SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2026, Vol. 69, Iss. 12, 222204:1–222204:21
https://doi.org/10.1007/s11432-024-5222-9

A sampling efficient optimization method in constrained reinforcement learning for cyber-physical systems

Siwei LIU & Qing-Shan JIA*

Center for Intelligent and Networked Systems, Department of Automation, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China

Received 23 December 2024/Revised 14 January 2026/Accepted 6 February 2026/Published online 24 November 2026

Abstract With the development of cyber-physical systems, optimizing policies with the pervasive existence of constraints has become a bottleneck. Thus, constrained reinforcement learning has become of great practical interest. Compared with constraints based on state space, simulation-based constraints are more challenging to deal with due to the time-consuming evaluation of both performance and feasibility. Considering this hard problem, we propose a sampling efficient algorithm based on the optimal computing budget allocation (OCBA) strategy and combine it with a typical reinforcement learning method, the deep Q-network (DQN). In addition, we provide theoretical analysis of convergence and test the algorithm experimentally with typical cyber-physical systems. According to the experimental results, the proposed algorithm can indeed achieve comparable returns with much fewer constraint violations during training.

Keywords constrained reinforcement learning, sampling efficiency, optimal computing budget allocation, cyber-physical systems, control policy optimization

Citation Liu S W, Jia Q-S. A sampling efficient optimization method in constrained reinforcement learning for cyber-physical systems. Sci China Inf Sci, 2026, 69(12): 222204, https://doi.org/10.1007/s11432-024-5222-9

1 Introduction

In the past decade, the constrained Markov decision process (CMDP) [1] has achieved great success in solving control and planning problems in cyber-physical systems, such as automated control tasks in robotic locomotion [2], navigation [3], transportation management [4,5], autonomous driving [6], and other complex systems [7–10]. Meanwhile, in recent years, numerous studies focusing on the broad learning system (BLS) [11, 12], dynamic coverage control [13], and the collaboration issues within the system [14–16] have achieved further advancements. CMDP can work better due to the pervasive existence of various constraints in applications of cyber-physical systems. Bertsekas [17] used Newton-type gradient-based methods to explain its success, whereas Cao [18] discussed it from the viewpoint of sensitivity analysis.

There are mainly three types of constraints in a constrained Markov decision process [19]. First, the constraint is directly on the state space, which means that certain states are forbidden for the system. Methods based on control barrier functions [6, 20] are examples provided to solve this type of problem. In addition, the constraint is on the behavior of the system. For those that can be described as a constraint function which is additive over time, a typical method is to use Lagrangian relaxation for equality constraints and the Karush-Kuhn-Tucker (KKT) conditions for equality and inequality constraints [1]. For variance constraints, Xia et al. [21–23] has conducted relevant research. Finally, some constraints are on the temporal logic of system behavior, such as temporal logic constraints (TLC) or computation tree logic (CTL) constraints [24–26].

In this article, we focus on the constraints on the performance of policies, especially those that can only be evaluated by simulations. These types of constraints will bring three challenges. First, the performance and feasibility of one policy must be evaluated by simulations, which could be time-consuming. Second, it is hard to use linear or non-linear programming solvers to deal with these problems, especially ones with large state and action spaces. Third, there is always an inherent conflict between optimization for performance and feasibility.

Among methods for CMDP, constrained reinforcement learning (CRL) is widely verified as a useful method. Safe reinforcement learning (RL) [1] is proposed to solve safety constraint problems and also improve the safety

* Corresponding author (email: jiaqs@tsinghua.edu.cn)

© Science China Press 2026

constraints of policies during training. There are usually two types of safe RL. The first is to use model-based RL (MBRL), which uses sample data to build transition models and assist in performance enhancement and judging feasibility during the training process [27, 28]. Some recent studies have already considered the combination of CMDP with MBRL [29,30]. In addition, several studies pay attention to the uncertainty of the learned model [31]. Although MBRL has achieved a lot for constraints on the state space, its insufficient understanding of the dynamics, especially for simulations for performance and feasibility for policies, is not suitable for our proposed problems. The other type is model-free RL (MFRL). This type of method always solves the problems by giving an additional cost signal indicating constraints and setting thresholds on the cost. Representative approaches include penalty function methods [32], Lagrange multiplier methods [33–35], and projection methods [36]. These methods enhance the expectation of safety by calculating cumulative costs during simulations. Others replace cumulative costs with safety certificates such as the energy function [37] or the reachability certificate [38] to ensure the satisfaction of the safety restrictions in each state. MFRL can deal with problems with high stochasticity, but also faces time-consuming challenges. We can use them to evaluate and predict the performance and feasibility of policies, but we still need to solve the difficulties in sampling efficiency.

The sampling efficiency in RL has also attracted growing attention in recent years, especially for simulation-based evaluation of performance and feasibility. Some methods consider using techniques such as addressing model error [39], transferring between scenarios [40, 41], safety evaluation [42, 43], etc. We have noticed that some methods based on computing budget allocation problems in simulation-based optimization are indeed suitable for our problems. Among the rich development, the optimal computing budget allocation (OCBA) [44] is particularly relevant to simulation-based CMDP and can well combine and utilize uncertainty. With uncertain information, OCBA aims to asymptotically maximize the probability of correctly selecting the best solution candidate [45]. This method can also be used in each step of optimization for RL methods.

In this work, we propose a constrained reinforcement learning method combined with OCBA-like techniques. This method first uses information sampled from the simulators to evaluate and predict the performance and feasibility of each action. Following the direction of the OCBA-like algorithm, it assigns a reasonable probability to each action, which can asymptotically maximize the probability of correctly selecting the best action with infinite sampling for the current situation. Finally, we apply it to some typical simulation-based constrained cyber-physical systems and find that it can indeed enhance the performance and feasibility of the policies during the training process and also has high sample efficiency, meaning it can obtain a good policy more quickly than baselines.

The main contributions are as follows.

- We propose a sampling efficient optimization method in CRL for cyber-physical systems. It can use OCBA-like methods to enhance the sampling process of the CRL.
- We give a method to provide the predictive-uncertainty estimation during the training process. It will help the OCBA-like method to be effective for both tabular representations and nonlinear function approximations. Thus, it can be applied to almost all common RL settings.
- We combine our method with a typical deep reinforcement learning method, DQN, as an example, and validate its effectiveness through numerical experiments with several representative cyber-physical systems. The results show that our algorithm remarkably improves the performance compared with the baseline algorithms both in feasibility improvement and sample efficiency.

The rest of this paper is organized as follows. We provide the preliminaries in Section 2, introduce the proposed algorithms, and give the convergence analysis in Section 3. In Section 4, we show the main experimental results. Finally, we give a brief conclusion in Section 5.

2 Preliminaries

In this section, we first introduce constrained reinforcement learning. Then, an outline of OCBA will be presented to develop our sampling strategy in the next section for reinforcement learning.

2.1 Constrained reinforcement learning

A sequential decision-making problem with inequality constraints in cyber-physical systems can be formulated as a CMDP $\mathcal{S}, \mathcal{A}, P, r, c, \gamma$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces. $P(s'|s,a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ denotes the one-step state transition probability from $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ with action $a \in \mathcal{A}$. $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, $r(s,a)$ is the immediate reward when action a is taken at state s. $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the constraint function when action a is taken at state s. Action a is often considered feasible for state s when $c(s,a) \leqslant 0$. $c(s,a)$ can be freely defined according

to the needs of the system. γ is the discount factor. At each time step $t \in \mathbb{N}$, the agent will choose action a_t with observation state s_t for the next time step according to the policy $\pi(s,a) : \mathcal{S} \times \mathcal{A} \to [0,1] \in \Pi$, which means the probability of selecting action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. Furthermore, we can use $\pi(s) := \arg\max_a \pi(s,a)$ to describe a deterministic policy. Given the initial state distribution $d_0(s)$, we can obtain a constrained Markov decision process as follows:

$$\begin{aligned} \max_{\pi \in \Pi} \quad & \mathbb{E}_{s_t,a_t \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) | d_0(s) \right] \\ \text{s.t.} \quad & \mathbb{E}_{s_t,a_t \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t c\left(s_t, a_t\right) | d_0(s) \right] \leqslant \eta, \end{aligned} \tag{1}$$

where $\eta \in \mathbb{R}$ is the cost threshold.

Considering that we often only pay attention to the feasibility of policies, we define a cost signal $h(s,a) = I(c(s,a) > 0)$, where $I(\cdot)$ is the indicator function. $h(s,a) = 1$ means that the constraints have been violated and the episode will always end at once in practice. Moreover, if we just replace $c(s,a)$ with $h(s,a)$ directly

$$\begin{aligned} & \mathbb{E}_{s_t,a_t \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t h\left(s_t, a_t\right) | d_0(s) \right] \\ & \quad = \mathbb{E}_{s_t,a_t \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t I(\{c(s_t,a_t) > 0\}) | d_0(s) \right] \\ & \quad = \sum_{t=0}^{\infty} \left\{ \mathbb{E}_{s_t,a_t \sim \pi} \left[\gamma^t I(\{c(s_t,a_t) > 0\}) | d_0(s) \right] \right\} \\ & \quad = \sum_{t=0}^{\infty} \Pr\left(c(s_t,a_t) > 0\right) \\ & \quad \geqslant \Pr\left(\bigcup_{t=0}^{\infty} c(s_t,a_t) > 0 \right) \\ & \quad = \Pr(\text{the constraints are violated}). \end{aligned} \tag{2}$$

According to the Bonferroni inequality, they are equal only when $\gamma = 1$ and the episode terminates with a violation of the constraint. Thus, we mainly consider the probability of constraint satisfaction and set $\gamma = 1$ for the constraint equation, which is also quite common in practical problems. Finally, we get the CMDP problems as follows:

$$\begin{aligned} \max_{\pi \in \Pi} \quad & \mathbb{E}_{s_t,a_t \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) | d_0(s) \right] \\ \text{s.t.} \quad & \mathbb{E}_{s_t,a_t \sim \pi} \left[\sum_{t=0}^{\infty} h\left(s_t, a_t\right) | d_0(s) \right] \leqslant \eta, \end{aligned} \tag{3}$$

where η is also the cost threshold, but $\eta \in [0,1]$ now and means a probability request for constraints.

Constrained reinforcement learning is one of the effective methods to solve the problem as (3). We can define the action value function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ of the policy π as follows:

$$Q^{\pi}(s,a) = r(s,a) + \gamma \sum_{s'} P(s'|s,\pi(s)) Q^{\pi}(s',\pi(s)). \tag{4}$$

Similarly, the action constraint value function $Q_H^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as follows:

$$Q_H^{\pi}(s,a) = h(s,a) + \sum_{s'} P(s'|s,\pi(s)) Q_H^{\pi}(s',\pi(s)). \tag{5}$$

However, the state transition probability is always unknown, and it may be the same for the reward function and cost signal. We can only learn from past interactions with the environment, which is the basic idea of model-free constrained reinforcement learning.

2.2 Optimal computing budget allocation

OCBA [44] is an effective method for resource allocation optimization in the field of simulation-based problems, which include those for cyber-physical systems. For the k alternatives, whose performance follows Gaussian distributions $\mathcal{N}(\mu_i, \sigma_i^2), i = 1, \ldots, k$, the method aims to choose the best alternative in the sense of mean performance with higher confidence, using given simulation replications.

Allocate the total T simulation replications to the k alternatives, each one gets N_i replications: $\{X_i^m\}_{m=1}^{N_i}$. So, we have

$$\begin{aligned} &\sum_{i=1}^{k} N_i = T, \\ &X := \{\{X_i^m\}_{m=1}^{N_i}\}_{i=1}^{k}. \end{aligned} \tag{6}$$

We can use a Gaussian distribution to approximate the posterior distribution of μ_i with a non-informative prior distribution

$$\hat{\mu}_i \sim \mathcal{N}\left(\frac{1}{N_i}\sum_{m=1}^{N_i} X_i^m, \frac{\sigma_i^2}{N_i}\right). \tag{7}$$

We define the probability of correct selection (PCS) as the confidence in selecting the best performance choice $b \in 1, \ldots, k$, which we want to maximize

$$\mathrm{PCS} = \Pr\left(\bigcap_{i \neq b} (\hat{\mu}_b > \hat{\mu}_i) \,|X\right). \tag{8}$$

For a simpler calculation, the approximate probability of correct selection (APCS) is introduced [45]

$$\mathrm{APCS} = 1 - \sum_{i \neq b} \Pr(\hat{\mu}_b \leqslant \hat{\mu}_i). \tag{9}$$

Finally, we can formulate the problem of OCBA with a given simulation replication amount

$$\begin{aligned} &\arg\max_{N_1,\ldots,N_k} \ \mathrm{APCS} \\ &\text{s.t.} \ \ \sum_{i=1}^{k} N_i = T, \end{aligned} \tag{10}$$

and the optimal solution is

$$\begin{aligned} &\frac{N_i}{N_j} = \left(\frac{\sigma_i/\delta_{b,i}}{\sigma_j/\delta_{b_j}}\right)^2, \quad i \neq b \text{ and } \ j \neq b, \\ &N_b = \sigma_b \left(\sum_{i \neq b} \frac{N_i^2}{\sigma_i^2}\right)^{\frac{1}{2}}, \\ &\sum_{i=1}^{k} N_i = T. \end{aligned} \tag{11}$$

3 Methodology

3.1 Sampling efficient constrained Q-learning

Q-learning is a typical reinforcement learning method, whose $Q(s,a)$ and $Q_H(s,a)$ can be described in a tabular way for the whole state space and action space. Thus, we can estimate $Q(s,a)$ and $Q_H(s,a)$ through iteration with sampled data, i.e., we can get the $(k+1)$-th estimation $\hat{Q}^{k+1}(s,a)$ and $\hat{Q}_H^{k+1}(s,a)$ as follows:

$$\begin{aligned} &\hat{Q}^{k+1}(s,a) = r(s,a) + \gamma \frac{1}{N_a}\sum_{i=1}^{N_a} \hat{Q}^k(s_i', \pi(s_i')), \\ &\hat{Q}_H^{k+1}(s,a) = h(s,a) + \frac{1}{N_a}\sum_{i=1}^{N_a} \hat{Q}_H^k(s_i', \pi(s_i')), \end{aligned} \tag{12}$$

where N_a is the number of replications of simulations assigned to action a, and s_i' is the next state obtained from the environment with the state-action pair s and a in the i-th simulation. With (12), we know all the action-value estimations can converge to Q^* and Q_H^* with probability 1 (w.p.1) [46]. Thus, we can convert the problem of (3) to the problem of finding the best feasible action in each step as follows:

$$\begin{aligned} &\max_a \ \hat{Q}^{k+1}(s,a) \\ &\text{s.t.} \quad \hat{Q}_H^{k+1}(s,a) \leqslant \eta. \end{aligned} \tag{13}$$

Similarly to the idea from the OCBA, we introduce the Bayesian viewpoint to (13). We can regard the samples $\hat{Q}^{k+1}(s,a)$ and $\hat{Q}_H^{k+1}(s,a)$ as fixed, and then consider the true values of $Q^{k+1}(s,a)$ and $Q_H^{k+1}(s,a)$ as random variables following the Gaussian posterior distribution as (14)

$$\begin{aligned} Q^{k+1}(s,a) &\sim \mathcal{N}\left(\hat{Q}^{k+1}(s,a), \sigma_a^2\right), \\ Q_H^{k+1}(s,a) &\sim \mathcal{N}\left(\hat{Q}_H^{k+1}(s,a), \sigma_{H,a}^2\right). \end{aligned} \tag{14}$$

We aim to find the observed best feasible action $\hat{a}_b$:

$$\hat{a}_b := \arg\max_{a\in\mathcal{A}} \left\{\hat{Q}^{k+1}(s,a) | a : \hat{Q}_H^{k+1}(s,a) \leqslant \eta\right\}, \tag{15}$$

and hope it is indeed the best feasible action a_b^{k+1} for the true value

$$a_b := \arg\max_{a\in\mathcal{A}} \left\{Q^{k+1}(s,a) | a : Q_H^{k+1}(s,a) \leqslant \eta\right\}. \tag{16}$$

We can also introduce a PCS problem into our problem as follows:

$$\begin{aligned} \text{PCS} &= \Pr\{a_b = \hat{a}_b\} \\ &= \Pr\left\{\left(\bigcap_{\text{feasible } a_i \neq a_b} Q^{k+1}(s,a_b) > Q^{k+1}(s,a_i)\right) \bigcap \left(\bigcap_{\text{infeasible } a_i} Q_H^{k+1}(s,a_i) > \eta\right)\right. \\ &\quad \left.\bigcap \left(\bigcap_{\text{feasible } a_i} Q_H^{k+1}(s,a_i) \leqslant \eta\right)\right\}. \end{aligned} \tag{17}$$

With the Bonferroni inequality, we have

$$\begin{aligned} \text{PCS} \geqslant 1 &- \sum_{\text{feasible } a_i \neq a_b} \left(1 - \Pr\left\{Q^{k+1}(s,a_b) > Q^{k+1}(s,a_i)\right\}\right) \\ &- \sum_{\text{infeasible } a_i} \left(1 - \Pr\left\{Q_H^{k+1}(s,a_i) > \eta\right\}\right) - \sum_{\text{feasible } a_i} \left(1 - \Pr\left\{Q_H^{k+1}(s,a_i) \leqslant \eta\right\}\right). \end{aligned} \tag{18}$$

Similarly, we call the right hand side of the above inequality APCS (for short). So our new problem is

$$\begin{aligned} &\max \ \text{APCS} \\ &\text{s.t.} \quad \sum_i N_i = N. \end{aligned} \tag{19}$$

Denote

$$\begin{aligned} \sigma_{b,i}^2 &= \frac{\sigma_b^2}{N_b} + \frac{\sigma_i^2}{N_i}, \\ \delta_{b,i} &= \hat{Q}^{k+1}(s,a_b) - \hat{Q}^{k+1}(s,a_i), \\ \delta_{H,i} &= \hat{Q}^{k+1}(s,a_i) - \eta, \\ \sigma_{H,i}^2 &= \frac{\sigma_{h,i}^2}{N_i}, \end{aligned} \tag{20}$$

where $\sigma_{h,i}^2$ is the variance of a single observation of $Q_H^{k+1}(s,a_i)$. Using similar analysis to that in the OCBA, we can obtain the following Theorem 1.

Theorem 1. The following computing budget allocation rule asymptotically maximizes APCS when $N \to \infty$ and $\delta_{b,i}/\sigma_i \neq \delta_{H,i}/\sigma_{h,i}$,

$$\beta_i = \begin{cases} \max\left[\left(\dfrac{\sigma_i}{\delta_{b,i}}\right)^2, \left(\dfrac{\sigma_{h,i}}{\delta_{H,i}}\right)^2\right], & \text{feasible } a_i \neq a_b, \\ \left(\dfrac{\sigma_{h,i}}{\delta_{H,i}}\right)^2, & \text{infeasible } a_i. \end{cases} \tag{21}$$

- For $i, j \neq b$,

$$\frac{N_i}{N_j} = \frac{\beta_i}{\beta_j}. \tag{22}$$

- For N_b, we have

$$N_b = \sigma_b \sqrt{\sum_{\text{feasible } a_i \neq \hat{a}_b} \frac{N_i^2}{\sigma_i^2}}. \tag{23}$$

Proof. Our problem is

$$\begin{aligned} &\max_{N_1,\ldots,N_{|\mathcal{A}|}} \quad \text{APCS} \\ &\text{s.t.} \quad \sum_i^{|\mathcal{A}|} N_i \leqslant N. \end{aligned} \tag{24}$$

Define the Lagrangian function

$$F = \text{APCS} + \lambda \left(\sum_i^{|\mathcal{A}|} N_i - N \right). \tag{25}$$

For a feasible action i, we have

$$\begin{aligned} \frac{\partial F}{\partial N_i} =& - \frac{\partial}{\partial N_i} \int_{-\infty}^{-\frac{\delta_{b,i}}{\sigma_{b,i}}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t - \frac{\partial}{\partial N_i} \int_{-\frac{\delta_{H,i}}{\sigma_{H,i}}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t + \lambda \\ =& - \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\right) \frac{\partial}{\partial \sigma_{b,i}} \left(-\frac{\delta_{b,i}}{\sigma_{b,i}}\right) \frac{\partial \sigma_{b,i}}{\partial N_i} + \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{H,i}^2}\right) \frac{\partial}{\partial \sigma_{H,i}} \left(-\frac{\delta_{H,i}}{\sigma_{H,i}}\right) \frac{\partial \sigma_{H,i}}{\partial N_i} + \lambda, \end{aligned} \tag{26}$$

where we have

$$\begin{aligned} &\frac{\partial}{\partial \sigma_{b,i}} \left(-\frac{\delta_{b,i}}{\sigma_{b,i}}\right) = \frac{\delta_{b,i}}{\sigma_{b,i}^2}, \\ &\frac{\partial \sigma_{b,i}}{\partial N_i} = \left(-\frac{1}{2}\right) \frac{\sigma_i^2}{\sigma_{b,i} N_i^2}, \\ &\frac{\partial}{\partial \sigma_{H,i}} \left(-\frac{\delta_{H,i}}{\sigma_{H,i}}\right) = \frac{\delta_{H,i}}{\sigma_{H,i}^2}, \\ &\frac{\partial \sigma_{H,i}}{\partial N_i} = \left(-\frac{1}{2}\right) \sigma_{h,i} N_i^{-\frac{3}{2}}. \end{aligned} \tag{27}$$

So we have

$$\frac{\partial F}{\partial N_i} = \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\right) \frac{\delta_{b,i}}{\sigma_{b,i}^3} \frac{\sigma_i^2}{N_i^2} - \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{H,i}^2}\right) \frac{\delta_{H,i}}{\sigma_{H,i}^2} \frac{\sigma_{h,i}}{N_i^{\frac{3}{2}}} + \lambda = 0. \tag{28}$$

For an infeasible action i, we have

$$\begin{aligned} \frac{\partial F}{\partial N_i} =& - \frac{\partial}{\partial N_i} \int_{-\infty}^{-\frac{\delta_{H,i}}{\sigma_{H,i}}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t + \lambda \\ =& - \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{H,i}^2}\right) \frac{\partial}{\partial \sigma_{H,i}} \left(-\frac{\delta_{H,i}}{\sigma_{H,i}}\right) \frac{\partial \sigma_{H,i}}{\partial N_i} + \lambda. \end{aligned} \tag{29}$$

With (27), we have

$$\frac{\partial F}{\partial N_i} = \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{H,i}^2}\right)\frac{\delta_{H,i}}{\sigma_{H,i}^2}\frac{\sigma_{h,i}}{N_i^{\frac{3}{2}}} + \lambda \;=\; 0. \tag{30}$$

For the observed best feasible action $\hat{a}_b$, we have

$$\begin{aligned}\frac{\partial F}{\partial N_b} &= -\sum_{\text{feasible } a_i \neq \hat{a}_b} \frac{\partial}{\partial N_b}\int_{-\infty}^{-\frac{\delta_{b,i}}{\sigma_{b,i}}} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{t^2}{2}\right)\mathrm{d}t - \frac{\partial}{\partial N_b}\int_{-\frac{\delta_{H,i}}{\sigma_{H,i}}}^{\infty}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{t^2}{2}\right)\mathrm{d}t + \lambda \\ &= \sum_{\text{feasible } a_i \neq \hat{a}_b} \frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\right)\frac{\delta_{b,i}}{\sigma_{b,i}^3}\frac{\sigma_b^2}{N_b^2} - \frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{H,b}^2}{\sigma_{H,b}^2}\right)\frac{\delta_{H,b}}{\sigma_{H,b}^2}\frac{\sigma_{h,b}}{N_b^{\frac{3}{2}}} + \lambda \\ &= 0.\end{aligned} \tag{31}$$

From (28), we have

$$\frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\right) = -\frac{N_i^2}{\sigma_i^2}\left(\lambda + \frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{H,i}^2}\right)\frac{\delta_{H,i}}{\sigma_{H,i}^2}\frac{\sigma_{h,i}}{N_i^{\frac{3}{2}}}\right). \tag{32}$$

Substituting this equation into (31), we have

$$\begin{aligned}\frac{\partial F}{\partial N_b} = &\sum_{\text{feasible } a_i \neq \hat{a}_b}\left[\frac{N_i^2}{\sigma_i^2}\frac{\sigma_b^2}{N_b^2}\frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{H,i}^2}\right)\frac{\delta_{H,i}}{\sigma_{H,i}^2}\frac{\sigma_{h,i}}{N_i^{\frac{3}{2}}}\right] - \frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{H,b}^2}{\sigma_{H,b}^2}\right)\frac{\delta_{H,b}}{\sigma_{H,b}^2}\frac{\sigma_{h,b}}{N_b^{\frac{3}{2}}} \\ &+ \left(1 - \sum_{\text{feasible } a_i \neq \hat{a}_b}\frac{N_i^2}{\sigma_i^2}\frac{\sigma_b^2}{N_b^2}\right)\lambda \\ = &\,0.\end{aligned} \tag{33}$$

Let

$$N_i = \alpha_i N, \quad i = 1, 2, \ldots, |\mathcal{A}|. \tag{34}$$

Then $\sum_i \alpha_i = 1$. With (20), we have

$$\begin{aligned}\frac{\partial F}{\partial N_b} = &\sum_{\text{feasible } a_i \neq \hat{a}_b}\left[\frac{\alpha_i^2}{\sigma_i^2}\frac{\sigma_b^2}{\alpha_b^2}\frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right)\frac{\delta_{H,i}}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_i N}}\right] \\ &- \frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{H,b}^2}{\sigma_{h,b}^2}\alpha_b N\right)\frac{\delta_{H,b}}{\sigma_{h,b}}\frac{1}{\sqrt{\alpha_b N}} + \left(1 - \sum_{\text{feasible } a_i \neq \hat{a}_b}\frac{\alpha_i^2}{\sigma_i^2}\frac{\sigma_b^2}{\alpha_b^2}\right)\lambda \\ = &\,0.\end{aligned} \tag{35}$$

When $N \to \infty$, we have

$$\lim_{N\to\infty}\frac{1}{\sqrt{N}} = 0, \tag{36}$$

and also, for feasible i:

- $\delta_{H,i} \neq 0$,

$$\begin{aligned}&-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i < 0, \\ &\Rightarrow \lim_{N\to\infty}\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right) = 0, \\ &\Rightarrow \lim_{N\to\infty}\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right)\frac{\delta_{H,i}}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_i N}} = 0;\end{aligned} \tag{37}$$

- $\delta_{H,i}=0$,

$$\begin{aligned}
&-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i=0,\\
&\Rightarrow \lim_{N\to\infty}\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right)=1<\infty,\\
&\Rightarrow \lim_{N\to\infty}\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right)\frac{\delta_{H,i}}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_i N}}=0.
\end{aligned}\tag{38}$$

So, we have

$$\lim_{N\to\infty}\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right)\frac{\delta_{H,i}}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_i N}}=0.\tag{39}$$

Then we have

$$\begin{aligned}
\lim_{N\to\infty}\frac{\partial F}{\partial N_b}=&\lim_{N\to\infty}\left\{\sum_{\text{feasible } a_i\neq\hat{a}_b}\left[\frac{\alpha_i^2}{\sigma_i^2}\frac{\sigma_b^2}{\alpha_b^2}\frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right)\frac{\delta_{H,i}}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_i N}}\right]\right.\\
&\left.-\frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{H,b}^2}{\sigma_{h,b}^2}\alpha_b N\right)\frac{\delta_{H,b}}{\sigma_{h,b}}\frac{1}{\sqrt{\alpha_b N}}+\left(1-\sum_{\text{feasible } a_i\neq\hat{a}_b}\frac{\alpha_i^2}{\sigma_i^2}\frac{\sigma_b^2}{\alpha_b^2}\right)\lambda\right\}\\
=&\left(1-\sum_{\text{feasible } a_i\neq\hat{a}_b}\frac{\alpha_i^2}{\sigma_i^2}\frac{\sigma_b^2}{\alpha_b^2}\right)\lambda\\
=&0.
\end{aligned}\tag{40}$$

So, we have

$$\begin{aligned}
&1-\sum_{\text{feasible } a_i\neq\hat{a}_b}\frac{\alpha_i^2}{\sigma_i^2}\frac{\sigma_b^2}{\alpha_b^2}=0,\\
&\Rightarrow \alpha_b=\sigma_b\sqrt{\sum_{\text{feasible } a_i\neq\hat{a}_b}\frac{\alpha_i^2}{\sigma_i^2}},\\
&\Rightarrow N_b=\sigma_b\sqrt{\sum_{\text{feasible } a_i\neq\hat{a}_b}\frac{N_i^2}{\sigma_i^2}}.
\end{aligned}\tag{41}$$

For action $i\neq b$, if i is infeasible, with (20), (34) and (30), we have

$$\begin{aligned}
\frac{\partial F}{\partial N_i}=&\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right)\frac{\delta_{H,i}}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_i N}}+\lambda\\
=&0.
\end{aligned}\tag{42}$$

If i is feasible, we assume N_b is much greater than N_i, i.e., $N_b\gg N_i$. Then we have

$$\sigma_{b,i}^2=\frac{\sigma_b}{N_b}+\frac{\sigma_i}{N_i}\approx\frac{\sigma_i}{N_i},\tag{43}$$

with (20), (34) and (28), we have

$$\begin{aligned}
\frac{\partial F}{\partial N_i}=&\frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{b,i}^2}{\sigma_i^2}\alpha_i N\right)\frac{\delta_{b,i}}{\sigma_i}\frac{1}{\sqrt{\alpha_i N}}-\frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right)\frac{\delta_{H,i}}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_i N}}+\lambda\\
=&0.
\end{aligned}\tag{44}$$

If action i is feasible, we have

$$\delta_{H,i} = \hat{Q}_H^{k+1} - H_0 < 0. \tag{45}$$

So, we have

$$\begin{aligned}\frac{\partial F}{\partial N_i} =& \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{\delta_{b,i}^2}{\sigma_i^2}\alpha_i N\right)\frac{\delta_{b,i}}{\sigma_i}\frac{1}{\sqrt{\alpha_i N}} + \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right)\frac{|\delta_{H,i}|}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_i N}} + \lambda \\ =& 0.\end{aligned} \tag{46}$$

For actions $i, j \neq b$.

- Action i, j are both infeasible.

With (42), we have

$$\begin{aligned}&\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right)\frac{\delta_{H,i}}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_i N}} = \exp\left(-\frac{1}{2}\frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j N\right)\frac{\delta_{H,j}}{\sigma_{h,j}}\frac{1}{\sqrt{\alpha_j N}}, \\ &\Rightarrow \exp\left[-\frac{1}{2}\left(\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i - \frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j\right)N\right] = \sqrt{\frac{\alpha_i}{\alpha_j}\cdot\frac{\delta_{H,j}^2/\sigma_{h,j}^2}{\delta_{H,i}^2/\sigma_{h,i}^2}}.\end{aligned} \tag{47}$$

We can find that when

$$\begin{aligned}&\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i - \frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j = 0 \Leftrightarrow \frac{\alpha_i}{\alpha_j}\cdot\frac{\delta_{H,j}^2/\sigma_{h,j}^2}{\delta_{H,i}^2/\sigma_{h,i}^2} = 1, \\ &\Rightarrow \exp\left[-\frac{1}{2}\left(\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i - \frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j\right)N\right] = 1 = \sqrt{\frac{\alpha_i}{\alpha_j}\cdot\frac{\delta_{H,j}^2/\sigma_{h,j}^2}{\delta_{H,i}^2/\sigma_{h,i}^2}}.\end{aligned} \tag{48}$$

Therefore, we have

$$\frac{N_i}{N_j} = \frac{\alpha_i}{\alpha_j} = \left(\frac{\delta_{H,i}/\sigma_{h,i}}{\delta_{H,j}/\sigma_{h,j}}\right)^2. \tag{49}$$

- Action i is feasible and j is infeasible.

With (46) and (42)

$$\begin{aligned}&\exp\left(-\frac{1}{2}\frac{\delta_{b,i}^2}{\sigma_i^2}\alpha_i N\right)\frac{\delta_{b,i}}{\sigma_i}\frac{1}{\sqrt{\alpha_i N}} + \exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i N\right)\frac{|\delta_{H,i}|}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_i N}} = \exp\left(-\frac{1}{2}\frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j N\right)\frac{\delta_{H,j}}{\sigma_{h,j}}\frac{1}{\sqrt{\alpha_j N}}, \\ \Rightarrow&\sqrt{\frac{\alpha_i}{\alpha_j}} = \exp\left[-\frac{1}{2}\left(\frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\alpha_i - \frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j\right)N\right]\frac{\delta_{b,i}/\sigma_i}{\delta_{H,j}/\sigma_{h,j}} + \exp\left[-\frac{1}{2}\left(\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i - \frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j\right)N\right]\frac{|\delta_{H,i}|/\sigma_{h,i}}{\delta_{H,j}/\sigma_{h,j}}.\end{aligned} \tag{50}$$

If $\delta_{b,i}/\sigma_i > |\delta_{H,i}|/\sigma_{H,i}$, let

$$\begin{aligned}&\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i = \frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j, \\ \Rightarrow&\frac{N_i}{N_j} = \frac{\alpha_i}{\alpha_j} = \left(\frac{\sigma_{h,i}/\delta_{H,i}}{\sigma_{h,j}/\delta_{H,j}}\right)^2,\end{aligned} \tag{51}$$

then, we have

$$\begin{aligned}&\frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\alpha_i - \frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j = \frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\alpha_i - \frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i > 0, \\ \Rightarrow&-\frac{1}{2}\left(\frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\alpha_i - \frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j\right) < 0, \\ \Rightarrow&\lim_{N\to\infty}\exp\left[-\frac{1}{2}\left(\frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\alpha_i - \frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j\right)N\right] = 0.\end{aligned} \tag{52}$$

So,

$$\begin{aligned}
&\lim_{N\to\infty}\sqrt{\frac{\alpha_i}{\alpha_j}}\\
&=\lim_{N\to\infty}\left\{\exp\left[-\frac{1}{2}\left(\frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\alpha_i-\frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j\right)N\right]\frac{\delta_{b,i}/\sigma_i}{\delta_{H,i}/\sigma_{h,i}}+\exp\left[-\frac{1}{2}\left(\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i-\frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j\right)N\right]\frac{|\delta_{H,i}|/\sigma_{h,i}}{\delta_{H,i}/\sigma_{h,i}}\right\}\\
&=\frac{|\delta_{H,i}|/\sigma_{h,i}}{\delta_{H,i}/\sigma_{h,i}},\\
\Rightarrow&\frac{\alpha_i}{\alpha_j}=\left(\frac{\sigma_{h,i}/\delta_{H,i}}{\sigma_{h,j}/\delta_{H,j}}\right)^2,
\end{aligned}\tag{53}$$

which is the same as (51). Thus, it is a solution to (50) under the condition $N\to\infty$. Furthermore, if $\delta_{b,i}/\sigma_i<|\delta_{H,i}|/\sigma_{H,i}$, we can have

$$\begin{aligned}
&\frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\alpha_i=\frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j,\\
\Rightarrow&\frac{N_i}{N_j}=\frac{\alpha_i}{\alpha_j}=\left(\frac{\sigma_i/\delta_{b,i}}{\sigma_{h,j}/\delta_{H,j}}\right)^2.
\end{aligned}\tag{54}$$

- Action i,j are both feasible.

With (46), we have

$$\begin{aligned}
&\exp\left(-\frac{1}{2}\frac{\delta_{b,i}^2}{\sigma_i^2}\alpha_iN\right)\frac{\delta_{b,i}}{\sigma_i}\frac{1}{\sqrt{\alpha_iN}}+\exp\left(-\frac{1}{2}\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_iN\right)\frac{|\delta_{H,i}|}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_iN}}\\
&=\exp\left(-\frac{1}{2}\frac{\delta_{b,j}^2}{\sigma_j^2}\alpha_jN\right)\frac{\delta_{b,j}}{\sigma_j}\frac{1}{\sqrt{\alpha_jN}}+\exp\left(-\frac{1}{2}\frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_jN\right)\frac{|\delta_{H,j}|}{\sigma_{h,j}}\frac{1}{\sqrt{\alpha_jN}}.
\end{aligned}\tag{55}$$

If $\delta_{b,i}/\sigma_i>\delta_{H,i}/\sigma_{h,i}$ and $\delta_{b,j}/\sigma_j>\delta_{H,j}/\sigma_{h,j}$, we can have

$$\begin{aligned}
&\lim_{N\to\infty}\exp\left[-\frac{1}{2}\left(\frac{\delta_{b,i}^2}{\sigma_{b,i}^2}\alpha_i-\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i\right)N\right]=0,\\
&\lim_{N\to\infty}\exp\left[-\frac{1}{2}\left(\frac{\delta_{b,j}^2}{\sigma_{b,j}^2}\alpha_j-\frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j\right)N\right]=0.
\end{aligned}\tag{56}$$

Let

$$\begin{aligned}
&\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\alpha_i=\frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_j,\\
&\Rightarrow\frac{N_i}{N_j}=\frac{\alpha_i}{\alpha_j}=\left(\frac{\sigma_{h,i}/\delta_{H,i}}{\sigma_{h,j}/\delta_{H,j}}\right)^2.
\end{aligned}\tag{57}$$

Let both sides of (55) divided by

$$\exp\left(-\frac{1}{2}\frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\alpha_jN\right)\frac{1}{\sqrt{N}},\tag{58}$$

we can get

$$\begin{aligned}
&\exp\left[-\frac{1}{2}\left(\frac{\delta_{b,i}^2}{\sigma_i^2}-\frac{\delta_{H,i}^2}{\sigma_{h,i}^2}\right)\alpha_iN\right]\frac{\delta_{b,i}}{\sigma_i}\frac{1}{\sqrt{\alpha_i}}+\frac{|\delta_{H,i}|}{\sigma_{h,i}}\frac{1}{\sqrt{\alpha_i}}\\
&=\exp\left[-\frac{1}{2}\left(\frac{\delta_{b,j}^2}{\sigma_j^2}-\frac{\delta_{H,j}^2}{\sigma_{h,j}^2}\right)\alpha_jN\right]\frac{\delta_{b,j}}{\sigma_j}\frac{1}{\sqrt{\alpha_j}}+\frac{|\delta_{H,j}|}{\sigma_{h,j}}\frac{1}{\sqrt{\alpha_j}}.
\end{aligned}\tag{59}$$

Under condition $N \to \infty$,

$$\begin{aligned}
&\lim_{N\to\infty} (\text{left hand side}) = \frac{|\delta_{H,i}|}{\sigma_{h,i}} \frac{1}{\sqrt{\alpha_i}},\\
&\lim_{N\to\infty} (\text{right hand side}) = \frac{|\delta_{H,j}|}{\sigma_{h,j}} \frac{1}{\sqrt{\alpha_j}},\\
\Rightarrow &\lim_{N\to\infty} (\text{left hand side}) = \lim_{N\to\infty} (\text{right hand side}),\\
\Leftrightarrow &\frac{|\delta_{H,i}|}{\sigma_{h,i}} \frac{1}{\sqrt{\alpha_i}} = \frac{|\delta_{H,j}|}{\sigma_{h,j}} \frac{1}{\sqrt{\alpha_j}},\\
\Leftrightarrow &\frac{N_i}{N_j} = \frac{\alpha_i}{\alpha_j} = \left(\frac{\sigma_{h,i}/\delta_{H,i}}{\sigma_{h,j}/\delta_{H,j}}\right)^2,
\end{aligned} \tag{60}$$

which is the same as (57). Thus, it is a solution to (55) under the condition $N \to \infty$.

It is similar when $\delta_{b,i}/\sigma_i \neq \delta_{H,i}/\sigma_{h,i}$ and $\delta_{b,j}/\sigma_j \neq \delta_{H,j}/\sigma_{h,j}$.

In summary,

$$\frac{N_i}{N_j} = \frac{\left[\min\left(\delta_{b,i}/\sigma_i, |\delta_{h,i}|/\sigma_{h,i}\right)\right]^{-2}}{\left[\min\left(\delta_{b,j}/\sigma_j, |\delta_{h,j}|/\sigma_{h,j}\right)\right]^{-2}} = \frac{\max\left[\left(\sigma_i/\delta_{b,i}\right)^2, \left(\sigma_{h,i}/\delta_{H,i}\right)^2\right]}{\max\left[\left(\sigma_j/\delta_{b,j}\right)^2, \left(\sigma_{h,j}/\delta_{H,j}\right)^2\right]}. \tag{61}$$

In conclusion, when $\delta_{b,i}/\sigma_i \neq \delta_{H,i}/\sigma_{h,i}$, we can have

$$\beta_i = \begin{cases} \max\left[\left(\dfrac{\sigma_i}{\delta_{b,i}}\right)^2, \left(\dfrac{\sigma_{h,i}}{\delta_{H,i}}\right)^2\right], & \text{feasible } a_i \neq a_b,\\ \left(\dfrac{\sigma_{h,i}}{\delta_{H,i}}\right)^2, & \text{infeasible } a_i. \end{cases} \tag{62}$$

- For $i, j \neq b$,

$$\frac{N_i}{N_j} = \frac{\beta_i}{\beta_j}. \tag{63}$$

- For N_b, we have

$$N_b = \sigma_b \sqrt{\sum_{\text{feasible } a_i \neq \hat{a}_b} \frac{N_i^2}{\sigma_i^2}}. \tag{64}$$

This completes the proof.

The convergence property is similar to that in [46].

Additionally, Theorem 1 can only be used when there is at least one known feasible action for each state. We notice a situation where the optimization starts with no feasible policies. In this situation, we want to first find one feasible policy, so we have a definition similar to Theorem 1 as follows:

$$\begin{aligned}
&\hat{a}_b := \arg\min_{a\in\mathcal{A}} \left\{\hat{Q}_H^{k+1}(s,a)\right\},\\
&a_b := \arg\min_{a\in\mathcal{A}} \left\{Q_H^{k+1}(s,a)\right\},\\
&\forall a \in \mathcal{A},\ \hat{Q}_H^{k+1}(s,a) > \eta,\ Q_H^{k+1}(s,a) > \eta.
\end{aligned} \tag{65}$$

Then, we can formulate PCS and APCS problems, too.

$$\begin{aligned}
\text{PCS} =& \Pr\{a_b = \hat{a}_b\} = \Pr\left\{\bigcap_{\text{feasible } a_i \neq a_b} Q_H^{k+1}(s,a_b) < Q_H^{k+1}(s,a_i)\right\},\\
\text{APCS} =& 1 - \sum_{\text{feasible } a_i \neq a_b} \left[1 - \Pr\left(Q_H^{k+1}(s,a_b) < Q_H^{k+1}(s,a_i)\right)\right].
\end{aligned} \tag{66}$$

Denote

$$\begin{aligned}\delta_{b,i} &= \hat{Q}_H^{k+1}(s,a_i) - \hat{Q}_H^{k+1}(s,a_b),\\ \sigma_{b,i}^2 &= \frac{\sigma_{h,i}^2}{N_i} + \frac{\sigma_{h,b}^2}{N_b},\end{aligned} \tag{67}$$

where $\sigma_{h,i}^2$ is the variance of a single observation of $Q_H^{k+1}(s,a_i)$. Using similar analysis to that in the OCBA, we can obtain the following Theorem 2.

Theorem 2. The following computing budget allocation rule asymptotically maximizes APCS when $N \to \infty$,

$$\beta_i = \left(\frac{\sigma_{b,i}}{\delta_{b,i}}\right)^2. \tag{68}$$

- For $i,j \neq b$,

$$\frac{N_i}{N_j} = \frac{\beta_i}{\beta_j}. \tag{69}$$

- For N_b, we have

$$N_b = \sigma_b \sqrt{\sum_{\text{feasible } a_i \neq \hat{a}_b} \frac{N_i^2}{\sigma_i^2}}. \tag{70}$$

The proof is similar to [45], so it will not be elaborated here.

We now have the following Algorithm 1. Theorems 1 and 2 guarantee that Algorithm 1 asymptotically maximizes the probability of correctly selecting the best feasible action at each state.

Algorithm 1 Sampling efficient constrained Q-learning algorithm.

```
Given each state sample number T, total sample budget 𝒯;
Initialize Q(s,a) and Q_H(s,a), turn counter k = 0;
while k < 𝒯 do
    observe the initial state s = s_0;
    while not done do
        if ∃ a ∈ 𝒜 s.t. Q_H(s,a) < η then
            Sample results {s', r, h} for state s_0 T times follow Theorem 1;
        else
            Sample results {s', r, h} for state s_0 T times follow Theorem 2;
        end if
        k ← k + T;
        Update Q(s,a) and Q_H(s,a) with (12);
        Get the best feasible action a_b = arg max_{a∈𝒜}{Q(s,a)|Q_H(s,a) < η} or best infeasible action a_b = arg min_{a∈𝒜}{Q_H(s,a)};
        Execute a_b and observe s = s' with done signal;
    end while
end while
```

3.2 Sampling efficient constrained DQN

Algorithm 1 gives an example application in reinforcement learning. However, in most cyber-physical systems, it is always very hard to describe value functions in a tabular way for the whole state space and action space. Moreover, sampling before executing actions is almost impossible, as the simulators are always physical object platforms or high-fidelity simulation platforms. Thus, deep reinforcement learning is introduced to solve problems like these.

There are various algorithms in reinforcement learning. They can generally be divided into three categories depending on whether the state space and the action space are continuous or discrete. We show typical methods in Table 1 [47–57]. Among those, methods with at least one continuous space are usually deep reinforcement learning methods.

For the purpose of more straightforward calculation and demonstrating the situations of different actions, we choose the DQN [49] method as the example in our algorithm, whose state space is continuous while the action space is discrete.

Table 1 Categorization of representative RL algorithms.

		$\mathcal{S}$	
		Discrete	Continuous
$\mathcal{A}$	Discrete	Q-learning [47], Sarsa [48]	DQN [49], Double DQN [50], Dueling DQN [51]
	Continuous	A3C [52], TRPO [53], PPO [54], DDPG [55], TD3 [56], SAC [57]	

Moreover, Algorithm 1 is not suitable for large-scale problems, especially ones with continuous state space. Here we introduce an ensemble-based predictive-uncertainty estimation method, which can be used in function approximations, such as networks in deep reinforcement learning. We use multiple networks $Q_m(s,a), Q_{H,m}(s,a), m = 1,2,\ldots,N$ to estimate the distribution of value functions in (14)

$$\begin{aligned}
&\hat{Q}^{k+1}(s,a) = \mathbb{E}_m[Q_m(s,a)],\\
&\hat{Q}_H^{k+1}(s,a) = \mathbb{E}_m[Q_{H,m}(s,a)],\\
&Q^{k+1}(s,a) \sim \mathcal{N}\left(\hat{Q}^{k+1}(s,a), \sigma^2(Q_m(s,a))\right),\\
&Q_H^{k+1}(s,a) \sim \mathcal{N}\left(\hat{Q}_H^{k+1}(s,a), \sigma^2(Q_{H,m}(s,a))\right).
\end{aligned} \tag{71}$$

Thus, we have one way to obtain predictive-uncertainty without many samples at once. At each time, we can obtain the best action according to Algorithm 1

$$a_b = \begin{cases} \arg\min\limits_{a\in\mathcal{A}} \left\{\mathbb{E}_m[Q_{H,m}(s,a)]\right\}, & \text{if } \forall a \in \mathcal{A}\ \mathbb{E}_m[Q_{H,m}(s,a)] > \eta,\\ \arg\max\limits_{a\in\mathcal{A}} \{\mathbb{E}_m[Q_m(s,a)] | a : \mathbb{E}_m[Q_{H,m}(s,a)] \leqslant \eta\}, & \text{o.w.} \end{cases} \tag{72}$$

For the training process, the sampling probability $p(a|s)$ can be used in every step of the sampling. With this probability, when we sample sufficiently many times, the simulation replications of different actions for one state will approximate those of Theorem 1, which is guaranteed by the law of large numbers.

First, we denote

$$\begin{aligned}
\delta_b(s,a_i) &= \mathbb{E}_m\left[Q_m(s,a_b)\right] - \mathbb{E}_m\left[Q_m(s,a_i)\right],\\
\delta_H(s,a_i) &= \mathbb{E}_m\left[Q_{H,m}(s,a_i)\right] - \eta,\\
\delta_{H,b}(s,a_i) &= \mathbb{E}_m\left[Q_{H,m}(s,a_i)\right] - \mathbb{E}_m\left[Q_{H,m}(s,a_b)\right].
\end{aligned} \tag{73}$$

Then, we can denote $\beta(s,a)$ referring to (21) and (23)

$$\begin{aligned}
\beta(s,a_i) &= \begin{cases} \max\left[\dfrac{\sigma^2(Q_m(s,a_i))}{\delta_b^2(s,a_i)}, \dfrac{\sigma^2(Q_{H,m}(s,a_i))}{\delta_H^2(s,a_i)}\right], & \text{feasible } a_i \neq a_b,\\ \dfrac{\sigma^2(Q_{H,m}(s,a_i))}{\delta_H^2(s,a_i)}, & \text{infeasible } a_i, \end{cases}\\
\beta(s,a_b) &= \sigma^2(Q_m(s,a_b))^{\frac{1}{2}} \cdot \left(\sum_{\text{feasible } a_j\neq a_b} \frac{\beta^2(s,a_j)}{\sigma^2(Q_m(s,a_j))}\right)^{\frac{1}{2}}.
\end{aligned} \tag{74}$$

Furthermore, for the situation where there is no feasible action, we can denote $\beta(s,a)$ referring to (68) and (70)

$$\begin{aligned}
\beta(s,a_i) &= \frac{\sigma^2(Q_{H,m}(s,a_i))}{\delta_{H,b}^2(s,a_i)}, \quad \text{for } i \neq b,\\
\beta(s,a_b) &= \sigma^2(Q_{H,m}(s,a_b))^{\frac{1}{2}} \cdot \left(\sum_{\text{feasible } a_j\neq a_b} \frac{\beta^2(s,a_j)}{\sigma^2(Q_{H,m}(s,a_j))}\right)^{\frac{1}{2}}.
\end{aligned} \tag{75}$$

Finally, we have the sampling probability as follows:

$$\begin{aligned}
z(s) &= \epsilon_p + \sum_{a\in\mathcal{A}} \beta(s,a),\\
p(a|s) &= \frac{\beta(s,a)}{z(s)} + \frac{\epsilon_p}{|\mathcal{A}|},
\end{aligned} \tag{76}$$

where ϵ_p is a tiny positive number for ϵ-greedy to alleviate the model error.

In DQN, the network parameters are updated with the temporal difference (TD) target y_t, and a fixed Q target network is also introduced into the training process. Here we use $Q_m(s,a|\theta_m^-)$ to represent the fixed Q target network of $Q_m(s,a|\theta_m)$ and $y_{t,m}$ to represent the TD target of the m-th value network in our framework. The multi-networks are regarded as a whole. Thus, the future value is sampled from the distribution formed by the multi-networks. For one sample $\langle s_t, a_t, s_{t+1}, r_t, h_t\rangle$, we obtain a_b in (72) with fixed Q target networks first. Then, we calculate the TD-target as follows:

$$\begin{aligned}
\overline{Q}(s_{t+1},a_b) &\sim \mathcal{N}\Big(\mathbb{E}_m[Q_m(s_{t+1},a_b|\theta_m^-)], \epsilon_\sigma\cdot\sigma^2[Q_m(s_{t+1},a_b|\theta_m^-)]\Big),\\
\overline{Q}_H(s_{t+1},a_b) &\sim \mathcal{N}\Big(\mathbb{E}_m[Q_{H,m}(s_{t+1},a_b|\theta_{H,m}^-)], \epsilon_\sigma\cdot\sigma^2[Q_{H,m}(s_{t+1},a_b|\theta_{H,m}^-)]\Big),\\
y_{t,m}(s_t,a_t) &= r_t+\gamma\overline{Q}(s_{t+1},a_b),\\
y_{t,H,m}(s_t,a_t) &= h_t+\overline{Q}_H(s_{t+1},a_b),
\end{aligned} \tag{77}$$

where $\epsilon_\sigma\in[0,1)$ is a factor used to guarantee convergence. Finally, we can realize value iteration by minimizing the residual error with one sample trajectory

$$\begin{aligned}
L(\theta_m) &= \mathbb{E}_t\left[\left(y_{t,m}(s_t,a_t)-Q_m(s_t,a_t|\theta_m)\right)^2\right],\\
L(\theta_{H,m}) &= \mathbb{E}_t\left[\left(y_{t,H,m}(s_t,a_t)-Q_{H,m}(s_t,a_t|\theta_{H,m})\right)^2\right].
\end{aligned} \tag{78}$$

Now, we have Algorithm 2, which combines our framework into the DQN method.

Algorithm 2 Sampling efficient constrained DQN.

Randomly initialize $\{Q^{\theta_m}\}_{m=1}^n$, $\{Q_H^{\theta_{H,m}}\}_{m=1}^n$, and set the target parameters $\theta_m^-\leftarrow\theta_m$, $\theta_{H,m}^-\leftarrow\theta_{H,m}$; set the step size for target networks κ, replay buffer $D=\emptyset$, turn counter $k=0$, and total sample budget $\mathcal{T}$;
while $k<\mathcal{T}$ **do**
  observe the initial state s_0;
  while not done **do**
    Obtain feasible and infeasible sets with $\{Q_H^{\theta_{H,m}}\}_{m=1}^n$;
    Sample action with the raised sampling strategy $a_t\sim p(\cdot|s_t)$ in (74)–(76);
    Execute a_t and observe s_{t+1}, r_t, h_t with done signal;
    Store $\langle s_t,a_t,r_t,h_t,s_{t+1}\rangle$ in D;
    Update the parameters of networks by performing gradient descent on $L(\theta_m)$ and $L(\theta_{H,m})$ with (77) and (78);
    Update the target networks with $\theta_m^-\leftarrow(1-\kappa)\theta_m^-+\kappa\theta_m$ and $\theta_{H,m}^-\leftarrow(1-\kappa)\theta_{H,m}^-+\kappa\theta_{H,m}$;
  end while
end while
Set the iteration counter $k\leftarrow k+1$.

Theorems 1 and 2 can also guarantee that Algorithm 2 asymptotically maximizes the probability of correctly selecting the best action in each state. The schematic diagram of the algorithm we proposed is shown in Figure 1.

3.3 Convergence property analysis

In this part, we will talk about the convergence property of the proposed algorithm. Here, we present some basic definitions in reinforcement learning and our update equations in a simple way, referring to [46, 58]. It should be noted that the convergence situations of the action value function and the constraint value function are consistent. Here, only the part of the value function is shown. The subsequent content aims to demonstrate that for the Q-functions of the algorithm proposed in this paper, their means can converge to the optimal Q-functions and their variances can converge to 0. Therefore, the Q-functions can converge to the optimal Q-functions.

The optimal action-value function is defined as (79) following the Bellman equation:

$$\begin{aligned}
Q^*(s,a) &= \max_\pi Q^\pi(s,a)\\
&= \mathbb{E}_{s'\sim P(\cdot|s,a)}[r(s,a)+\gamma\max_{a'}Q^*(s',a')].
\end{aligned} \tag{79}$$

The main theorem is developed on the following lemmas.

Lemma 1 ([58]). The random process Δ_t taking values in $\mathbb{R}^n$ is defined as

$$\Delta_{t+1}(x) = (1-\alpha_t(x))\Delta_t(x)+\alpha_t(x)F_t(x), \tag{80}$$

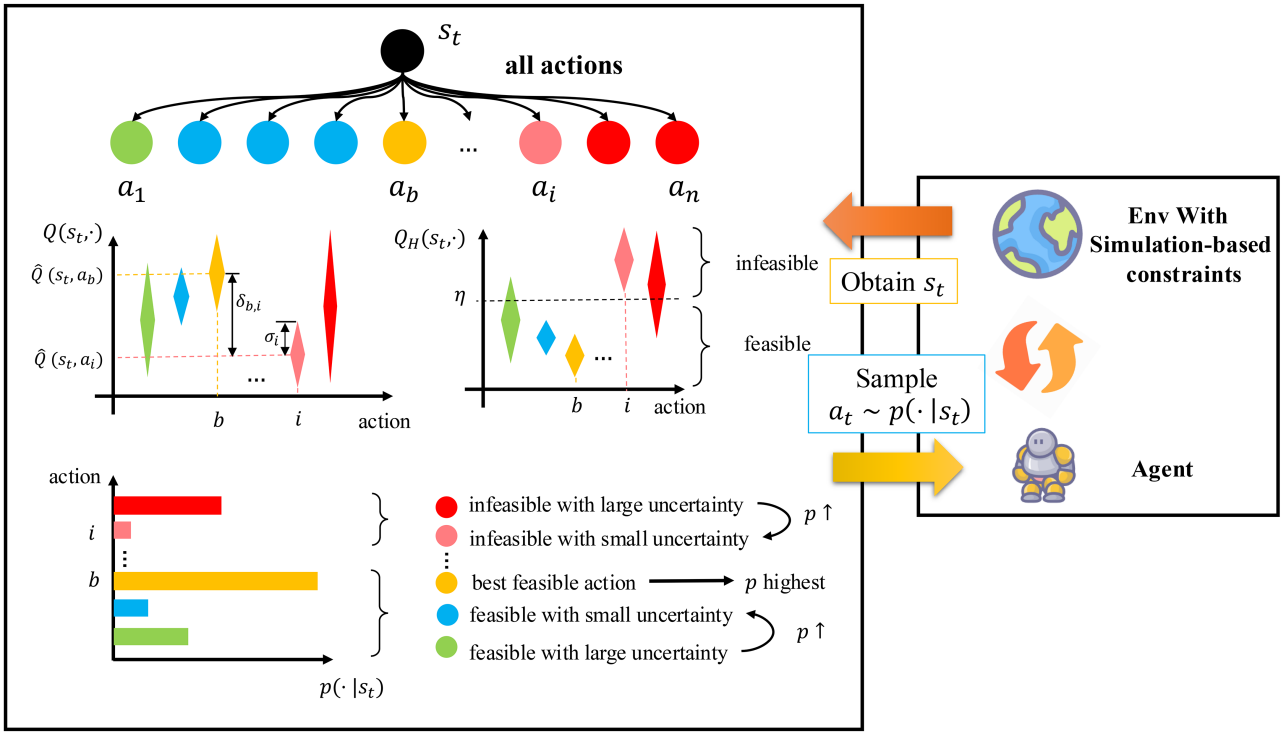

Figure 1 (Color online) Sample efficiency constrained deep reinforcement learning: we illustrate the main ideas of the proposed algorithm. For s_t obtained in each step t, we first calculate the $Q(s_t,\cdot), Q_H(s_t,\cdot)$ (including mean and variance) with ensemble networks. According to the results, we will divide all actions into feasible and infeasible sets. Different sets will use different probability calculation methods (as shown in (74)). Finally, we sample actions with the probability calculated by (76). After executing the action, we update the networks with the results. Note that uncertainty in the figure means $\beta(s_t,\cdot)$ in (74). Additionally, this is the sampling strategy for the training process. When we use the test mode, we can directly use the best action in (72) without sampling.

where $\alpha_t(x)$ is the step size for x at time step t, and F_t is a random process. Let $F_t = \{F_i|\forall i < t\}$, then Δ_t converges to 0 w.p.1 if the following conditions are satisfied for all x.

(1) $0 \leqslant \alpha_t(x) \leqslant 1, \quad \sum_{t=0}^{\infty} \alpha_t(x) = \infty, \quad \sum_{t=0}^{\infty} \alpha_t^2(x) < \infty.$

(2) $\exists\, \vartheta < 1,$ so that $|\mathbb{E}[F_t(x)|\mathcal{F}_t]| \leqslant \vartheta \|\Delta_t\|_\infty.$

(3) $\exists\, \mathcal{C} > 0,$ so that $\mathrm{var}[F_t(x)|\mathcal{F}_t] \leqslant \mathcal{C}(1 + \|\Delta_t\|_\infty^2).$

Lemma 2 ([46]). Given a finite MDP, $\langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$, the Q-learning algorithm given by the update rule

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(s_t, a_t)[y_t - Q_t(s_t, a_t)], \tag{81}$$

where $y_t = r_t + \gamma \max_{a'} Q_t(s_{t+1}, a')$, converges w.p.1 to the optimal action-value function as long as each state-action pair is performed infinitely often, and

$$\begin{aligned} &\sum_t \alpha_t(s,a) = \infty, \\ &\sum_t \alpha_t^2(s,a) < \infty, \end{aligned} \tag{82}$$

holds for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.

Lemma 1 gives a general criterion for the convergence of random processes, and Lemma 2 further proves the convergence property of Q-learning.

Theorem 3. Given a finite constrained MDP $\mathcal{S}, \mathcal{A}, P, r, h, \gamma$. The sample efficiency Q-learning algorithm with a Gaussian distribution and bounded variance

$$Q_t(s,a) \sim \mathcal{N}\left(\mu(Q_t(s,a), \sigma^2(Q_t(s,a)))\right), \tag{83}$$

and select actions following the policy of (76). The estimations are updated with

$$\begin{aligned} \mu(Q_{t+1}(s,a)) =& (1 - \alpha(s_t,a_t))\, \mu(Q_t(s,a)) + \alpha_t(s_t,a_t) \left(r_t + \gamma \max_a \mu(Q_t(s_{t+1},a))\right), \\ \sigma^2\left(Q_{t+1}(s,a)\right) =& \epsilon_\sigma \left[(1 - \alpha(s_t,a_t))^2\, \sigma^2(Q_t(s,a))\; \alpha_t^2(s_t,a_t)\; (\sigma^2(r_t) + \gamma^2 \max_a \sigma^2(Q_t(s_{t+1},a)))\right], \end{aligned} \tag{84}$$

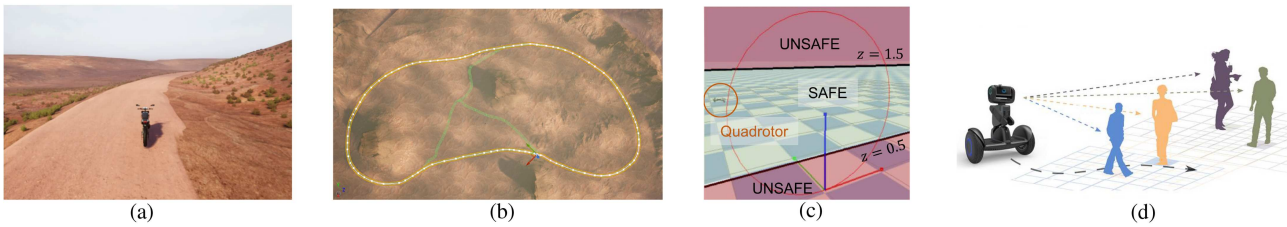

Figure 2 (Color online) Schematic diagrams of three tasks. (a) CarControl task; (b) CarControl route; (c) quadrotor task; (d) CrowdNav task.

where $\epsilon_\sigma \in (0,1)$ and $\alpha_t(s,a) \in [0,1]$ take a nonzero value only on $(s,a) = (s_t,a_t)$, and $\sum_t \alpha_t(s,a) = \infty$ and $\sum_t \alpha_t^2(s,a) < \infty$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. Then the estimations converge w.p.1 to the optimal action-value functions

$$\lim_{t\to\infty} Q_t(s,a) = Q^*(s,a), \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}. \tag{85}$$

Proof. First, we prove that $\mu(Q_t(s,a))$ converges to the optimal action-value functions Q^*. From (76), we know that $p_t(a|s) \geqslant \frac{\epsilon_p}{|\mathcal{A}|} > 0$ holds for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. Thus, each reachable state-action pair will be performed infinitely often. From Lemma 2 where we replace Q_t with $\mu(Q_t)$, we obtain from (84)

$$\lim_{t\to\infty} \mu(Q_t(s,a)) = Q^*(s,a), \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}. \tag{86}$$

Then, we prove that the variance of the action value functions will converge to 0,

$$\begin{aligned} \sigma^2(Q_{t+1}(s,a)) =& \epsilon_\sigma \left[(1-\alpha(s_t,a_t))^2 \sigma^2(Q_t(s,a)) \, \alpha_t^2(s_t,a_t) \, (\sigma^2(r_t) + \gamma^2 \max_a \sigma^2(Q_t(s_{t+1},a))) \right] \\ \leqslant& \epsilon_\sigma \max_{(s,a)} \sigma^2(Q_t), \\ \Rightarrow \max_{(s,a)} \sigma^2(Q_{t+1}) \leqslant& \epsilon_\sigma \max_{(s,a)} \sigma^2(Q_t), \\ \Rightarrow \max_{(s,a)} \sigma^2(Q_{t+1}) \leqslant& \epsilon_\sigma^{t+1} \max_{(s,a)} \sigma^2(Q_0). \end{aligned} \tag{87}$$

Since $\epsilon_\sigma \in (0,1)$ and $\sigma^2(Q_0)$ is bounded, we can have

$$\begin{aligned} &\lim_{t\to\infty} \max_{(s,a)} \sigma^2(Q_t) \to 0, \\ \Rightarrow &\lim_{t\to\infty} \sigma^2(Q_t(s,a)) \to 0, \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}. \end{aligned} \tag{88}$$

Based on (86) and (88), we can prove that all of the action-value estimations converge to Q^*.

Finally, the constraint value function will also converge to Q_H^* with similar update methods, so our proposed algorithm will converge to the best policy.

4 Experimental results

We evaluated our proposed Algorithm 2 in classical cyber-physical systems, including autonomous vehicles and quadrotors. Our goal is to validate: (1) whether our algorithm is able to learn a policy that satisfies the given simulation-based constraints; (2) whether the proposed algorithm is able to find an equally good or even better policy with fewer samples.

4.1 Tasks

We tested our algorithm on three tasks compared with two baselines mentioned later. Note that each episode will be terminated if the constraints are violated during training. Additionally, for convenience in training, the maximum sequence length has also been set.

CarControl is a task of controlling an autonomous vehicle to drive along a given route in a wide scene for a period of time with a given velocity of 10 m/s (the schematic diagram of the actual physical meaning is shown in Figure 2(a) [43]). The vehicle dynamics are given according to [59], which is a complex dynamic model considering angular

momentum, etc. The goal is to let the vehicle track and drive along the specified road (shown in Figure 2(b)) for a period of time. In this task, the state space includes five necessary pieces of information: the distance offset from the center of the road, the angle between the vehicle's orientation and the tangential direction of the road, the curvature of the road, and the actual position of the vehicle. Additionally, we give the vehicle twenty-one actions representing the vehicle's steering commands with equal intervals ranging from 1.0 to -1.0, which correspond to the maximum and minimum turning angles, respectively. The reward consists of two parts: the reward for the vehicle being able to move forward continuously while satisfying the constraints, and the linear reward related to the distance offset from the center line of the road, where the closer, the better. Here, the task constraint is that the offset distance cannot exceed the road width of 5 m. If the vehicle is out of the road, we will add another cost to this bad decision.

Quadrotor is a 2D quadrotor tracking task in safe-control-gym [60]. In this task, the policy needs to control a quadrotor to track the circle trajectory (red line in Figure 2(c)) as accurately as possible. We will give the necessary flying attitude to the policy, and the actions are divided into 9 parts, respectively, in two dimensions, representing normalized torque. The task constraints in this task are two limits in the zone (black straight line in Figure 2(c)). Note that the constraint contradicts the reward. This is a more challenging task than CarControl to further test the algorithm's ability to handle the contradiction between constraints and rewards.

CrowdNav is a task to control a robot to navigate in a crowded environment [61] (the schematic diagram of the actual physical meaning is in Figure 2(d)). The robot needs to reach the target position while avoiding collisions with pedestrians. The state space includes the robot's position, the target position, and the pedestrian positions. The action space includes the robot's velocity and rotation. In our experiment, the velocity is divided into five discrete parts in exponential form, while the rotation is divided into 24 discrete parts in linear form. The reward is given based on the distance to the target. The simulated pedestrians are controlled by ORCA (reciprocal n-body collision avoidance) [62]. The parameters of ORCA are sampled from a Gaussian distribution to introduce behavioral diversity. In our experiment, we randomly generate three pedestrians on the circumference of a circle with a radius of 4 m and let the robot and goal position be the two endpoints of the diameter, respectively. The task constraint is that the robot must not collide with any pedestrian. This task is more complex than the previous two tasks because the robot needs to consider the dynamic environment and the behavior of pedestrians.

We must note that, in the tasks above, the simulation-based constraint is to ensure that the probability that the policy does not satisfy the task constraints is less than 10%. This design is based on the characteristic that control strategies in these application scenarios are usually evaluated by metrics such as "pass rate" and "completion rate".

4.2 Baselines

The baseline for comparison includes two types of algorithms. First, we want to verify that the method we proposed has an advantage in the ability to satisfy constraints compared to the mainstream constraint-oriented methods. The most representative method is the Lagrange multiplier method. We used DQN with the Lagrange multiplier (DQN-lambda) as the first baseline. Second, verifying that the idea from OCBA can effectively improve the sample utilization efficiency is another objective. Therefore, it is necessary to use methods that also have a constraint evaluation network as a baseline. Here, we use the constrained-DQN method, which adds another neural network to learn whether the constraints are satisfied (baseline CRL). Constrained-DQN only uses one neural network for learning the action-value function and constraint function.

For the simulation-based constraint, different methods use different learning strategies. The DQN-λ method cannot handle this type of constraint, so it will use task constraints and cost directly instead. In contrast, our proposed algorithm and Constrained-DQN indeed consider simulation-based constraints with constraint networks.

4.3 Results

We evaluated all algorithms with three metrics: (1) the episode return (ER), which indicates the performance of each algorithm; (2) the percentage of episodes that violate the constraints (PVC), which indicates the ability of each algorithm to satisfy the constraints; (3) the length of the episode (LE), which indicates the ability to keep the constraints satisfied. Each metric is averaged over 100 different random runs, and we record the average and standard deviation of each metric.

The actual experimental parameters are shown in Table 2. In our experiments, the ensemble size was set to 5. A detailed analysis of its precise meaning and the rationale behind this choice is provided in Appendix A. The results are shown in Figure 3. As expected, our proposed algorithm has the best performance in terms of ER and PVC. The ER of our algorithm is higher than that of the other two baselines, which means that our algorithm can learn a better policy. The PVC of our algorithm is lower than the other two baselines, which means that our algorithm

Table 2 Experimental parameter settings for three tasks. 256×4 means that it corresponds to four sequential linear layers, each consisting of 256 neurons.

	CarControl	Quadrotor	CrowdNav
Lambda for cost function	1.0	1.0	1.0
Hidden size of value networks and constraint networks	256×4	256×2	128×3
Learning rate of value networks	1e−4	1e−4	3e−4
Learning rate of constraint networks	5e−2	1	1e−2
Batch size	128	128	256

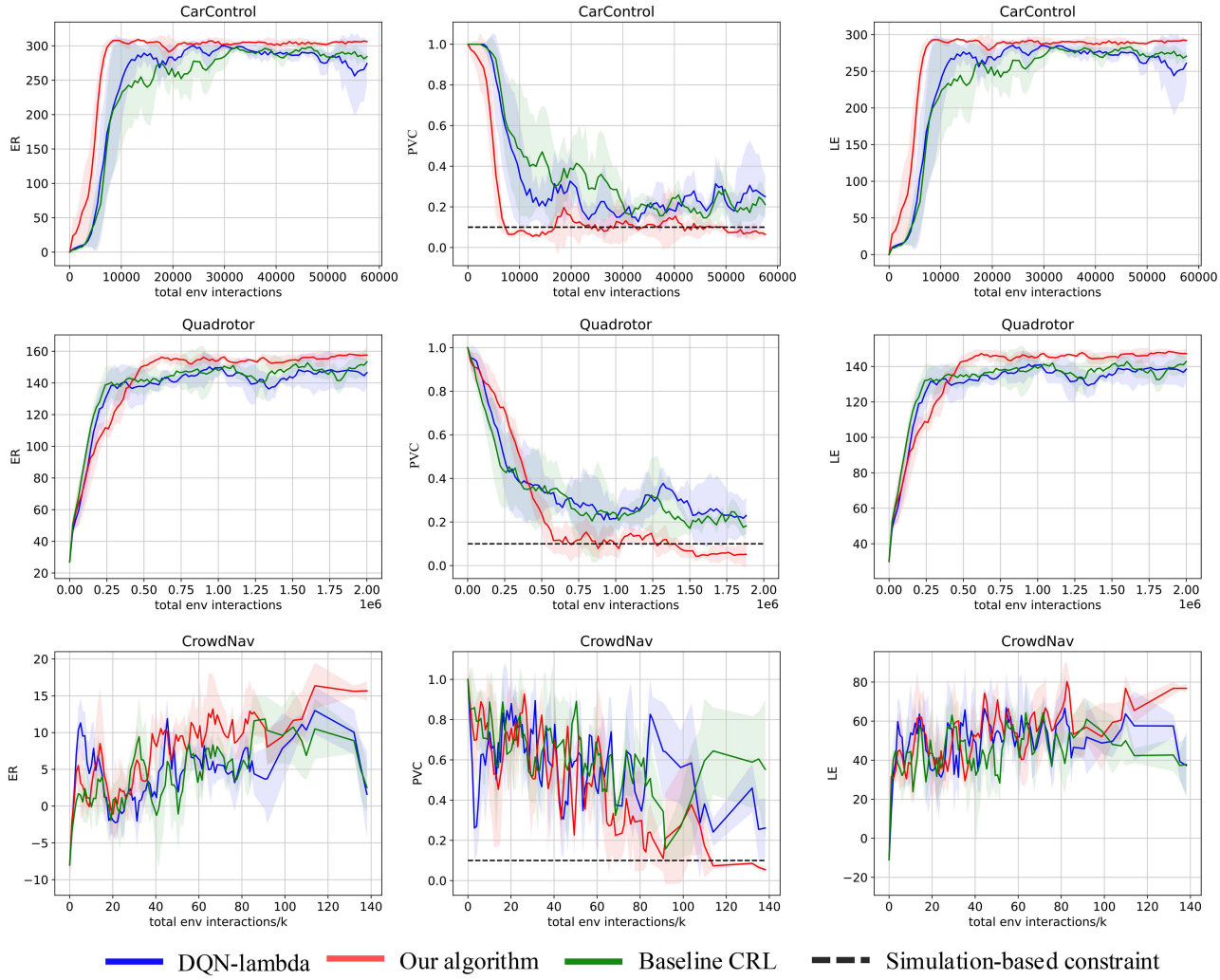

Figure 3 (Color online) Training results on the three tasks. The solid lines represent the mean performance and the shaded regions represent the standard deviation. Each row corresponds to three metrics of one task. Each column corresponds to one metric, from left to right: ER, PVC, and LE, respectively.

can better satisfy the constraints. The LE of our algorithm is also higher than the other two baselines, which means that our algorithm can keep satisfying the constraints better. Additionally, our proposed algorithm can achieve higher ER and lower PVC in a faster way than the other two baselines. This means that our proposed algorithm can learn a better policy with fewer samples.

5 Conclusion

In this paper, we proposed a novel algorithm for constrained reinforcement learning in cyber-physical systems. We introduced a new framework that combines the OCBA method with deep reinforcement learning, which can improve the efficiency of sample utilization. We proved that our proposed algorithm can asymptotically maximize

the probability of correctly selecting the best action at each state. We also proved that our proposed algorithm can converge to the optimal action value and constraint functions. We evaluated our proposed algorithm on three classical cyber-physical system tasks and compared it with two baselines based on the DQN framework. The results show that our proposed algorithm can learn a better policy with fewer samples and satisfy the simulation-based constraints better. However, the current work is only confined to problems with discrete action spaces and lacks sufficient discussion on continuous action spaces. In the future, we will further explore the application of our proposed algorithm in more complex and real cyber-physical systems tasks and extend our method to continuous action spaces, for example, by using the line search method.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62125304, 62073182), National Key Research and Development Program of China (Grant No. 2022YFA1004600), and Beijing Natural Science Foundation (Grant No. L233005).

References

1 Altman E. Constrained Markov Decision Processes. New York: Routledge, 2021
2 Hafner D, Lillicrap T, Ba J, et al. Dream to control: learning behaviors by latent imagination. ArXiv:1912.01603
3 Liu Z, Liu Q, Tang L, et al. Visuomotor reinforcement learning for multirobot cooperative navigation. IEEE Trans Automat Sci Eng, 2021, 19: 3234–3245
4 Brittain M, Wei P. Scalable autonomous separation assurance with heterogeneous multi-agent reinforcement learning. IEEE Trans Automat Sci Eng, 2022, 19: 2837–2848
5 Yan Z, Kreidieh A R, Vinitsky E, et al. Unified automatic control of vehicular systems with reinforcement learning. IEEE Trans Automat Sci Eng, 2022, 20: 789–804
6 Jin X Z, Li K, Jia Q S. Constrained reinforcement learning with statewise projection: a control barrier function approach. Sci China Inf Sci, 2024, 67: 132202
7 Yang Y F, Huang T, Wang T Q, et al. Sampling-efficient path planning and improved actor-critic-based obstacle avoidance for autonomous robots. Sci China Inf Sci, 2024, 67: 152204
8 Zhu J, Wei Y T, Kang Y, et al. Adaptive deep reinforcement learning for non-stationary environments. Sci China Inf Sci, 2022, 65: 202204
9 Li J N, Nie H, Chai T Y, et al. Reinforcement learning for optimal tracking of large-scale systems with multitime scales. Sci China Inf Sci, 2023, 66: 170201
10 Xu Y, Wu Z G, Che W W, et al. Reinforcement learning-based unknown reference tracking control of HMASs with nonidentical communication delays. Sci China Inf Sci, 2023, 66: 170203
11 Jin J, Chang S, Duan J, et al. Groupwise label enhancement broad learning system for image classification. IEEE Trans Cybern, 2025, 55: 3431–3444
12 Jin J, Geng B, Li Y, et al. Flexible label-induced manifold broad learning system for multiclass recognition. IEEE Trans Neural Netw Learn Syst, 2024, 35: 16076–16090
13 Yu D, Xu H, Chen C L P, et al. Dynamic coverage control based on K-means. IEEE Trans Ind Electron, 2022, 69: 5333–5341
14 Jia D Y, Dai X F, Xing J L, et al. Asymmetric interaction preference induces cooperation in human-agent hybrid game. Sci China Inf Sci, 2025, 68: 212201
15 Wang Z, Jusup M, Wang R W, et al. Onymity promotes cooperation in social dilemma experiments. Sci Adv, 2017, 3: 1601444
16 Wang Z, Song P X, Peng D T, et al. Manipulate holistic cooperation of population game: optimal induction strategy for a few irrational agents in social networks. Sci China Inf Sci, 2025, 68: 142202
17 Bertsekas D. Lessons from AlphaZero for Optimal, Model Predictive, and Adaptive Control. Belmont: Athena Scientific, 2022
18 Cao X R. Stochastic learning and optimization—a sensitivity-based approach. IFAC Proc Volumes, 2008, 41: 3480–3492
19 Jia Q S. On sampling efficiency optimization in constrained reinforcement learning. In: Proceedings of the 2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2024. 966–971
20 Ames A D, Xu X, Grizzle J W, et al. Control barrier function based quadratic programs for safety critical systems. IEEE Trans Automat Contr, 2017, 62: 3861–3876
21 Ma S, Ma X, Xia L. A unified algorithm framework for mean-variance optimization in discounted Markov decision processes. Eur J Oper Res, 2023, 311: 1057–1067
22 Xia L. Mean-variance optimization of discrete time discounted Markov decision processes. Automatica, 2018, 88: 76–82
23 Xia L. Risk-sensitive Markov decision processes with combined metrics of mean and variance. POM, 2020, 29: 2808–2827
24 Baier C, Katoen J P. Principles of Model Checking. Cambridge: MIT Press, 2008
25 Gao Y, Abate A, Jiang F J, et al. Temporal logic trees for model checking and control synthesis of uncertain discrete-time systems. IEEE Trans Automat Contr, 2021, 67: 5071–5086
26 Lennartson B, Jia Q S. Reinforcement learning with temporal logic constraints. IFAC-PapersOnLine, 2020, 53: 485–492
27 Chua K, Calandra R, McAllister R, et al. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 4759–4770
28 Janner M, Fu J, Zhang M, et al. When to trust your model: model-based policy optimization. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 12498–12509
29 Thomas G, Luo Y, Ma T. Safe reinforcement learning by imagining the near future. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 13859–13869
30 Zanger M A, Daaboul K, Zöllner J M. Safe continuous control with constrained model-based policy optimization. In: Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021. 3512–3519
31 Yu D, Zou W, Yang Y, et al. Safe model-based reinforcement learning with an uncertainty-aware reachability certificate. IEEE Trans Automat Sci Eng, 2024, 21: 4129–4142
32 Guan Y, Ren Y, Sun Q, et al. Integrated decision and control: toward interpretable and computationally efficient driving intelligence. IEEE Trans Cybern, 2022, 53: 859–873
33 Duan J, Liu Z, Li S E, et al. Deep adaptive dynamic programming for nonaffine nonlinear optimal control problem with state constraints. ArXiv:1911.11397
34 Tessler C, Mankowitz D J, Mannor S. Reward constrained policy optimization. In: Proceedings of the 7th International Conference on Learning Representations, 2019
35 Ray A, Achiam J, Amodei D. Benchmarking safe exploration in deep reinforcement learning. ArXiv:1910.01708
36 Yang T Y, Rosca J, Narasimhan K, et al. Projection-based constrained policy optimization. ArXiv:2010.03152
37 Ma H, Liu C, Li S E, et al. Joint synthesis of safety certificate and safe control policy using constrained reinforcement learning. In: Proceedings of Learning for Dynamics and Control Conference, 2022. 97–109
38 Yu D, Ma H, Li S, et al. Reachability constrained reinforcement learning. In: Proceedings of the 39th International Conference on Machine Learning, 2022. 25636–25655
39 Li J, Hu X, Xu H, et al. Mind the gap: offline policy optimization for imperfect rewards. ArXiv:2302.01667

40 Li K, Jia Q S. Bridging scenarios in reinforcement learning with continuously generated relaying predictive models. In: Proceedings of the 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), 2022. 10–15
41 Li K, Jin X, Jia Q S, et al. An OCBA-based method for efficient sample collection in reinforcement learning. IEEE Trans Automat Sci Eng, 2024, 21: 3615–3626
42 Feng S, Sun H, Yan X, et al. Dense reinforcement learning for safety validation of autonomous vehicles. Nature, 2023, 615: 620–627
43 Liu S, Li K, Jia Q S. An accelerated safety probability estimation method for control policies of autonomous vehicles in open environments. In: Proceedings of the 2023 China Automation Congress (CAC), 2023. 2412–2417
44 Chen C H, Lin J, Yücesan E, et al. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. Discrete Event Dynamic Syst, 2000, 10: 251–270
45 Chen C H, Lee L H. Stochastic Simulation Optimization: an Optimal Computing Budget Allocation. Singapore: World Scientific, 2011
46 Melo F S. Convergence of Q-Learning: A Simple Proof. Tech Rep. 2001
47 Watkins C J, Dayan P. Q-learning. Mach Learn, 1992, 8: 279–292
48 Rummery G A, Niranjan M. On-Line Q-Learning Using Connectionist Systems. CUED/F-INFENG/TR 166. 1994
49 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518: 529–533
50 Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016. 2094–2100
51 Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning, 2016. 1995–2003
52 Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning, 2016. 1928–1937
53 Schulman J, Levine S, Moritz P, et al. Trust region policy optimization. In: Proceedings of the 32nd International Conference on Machine Learning, 2015. 1889–1897
54 Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms. ArXiv:1707.06347
55 Lillicrap T, Hunt J, Pritzel A, et al. Continuous control with deep reinforcement learning. In: Proceedings of the 4th International Conference on Learning Representations, 2016
56 Fujimoto S, van Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: Proceedings of the 35th International Conference on Machine Learning, 2018. 1587–1596
57 Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proceedings of the 35th International Conference on Machine Learning, 2018. 1861–1870
58 Jaakkola T, Jordan M I, Singh S P. On the convergence of stochastic iterative dynamic programming algorithms. Neural Comput, 1994, 6: 1185–1201
59 Ge Q, Sun Q, Li S E, et al. Numerically stable dynamic bicycle model for discrete-time control. In: Proceedings of the 2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops), 2021. 128–134
60 Yuan Z, Hall A W, Zhou S, et al. Safe-control-gym: a unified benchmark suite for safe learning-based control and reinforcement learning in robotics. IEEE Robot Autom Lett, 2022, 7: 11142–11149
61 Chen C, Liu Y, Kreiss S, et al. Crowd-robot interaction: crowd-aware robot navigation with attention-based deep reinforcement learning. In: Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), 2019. 6015–6022
62 Van den Berg J, Lin M, Manocha D. Reciprocal velocity obstacles for real-time multi-agent navigation. In: Proceedings of the 2008 IEEE International Conference on Robotics and Automation, 2008. 1928–1935
63 Barnard G A. Statistical inference. J R Stat Soc Ser B, 1949, 11: 115–139

Appendix A Choice of ensemble size

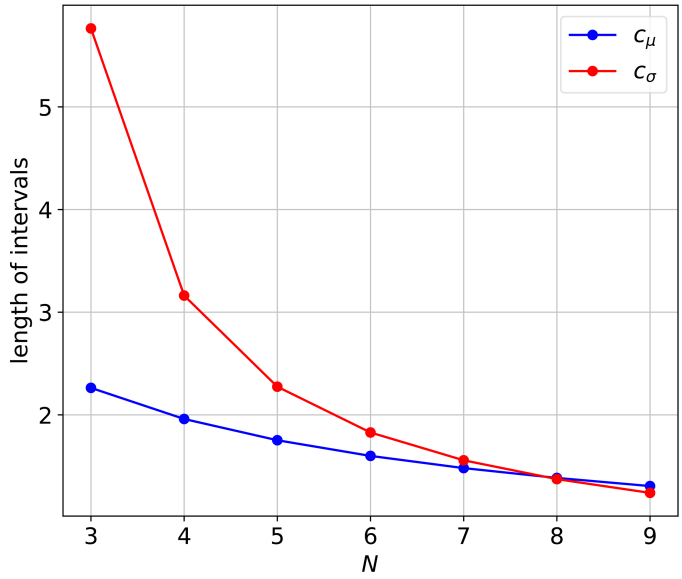

Figure A1 (Color online) Length of confidence intervals with respect to N.

In this section, we will discuss the choice of ensemble size, which is the number of value networks and constraint networks in our proposed algorithm. The ensemble size is a trade-off between the sampling accuracy and the computational cost.

In our proposed framework, ensemble size represents the number of sampling times for the distribution of the value function and constraint function. From [63], we know the expected length of the confidence interval of the mean and variance of the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ is as follows:

$$\begin{aligned} c_\mu &= \frac{2\sigma \mathcal{Z}_{v/2}}{\sqrt{N}}, \\ c_\sigma &= \sigma \left(\sqrt{\frac{N-1}{\mathcal{X}^2_{N-1,1-v/2}}} - \sqrt{\frac{N-1}{\mathcal{X}^2_{N-1,v/2}}} \right), \end{aligned} \tag{A1}$$

where $\mathcal{Z}$ and $\mathcal{X}^2$ are quantiles of Gaussian and chi-square distributions, N is the ensemble size. An example of the expected length trend of the confidence interval of the mean and variance is shown in Figure A1. Additionally, there is a linear relationship between the amount of computation and ensemble size. In serial computation, it is manifested as computation time, while in parallel computation, it is manifested as the requirement for computing power or the number of computing devices.

Based on the computing ability the authors have and the accuracy requirement, we choose the ensemble size as 5 in our experiments.