УДК: 004.056
doi: 10.26583/bit.2024.2.06

Mohsen Abdollahzadeh Aghbolagh
School of Information Technology and Data Science,
Irkutsk National Research Technical University,
Lermontova Str., Irkutsk 664074, Russia
e-mail: mohsen.abdollahzadeh001@gmail.com, https://orcid.org/0000-0002-6414-2847

ENSURING SAFETY WHILE ENHANCING PERFORMANCE: ENCOURAGING REINFORCEMENT LEARNING BY ADDRESSING CONSTRAINTS AND UNCERTAINTY

Abstract. Striking a balance between safety and performance remains a critical concern, despite advancements in the field. To address this issue, a versatile framework named Safety Goes Along with Performance (SGAWP) is proposed, centered on off-policy algorithms grounded in value function optimization. SGAWP utilizes reinforcement learning to navigate the data space, emphasizing high task performance while addressing risks (such as undesirable states) by incorporating safety costs into the value function. By integrating uncertainty management and task performance constraints, SGAWP aims to achieve improved safety performance alongside respectable task performance. Moreover, SGAWP leverages curiosity-driven exploration to expand the data space and employs task policies to enhance safety policy performance. As a result, SGAWP enhances safety performance with minimal loss in task performance. Beyond its success in reinforcement learning, SGAWP holds promise for applications like autonomous driving, where safety is paramount. Through rigorous experimentation across various off-policy algorithms, SGAWP demonstrates robust generalization and achieves its objectives effectively.
Keywords: safe reinforcement learning constraint, off-policy, exploration, risk assessment.
For citation: AGHBOLAGH, Mohsen Abdollahzadeh. ENSURING SAFETY WHILE ENHANCING PERFORMANCE: ENCOURAGING REINFORCEMENT LEARNING BY ADDRESSING CONSTRAINTS AND UNCERTAINTY. IT Security (Russia), [S.l.], v. 31, no. 2, p. 90–110, 2024. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1635. DOI: http://dx.doi.org/10.26583/bit.2024.2.06.

Introduction

Reinforcement learning achieves optimal policy through continuous interaction with the environment. When exploration exceeds safe boundaries, it may lead to dangerous states due to risky actions, posing serious consequences. Crafting and training policies meticulously become imperative to proactively mitigate unsafe actions and emergence of unsafe states. Real-world applications of reinforcement learning, such as autonomous driving [1], financial trading [2], and medical diagnosis [3], necessitate high safety requirements for their success, heavily relying on safety assurance.

Despite considerable attention to safe reinforcement learning, unresolved issues persist, particularly the inherent trade-off between safety performance and task performance. Many applications hesitate to adopt reinforcement learning due to perceived trade-offs between safety and improved task performance. This hesitation arises from concerns that excessive focus on safety may limit agent exploration and innovation, resulting in inefficient task execution. Additionally, stringent safety requirements may cause overreaction to potential threats, leading to resource wastage. Therefore, it is crucial for algorithms to prioritize safety performance while equally emphasizing task performance. Maintaining this balance presents a challenge in safe reinforcement learning.

Safety Goes Along with Performance

Existing methods for safe reinforcement learning can be categorized into two groups based on implementation methods: those modifying learning objectives [4] and those altering exploration

strategies [5]. Methods modifying learning objectives introduce risk-related factors into reinforcement learning objective functions through online interactive feedback mechanisms, transforming constrained problems into unconstrained ones. One such method, the Lagrangian method [6], incorporates risk as opportunity constraints and conditional risk values, converting constrained Markov Decision Process (CMDP) problems into unconstrained optimization problems. However, this method relies on strict assumptions and may struggle with selecting appropriate Lagrange multipliers, limiting its applicability in complex scenarios.

Exploration-based methods modify exploration strategies and integrate risk indicators to ensure safety during exploration and exploitation [7]. Nevertheless, inherent risks persist with this approach. Offline reinforcement learning represents a different extreme by training solely on static datasets without environment interaction, avoiding exploration [8]. However, deployment safety lacks constraints in offline reinforcement learning, often resulting in safety compromises during deployment [9]. Additionally, offline reinforcement learning methods may encounter out-of-distribution (OOD) actions during real-world interaction, posing a distribution shift problem.

In this study, our primary goal is to enhance safety performance while maintaining task performance standards. We combine objective-based and exploration-based ideas, training the safety policy in a high task performance data space constrained by the task policy. The safety policy then restricts a subset of the data space with safety performance. Our method incorporates task and safety policies, where the task policy utilizes existing reinforcement learning algorithms with high task performance to constrain the state-action space with high rewards but high safety risks for the safety policy. Conversely, the safety policy employs a value constraint method from reinforcement learning, introducing risk factors into the Q value to further constrain the state-action space. This approach focuses on limiting the state-action space to areas with low risk while balancing high reward and high risk areas. Ultimately, the safety policy learns safety concepts, even with assistance from unsafe trajectories. Training and testing results demonstrate superior safety performance and improved task performance of our algorithm.

Contributions

Our contributions include:

- Enhancing safety performance without significant task performance loss by introducing safety constraints, achieving a balance between safety and task performance.
- Designing a framework compatible with off-policy reinforcement learning methods based on value function optimization.
- Employing the three-point estimation technique from risk assessment to address uncertainty challenges, applying task constraints to the safety policy and enhancing its task performance.
- Introducing exploration mechanisms to encourage agents to explore beyond their comfort zones, ensuring diversity within the data space.
- Designing a promising solution for safety-sensitive applications like autonomous driving, aiming to enhance safety while maintaining task performance.

The paper is organized as follows: Section 2 covers the methodology, Section 3 explains our method in detail, Section 4 outlines our experiment construction, Section 5 presents and analyzes experimental results, Section 6 describes the application prospects of the algorithm, and finally, Section 7 summarizes and discusses future directions.

1. Related work

1.1. Safe Reinforcement Learning

Garcia and Fernandez [10] defined safe reinforcement learning as a paradigm that integrates safety and risk concepts into reinforcement learning frameworks. The primary goal is to

enable agents to acquire optimal policies maximizing task performance while preventing hazardous behaviors or unsafe outcomes during decision-making processes. Since 2015, researchers have introduced numerous secure reinforcement learning algorithms, building upon this foundational research.

To enhance agent safety, researchers have proposed various safe reinforcement learning algorithms. One common approach is to introduce constraints or penalty mechanisms limiting the agent's action space to prevent dangerous actions. These methods often fall under the category of those based on value constraints [11, 12] Additionally, designing a suitable safety reward function can motivate agents to avoid dangerous behaviors, such as rewarding stay time in safe zones while penalizing entry into dangerous zones [13]. Moreover, addressing uncertainty in reinforcement learning is crucial for safety. Handling uncertain environmental conditions judiciously can significantly improve an agent's robustness, enabling discerning decisions in its surroundings and enhancing overall safety performance [14].

Safe reinforcement learning is essential for addressing safety challenges in real-world tasks. By combining constraints, reward function design, uncertainty processing, and other methods, safe reinforcement learning effectively improves agent safety, ensuring the application of reinforcement learning in complex tasks. However, balancing safety and performance optimization remains a critical research direction.

1.2. The Actor-Critic Architecture

The value function [15] is a fundamental concept in reinforcement learning, and various methods, such as Monte Carlo method [16] and Temporal Difference learning [17], approximate state value or action-value functions using parameters. Deep reinforcement learning frameworks utilize neural networks to fit the value function, exemplified by early examples like Deep Q-Networks (DQN) [18]. Subsequent variants, including Double DQN [19], address challenges such as value function overestimation in action states encountered in Q-learning [20].

Unlike the indirect policy selection method based on the value function, policy-based methods represent policies linearly or nonlinearly and find parameter values maximizing the learning objective, such as expected cumulative discount reward. Trust Region Policy Optimization (TRPO) [21] is a notable algorithm in this category, employing KL divergence to enforce constraints on policy proximity [22].

Algorithms based on the value function framework face challenges in directly producing action value outputs, especially for continuous action spaces, and they may suffer from high deviation and instability. Policy gradient-based algorithms require sampling many trajectories, leading to high variance and gradient noise, making training unstable and policy convergence difficult. The actor-critic architecture [23] combines the advantages of value function and policy gradient algorithms, mitigating their shortcomings to some extent and forming a more comprehensive agent. This architecture comprises actor and critic networks, with the former generating policies and the latter evaluating policies. Several algorithms leverage the Actor-Critic framework, such as Soft Actor-Critic (SAC) [24], from which our method draws inspiration.

1.3. Constrained Markov Decision Process

A typical problem encountered in reinforcement learning involves framing an infinite-horizon deterministic Markov Decision Process (MDP). An MDP is typically described as a 6-tuple, denoted as $M = (S, A, \gamma, r, \mu, T)$, where S represents the state space, A is the action space, $r : S \times A \rightarrow R$ denotes the reward function, $0 \leq \gamma < 1$ is the discount factor, μ is the initial state distribution, and $T : \mathrm{S} \times \mathrm{A} \rightarrow \mathrm{S}$ refers to the state transition function used to describe the dynamical model. An agent interacts with the environment by executing its policy $\pi : \mathrm{S} \rightarrow \mathrm{A}$ to obtain a reward r, with the expected discounted cumulative reward given by:

$$J(\pi) = E\mu,\pi,T\ [\sum\infty\ t=0\ \gamma\ t\ r\ (\ st\ ,\ at\ )], \quad (1)$$

The primary objective is to find a strategy π that maximizes the expected discounted cumulative reward $J(\pi)$.

The standard safe reinforcement learning problem is typically defined as a Constrained Markov Decision Process (CMDP) [25]. The cost constraint $C = \{c, d\}$ is added to the basis of the standard Markov decision process M, where $c : S \times A \rightarrow R$ represents a cost function, and $d \in R$ is the safety threshold. Therefore, CMDP = (S, A, γ, r, c, d, μ, T). In this context, the safe reinforcement learning problem aims to solve for an optimal policy that satisfies both the expected discounted cumulative reward and the safety constraint. The goal is to find a policy that satisfies the following conditions:

$$\max \pi\ J(\pi)\ \text{s.t.}\ Jc\ (\pi) = E\ [\sum\infty\ t=0\ \gamma\ t\ c\ (\ st\ ,\ at\ )\ ] \leq d \quad (2)$$

Setting $d = 0$ indicates perfect constraint satisfaction, which is an ideal scenario. However, it is challenging to achieve perfect constraints, as they often impact task performance. Therefore, a compromise method is employed to find strategies that maintain a certain degree of task performance while significantly improving safety performance.

The data space is defined as $\Pi = \{(st, at, st+1)|st \in, at \sim \pi(\cdot|st), st+1 \sim T(\cdot|st, at)\}$. The data space with high reward and high-risk features is denoted as $\Pi task$, and the corresponding policy is denoted as the task policy $\pi task$. In contrast, the data space with low-risk features is denoted as $\Pi safe$, and the corresponding policy is denoted as $\pi safe$.

The goal is to find the common intersection – the set of targets that satisfy both high reward and low risk, denoted as $\Pi target := \Pi task \cap \Pi safe$, and the policy to constrain the intersection region is our target policy $\pi target$.

1.4. Three-Point Estimation Approach

The Three-Point Estimation approach is a method commonly utilized in project management [26] and risk assessment to tackle the uncertainties inherent in conventional single-point estimation techniques. In project management, accurately estimating progress and costs can be challenging due to limited historical data, leading to uncertainty and risk in estimation. To address this challenge and improve the accuracy of duration estimation, the Three-Point Estimation approach is employed to enhance the implementation of techniques like the Program Evaluation and Review Technique (PERT) [27].

In various fields, including project planning and schedule management, the Three-Point Estimation approach is frequently employed. The core idea involves introducing uncertainty into the project plan and obtaining more precise forecasts of time and resource requirements by considering estimates under different scenarios. This method aids project managers in gaining a deeper understanding of the project's risk profile, allowing them to implement suitable measures to ensure smooth project progression.

When estimating the duration or cost of an activity, a thorough assessment typically involves considering three scenarios: the most optimistic (representing a favorable outcome), the worst-case (indicating an unfavorable outcome), and a general estimate. These assessments result in three key values: the most optimistic duration (Tp), the most pessimistic duration (To), and the most probable duration (Tm). Utilizing these three estimates, formulas such as the "triangle distribution" and Beta distribution are employed to calculate the expected duration (average duration Te).

Therefore, the Three-Point Estimation approach provides more precise estimation points and holds significant relevance across diverse domains. It excels in managing uncertainties and serves as a reliable basis for project planning and decision-making, offering valuable insights and guidance.

2. Methodology

We begin by presenting an overview of the comprehensive architecture of SGAWP. Subsequently, we offer detailed explanations for the task strategy, safety strategy, and methods to address uncertainty and diversity, respectively.

2.1. Overview

At a conceptual level, the SGAWP framework is a collaborative dual-policy framework comprising the task strategy and safety strategy. The task strategy integrates existing algorithms with high task performance in reinforcement learning, ensuring task performance within the framework. The safety strategy guarantees safety performance by employing the method of value constraint and introducing risk factors to further confine the safety space within the high-performance space constrained by the task strategy, thereby maintaining a certain level of task performance while enhancing safety performance. Moreover, for the safety policy, our method incorporates the uncertainty factor and utilizes the three-point estimation method from project management to introduce task performance constraints. To enhance data space diversity, our method includes an exploration mechanism. By leveraging the task policy, safety policy using the three-point estimation method, and exploration mechanism, the resulting safety policy becomes the final target policy.

Algorithm 1 Safety Goes Along with Performance

1: **Input:**Initialize replay buffer R, the task strategy $\pi task$, the safety policy $\pi saf\ e$, the list $ep\_min\_return\_list$, the task performance constraint threshold ep_min_return.
2: **Output:**The safety policy $\pi saf\ e$ with task performance constraints.
3: **for** $episode$ = 1, **do**
4: Enrich the data space $\Pi task$ with high-performance and high-cost data through the implementation of the task strategy $\pi task$ combined with an exploration mechanism.
5: $R = R \cup \Pi task$
6: Update $\pi task$
7: Incorporate low-cost data with uncertainty constraints into the data space $\Pi saf\ e$ by implementing the task strategy $\pi saf\ e$ in conjunction with an exploration mechanism.
8: $R = R \cup \Pi saf\ e$
9: Update $\pi saf\ e$, $ep\_min\_return\_list$
10: Update the threshold ep_min_return (Three-Point Estimation Algorithm)
11: **end for**

The flowchart of SGAWP is depicted in Fig. 1, and the algorithmic framework is presented in Algorithm 1. We summarize the general flow as follows: We initialize several components, including the replay buffer, task policy, safety policy, a list of minimum returns for one episode, and the minimum return threshold as a task performance constraint (Line 1). Subsequently, we commence the iterative process of constraining the target data space with the dual objectives of achieving high task performance and high safety performance (Lines 3-11). Within the task strategy, we impose constraints on the diverse data space characterized by high task performance and low safety performance (Line 4). We then update both the task policy and the replay buffer (Lines 5-6). Simultaneously, in the safety strategy, we constrain the low-cost diverse data space, incorporating uncertainty constraints (Line 7). Here, we also update the safety policy and the

replay buffer (Line 8-9). Finally, we refine the threshold value using the three-point estimation method (Line 10).

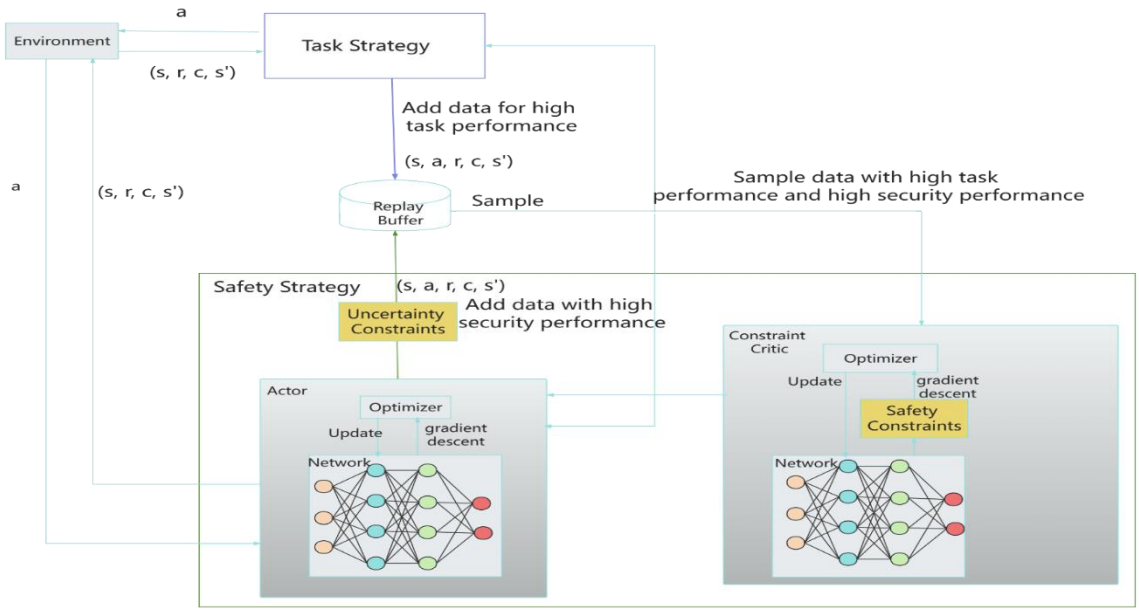


Fig. 1. Overview of SGAWP's architecture. SGAWP comprises four main components: the environment, the task strategy employing off-policy methods, the safety strategy utilizing the actor-critic framework, and the data space

2.2. Task Strategy

The current reinforcement learning methods have reached notable task performance levels and achieved rewards close to the environmental limit across various simulation environments. As a result, SGAWP integrates the existing off-policy.

Our framework leverages these algorithms to restrict the data space with high task performance.

2.3. Safety Strategy

The next phase involves further narrowing down the space of high safety performance while maintaining high task performance. This is achieved through the implementation of the safety policy, a pivotal component of our framework.

The safety policy strategy can be seamlessly integrated into other off-policy algorithms centered on value function optimization, such as Deep Deterministic Policy Gradient (DDPG) [28]. In DDPG, comprising policy and value function networks, both networks have target and actual components, updated using a soft update strategy to ensure training stability and convergence. The policy network acts as the actor, generating deterministic actions, while the value function network serves as the critic, evaluating state-action values.

Our safety strategy, termed the "conservative-critic," is incorporated into the value function network. It introduces safety soft constraints by integrating them with the action-state value function, adding an immediate cost alongside the conventional reward when evaluating the value of a state subsequent to an action. This cost influences the decision-making process, enabling actions to consider safety concerns and determine secure courses of action.

The update process for the value function involves utilizing both the conservative-Q value and the Conservative-Q-target value from the target network, alongside the output of the estimated

network, incorporating actual rewards and immediate costs. The conservative-critic estimated network is updated using gradient descent of temporal difference, while the target network undergoes gradual updates via a soft update strategy. The policy network is updated through gradient ascent to optimize Q-value maximization.

$$Q(s, a) = r + \gamma * \max a' Q (s' , a') – c \quad (3)$$

The core formula for fitting the conservative-critic is as follows: $(s,)=r+\gamma*\max Q\ s',a' -c$, where c represents the immediate cost. By incorporating a safety cost into the value function calculation at each step, a soft safety constraint is imposed, gradually enhancing safety performance. We illustrate the integration of SGAWP into DDPG through pseudocode in Algorithm 2 and a flow diagram in Fig. 2. Notably, we utilize the task performance threshold calculated with uncertainty as a screening criterion for the target data space and incorporate the immediate cost as a soft safety constraint within the expected Q value in the safety strategy. This process results in the generation of a safety data space that considers task performance.

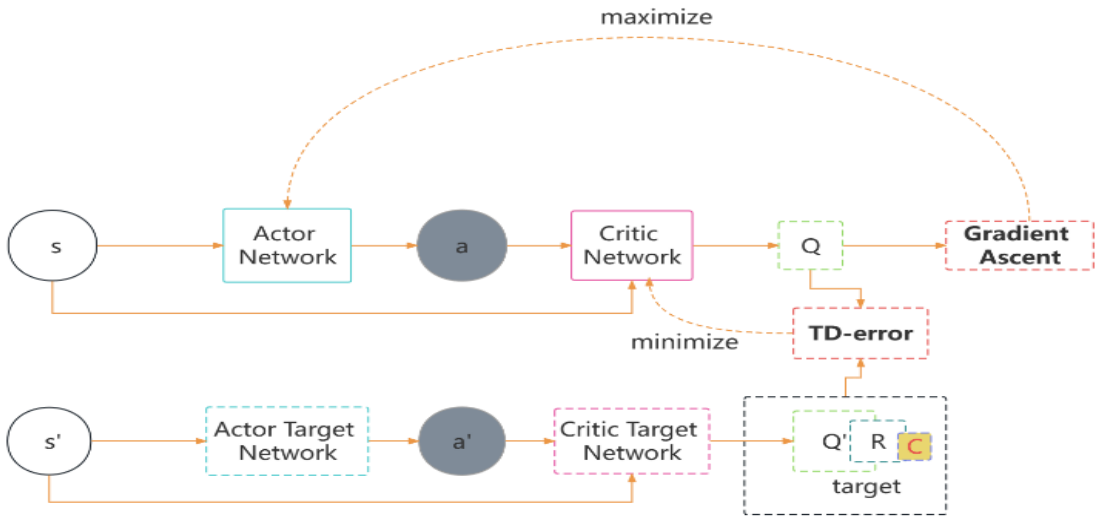


Fig. 2. Depicts DDPG with SGAWP, where a cost component (depicted by the yellow square) is integrated into the calculation of the expected Q value for optimization

2.4. Uncertainty Constraint and Diversity

In the realm of three-point estimation, the beta distribution emerges as a continuous probability distribution. Here, the pessimistic and optimistic estimates correspond to the upper and lower limits of this distribution, while the most probable estimate coincides with its peak. Unlike the triangular distribution, the beta distribution offers a more precise handling of uncertainty. Hence, we embrace the beta distribution and utilize the subsequent calculation formula:

$$T_e = T_o + 4 * T_m + T_p⁄6 \quad (4)$$

Our approach aims to reinforce the influence of the space characterized by high task performance and high cost, as constrained by the task policy, through integration with the safety policy. Additionally, we aim to further constrain the space characterized by high safety performance. To achieve this objective, we utilize the experience pool to filter the space constrained by the safety policy and set a threshold to filter the experience generated by the safety policy. Since the safety policy inherently includes soft safety constraints, which offer some assurance regarding safety performance, our screening process primarily focuses on optimizing

task performance. Consequently, the threshold used for screening corresponds to the lower limit of task performance, typically represented by the reward value.

As the training progresses, the threshold of task performance dynamically changes, but the overall trend shows improvement, making it challenging to accurately determine the threshold. Therefore, in conjunction with the three-point estimation method, we set the frequency of updating the threshold to k episodes. During each episode, we calculate the lowest task performance of the current round. Once the update frequency is reached, we consider the minimum value of the lowest task performance set as the most pessimistic estimate in the three-point estimation method, the average ensemble value as the most probable estimate, and the maximum ensemble value as the most optimistic estimate. The threshold is then updated according to the estimation formula.

By employing the three-point estimation method, the strategy effectively mitigates uncertainty's impact on the lower bound of task performance (Lockwood and Si, 2022), yielding a more precise assessment of task performance's lower limit and aiding safety policies in constraining high-safety performance areas.

We outline the Algorithm 3 for the three-point estimation method.

Following the central policy's restriction of a state action space with high reward and cost, combined with the safety policy, the low-cost space is further restricted. However, the safety policy tends to lean towards conservatism. Introducing the safety policy may lead the agent into a local safety environment, forming a "comfort zone" dictated by the safety policy [29]. Thus, to prevent the agent from being confined solely to the safe action space and encourage exploration beyond its "comfort zone," we integrate an exploration mechanism [30].

This mechanism encourages the agent to explore beyond its current boundaries, enabling a more comprehensive understanding of environmental information and enhancing the agent’s learning efficiency.

Exploration has yielded many impressive results, such as the curiosity-driven mechanism ICM [31] and the never give up (NGU) [32]. These exploration algorithms mainly realize intrinsic reward-driven exploration in reinforcement learning. In our experiments, we utilize a curiosity-driven mechanism to incentivize the agent to explore outward.

The curiosity mechanism prompts the agent to take actions aimed at reducing uncertainty regarding its ability to predict outcomes. Curiosity arises as an intrinsic motivator, stemming from disparities in the agent’s capacity to forecast the consequences of its actions within its current state. Quantifying curiosity error requires constructing an environmental dynamics model, predicting the subsequent state based on the current state and action. The curiosity mechanism is designed to generate the agent's curiosity reward, driving it out of the comfort zone.

The fundamental equation of the curiosity mechanism is expressed as Curiosity = ϕpredict (st+1) - (st+1), where (st) represents the feature representation of the current state st.

In our approach, the prediction error of the forward dynamic model (which indicates the variance between predicted and actual subsequent states) is provided separately to both the task policy and the safety policy. This acts as an inherent reward, intended to encourage curiosity and promote exploration within each policy.

Algorithm 2 Reinforcement Learning Improved by SGAWP

Input: Randomly Initialize constraint critic network $Q_c(s,a|\theta^{Q_c})$ and actor $\mu(s|\theta^{\mu})$ with weights θ^{Q_c} and θ^{μ}.
Initialize target constraint network Q_c' and μ' with weights $\theta^{Q_c'} \leftarrow \theta^{Q_c}$, $\theta^{\mu'} \leftarrow \theta^{\mu}$.
Replay buffer R in Algorithm 1.
The expected minimum return per episode *ep_min_return* in Algorithm 3.
Output: Π_{safe}
for $episode = 1, N$ **do**
  Initialize a random process $\mathcal{N}$ for action exploration.
  Receive initial observation state s_0.
  for $t = 1, T$ **do**
    Select action $a_t = \mu(s_t|\theta^{\mu}) + \mathcal{N}_t$ according to the current policy π_{safe} and exploration noise.
    Execute action a_t and observe reward r_t and observe cost c_t and observe next state s_{t+1}.
    if $r_t \geq ep\_min\_return$ **then**
      $\Pi_{safe} = \Pi_{safe} \cup (s_t, a_t, r_t, c_t, s_{t+1})$.
    end if
    Set $y_i = r_i + \gamma * Q_c'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q_c'}) - c_i$.
    Update constraint critic by minimizing the loss: $L = \frac{1}{N}\sum_i \left(y_i - Q_c\left(s_i, a_i \mid \theta^{Q_c}\right)\right)^2$.
    Update the actor policy π_{safe} using the sampled policy gradient.
    Update the target network: $\theta^{Q_c'} \leftarrow \tau\theta^{Q_c} + (1-\tau)\theta^{Q_c'}$, $\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$
  end for
end for

Algorithm 3 Safeguarding Uncertainties through Risk Estimation in Safe Reinforcement Learning

Input: The list *ep_min_return_list* consisting of the minimum per-episode returns of K episodes is used as input.
Output: The expected minimum return per episode *ep_min_return* as a task performance constraint for the next K episodes.
Sort the list *ep_min_return_list* in ascending order in place.
$T_p = ep\_min\_return\_list[0]$
$T_o = ep\_min\_return\_list[-1]$
$T_m = the average of\ ep\_min\_return\_list$
$ep\_min\_return = (T_p + T_m * 4 + T_o)/6$

3. Experiment

In our experiments, we primarily assess our approach on navigation tasks, aiming to determine:

- Whether our method can sustain comparable rewards to the comparison methods, thereby ensuring consistent task performance.
- Whether our method can decrease safety costs and substantially enhance safety performance in comparison to the comparison methods.

3.1. Environment Configuration

All our experiments are conducted using Safety-Gym, a tool developed by OpenAI to facilitate research in safety exploration [33]. This platform serves as a benchmark for constrained robot navigation, assessing how effectively an agent adheres to safety constraints during reinforcement learning training, such as how a self-driving car learns to avoid accidents during training.

The SafetyGym toolkit, built upon the MuJoCo physics engine and leveraging OpenAI's Gym interface, comprises two components. The first is an environment creator that enables users to design new environments by combining various physics elements, goals, and safety

requirements. The second part consists of pre-configured benchmark environments primarily aimed at standardizing the quantitative evaluation of safe reinforcement learning methods.

SafetyGym features three prebuilt robots (point, car, doggo) and three primary tasks (goal, button, push), each with two difficulty levels. This platform gradually challenges the target AI by providing rewards or penalties, allowing them to learn through trial and error, which can sometimes lead to risky behavior as the agent seeks to maximize rewards.

In our experiments, we opt for the maximum difficulty level to facilitate the accumulation of safety costs and enable a more straightforward comparison of the effectiveness of our method in hazardous environments. We explore combinations of three agents and three tasks, thereby evaluating our method and comparison methods across nine environments in navigation tasks. In the following sections, we denote environments by concatenating the names of different agents, tasks, and difficulty levels, such as PointGoal2.

Fig. 3 presents an overview of various environment types based on robot classification, with nine specific environments corresponding to each robot type based on task and difficulty level combinations. In Fig. 4, we use the pointgoal environment as an example to provide a detailed introduction.

Environment setup occurs either at the beginning of each trial or once the agent reaches the goal. Multiple hazards are present in the environment, necessitating constraints. In these environments, both reward and safety costs are measured by the signed distance between objects. Dynamics for cars and doggo are more complex compared to point environments.

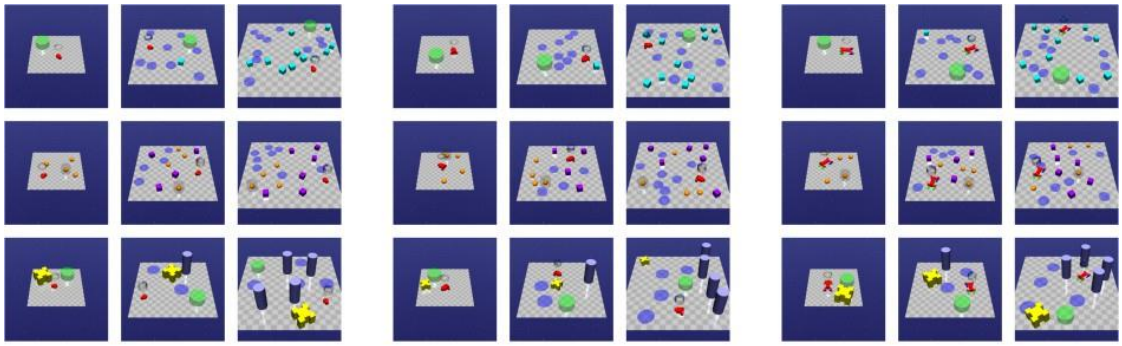


Fig. 3. Panoramic views of all environments for Point Benchmark, Car Benchmark, and Doggo Benchmark

3.2. Baselines and Comparisons

We employ Proximal Policy Optimization (PPO) (Schulman, Wolski, Dhariwal, Radford, and Klimov, 2017), TRPO, and SAC as comparison methods, where:

- PPO is a policy-based reinforcement learning algorithm. It introduces a constraint on policy updates during optimization to maintain similarity between policies before and after updates. This constraint aims to improve the algorithm's stability and sampling efficiency.
- TRPO is a policy-based reinforcement learning algorithm. By restricting the size of policy updates, TRPO utilizes the trust region to ensure policy stability during the update process and prevent excessively large changes, thereby enhancing training stability and convergence.
- SAC is a reinforcement learning algorithm based on the value function. It encourages policy exploration by maximizing entropy and considers the trade-off between reward and exploration, enabling the agent to learn a robust and efficient policy.

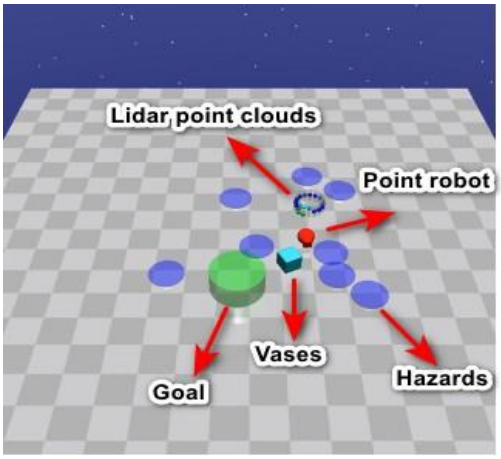


Fig. 4. Illustration of Safexp-PointGoal2. Employing the Point cloud dataset (Lidar point clouds) gathered from the sensor, the agent (Point) traverses the environment, avoiding hazardous areas (Hazards, Vases), and ultimately reaching its destination (Goal)

In addition, as we introduced a flexible framework for off-policy algorithms based on value function optimization, we experimented with various combinations of task policies and safety policies for SAC and DDPG algorithms, in addition to the experiments mentioned above. Each algorithm can function as either a task policy or a safety policy with the imposition of safety strategies. Consequently, there are four combinations under each task, denoted by {task policy benchmark algorithm}-SP{safety policy benchmark algorithm}. Regarding the experiments involving different combinations, we present and analyze them as part of the ablation experiment.

3.3. Metrics

This section introduces the two metrics utilized to evaluate our experiments:

• Episode Reward (ER): The cumulative reward obtained by the agent in each iteration reflects the total rewards accumulated throughout its adherence to the ongoing interim strategy within that iteration. This metric serves as a straightforward tool for assessing the algorithm's task performance in each round. It aids in evaluating the algorithm's overall impact on a specific task and facilitates the comparison of algorithmic effectiveness across different tasks. A round concludes under two specific conditions: when the task successfully achieves its objectives and concludes, or when the maximum allowable round steps reach a limit of 1000. The expression $\sum T_{ept=0}\, rt$ represents the task performance of an agent on a particular task for one round guided by an intermediate policy, where $T_{ep} \leq 1000$.

• Episode Safety Cost (ESC): The total safety cost incurred by the agent in each round refers to the sum of all safety costs under the guidance of the current intermediate strategy in an episode. This metric is commonly used in risk-sensitive tasks and can measure the safety performance of the algorithm on a specific task. The magnitude of the total safety cost in each episode can intuitively indicate whether the agent avoids more unsafe behaviors in the current episode. The expression $\sum T_{ept=0}\, ct$ represents the safety performance of an agent on a particular task for one episode guided by an intermediate policy.

It is important to note that SafetyGym's environment features sparse rewards. To ensure robustness and statistical significance, we conducted three random seed replicates for each run of every algorithm in each environment for every task. Subsequently, we recorded and reported the average results, enhancing credibility and statistical reliability.

4. Results and Analysis

Our experimental study presents the task performance and safety performance of four algorithms across nine combinations.

For visual representation, each performance is illustrated across nine separate graphs. Within these graphs, there are three rows, each corresponding to a specific type of agent, and three columns, each representing a distinct type of task. All graphs are generated at a consistent difficulty level of 2. Since the reward is sparse in the environment, the total return of the round is low after smoothing. We conducted three independent runs for each algorithm to ensure plausibility and statistical significance. The solid curve represents the average of three runs, while the shaded area encompasses the mean with one standard deviation.

4.1. Task Performance

In Fig. 5, from the perspective of agent and task, it can be observed that in all tasks of the point agent, the rewards of our method and the other three comparison methods are mostly consistent, and the task performance is similar in the Goal task and Push task. For the Button task, due to the nature of the task, the task performance of our algorithm has a gap compared to other methods. In all tasks of the Car agent, there are apparent differences in task performance between different algorithms. The task performance of our method is basically at the top, second only to or close to the best-performing TRPO. In terms of stability, the curve volatility of our method is relatively small, while the task performance of SAC in the three tasks is poor, especially on the Goal task. For the Doggo agent, our method has significant advantages in task performance, which is similar to SAC and far superior to PPO and TRPO, but also exhibits minimal fluctuation and far superior stability compared to SAC.

4.2. Safety performance

From Fig. 6, it is evident that our approach exhibits minimal total round cost across all experiments, demonstrating a certain level of generalization along with exceptionally high safety performance. Moreover, our method shows minimal volatility and high stability.

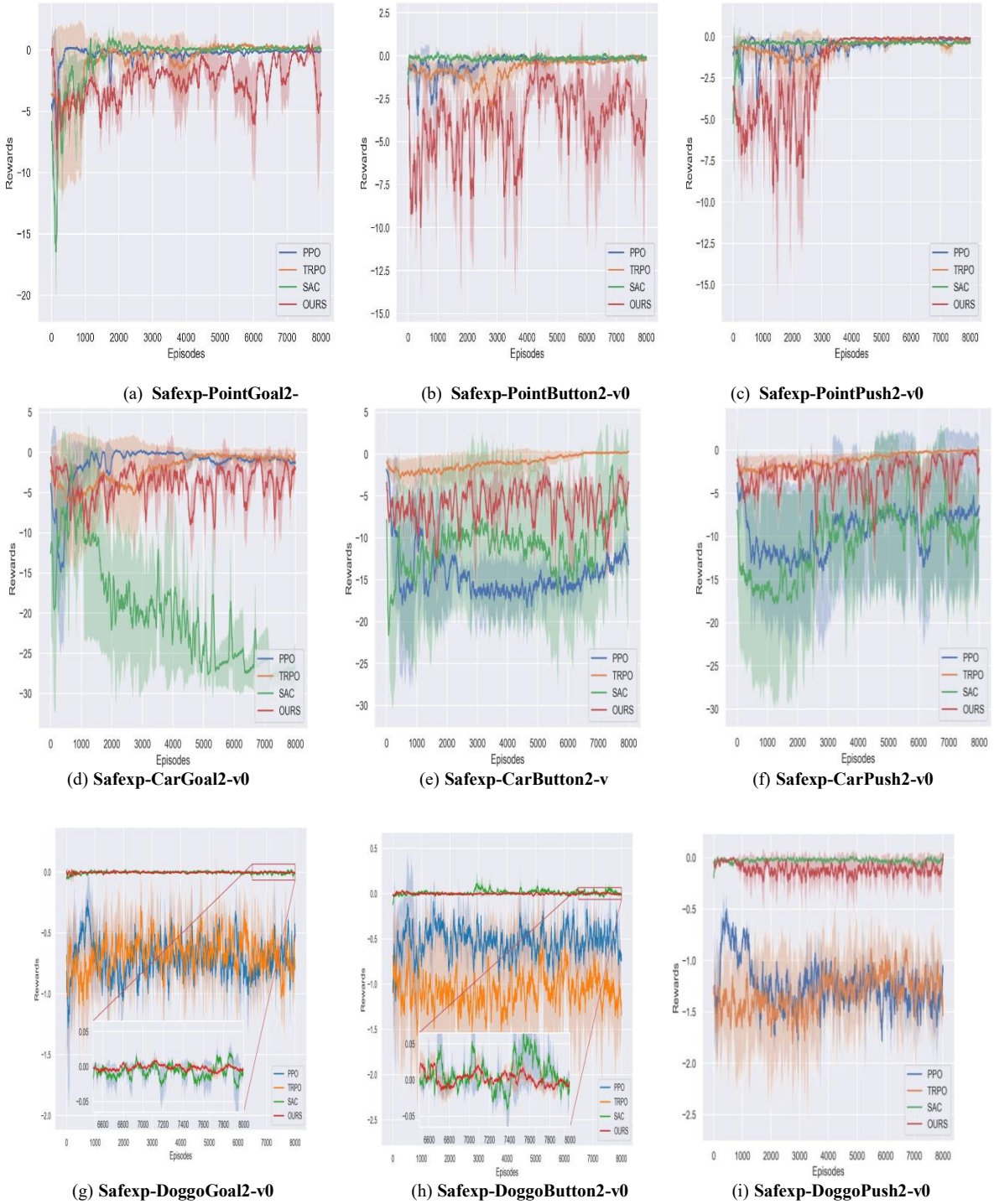
Table 1 presents comparative experiments regarding the exploration mechanism, where SGAWP+Eploration indicates the utilization of exploration in SGAWP, while SGAWP-Exploration denotes the absence of exploration in SGAWP.

Table 1. Comparative experiments about the exploration mechanism, where SGAWP+Eploration means using exploration in SGAWP and SGAWP-Exploration means not using exploration in SGAWP

Environment	Algorithm	Average Episode Reward	Average Episode Cost
Safexp-PointPush2-v0	SGAWP+Eploration	$-\mathbf{1.929} \pm 0.263$	$\mathbf{22.428} \pm 2.348$
	SGAWP-Eploration	-3.207 ± 2.524	23.647 ± 6.928
Safexp-CarGoal2-v0	SGAWP+Eploration	$-\mathbf{3.930} \pm 0.559$	$\mathbf{21.608} \pm 10.328$
	SGAWP-Eploration	-4.117 ± 0.384	30.373 ± 2.778
Safexp-CarPush2-v0	SGAWP+Eploration	$-\mathbf{2.963} \pm 0.930$	$\mathbf{27.870} \pm 10.445$
	SGAWP-Eploration	-3.724 ± 0.998	31.133 ± 12.449

Furthermore, it consistently achieves 0 safety violation constraints across all tasks associated with the Doggo agent. In contrast, when comparing these results to those obtained from experiments involving the other three comparison methods, most of them exhibit higher total round costs and demonstrate inferior safety performance. Moreover, there is significant

volatility, indicating low stability. The safety performance of PPO and TRPO is relatively poor across all tasks. SAC shows comparable safety performance to our method only in all tasks of the Doggo agent. However, safety risks persist, and the generalization is significantly inferior to our method.

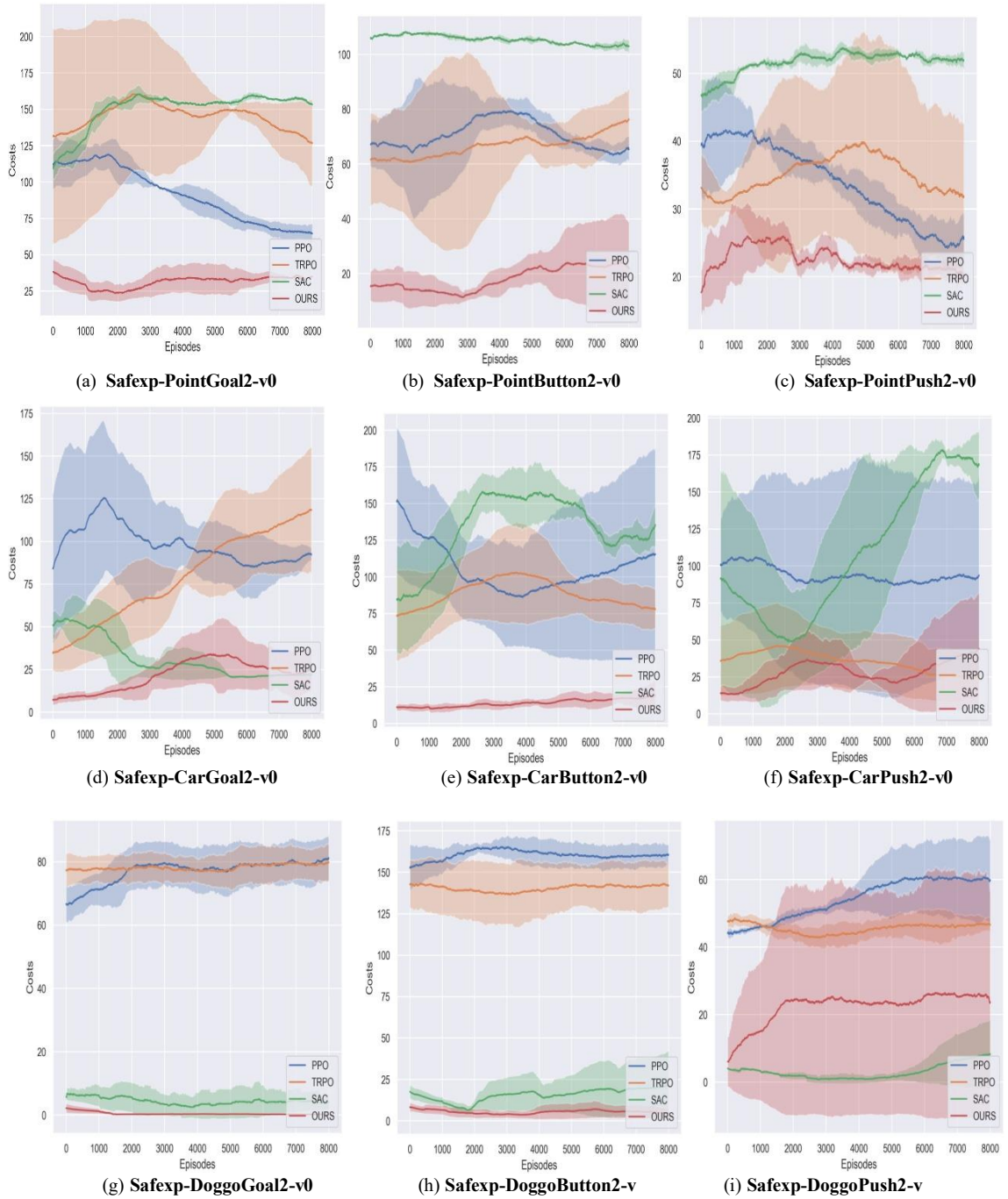


(a) **Safexp-PointGoal2-** (b) **Safexp-PointButton2-v0** (c) **Safexp-PointPush2-v0**

(d) **Safexp-CarGoal2-v0** (e) **Safexp-CarButton2-v** (f) **Safexp-CarPush2-v0**

(g) **Safexp-DoggoGoal2-v0** (h) **Safexp-DoggoButton2-v0** (i) **Safexp-DoggoPush2-v0**

Fig. 5. Comparisons between our method and the baseline with respect to the total return of one episode



(a) **Safexp-PointGoal2-v0** (b) **Safexp-PointButton2-v0** (c) **Safexp-PointPush2-v0**

(d) **Safexp-CarGoal2-v0** (e) **Safexp-CarButton2-v0** (f) **Safexp-CarPush2-v0**

(g) **Safexp-DoggoGoal2-v0** (h) **Safexp-DoggoButton2-v** (i) **Safexp-DoggoPush2-v**

Fig. 6. Comparisons between our method and the baseline with respect to the total safety cost of a round

4.3. Ablation Experiment

This section delves into ablation experiments focusing on two distinct aspects: the exploration mechanism and the integration methodology of the task policy with the safety policy.

Exploration Mechanism Evaluation: We conduct ablation experiments to assess the effectiveness of our added exploration mechanism. Using combinations such as DDPG-SPDDPG as an example, we conduct ablation experiments on environments PointPush2, CarGoal2, and CarPush2. Three experiments are performed using different random seeds. In each experiment, we

calculate the average rewards and costs for each of the 8000 episodes. Subsequently, we compute the mean and variance across the three sets of random seeds. The results are presented in Table 1.

• Task Performance: Across the three environments, the inclusion of the exploration mechanism notably enhances task performance, as indicated by higher average reward values. This improvement stems from the increased diversity of the data space facilitated by the exploration mechanism, allowing SGAWP to explore high-reward data effectively.

• Safety Performance: The presence of the exploration mechanism not only increases high-reward data but also augments low-cost data within the data space. Consequently, SGAWP with the exploration mechanism outperforms SGAWP without the exploration mechanism across all three environments.

Table 2. Comparative experiments about different combinations

Environment	Algorithm	Average Episode Reward	Average Episode Cost
Safexp-PointGoal2-v0	SAC-SPSAC	**−1.800** ± 0.476	38.264 ± 5.347
	DDPG-SPSAC	−1.890 ± 0.414	37.173 ± 5.426
	SAC-SPDDPG	−1.833 ± 0.742	40.159 ± 5.817
	DDPG-SPDDPG	−2.549 ± 0.639	**31.741** ± 7.499
Safexp-PointButton2-v0	SAC-SPSAC	**−3.010** ± 0.722	21.896 ± 2.770
	DDPG-SPSAC	−4.320 ± 1.032	**13.759** ± 0.353
	SAC-SPDDPG	−4.271 ± 0.710	19.001 ± 1.438
	DDPG-SPDDPG	−3.853 ± 0.760	18.096 ± 8.272
Safexp-PointPush2-v0	SAC-SPSAC	−4.210 ± 2.304	**16.423** ± 2.433
	DDPG-SPSAC	−2.358 ± 0.789	22.069 ± 2.468
	SAC-SPDDPG	−2.636 ± 0.206	20.030 ± 3.413
	DDPG-SPDDPG	**−1.929** ± 0.263	22.428 ± 2.348
Safexp-CarGoal2-v0	SAC-SPSAC	−2.932 ± 1.175	39.412 ± 15.195
	DDPG-SPSAC	−2.980 ± 1.964	23.319 ± 14.969
	SAC-SPDDPG	**−2.534** ± 0.546	53.282 ± 16.361
	DDPG-SPDDPG	−3.930 ± 0.559	**21.608** ± 10.328
Safexp-CarButton2-v0	SAC-SPSAC	−7.487 ± 3.078	11.326 ± 2.381
	DDPG-SPSAC	−6.948 ± 9.105	**8.487** ± 6.238
	SAC-SPDDPG	−6.679 ± 0.343	14.302 ± 3.942
	DDPG-SPDDPG	**−6.056** ± 0.407	14.048 ± 1.527
Safexp-CarPush2-v0	SAC-SPSAC	−4.661 ± 1.480	19.158 ± 4.106
	DDPG-SPSAC	−6.066 ± 2.237	**12.443** ± 3.081
	SAC-SPDDPG	−4.956 ± 0.315	19.678 ± 2.429
	DDPG-SPDDPG	**−2.963** ± 0.930	27.870 ± 10.445
Safexp-DoggoGoal2-v0	SAC-SPSAC	**0.000** ± 0.002	1.678 ± 1.284
	DDPG-SPSAC	−0.002 ± 0.001	0.274 ± 0.049
	SAC-SPDDPG	−0.002 ± 0.002	**0.130** ± 0.046
	DDPG-SPDDPG	−0.003 ± 0.001	0.483 ± 0.164
Safexp-DoggoButton2-v0	SAC-SPSAC	**0.000** ± 0.005	8.927 ± 0.927
	DDPG-SPSAC	−0.003 ± 0.003	**4.325** ± 3.825
	SAC-SPDDPG	−0.006 ± 0.006	12.232 ± 3.009
	DDPG-SPDDPG	−0.001 ± 0.002	5.560 ± 3.479
Safexp-DoggoPush2-v0	SAC-SPSAC	−0.022 ± 0.010	2.371 ± 3.100
	DDPG-SPSAC	−0.025 ± 0.004	0.989 ± 0.162
	SAC-SPDDPG	**−0.016** ± 0.007	**0.276** ± 0.206
	DDPG-SPDDPG	−0.116 ± 0.171	22.049 ± 37.929

Generalization Assessment: In this section, we employ SAC and DDPG, two algorithms grounded in value function optimization, as the task and safety strategies, respectively, resulting in a total of four combinations. Ablation experiments are conducted on different combinations of three agent types, three task types, and a difficulty level of 2. We also perform three experiments with different random seeds, calculating the mean and standard deviation for each round. The outcomes of these experiments are depicted in Table 2.

• Task Performance: Across the six environments, the task performance of the combinations using SAC as the task strategy tends to outperform those with DDPG as the task strategy. This disparity arises from the fact that DDPG employs a deterministic strategy, which can limit its ability to explore unknown environments. DDPG tends to generate relatively certain actions and may struggle to effectively explore potential high-return areas. In contrast, SAC incorporates entropy regularization, which allows for the generation of somewhat random action strategies. Since SGAWP initially constrains the data space to prioritize high task performance, the combinations utilizing DDPG as the task strategy face inherent limitations within the data space constraints. This, in turn, results in lower task performance for these combinations.

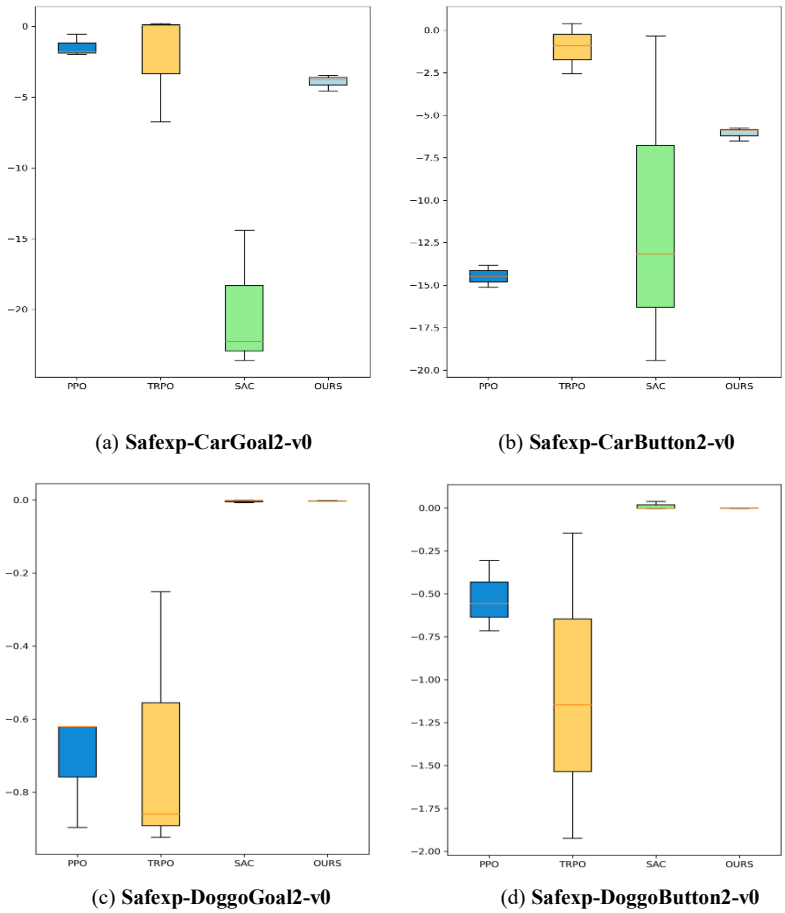
• Safety Performance: Regarding safety performance, all four combination methods demonstrate relatively robust safety performance across all environments, consistently maintaining costs at lower levels with minor fluctuations. Remarkably, in six of these environments, the combination employing DDPG as the task strategy exhibits superior safety performance. This result can be attributed to the fact that while DDPG may not fully explore high-return areas, it effectively reduces risk within the constrained data space. Consequently, this enhances the safety of subsequent learning strategies. Additionally, as a safety policy, DDPG tends to make deterministic decisions, streamlining the process of selecting safe choices. Consequently, in four of the environments, the combination utilizing DDPG as the safety policy demonstrates superior safety performance.

4.4. Data Analysis

In this segment, we employ the boxplot method (Michael Frigge and Iglewicz, 1989) for data analysis. Within Fig. 7 and Fig. 8, the median value is represented by the orange line in the box plot, while the lower and upper boundaries of the box plot denote the 25 and 75 percentages of all collected values, respectively.

In Fig. 7, our algorithms demonstrate compact distributions, indicating algorithmic stability. Notably, the median line of our algorithm often falls within the first quartile range in many environments, suggesting comparable task performance with other algorithms. This aligns with our experimental goal of maintaining consistent task performance levels.

In Fig. 8, our algorithm consistently shows the lowest median line, indicating superior safety performance with notably low safety costs, as intended. Furthermore, the box height for our algorithm remains relatively small, reflecting its stability, largely attributed to DDPG within our framework. Additionally, our algorithm exhibits the lowest upper quartile distribution, indicating minimal security costs even in worst-case scenarios. In contrast, other comparison methods perform less favorably, with higher median lines implying increased average safety costs. Moreover, their larger box heights suggest inferior stability compared to our algorithm.



(a) **Safexp-CarGoal2-v0** (b) **Safexp-CarButton2-v0**

(c) **Safexp-DoggoGoal2-v0** (d) **Safexp-DoggoButton2-v0**

Fig. 7. The boxplot was generated by computing the average of each step over 8,000 iterations using 3 distinct random seeds for each of the 4 algorithms

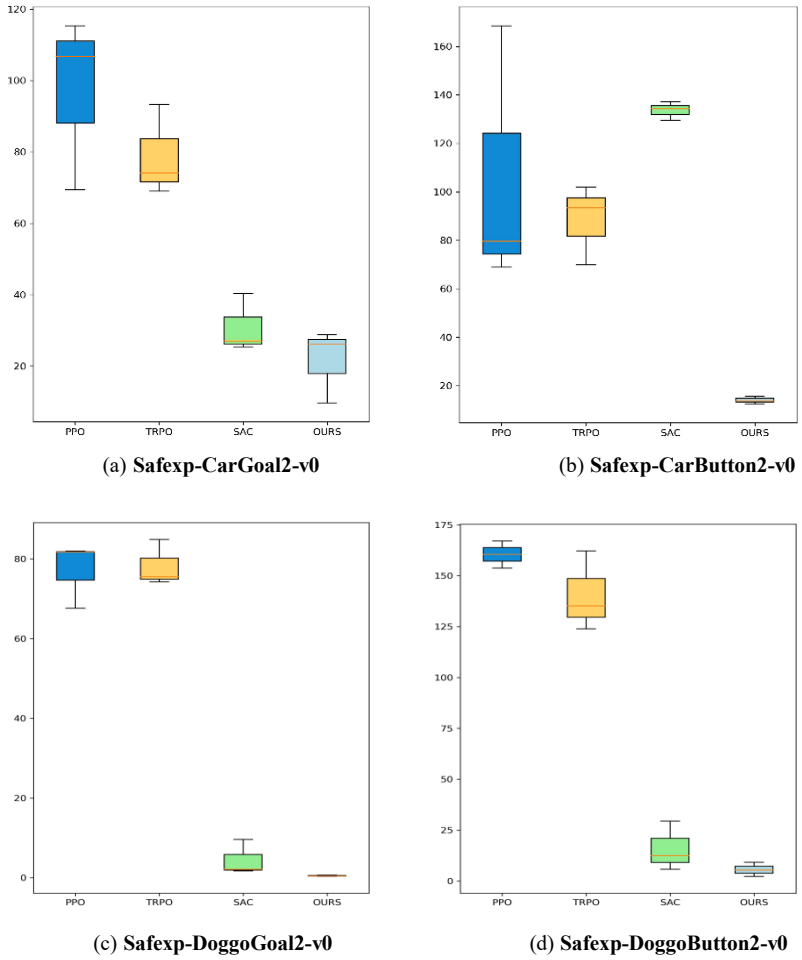
4.5. Comprehensive Analysis

In our comprehensive analysis conducted on the SafetyGym platform, various combinations of agents, task types, and difficulty levels were explored. Round-by-round reward values and costs were utilized to assess task and safety performance, respectively. The integration of safety constraints within SGAWP significantly enhanced safety performance compared to comparison methods, with some environments even achieving zero safety constraint violations. Task performance across most environments demonstrated capabilities on par with or superior to the best comparison method. Moreover, the implementation of uncertainty handling through the three-point estimation method notably improved the stability of task performance across diverse scenarios.

Furthermore, SGAWP conducted ablation experiments on three different random seeds in environments such as CarGoal2 and DoggoButton2 to evaluate the impact of the added exploration mechanism. These experiments illustrated that the exploration mechanism expanded the data space, resulting in enhanced data diversity. Consequently, actions with reduced risk and greater reward potential were selected. Thus, SGAWP exhibited comprehensive leadership in safety performance across various random seeds and environments without the addition of an exploration mechanism, while also demonstrating improved task performance in most enviro nments.

Through different combinations of experiments, although variations were observed, the overall safety performance of SGAWP surpassed baseline methods, while task performance was

generally maintained to a certain extent. This underscores the generalization capabilities of the SGAWP framework.



(a) **Safexp-CarGoal2-v0** (b) **Safexp-CarButton2-v0**

(c) **Safexp-DoggoGoal2-v0** (d) **Safexp-DoggoButton2-v0**

Fig. 8. The boxplot was generated by computing the average costs of each step over 8,000 iterations using 3 distinct random seeds for each of the 4 algorithms

5. Potential Applications

The continuous advancement of autonomous driving technology emphasizes the growing need to balance vehicle task performance with driving safety. It's important to note that our experiments were conducted within the SafetyGym environment, a meticulously designed navigation simulation platform prioritizing safety. This environment allows for the simulation of various scenarios, including obstacle avoidance and handling hazardous conditions, facilitating rapid algorithm development and validation crucial for autonomous driving advancement.

The outcomes of these experiments are highly promising, showcasing significant advancements made by our SGAWP algorithm in enhancing the safety performance of autonomous driving systems. These simulation-based experiments provide a robust foundation for the eventual deployment of autonomous driving technology on real-world roads. Moreover, they contribute significantly to the overarching objective of achieving superior safety performance while maintaining high task performance levels in real-world applications.

Conclusion

In this paper, we introduce a versatile framework for enhancing the safety performance of reinforcement learning algorithms, particularly focusing on off-policy algorithms with value function optimization. Our framework aims to integrate with existing algorithms to significantly improve safety while maintaining a certain level of task performance. Initially, we select an off-policy algorithm to serve as the task policy component, which acts as the behavior policy. We then constrain the action-state space to regions characterized by both high rewards and high costs. Subsequently, we integrate our safety strategy into the off-policy algorithm using the value function framework. This integration involves employing a "conservative-critic" to model conservative-Q values, introducing the risk factor into the learning process. Additionally, we introduce an exploration mechanism to encourage the agent to explore unfamiliar territories, along with an uncertainty module incorporating the three-point estimation method. These mechanisms help deal with uncertainty during training while maintaining task performance and reducing safety costs.

We evaluate our method alongside three baseline methods within the Safety-Gym environment, showcasing its robust generalization capability. Our method not only significantly outperforms other methods in terms of safety performance but also maintains a respectable level of task performance, which is crucial for real-world applications. Ablation experiments further confirm the positive impact of our exploration mechanism. The versatility and adaptability of our framework are evident through consistent success across different combination experiments, indicating its potential for various applications.

It's noteworthy that our method imposes minimal constraints on the initial policy, prompting exploration into leveraging relevant prior knowledge tailored for navigation tasks to further enhance performance. Additionally, there's room for exploration in selecting more appropriate safety constraints based on specific environments. The synergy between tasks and safety policies also presents significant potential for expansion and refinement in future research.

REFERENCES:

[1] G. Li, Y. Yang, S. Li, X. Qu, N. Lyu, S. E. Li, Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness, Transportation Research Part C: Emerging Technologies 134 (2022) 103452. DOI: https://doi.org/10.1016/j.trc.2021.103452.

[2] C. Shiranthika, K. -W. Chen, C. -Y. Wang, C. -Y. Yang, B. H. Sudantha and W. -F. Li. Supervised Optimal Chemotherapy Regimen Based on Offline Reinforcement Learning. IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 9, p. 4763–4772, Sept. 2022. DOI: 10.1109/JBHI.2022.3183854.

[3] M.Abdollahzadeh, Yahya Dorostkar Navaei, Anomaly detection in heart disease using a density based unsupervised approach. DOI: https://doi.org/10.1155/2022/6913043.

[4] M. Selim, A. Alanwar, S. Kousik, G. Gao, M. Pavone and K. H. Johansson. Safe Reinforcement Learning Using Black-Box Reachability Analysis. IEEE Robotics and Automation Letters, vol. 7, no. 4, p. 10665–10672, Oct. 2022. DOI: 10.1109/LRA.2022.3192205.

[5] Y. Chow, M. Ghavamzadeh, L. Janson, M. Pavone, Risk-constrained reinforcement learning with percentile risk criteria, J. Mach. Learn. Res. 18 (2017) 60706120. URL: https://archive.org/details/arxiv-1512.01629 (accessed: 01.04.2024).

[6] C. Gehring, D. Precup, Smart exploration in reinforcement learning using absolute temporal difference errors, in: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2013, p. 10371044. URL: https://www.researchgate.net/publication/262164966_Smart_exploration_in_reinforcement_learning_using_absolute_temporal_difference_errors (accessed: 01.04.2024).

[7] S. Fujimoto, D. Meger, D. Precup, Off-policy deep reinforcement learning without exploration, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, PMLR. 2019, p. 2052–2062. URL: https://proceedings.mlr.press/v97/fujimoto19a.html (accessed: 01.04.2024).

[8] J. García, Fern, o Fernández, A comprehensive survey on safe reinforcement learning, Journal of Machine Learning Research 16 (2015), p. 1437–1480. URL: https://www.semanticscholar.org/paper/A-comprehensive-

survey-on-safe-reinforcement-García-Fernández/c0f2c4104ef6e36bb67022001179887e6600d24d (accessed: 01.04.2024).

[9] A. Kumar, A. Zhou, G. Tucker, S. Levine, Conservative q-learning for offline reinforcement learning, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc. 2020, p. 1179–1191. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf (accessed: 01.04.2024).

[10] Xu, H., Zhan, X., & Zhu, X. (2022). Constraints Penalized Q-learning for Safe Offline Reinforcement Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 36(8), p. 8753–8760. DOI: https://doi.org/10.1609/aaai.v36i8.20855.

[11] G. Thomas, Y. Luo, T. Ma, Safe reinforcement learning by imagining the near future, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc. 2021, p. 13859–13869. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/73b277c11266681122132d024f53a75b-Paper.pdf. (accessed: 01.04.2024).

[12] Y. J. Ma, A. Shen, O. Bastani, J. Dinesh, Conservative and adaptive penalty for model-based safe reinforcement learning, in: AAAI. 2022, p. 5404–5412. DOI: 10.1609/aaai.v36i5.20478.

[13] M. L. Littman, Value-function reinforcement learning in markov games, Cognitive Systems Research 2 (2001) 55–66. DOI: https://doi.org/10.1016/S1389-0417(01)00015-8.

[14] Luengo, D., Martino, L., Bugallo, M. et al. A survey of Monte Carlo methods for parameter estimation. EURASIP J. Adv. Signal Process. 25 (2020). DOI: https://doi.org/10.1186/s13634-020-00675-6.

[15] L. Wang, Z. Tong, B. Ji, G. Wu, Tdn: Temporal difference networks for efficient action recognition, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021, p. 1895–1904. DOI: 10.1109/CVPR46437.2021.00193.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning. DOI: https://doi.org/10.48550/arXiv.1312.5602.

[17] van Hasselt, H., Guez, A., & Silver, D. (2016). Deep Reinforcement Learning with Double Q-Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 30(1). DOI: https://doi.org/10.1609/aaai.v30i1.10295.

[18] Clifton, Jesse and Laber, Eric B., Q-Learning: Theory and Applications (March 2020). Annual Review of Statistics and Its Application. V. 7, Issue 1, p. 279–301, 2020. DOI: http://dx.doi.org/10.1146/annurev-statistics-031219-041220.

[19] J. Schulman, S. Levine, P. Moritz, M. Jordan, P. Abbeel, Trust region policy optimization, in: Proceedings of the 32nd International Conference on International Conference on Machine Learning – Vol. 37, ICML'15, JMLR.org, 2015, p. 1889–1897. URL: https://proceedings.mlr.press/v37/schulman15.pdf (accessed: 01.04.2024).

[20] N. Vieillard, T. Kozuno, B. Scherrer, O. Pietquin, R. Munos, M. Geist, Leverage the average: an analysis of kl regularization in reinforcement learning, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc. 2020, p. 12163–12174. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/8e2c381d4dd04f1c55093f22c59c3a08-Paper.pdf (accessed: 01.04.2024).

[21] V. Konda, J. Tsitsiklis, Actor-critic algorithms, in: S. Solla, T. Leen, K. Müller (Eds.), Advances in Neural Information Processing Systems, volume 12, MIT Press, 1999. URL: https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf (accessed: 01.04.2024).

[22] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, vol. 80 of Proceedings of Machine Learning Research, PMLR. 2018, p. 1861–1870. URL: https://proceedings.mlr.press/v80/haarnoja18b.html (accessed: 01.04.2024).

[23] E. Altman, Constrained Markov Decision Processes, 1st ed., Routledge, 1999. DOI: 10.1201/9781315140223.

[24] J. Taylor, Project scheduling and cost control: planning, monitoring and controlling the baseline, J. Ross Publishing, 2008. – 280 p.

[25] K. R. MacCrimmon, C. A. Ryavec, An analytical study of the pert assumptions, Operations Research 12 (1964) 16–37. DOI: https://doi.org/10.1287/opre.12.1.16.

[26] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, M. Riedmiller, Deterministic policy gradient algorithms, in: E. P. Xing, T. Jebara (Eds.), Proceedings of the 31st International Conference on Machine Learning, volume 32 of Proceedings of Machine Learning Research, PMLR, Bejing, China. 2014, p. 387–395. URL: https://proceedings.mlr.press/v32/silver14.html (accessed: 01.04.2024).

[27] Lockwood, O., & Si, M. (2022). A Review of Uncertainty for Deep Reinforcement Learning. Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 18(1), p. 155–162. DOI: https://doi.org/10.1609/aiide.v18i1.21959.
[28] J. Hao et al. Exploration in Deep Reinforcement Learning: From Single-Agent to Multiagent Domain. IEEE Transactions on Neural Networks and Learning Systems. DOI: 10.1109/TNNLS.2023.3236361.
[29] D. Pathak, P. Agrawal, A. A. Efros, T. Darrell, Curiosity-driven exploration by self-supervised prediction. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2017, p. 488–489. DOI: 10.1109/CVPRW.2017.70.
[30] A. P. Badia, P. Sprechmann, A. Vitvitskyi, D. Guo, B. Piot, S. Kapturowski, O. Tieleman, M. Arjovsky, A. Pritzel, A. Bolt, C. Blundell, Never give up: Learning directed exploration strategies, in: International Conference on Learning Representations, 2020. URL: https://arxiv.org/pdf/2002.06038 (accessed: 01.04.2024).
[31] A. Ray, J. Achiam, D. Amodei, Benchmarking safe exploration in deep reinforcement learning, Preprint, OpenAI, San Francisco, CA (2019). URL: https://openai.com/index/benchmarking-safe-exploration-in-deep-reinforcement-learning (accessed: 01.04.2024).
[32] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms. DOI: https://doi.org/10.48550/arXiv.1707.06347.
[33] Frigge, M., Hoaglin, D. C. and Iglewicz, B. (1989). Some Implementations of the Boxplot. The American Statistician, 43(1), p. 50–54. DOI: 10.1080/00031305.1989.10475612.

Поступила в редакцию – 10 апреля 2024 г. Окончательный вариант – 24 мая 2024 г.
Received – April 10, 2024. The final version – May 24, 2024.