D})$ always improves from the original policy π_0. We first replace $G(s,a)$ in equation (13), to obtain

$$\begin{aligned}&\bar{G}(s,a)=\frac{1}{n}\sum G(s,a)=\frac{1}{n}\sum\left(\hat{G}(s,a)-\theta\right)=\frac{1}{n}\sum\hat{G}(s,a)-\frac{1}{n}\sum\theta,\frac{1}{n}\sum\theta\to 0\\&\Rightarrow\hat{G}(s,a)=\frac{1}{n}\sum G(s,a)\to\tilde{Q}_{\pi}(s,a)\Rightarrow\mathbb{E}(\mathcal{D}_k)=\mathbb{E}(\hat{\mathcal{D}}_k)\end{aligned}\tag{40}$$

where $\hat{\mathcal{D}}_k$ denotes the dataset with the zero-mean noise distribution. According to the probability theory, the variance of $\hat{\mathcal{D}}_k$ should satisfy that $\sigma(\hat{\mathcal{D}}_k)\geq\sigma(\mathcal{D}_k)$.Then, combining equation (28), it can conclude that

$$\begin{aligned}&\hat{Q}^{-}_{\pi}(s,\pi(s),\hat{\mathcal{D}}_k)=\mathbb{E}(\hat{\mathcal{D}}_k)-j\sigma(\hat{\mathcal{D}}_k)\\&=\mathbb{E}(\mathcal{D}_k)-j\sigma(\hat{\mathcal{D}}_k)\leq\mathbb{E}(\mathcal{D}_k)-j(\delta)\sigma(\mathcal{D}_k)=Q^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)\\&\Rightarrow\hat{Q}^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)\leq Q^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)\end{aligned}\tag{41}$$

Similarly, for the BCV of the baseline policy, we obtain

$$\begin{aligned}&\hat{Q}^{+}_{\pi}(s,\pi(s),\hat{\mathcal{D}}_k)=\mathbb{E}(\hat{\mathcal{D}}_k)+j\sigma(\hat{\mathcal{D}}_k)=\mathbb{E}(\mathcal{D}_k)+j\sigma(\hat{\mathcal{D}}_k)\\&\geq\mathbb{E}(\mathcal{D}_k)+j\sigma(\mathcal{D}_k)=Q^{+}_{\pi}(s,\pi(s),\mathcal{D}_k)\\&\Rightarrow\hat{Q}^{+}_{\pi}(s,\pi(s),\mathcal{D}_k)\geq Q^{+}_{\pi}(s,\pi(s),\mathcal{D}_k)\end{aligned}\tag{42}$$

Then, if the outperforming condition in equation (19) satisfies, it will become

$$\begin{aligned}&\hat{Q}^{-}_{\pi_{ci(\mathcal{D})}}(s,\pi_{ci}(s,\mathcal{D}),\mathcal{D})-\hat{Q}^{+}_{\pi_0}(s,\pi_0(s),\mathcal{D})\geq 0\\&\Rightarrow Q^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)-\hat{Q}^{+}_{\pi_0}(s,\pi_0(s),\mathcal{D})\geq 0\\&\Rightarrow Q^{-}_{\pi}(s,\pi(s),\mathcal{D}_k)-Q^{+}_{\pi}(s,\pi_0(s),\mathcal{D}_k)\geq 0\end{aligned}\tag{43}$$

Thus, under a fixed feedback error, the DCARL policy satisfies the outperforming requirements, namely, for more than $(1-\delta)(1-\delta_0)$ probability:

$$\forall s,\mathcal{D}_k,\mathbb{E}_{\pi(\mathcal{D}_k)}\left[\sum_{t=h}^{h+H-1}r_t(s_t,a_t)|s_h=\hat{s}\right]\geq\mathbb{E}_{\pi_0}\left[\sum_{t=h}^{h+H-1}r_t(s_t,a_t)|s_h=\hat{s}\right]\tag{44}$$

(2) When $\theta_0=0$, more $\mathcal{D}$ results in a higher WCV.

Similarly, equation (22) can be rewritten as:

$$\begin{aligned}&\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_{k+1}),\hat{\mathcal{D}}_{k+1})=\arg\max_{\pi\in\Pi}\left(\hat{Q}^{c}_{\pi}(s,\pi(s),\hat{\mathcal{D}}_{k+1})\right)\\&\geq\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_k),\hat{\mathcal{D}}_{k+1})\geq\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_k),\hat{\mathcal{D}}_k)\geq\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_{k+1}),\hat{\mathcal{D}}_k)\end{aligned}\tag{45}$$

Then

$$\begin{aligned}&\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_{k+1}),\hat{\mathcal{D}}_{k+1})-\hat{Q}^{-}_{\pi}(s,\pi_{ci}(s,\hat{\mathcal{D}}_{k+1}),\hat{\mathcal{D}}_k)\geq 0\\&\Rightarrow\mathbb{E}(\hat{\mathcal{D}}_k)-j\sigma(\hat{\mathcal{D}}_k)-(\mathbb{E}(\hat{\mathcal{D}}_{k+1})-j\sigma(\hat{\mathcal{D}}_{k+1}))\geq 0\end{aligned}\tag{46}$$

As $p(\theta)$ has the same effect on the dataset, $\sigma(\hat{\mathcal{D}}_k)-\sigma(\hat{\mathcal{D}}_{k+1})$is equal to $\sigma(\mathcal{D}_k)-\sigma(\mathcal{D}_{k+1})$.

Then, combining with the zero-mean condition of $p(\theta)$, it can be concluded that

$$Q^{-}_{\pi}(s,\pi_{ci}(s,\mathcal{D}_{k+1}),\mathcal{D}_{k+1})\geq Q^{-}_{\pi}(s,\pi_{ci}(s,\mathcal{D}_{k+1}),\mathcal{D}_k)\tag{47}$$

From all the above, when $\theta_0=0$, this theorem is proved. If $\theta_0\neq 0$, then following the previous theorem, we can also conclude this theorem.

Theorem 6

Assuming two DCARL agents with a different δ (or δ_0) value, denoted as $\pi^{a}_{ci}(s,\mathcal{D})$ and $\pi^{b}_{ci}(s,\mathcal{D})$ (or $\pi^{a_0}_{ci}(s,\mathcal{D})$ and $\pi^{b_0}_{ci}(s,\mathcal{D})$), respectively, we can prove that decreasing the δ or δ_0 value will increase the required data amount for activation, that is

$$\begin{aligned}&\text{If }\delta_a>\delta_b\text{ and given dataset }\mathcal{D},\text{then }\forall s,n\left(\pi^{a}_{ci}(s,\mathcal{D})\right)\leq n\left(\pi^{b}_{ci}(s,\mathcal{D})\right)\\&n\left(\pi_{ci}(s,\mathcal{D}=\{d_1,d_2,\ldots\})\right)=\inf\mathcal{N}(s,\mathcal{D})\\&s.t.\mathcal{N}(s,\mathcal{D})=\{k\in\mathbf{N}|\pi_{ci}(s,\mathcal{D}_k=\{d_1,d_2,\ldots d_k\})\neq\pi_0(s)\}\end{aligned}\tag{48}$$

where $n(\pi_{ci}(s,\mathcal{D}))$ denotes the data amount when the DCARL agent starts to deviate from the baseline policy. Here, a and b are the symbols to distinguish different δ values. $\mathcal{N}$ denotes the normal distribution.

Proof of Theorem 6.

(1) Given $\delta_a>\delta_b$ with a fixed δ_0
According to equation (1), the confidence value for a policy should satisfy that

$$\begin{aligned}&\begin{cases}\mathcal{Q}^{-}_{\delta_a}(s,\pi(s),\mathcal{D})=\{q\in\mathbf{R}|P(\tilde{Q}_{\pi}(s,\pi(s))\geq q|\mathcal{D})\geq 1-\delta_a\}\\\mathcal{Q}^{-}_{\delta_b}(s,\pi(s),\mathcal{D})=\{q\in\mathbf{R}|P(\tilde{Q}_{\pi}(s,\pi(s))\geq q|\mathcal{D})\geq 1-\delta_b\}\\\delta_a>\delta_b\end{cases}\\&\Rightarrow\mathcal{Q}^{-}_{\delta_b}(s,\pi(s),\mathcal{D})\subseteq\mathcal{Q}^{-}_{\delta_a}(s,\pi(s),\mathcal{D})\end{aligned}\tag{49}$$

we obtain

$$\begin{aligned}&\forall Q_{\pi}^{-}\left(s,\pi\left(s\right),\mathcal{D}_{k},\delta_{b}\right)\in\mathcal{Q}_{\delta_{b}}^{-}\left(s,\pi\left(s\right),\mathcal{D}\right)\\&\Rightarrow Q_{\pi}^{-}\left(s,\pi\left(s\right),\mathcal{D}_{k},\delta_{b}\right)\in\mathcal{Q}_{\delta_{a}}^{-}\left(s,\pi\left(s\right),\mathcal{D}\right)\end{aligned}\tag{50}$$

Thus, when $\pi_{ci}^{b}(s,\mathcal{D}_{k})\neq\pi_{0}(s)$, according to equation (3), we obtain

$$Q_{\pi}^{-}\left(s,\pi\left(s\right),\mathcal{D}_{k},\delta_{b}\right)\in\mathcal{Q}_{\delta_{a}}^{-}\left(s,\pi\left(s\right),\mathcal{D}\right)>Q_{\pi_{0}}^{+}\left(s,\pi_{0}\left(s\right),\mathcal{D}_{k},\delta_{0}\right)\tag{51}$$

$$\Rightarrow Q_{\pi}^{-}\left(s,\pi\left(s\right),\mathcal{D}_{k},\delta_{a}\right)>Q_{\pi_{0}}^{+}\left(s,\pi_{0}\left(s\right),\mathcal{D}_{k},\delta_{0}\right)\tag{52}$$

Then, it can be concluded that:

$$\pi_{ci}^{b}(s,\mathcal{D}_{k})\neq\pi_{0}(s)\Rightarrow\pi_{ci}^{a}(s,\mathcal{D}_{k})\neq\pi_{0}(s)\tag{53}$$

Namely, if the DCARL has satisfied the δ_b deviation condition given $\mathcal{D}_k$, then it must satisfy the δ_a deviation condition, that is, $\mathcal{N}_b(s,\mathcal{D})\subseteq\mathcal{N}_a(s,\mathcal{D})$. Thus, combining with equation (48), we obtain:

$$\begin{aligned}&\mathcal{N}_{b}\left(s,\mathcal{D}\right)\subseteq\mathcal{N}_{a}\left(s,\mathcal{D}\right)\Rightarrow\inf\mathcal{N}_{b}\left(s,\mathcal{D}\right)\geq\inf\mathcal{N}_{a}\left(s,\mathcal{D}\right)\\&\Rightarrow n\left(\pi_{ci}^{a}\left(s,\mathcal{D}\right)\right)\leq n\left(\pi_{ci}^{b}\left(s,\mathcal{D}\right)\right)\end{aligned}\tag{54}$$

It is proved that when $\delta_a>\delta_b$ with a fixed δ_0, then $n\left(\pi_{ci}^{a}(s,\mathcal{D})\right)\leq n\left(\pi_{ci}^{b}(s,\mathcal{D})\right)$

(2) Given $\delta_{a0}>\delta_{b0}$ with a fixed δ

Similarly, when $\pi_{ci}^{b_0}(s,\mathcal{D}_k)\neq\pi_0(s)$, according to equation (7):

$$\begin{aligned}&\begin{cases}\mathcal{Q}_{\delta_{a}}^{+}\left(s,\pi\left(s\right),\mathcal{D}\right)=\left\{q\in\mathbf{R}|P\left(\tilde{Q}_{\pi}\left(s,\pi\left(s\right)\right)\leq q|\mathcal{D}\right)\geq1-\delta_{a0}\right\}\\\mathcal{Q}_{\delta_{b}}^{+}\left(s,\pi\left(s\right),\mathcal{D}\right)=\left\{q\in\mathbf{R}|P\left(\tilde{Q}_{\pi}\left(s,\pi\left(s\right)\right)\leq q|\mathcal{D}\right)\geq1-\delta_{b0}\right\}\\\delta_{a0}>\delta_{b0}\end{cases}\\&\Rightarrow\mathcal{Q}_{\delta_{b}}^{+}\left(s,\pi\left(s\right),\mathcal{D}\right)\subseteq\mathcal{Q}_{\delta_{a}}^{+}\left(s,\pi\left(s\right),\mathcal{D}\right)\end{aligned}\tag{55}$$

we obtain

$$\begin{aligned}&\forall Q_{\pi}^{+}\left(s,\pi\left(s\right),\mathcal{D}_{k},\delta_{b0}\right)\in\mathcal{Q}_{\delta_{b0}}^{+}\left(s,\pi\left(s\right),\mathcal{D}\right)\\&\Rightarrow Q_{\pi}^{+}\left(s,\pi\left(s\right),\mathcal{D}_{k},\delta_{b0}\right)\in\mathcal{Q}_{\delta_{a0}}^{+}\left(s,\pi\left(s\right),\mathcal{D}\right)\end{aligned}\tag{56}$$

Thus, when $\pi_{ci}^{b}(s,\mathcal{D}_k)\neq\pi_0(s)$, according to equation (3), we obtain

$$Q_{\pi}^{-}\left(s,\pi\left(s\right),\mathcal{D}_{k},\delta\right)>Q_{\pi_{0}}^{+}\left(s,\pi_{0}\left(s\right),\mathcal{D}_{k},\delta_{b0}\right)\in\mathcal{Q}_{\delta_{a0}}^{+}\left(s,\pi\left(s\right),\mathcal{D}\right)\tag{57}$$

$$\Rightarrow Q_{\pi}^{-}\left(s,\pi\left(s\right),\mathcal{D}_{k},\delta\right)>Q_{\pi_{0}}^{+}\left(s,\pi_{0}\left(s\right),\mathcal{D}_{k},\delta_{a0}\right)\tag{58}$$

Combining with equation (3), it can be concluded that:

$$\pi_{ci}^{b_0}(s,\mathcal{D}_{k})\neq\pi_{0}(s)\Rightarrow\pi_{ci}^{a_0}(s,\mathcal{D}_{k})\neq\pi_{0}(s)\tag{59}$$

Similarly, with equation (48), we obtain

$$\begin{aligned}&\mathcal{N}_{b0}\left(s,\mathcal{D}\right)\subseteq\mathcal{N}_{a0}\left(s,\mathcal{D}\right)\Rightarrow\inf\mathcal{N}_{b0}\left(s,\mathcal{D}\right)\geq\inf\mathcal{N}_{a0}\left(s,\mathcal{D}\right)\\&\Rightarrow n\left(\pi_{ci}^{a_0}\left(s,\mathcal{D}\right)\right)\leq n\left(\pi_{ci}^{b_0}\left(s,\mathcal{D}\right)\right)\end{aligned}\tag{60}$$

It is proved that when $\delta_{a0}>\delta_{b0}$ with a fixed δ, $n\left(\pi_{ci}^{a_0}(s,\mathcal{D})\right)\leq n\left(\pi_{ci}^{b_0}(s,\mathcal{D})\right)$.

Combining equations (54) and (60), this theorem is proved.

Data availability

The Supplementary Software file contains the minimum data to run and render the results for all three experiments. These data are also available in a public repository at https://github.com/zhcao92/DCARL (ref. [37]).

Code availability

The source code of the self-driving experiments is available at https://github.com/zhcao92/DCARL (ref. [38]). It contains the proposed DCARL planning algorithms as well as the used perception, localization and control algorithms in our self-driving cars.

References

1. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (MIT Press, 2018).
2. Silver, D. et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **362**, 1140–1144 (2018).
3. Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
4. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
5. Ye, F., Zhang, S., Wang, P. & Chan, C.-Y. A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles. *In 2021 IEEE Intelligent Vehicles Symposium (IV)* 1073–1080 (IEEE, 2021).
6. Zhu, Z. & Zhao, H. A survey of deep RL and IL for autonomous driving policy learning. *IEEE Trans. Intell. Transp. Syst.* **23**, 14043–14065 (2022).
7. Aradi, S. Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **23**, 740–759 (2022).
8. Cao, Z. et al. Highway exiting planner for automated vehicles using reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* **22**, 990–1000 (2020).
9. Stilgoe, J. Self-driving cars will take a while to get right. *Nat. Mach. Intell.* **1**, 202–203 (2019).
10. Kalra, N. & Paddock, S. M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. Part A* **94**, 182–193 (2016).
11. Disengagement reports. *California DMV* https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/disengagement-reports/ (2021).
12. Li, G. et al. Decision making of autonomous vehicles in lane change scenarios: deep reinforcement learning approaches with risk awareness. *Transp. Res. Part C* **134**, 103452 (2022).
13. Shu, H., Liu, T., Mu, X. & Cao, D. Driving tasks transfer using deep reinforcement learning for decision-making of autonomous vehicles in unsignalized intersection. *IEEE Trans. Veh. Technol.* **71**, 41–52 (2021).
14. Pek, C., Manzinger, S., Koschi, M. & Althoff, M. Using online verification to prevent autonomous vehicles from causing accidents. *Nat. Mach. Intell.* **2**, 518–528 (2020).
15. Xu, S., Peng, H., Lu, P., Zhu, M. & Tang, Y. Design and experiments of safeguard protected preview lane keeping control for autonomous vehicles. *IEEE Access* **8**, 29944–29953 (2020).
16. Yang, J., Zhang, J., Xi, M., Lei, Y. & Sun, Y. A deep reinforcement learning algorithm suitable for autonomous vehicles: double bootstrapped soft-actor-critic-discrete. *IEEE Trans. Cogn. Dev. Syst.* https://doi.org/10.1109/TCDS.2021.3092715 (2021).
17. Schwall, M., Daniel, T., Victor, T., Favaro, F. & Hohnhold, H. Waymo public road safety performance data. Preprint at arXiv https://doi.org/10.48550/arXiv.2011.00038 (2020).

18. Fan, H. et al. Baidu Apollo EM motion planner. Preprint at arXiv https://doi.org/10.48550/arXiv.1807.08048 (2018).
19. Kato, S. et al. Autoware on board: enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems* 287–296 (IEEE, 2018).
20. Cao, Z., Xu, S., Peng, H., Yang, D. & Zidek, R. Confidence-aware reinforcement learning for self-driving cars. *IEEE Trans. Intell. Transp. Syst.* **23**, 7419–7430 (2022).
21. Thomas, P. S. et al. Preventing undesirable behavior of intelligent machines. *Science* **366**, 999–1004 (2019).
22. Levine, S., Kumar, A., Tucker, G. & Fu, J. Offline reinforcement learning: tutorial, review, and perspectives on open problems. Preprint at arXiv https://doi.org/10.48550/arXiv.2005.01643 (2020).
23. Garcıa, J. & Fernández, F. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.* **16**, 1437–1480 (2015).
24. Achiam, J., Held, D., Tamar, A. & Abbeel, P. Constrained policy optimization. In *International Conference on Machine Learning* 22–31 (JMLR, 2017).
25. Berkenkamp, F., Turchetta, M., Schoellig, A. & Krause, A. Safe model-based reinforcement learning with stability guarantees. *Adv. Neural Inf. Process. Syst.* **30**, 908-919 (2017).
26. Ghadirzadeh, A., Maki, A., Kragic, D. & Björkman, M. Deep predictive policy training using reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems* 2351–2358 (IEEE, 2017).
27. Abbeel, P. & Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proc. Twenty-first International Conference on Machine Learning*, 1 (Association for Computing Machinery, 2004).
28. Abbeel, P. & Ng, A. Y. Exploration and apprenticeship learning in reinforcement learning. In *Proc. 22nd International Conference on Machine Learning* 1–8 (Association for Computing Machinery, 2005).
29. Ross, S., Gordon, G. & Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In Gordon, G., Dunson, D. & Dudík, M. (eds) *Proc. Fourteenth International Conference on Artificial Intelligence and Statistics*, 627–635 (JMLR, 2011).
30. Zhang, J. & Cho, K. Query-efficient imitation learning for end-to-end autonomous driving. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, 2891–2897 (AAAI Press, 2017).
31. Bicer, Y., Alizadeh, A., Ure, N. K., Erdogan, A. & Kizilirmak, O. Sample efficient interactive end-to-end deep learning for self-driving cars with selective multi-class safe dataset aggregation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems* 2629–2634 (IEEE, 2019).
32. Alshiekh, M. *et al* Safe reinforcement learning via shielding. In *Proc. Thirty-Second AAAI Conference on Artificial Intelligence* Vol. 32, 2669-2678 (AAAI Press, 2018).
33. Brun, W., Keren, G., Kirkeboen, G. & Montgomery, H. *Perspectives on Thinking, Judging, and Decision Making* (Universitetsforlaget, 2011).
34. Dabney, W. et al. A distributional code for value in dopamine-based reinforcement learning. *Nature* **577**, 671–675 (2020).
35. Cao, Z. et al. A geometry-driven car-following distance estimation algorithm robust to road slopes. *Transp. Res. Part C* **102**, 274–288 (2019).
36. Xu, S. et al. System and experiments of model-driven motion planning and control for autonomous vehicles. *IEEE Trans. Syst. Man. Cybern. Syst.* **52**, 5975–5988 (2022).
37. Cao, Z. Codes and data for dynamic confidence-aware reinforcement learning. DCARL. *Zenodo* https://zenodo.org/badge/latestdoi/578512035 (2022).
38. Kochenderfer, M. J. *Decision Making Under Uncertainty: Theory and Application* (MIT Press, 2015).
39. Ivanovic, B. et al. Heterogeneous-agent trajectory forecasting incorporating class uncertainty. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 12196–12203 (IEEE, 2022).
40. Yang, Y., Zha, K., Chen, Y., Wang, H. & Katabi, D. Delving into deep imbalanced regression. *In International Conference on Machine Learning* 11842–11851 (PMLR, 2021).
41. Efron, B. & Tibshirani, R. J. *An Introduction to the Bootstrap* (CRC Press, 1994).
42. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A. & Koltun, V. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, 1–16 (PMLR, 2017).

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) (U1864203 (D.Y.), 52102460 (Z.C.), 61903220 (K.J.)), China Postdoctoral Science Foundation (2021M701883 (Z.C.)) and Beijing Municipal Science and Technology Commission (Z221100008122011 (D.Y.)). It is also funded by the Tsinghua University-Toyota Joint Center (D.Y.).

Author contributions

Z.C., S.X., D.Y. and H.P. developed the performance improvement technique, which can outperform the existing self-driving policy. Z.C. and W.Z. developed the continuous improvement technique using the worst confidence value. Z.C., S.X. and W.Z. designed the whole self-driving platform in the real world. Z.C., K.J. and D.Y. designed and conducted the experiments and collected the data.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-023-00610-y.

Correspondence and requests for materials should be addressed to Diange Yang.

Peer review information *Nature Machine Intelligence* thanks Ali Alizadeh and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.