Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.086535

ARTICLE

An Improved Safe Soft Actor-Critic Path Planning Algorithm for Autonomous Vehicles Based on a Dual-Stream Q-Network and Dynamic Analytic Hierarchy Process

Shengxuan Dong and Xiongwei Li*

Shijiazhuang Campus, Army Engineering University of PLA, Shijiazhuang, China

*Corresponding Author: Xiongwei Li. Email: lxwys@aeu.edu.cn

Received: 01 June 2026; Accepted: 21 July 2026; Published: 15 September 2026

ABSTRACT: To address the conflict between navigation performance and safety constraints in safe reinforcement learning, this paper proposes Dual Stream-Analytic Hierarchy Process-Safe Soft Actor (DS-AHP-SAC), a safe soft actor-critic algorithm based on a dual-stream Q-network and dynamic Analytic Hierarchy Process (AHP) stratified experience replay. The algorithm achieves a balance between reward maximization and constraint satisfaction through three synergistic designs: (1) decoupling the Q-network into independent navigation and safety value streams to eliminate gradient interference at the Critic level and mitigate gradient competition at the Actor level; (2) constructing a three-criterion dynamic sampling strategy based on AHP, incorporating safety urgency, information value, and scarcity to enable phase-adaptive experience replay; (3) designing a curriculum-scheduling scheme that linearly increases the safety constraint weight during training, preventing policy degradation caused by premature imposition of high safety penalties. Experimental results in a two-dimensional continuous navigation environment demonstrate that DS-AHP-SAC reduces the violation rate by 20.0% compared to unconstrained SAC without sacrificing navigation success rate, while avoiding the policy collapse observed in SAC-Lagrangian. Ablation studies validate the necessity of the dual-stream Q-network and the convergence acceleration effect of dynamic AHP.

KEYWORDS: Safe reinforcement learning; dual-stream Q-network; analytic hierarchy process; experience replay; constrained Markov decision process

1 Introduction

The demand for autonomous navigation of unmanned vehicles in scenarios such as indoor warehouse logistics and outdoor material delivery continues to grow. Deep Reinforcement Learning (DRL) [1] has become an important approach for solving path planning and obstacle avoidance problems for unmanned vehicles in complex environments due to its end-to-end decision-making capability [2,3]. However, unmanned vehicles face strict collision constraints during operation. Achieving efficient navigation while ensuring safety remains a core challenge in the field of Safe Reinforcement Learning (Safe RL) [4].

Deep reinforcement learning path planning methods have evolved from discrete action spaces (DQN) to continuous action spaces (DDPG, TD3, SAC). DQN combines Q-learning with deep neural networks, laying the foundation of deep reinforcement learning, but is limited to discrete actions. DDPG and TD3 extend deterministic policy gradients to continuous action spaces, while SAC introduces entropy regularization to achieve a more thorough exploration-exploitation balance. Building upon these, safety constraints have been gradually introduced, giving rise to different technical routes such as trust-region methods based

on Constrained Policy Optimization (CPO), SAC-Lagrangian based on Lagrangian multipliers [5], and Recovery RL based on runtime recovery strategies. These methods each have their own focus but exhibit deficiencies in gradient conflicts at the Critic level, dynamic adaptation of experience utilization efficiency, and constraint scheduling stability. This paper addresses the aforementioned issues.

FOCOPS [6] and CPO [7] are based on trust-region on-policy optimization, directly imposing safety constraints during policy updates, but the on-policy mechanism suffers from low sample efficiency. CVPO [8] reformulates constraints through variational inference, belonging to a different constraint-handling route. Recovery RL maintains an independent recovery policy to ensure runtime safety, where the additional policy switching mechanism increases engineering complexity. The work in this paper, under the off-policy SAC framework, differs from on-policy methods and recovery-strategy methods in design philosophy; direct cross-category numerical comparison requires additional evaluation protocols, and such comparison is left for future work.

Existing safe RL algorithms are mostly based on the Constrained Markov Decision Process (CMDP) framework [4] and handle safety constraints through several distinct technical routes: trust-region on-policy methods such as CPO (Constrained Policy Optimization) [7] directly impose cost constraints during policy updates; runtime-recovery methods such as Recovery RL maintain an independent safety-recovery policy; and Control Barrier Function (CBF)-based methods [9] act as a barrier-filter safety shield. Among these, the Lagrangian family—most prominently SAC-Lagrangian—converts the safety cost into a penalty term on the reward via a Lagrangian multiplier and folds it into a single-stream value function. These algorithms face three limitations, the first of which is specific to the Lagrangian single-stream formulation: (1) in Lagrangian-Critic methods such as SAC-Lagrangian, when the gradients of task rewards and safety costs backpropagate through the same Critic parameter space, they counteract each other, leading to oscillating policy convergence or even degeneration into conservative policies that simply stop moving; (2) Prioritized Experience Replay (PER) [10] updates priorities only for the transitions sampled in each batch, yet its priority scores still reflect instantaneous TD errors and therefore track the rapidly changing cost landscape during safe RL training, making it difficult to provide phase-adaptive sampling; (3) in adaptive multiplier algorithms, the multiplier can grow uncontrollably leading to policy collapse, while fixed hyperparameters cannot accommodate the differentiated needs of different training stages.

To address the above issues, this paper proposes an improved Safe SAC algorithm based on a dual-stream Q-network and dynamic AHP-based experience replay. The main contributions are as follows: (1) a dual-stream Q-network structure is proposed, which decouples the Q-network into independent navigation value and safety value streams, eliminating gradient conflicts at the Critic structural level and mitigating gradient competition at the Actor level; (2) an AHP-based [11] three-criteria evaluation model for safety urgency, information value, and scarcity is established, driven by training progress to dynamically interpolate criteria weights and achieve stage-adaptive adjustment of experience replay; (3) a curriculum-based scheduling strategy is designed to linearly increase the safety constraint weight, achieving a smooth transition from navigation exploration to safe exploitation.

2 Theoretical Foundations

2.1 Constrained Markov Decision Process

Safe reinforcement learning typically models the problem as a Constrained Markov Decision Process (CMDP), described by a septuple $(\mathcal{S}, \mathcal{A}, P, R, C, \rho_0, \gamma)$ [4], where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the state transition probability, R is the task reward function, C is the cost function, ρ_0 is the initial state distribution, and γ is the discount factor. The optimization objective of the policy is to maximize the expected

cumulative discounted reward while satisfying the constraint that the expected cumulative discounted cost does not exceed a threshold:

$$\max_{\pi} \quad \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] \tag{1}$$

$$\text{s.t.} \quad \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right]\le d \tag{2}$$

Unlike standard MDPs that only maximize rewards, CMDP explicitly introduces cost constraints, separating safety from its implicit encoding in the reward function and establishing safety constraints as a hard boundary for policy optimization. Algorithms based on Lagrangian relaxation introduce a dual variable λ to convert the constrained problem into an unconstrained problem, and adaptively update λ through gradient ascent to approximate constraint satisfaction. However, in practice, the adaptive multiplier can grow uncontrollably, leading to policy collapse.

2.2 Soft Actor-Critic Algorithm

The Soft Actor-Critic (SAC) [12] is an off-policy actor-critic algorithm based on the maximum entropy framework, which has been applied in robot navigation [13]. It maximizes the cumulative reward while simultaneously maximizing the policy entropy to enhance exploration and avoid premature convergence to local optima:

$$\max_{\pi} \quad \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t\big(r(s_t,a_t)+\alpha\mathcal{H}(\pi(\cdot|s_t))\big)\right] \tag{3}$$

where α is the temperature coefficient controlling the strength of entropy regularization, and $\mathcal{H}(\pi(\cdot|s_t))$ is the policy entropy.

SAC employs a twin Q-network structure Q_1, Q_2, combined with an improved experience replay mechanism [14] to store and reuse interaction data. It uses Clipped Double-Q Learning [15] to mitigate overestimation bias, employing $\min(Q_1, Q_2)$ to suppress optimistic bias, and updates by minimizing the soft Bellman residual. Target network parameters are slowly tracked through soft updates. The policy network is updated by minimizing the KL divergence loss, using the reparameterization trick to make gradients estimable.

2.3 Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP) is a multi-criteria decision-making algorithm that combines qualitative and quantitative analysis [11]. Its core idea is to decompose a complex decision problem into a goal layer, a criteria layer, and an alternative layer, quantifying the relative importance of each criterion through pairwise comparison matrices. For K evaluation criteria, a $K \times K$ pairwise comparison matrix $\mathbf{A}$ is constructed, where element a_{ij} represents the importance scale of criterion i relative to criterion j, using a 1–9 scale method, satisfying reciprocity $a_{ij} = 1/a_{ji}$ and $a_{ii} = 1$. By solving the normalized right eigenvector corresponding to the largest eigenvalue of the matrix, the weight vector $\mathbf{w}$ for each criterion is obtained. The Consistency Ratio (CR) is introduced to verify the logical consistency of the judgment matrix: CR = CI/RI, where CI is the Consistency Index and RI is the average consistency index of random matrices of the same order. When CR < 0.1, the judgment matrix is considered to have satisfactory consistency. Traditional AHP is mostly used in offline static decision-making scenarios where weights remain fixed once determined.

3 The Proposed Algorithm

The DS-AHP-SAC algorithm consists of three synergistic modules: a dual-stream Q-network, a dynamic AHP tiered experience replay mechanism, and a curriculum-based scheduling strategy. The overall architecture is shown in Fig. 1. All three modules are driven by a unified training progress signal, ensuring synchronization between the timing of safety constraint intervention and the focus of experience sampling. The dual-stream Critic computes the TD targets for the navigation and safety streams separately during each gradient update; the dynamic AHP module interpolates criteria weights based on the current training progress, assigning differentiated sampling probabilities to four types of experiences in the tiered buffers; and the three-stage scheduler outputs the current safety weight based on global training progress for the Actor's policy update.

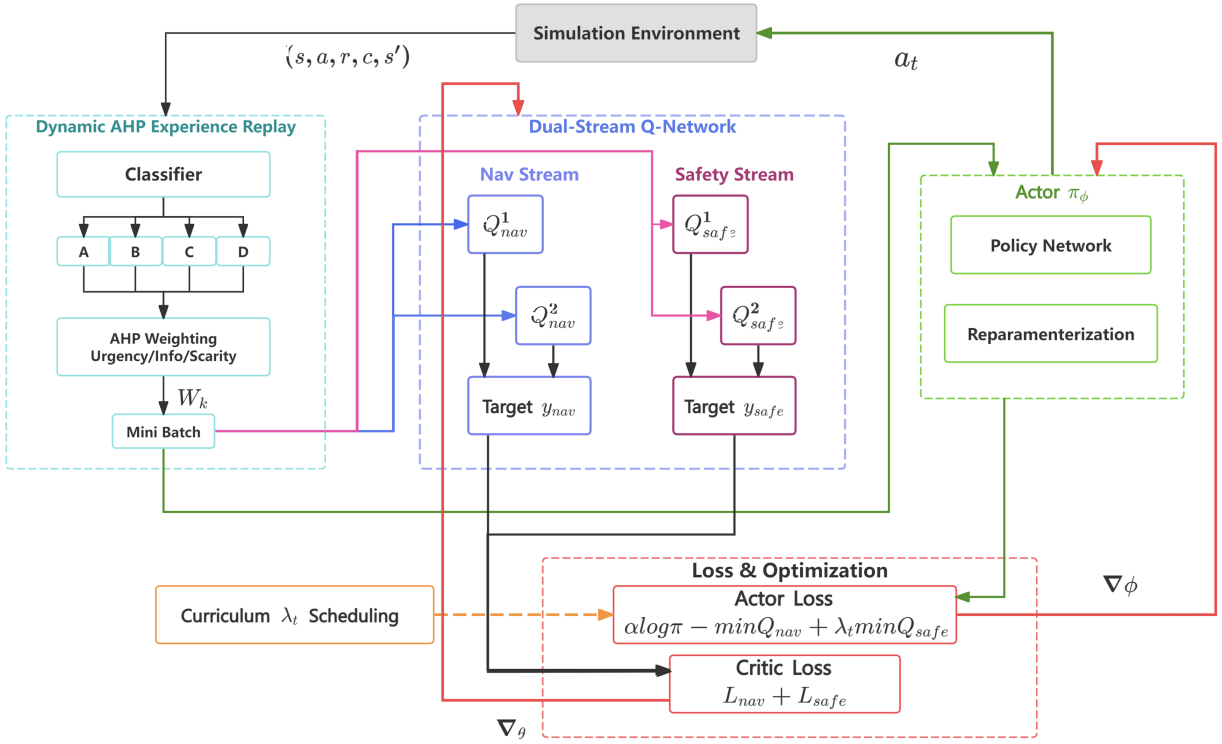


Figure 1: Overall architecture of the DS-AHP-SAC algorithm. Solid arrows indicate forward data flow; red dashed arrows indicate gradient feedback for parameter updates. The three modules are driven by a unified training progress signal p: (1) Dynamic AHP Experience Replay classifies and prioritizes experiences; (2) Dual-Stream Q-Network decouples navigation and safety value estimation; (3) Actor fuses dual-stream Q-values weighted by curriculum-scheduled λ_t.

3.1 Dual-Stream Q-Network Design

In the Lagrangian formulation of safe SAC (i.e., SAC-Lagrangian), the safety cost is converted into a penalty term on the reward by a Lagrangian multiplier and folded into a single-stream Q-network, so that the modified reward drives the Q-network parameters with two opposing gradient signals simultaneously. (Standard SAC, as described in Section 2.2, is an unconstrained maximum-entropy algorithm with no safety-cost component and therefore does not suffer from this conflict; it is included in the experiments of Section 4.2 as an unconstrained upper-bound reference for navigation performance.) During Critic backpropagation in SAC-Lagrangian, the gradient from the reward TD error pushes Q-values upward, while

the gradient from the cost TD error pushes Q-values downward. These two directly counteract each other in the same parameter space, causing value estimation oscillation or even convergence failure.

To eliminate the internal gradient conflict in the Critic, we propose a dual-stream Q-network structure that decouples value estimation into two completely independent sub-networks, containing a total of four Q-networks: $Q^{1,2}_{\text{nav}}$ and $Q^{1,2}_{\text{safe}}$, each with independent parameters.

Navigation stream: Estimates the task completion value of state-action pairs using the standard SAC soft Bellman update, with the target value:

$$y_{\text{nav}} = r + \gamma \min_{i=1,2}\left(Q^{i}_{\text{nav,targ}}(s', a') - \alpha \log \pi_{\phi}(a'|s') \right) \tag{4}$$

Safety stream: Estimates the cumulative safety cost of state-action pairs using a cost Bellman update without the entropy term. Consistent with the navigation stream, the safety stream adopts the clipped double-Q minimum estimate ($\min_{i=1,2}$) as the Bellman target, primarily to suppress overestimation bias of the safety-cost Q-values. In TD learning, Q-value overestimation accumulates progressively through Bellman iterations, causing training instability; for safety costs, overestimation additionally makes the policy overly conservative regarding states near obstacles, analogous to the mechanism in SAC-Lagrangian where the multiplier tops out and the policy tends toward stationarity. The minimum estimate entails a certain underestimation risk, which is compensated by the curriculum-based λ scheduling: in the later training phase λ_t approaches $\lambda_{\max}$, and the safety penalty term remains sufficient to effectively constrain policy behavior. Experiments show that the final VR of DS-AHP-SAC is significantly lower than that of unconstrained SAC, indicating that this compensation mechanism is effective. The target value is:

$$y_{\text{safe}} = c + \gamma \min_{i=1,2} Q^{i}_{\text{safe,targ}}(s', a') \tag{5}$$

Both streams share the same input (the concatenation vector of state and action), but their network parameters are completely independent. The total Critic loss is the sum of mean squared errors of the four independent TD errors. Since the parameters are completely independent, the gradients of the navigation and safety streams do not counteract each other in the same parameter space, thereby eliminating gradient conflicts at the Critic level.

The Actor performs policy updates by fusing the Q-values from both navigation and safety streams, using the Clipped Double-Q min operation to suppress overestimation bias:

$$\mathcal{L}_{\text{Actor}} = \mathbb{E}_{a\sim\pi}\left[\alpha \log \pi(a|s) - \min_{i=1,2} Q^{i}_{\text{nav}}(s, a) + \lambda_t \min_{i=1,2} Q^{i}_{\text{safe}}(s, a) \right] \tag{6}$$

where the navigation stream enters the Actor loss with a negative sign (maximizing navigation value), and the safety stream enters with a positive sign weighted by λ_t (minimizing safety cost), with the weight controlled by curriculum-based scheduling.

It should be noted that although the dual-stream design eliminates gradient conflicts within the Critic, the Actor still receives gradient signals from both Q_{nav} and Q_{safe} simultaneously. When the two types of gradients have opposite directions in the Actor parameter space, the Actor still faces a competitive gradient problem. The curriculum-based λ scheduling mitigates this competition by gradually increasing the weight of the safety gradient rather than completely eliminating it. We explicitly distinguish this as: the dual-stream design eliminates gradient conflicts at the Critic level and mitigates gradient competition at the Actor level.

3.2 Dynamic AHP Tiered Experience Replay

Standard experience replay uses uniform random sampling, which cannot distinguish the safety urgency and learning value of experiences. We propose an AHP-based tiered experience replay mechanism that achieves multi-criteria dynamic sampling through a combination of offline construction and online interpolation.

3.2.1 Experience Tiering Strategy

Based on safety cost and task reward, interaction experiences are divided into four categories and stored in independent FIFO queues. The classification criteria are shown in Table 1.

Table 1: Classification criteria and semantics for experience tiering strategies.

Category	Condition	Semantic Meaning
A	$c = 0$ and $r \geq \bar{r}_{30}$	Safe and efficient
B	$c = 0$ and $r < \bar{r}_{30}$	Safe but inefficient
C	$c > 0$ and $r \geq \bar{r}_{30}$	Approaching danger zone
D	$c > 0$ and $r < \bar{r}_{30}$	Collision occurred

Here $\bar{r}_{30}$ is the 30th percentile of rewards from the most recent 2000 experiences, dynamically updated through a sliding window.

3.2.2 AHP Multi-Criteria Weight Computation

We construct a three-criteria AHP model to assign sampling weights for four categories of experiences:

Criterion u_1 (Safety Urgency): An offline-constructed 4×4 pairwise comparison matrix reflecting the safety priority ordering $D > C \gg A > B$. The matrix is:

$$\mathbf{A}_{u_1} = \begin{pmatrix} 1 & 3 & 1/5 & 1/7 \\ 1/3 & 1 & 1/7 & 1/9 \\ 5 & 7 & 1 & 1/3 \\ 7 & 9 & 3 & 1 \end{pmatrix} \tag{7}$$

The weight vector is computed using the geometric mean method, and consistency verification confirms acceptable consistency. The safety urgency scores for each category are denoted as:

$$\mathbf{w}^{(u_1)} = (w_A^{(u_1)}, w_B^{(u_1)}, w_C^{(u_1)}, w_D^{(u_1)}) \tag{8}$$

where the sum of all category scores equals 1.

Criterion u_2 (Information Value): Computed online, proportional to the exponential moving average TD error of each category, reflecting the contribution of experiences to network updates:

$$w_k^{(u_2)} = \frac{\overline{\delta}_k}{\sum_{j \in \{A,B,C,D\}} \overline{\delta}_j} \tag{9}$$

Criterion u_3 (Scarcity): Computed online, inversely proportional to the current storage volume of each category's buffer:

$$w_k^{(u_3)} = \frac{1/n_k}{\sum_{j\in\{A,B,C,D\}} 1/n_j} \tag{10}$$

3.2.3 Dynamic Interpolation of Criteria Weights

The criteria-level weights are linearly interpolated with training progress p:

$$\begin{aligned}\beta_1(p) &= \beta_1^{\text{init}} + p \cdot (\beta_1^{\text{final}} - \beta_1^{\text{init}}) \\ \beta_2(p) &= 1 - \beta_1(p) - \beta_3(p) \\ \beta_3(p) &= \beta_3^{\text{init}} + p \cdot (\beta_3^{\text{final}} - \beta_3^{\text{init}})\end{aligned} \tag{11}$$

where $\beta_1^{\text{init}} > \beta_1^{\text{final}}$ (emphasizing safety urgency early), and $\beta_3^{\text{final}} > \beta_3^{\text{init}}$ (emphasizing information value and scarcity later). The final composite sampling weight for each category is:

$$W_k = \beta_1 \cdot w_k^{(u_1)} + \beta_2 \cdot w_k^{(u_2)} + \beta_3 \cdot w_k^{(u_3)}, \quad k \in \{A, B, C, D\} \tag{12}$$

This dynamic mechanism enables AHP to adapt to the differentiated needs of different stages of safe reinforcement learning. In the early training phase, data is predominantly Category B (safe but inefficient), and prioritized sampling of D/C categories accelerates safety awareness establishment. In the later phase, D-type data accumulates, and the focus shifts to the u_2 criterion, prioritizing experiences with large TD errors to improve sample efficiency. Whereas PER updates only the sampled transitions' priorities per batch but ties those priorities to instantaneous TD errors (and thus to the rapidly shifting cost distribution in safe RL), AHP weight updates require only matrix multiplication ($O(K^2)$), significantly reducing computational overhead.

Regarding the robustness of the AHP comparison matrix: The priority structure ($D > C \gg A > B$) of the matrix used in this paper derives from prior knowledge of cost grading in safe RL, constituting a qualitative constraint rather than the result of precise tuning. As long as the priority ordering does not reverse, reasonable perturbations of the matrix elements do not change the ordering of the weight vector, exerting limited influence on sampling behavior; the consistency ratio of this paper is CR = 0.025, far below the acceptable threshold of 0.1, indicating good matrix consistency. Moreover, the output of AHP is the relative sampling weight ratio of each category, whose absolute value is adaptively adjusted by the three-criterion dynamic interpolation with training progress, requiring no separately configured fixed replay weighting coefficient. This adaptive characteristic endows the algorithm with inherent robustness to small perturbations of matrix elements and also constitutes an improvement over manually fixed weighting schemes.

3.3 Curriculum-Based Scheduling Strategy

The safety penalty weight λ_t in the dual-stream Q-value fusion directly affects the balance between navigation performance and safety. A fixed λ presents a dilemma: too small a value results in insufficient safety constraints, while too large a value causes the policy to degenerate conservatively. Inspired by curriculum learning [16], we design a curriculum-based scheduling strategy that enables λ_t to gradually transition from the navigation exploration phase to the safe exploitation phase:

$$\lambda_t = \begin{cases} \lambda_{\min}, & t < T_1 \\ \lambda_{\min} + (\lambda_{\max} - \lambda_{\min}) \cdot \dfrac{t - T_1}{T_2 - T_1}, & T_1 \le t < T_2 \\ \lambda_{\max}, & t \ge T_2 \end{cases} \tag{13}$$

where $\lambda_{\min} = 0.1$, $\lambda_{\max} = 0.7$, T_1 and T_2 are the transition boundaries, and T is the total number of training steps. This design is based on three considerations: (1) In the early training phase ($t < T_1$), λ is kept low, and the Actor is primarily driven by navigation Q-values, allowing the agent to freely explore and learn strategies for reaching the goal. This phase approximates unconstrained SAC training. (2) In the mid-training phase ($T_1 \le t < T_2$), λ linearly increases, progressively strengthening safety constraints on top of the already acquired navigation capability, avoiding the impact of a sudden introduction of safety gradients on the learned navigation policy. (3) In the late training phase ($t \ge T_2$), λ is fixed at $\lambda_{\max}$ to prevent performance oscillation caused by continuously increasing safety penalties.

In terms of theoretical properties, the convergence of the dual-stream Q-network can inherit the existing analytical framework of SAC. The two streams each satisfy the contraction mapping condition of their respective Bellman operators, and the completely independent parameters guarantee that the fixed-point iteration processes of the two streams do not interfere with each other. The navigation stream uses the SAC soft Bellman operator (Eq. (4)) and its convergence is therefore equivalent to that of single-stream SAC; the safety stream instead uses the standard cost Bellman operator without the entropy term (Eq. (5)), so it converges to the standard cost Q-function rather than to SAC's soft Q-function. Thus the contraction property holds for both streams, but their fixed points and operators differ—only the navigation stream is strictly equivalent to SAC. Non-uniform tiered sampling introduces a sampling bias into the learned Q-values, which in theory requires importance-weight correction. This paper does not apply explicit correction because the AHP weights vary dynamically with training progress, and introducing importance weights would add extra variance. No significant bias accumulation was observed in the experiments, which warrants further analysis. The design motivation of the curriculum-based λ scheduling is analogous to curriculum learning and annealing strategies. The low λ in the early training phase maintains sufficient navigation exploration, while the high λ in the later phase strengthens safety constraints. This differs from the policy degradation after multiplier saturation in SAC-Lagrangian and indicates that progressively increasing the safety weight is more stable than a fixed high penalty.

The complete training procedure of DS-AHP-SAC is summarized in Algorithm 1.

Algorithm 1: DS-AHP-SAC training procedure

Require: tiered replay buffers $\mathcal{B}_{A,B,C,D}$; Actor π_ϕ; dual-stream Critics Q^θ_{nav}, Q^θ_{safe} and targets; AHP matrix $\mathbf{A}_{u_1}$; schedule $[\lambda_{\min}, \lambda_{\max}]$

Ensure: trained safe navigation policy π_ϕ

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Execute $a \sim \pi_\phi(\cdot|s)$ in the environment, observe (r, c, s')
3: Classify (s, a, r, c, s') by (c, r) into tier $k \in \{A, B, C, D\}$ and store it in $\mathcal{B}_k$
4: **if** t mod $\Delta_{\text{AHP}} = 0$ **then**
5: Interpolate criteria weights $\beta_1(t), \beta_2(t), \beta_3(t)$ by progress $p = t/T$ ▷ Eq. (11)
6: Compute composite weights $W_k = \beta_1 w_k^{(u_1)} + \beta_2 w_k^{(u_2)} + \beta_3 w_k^{(u_3)}$ ▷ Eq. (12)
7: **end if**
8: Sample mini-batch from $\{\mathcal{B}_k\}$ with probability proportional to W_k

(Continued)

Algorithm 1 (continued)

9: Update dual-stream Critic by $\mathcal{L}_{\text{Critic}} = \sum_{i=1,2} \text{MSE}(y^i_{\text{nav}}) + \sum_{i=1,2} \text{MSE}(y^i_{\text{safe}})$
10: Compute curriculum weight λ_t by the schedule ▷ Eq. (13)
11: Update Actor by $\nabla_\phi \left[\alpha \log \pi_\phi(a|s) - \min_i Q^i_{\text{nav}}(s,a) + \lambda_t \min_i Q^i_{\text{safe}}(s,a) \right]$
12: Soft-update target networks: $\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}$, with $\tau = 0.005$
13: **end for**

4 Experimental Design

4.1 Simulation Environment and Evaluation Metrics

4.1.1 Simulation Environment

We constructed a two-dimensional continuous-space navigation simulation environment based purely on NumPy to validate the path planning performance of the proposed algorithm under safety constraints. The environment simulates a 15×15 m rectangular workspace. An unmanned vehicle (radius 0.3 m) is initialized at a random starting pose in the lower-left region, with the task goal of navigating to a fixed target point $(12.25, 12.25)$ with a capture radius of 1.5 m. Ten circular obstacles are randomly placed in the workspace, with radii uniformly sampled from $[0.4, 0.8]$ m, ensuring a minimum distance of 2.5 m between obstacle centers and the starting/target points during generation.

The unmanned vehicle adopts a differential-drive model with linear velocity range $[0, 1.5]$ m/s, angular velocity range $[-1.5, 1.5]$ rad/s, time step 0.1 s, and robot radius 0.3 m. The sensing module uses 16-ray uniformly distributed lidar with a maximum range of 8.0 m, with readings normalized to $[0, 1]$ and no noise modeling. Obstacles are randomly regenerated in position and radius at each episode reset; collisions do not terminate the episode, and upon collision the agent is bounced back to a safe region to maintain a continuous training signal. Hyperparameter settings rationale: the batch size of 256 and soft update coefficient $\tau = 0.005$ follow the original SAC paper; $\lambda_{\min} = 0.1$ and $\lambda_{\max} = 0.7$ were determined by validation-set scanning over candidate values $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, with the selection criterion being SR no lower than that of unconstrained SAC by more than 3% and the lowest VR; the priority structure of the AHP comparison matrix ($D > C \gg A > B$) comes from prior knowledge of cost grading in safe RL, with the specific values determined subject to the constraint CR < 0.1 (CR = 0.025 in this paper). Training was conducted on an Intel i7-12700 processor; all five seeds with 500 episodes each required approximately 8 to 12 h. Peak memory consumption is dominated by the replay buffers: the four tiered buffers (2.5×10^5 transitions each) together with the main buffer (5×10^5 transitions) store about 1.5×10^6 transitions, occupying roughly 300–400 MB of RAM, while the neural networks (an Actor with two $[128, 128]$ layers and four Q-networks each with two $[64, 64]$ layers, on the order of 4×10^4 parameters in total) account for a negligible fraction. During inference, the Actor is a single forward pass, identical to standard SAC, and the Critic adds one extra safety-stream forward pass, exerting negligible impact on real-time control. The current evaluation uses a custom two-dimensional simulation environment with a fully controllable state space and cost functions, facilitating precise ablation analysis. This setting still cannot replace standard benchmark validation because Safety-Gymnasium and related benchmarks include richer dynamics, different observation definitions, and environment specific cost semantics. We therefore treat the custom environment as the main controlled testbed and add a short Safety-Gymnasium check in Section 5.4 only as preliminary external evidence, rather than as a complete benchmark study.

The state space is defined as a 23-dimensional continuous vector, consisting of normalized position coordinates (2D), normalized linear velocity components (2D), normalized heading angle (1D), target

relative direction vector (2D), and normalized distance readings from 16 uniformly distributed lidar beams (16D), encompassing self-motion state, target bearing information, and environmental obstacle distribution.

The action space is defined as a 2-dimensional continuous vector, corresponding to linear velocity and angular velocity commands, respectively, with a value range of $[-1, 1]$. After internal clipping, these map to maximum linear velocity $v_{\max}$ and maximum angular velocity $\omega_{\max}$, with time step $\Delta t = 0.1$ s.

The reward function consists of three terms:

$$r(s, a, s') = r_{\text{goal}} \cdot \mathbb{1}[\|s' - s_g\| < r_c] + \alpha_r(\|s - s_g\| - \|s' - s_g\|) - \beta_r(\|a\|^2) \tag{14}$$

where the first term is the sparse reward for reaching the goal, the second term is the dense progress reward based on the change in goal distance, and the third term is the regularization penalty for action magnitude.

The safety cost function takes a piecewise linear form, with the net distance from the vehicle surface to the nearest obstacle as input:

$$c(s) = \begin{cases} 1, & d_{\text{safe}} \le 0 \\ 1 - \dfrac{d_{\text{safe}}}{d_{\text{th}}}, & 0 < d_{\text{safe}} < d_{\text{th}} \\ 0, & d_{\text{safe}} \ge d_{\text{th}} \end{cases} \tag{15}$$

where $d_{\text{th}} = 1.0$ m is the safety distance threshold. This design introduces a linear transition zone between the collision region and the safe region, providing continuous safety gradient signals for policy optimization. Collision events do not trigger episode termination, allowing the agent to continue learning recovery strategies after collisions and avoiding the scarcity of safety experiences caused by sparse termination signals.

4.1.2 Evaluation Metrics

We adopt the following two evaluation metrics:

(1) **Success Rate (SR)**: Records whether the goal is successfully reached at the end of each episode, taking the mean over the last 100 episodes. SR reflects the task completion capability of the algorithm.

(2) **Violation Rate (VR)**: The proportion of steps with safety cost $c > 0$ out of the total steps in each episode, taking the mean over the last 100 episodes. VR reflects the degree of constraint satisfaction of the algorithm.

It should be noted that SR and VR are not independent metrics. When the agent's policy degenerates into immobility, VR approaches zero, but SR is also zero at this point. This phenomenon is termed spurious safety. Therefore, the validity of the VR metric is predicated on SR > 0, and comparing safety differences only makes practical sense under the premise that the task is completable.

4.2 Baseline Algorithms and Ablation Variants

Baseline algorithms: The main experiment selects the following four baseline algorithms (PPO-Lagrangian is additionally introduced in the generalization experiment of Section 5.4): (1) SAC: the standard soft actor-critic algorithm, which completely ignores the safety cost signal and only maximizes the cumulative task reward, serving as an unconstrained upper-bound reference for navigation performance; (2) SAC-Lagrangian: introduces an adaptive Lagrange multiplier on the SAC framework, updated by gradient ascent, converting the constraint satisfaction problem into unconstrained optimization of a modified reward, where the cost threshold $\kappa = 0.30$, the multiplier learning rate is 0.01, and the multiplier upper bound is 1.5; (3) CPO: Constrained Policy Optimization [7], an on-policy method that constrains the policy update step

length per round via a trust region and directly imposes cost constraints through bisection search on the dual variable, with the cost threshold set to 0.30 and the λ upper bound set to 10.0; (4) CVPO: Constrained Variational Policy Optimization [8], an off-policy method that decomposes constrained optimization into an E-step (determining the optimal sampling distribution) and an M-step (maximizing the weighted policy likelihood) via an EM framework; this paper adopts an SAC-style M-step Actor update, determining the safety multiplier λ per batch by comparing the batch mean cost with the constraint threshold, with the cost threshold set to 0.30 and the λ upper bound set to 5.0.

Ablation variants: (5) w/o DS (removing the dual-stream structure, degenerating to a single-stream Lagrangian Q-network); (6) w/o AHP (replacing AHP weighting with uniform tiered sampling, retaining the dual-stream structure); (7) w/Static-AHP (always using fixed criterion weights from the early training phase).

4.3 Hyperparameter Settings

All methods share the core hyperparameters: learning rate 3×10^{-4}, discount factor $\gamma = 0.99$, soft update coefficient $\tau = 0.005$, entropy coefficient α (auto-tuned), batch size 256, warm-up steps 1000, and a gradient update frequency of one parameter update per 4 environment interaction steps. For DS-AHP-SAC, the curriculum-based scheduling parameters are set to $\lambda_{\min} = 0.1$ and $\lambda_{\max} = 0.7$, and the AHP criterion weight update interval is 100 gradient steps. Each method is trained independently for 500 episodes, repeated over 5 random seeds. The main hyperparameter settings are shown in Table 2.

Table 2: Main hyperparameter settings.

Hyperparameter	Value
Actor hidden layer structure	$[128, 128]$
Dual-stream Q-network (per sub-stream) hidden layers	$[64, 64]$
Learning rate (Actor/Critic)	3×10^{-4}
Discount factor γ	0.99
Soft update coefficient τ	0.005
DS-AHP-SAC tiered buffer (per category)	2.5×10^5
Buffer capacity	5×10^5
Batch size	256
λ scheduling $[\lambda_{\min}, \lambda_{\max}]$	$[0.1, 0.7]$ (default)
AHP criterion weight update interval	100 gradient steps
Training episodes	500 (max 300 steps per episode)

5 Experimental Results and Analysis

5.1 Overall Performance Comparison

Table 3 summarizes the average performance and convergence speed of each algorithm over the last 100 episodes after 500 episodes of training.

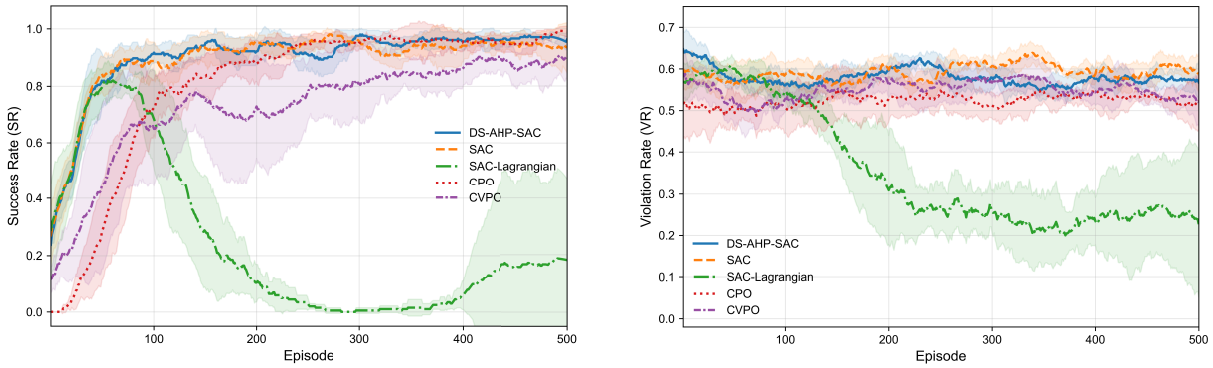
Fig. 2a and b shows the success rate and violation rate learning curves for all algorithms, respectively.

Navigation performance analysis. The steady-state SR of DS-AHP-SAC is 0.991, comparable to (i.e., within the inter-seed standard-deviation overlap of) unconstrained SAC (0.990), indicating that the proposed dual-stream decoupling structure does not introduce additional task performance loss after separating the navigation value and safety value into independent parameter spaces. The SR of SAC-Lagrangian is

only 0.143, and its standard deviation of 0.228 is also significantly higher than that of the other algorithms, exhibiting severe training instability. The root cause is that the Lagrange multiplier increases monotonically with the cost signal; once the multiplier exceeds the effective threshold, the modified reward becomes dominated by the safety penalty term, and the policy gradient continuously drives the Actor to select low-cost actions, ultimately degenerating into a stationary policy. The SR of CPO is 0.951; under the trust-region constraint, its navigation capability is strong but slightly lower than that of DS-AHP-SAC (0.991) and unconstrained SAC (0.990), because the on-policy mechanism has lower sample efficiency and accumulates effective training data more slowly. The SR of CVPO is 0.829; the per-batch dynamic adjustment of the safety multiplier λ makes its policy updates more conservative, and its navigation performance is lower than that of CPO (0.951) and DS-AHP-SAC (0.991).

Table 3: Final performance comparison results of all algorithms (Conv. eps. denotes the number of episodes to first reach SR $\geq$ 0.90).

Algorithm	SR (↑)	VR (↓)	Conv. Eps.
DS-AHP-SAC	0.991 ± 0.004	0.574 ± 0.011	56
SAC	0.990 ± 0.008	0.717 ± 0.005	79
SAC-Lagrangian	0.143 ± 0.228	0.447 ± 0.090	—
CPO	0.951 ± 0.028	0.529 ± 0.027	90
CVPO	0.829 ± 0.079	0.556 ± 0.018	103



a Success rate learning curves of all algorithms (shaded area represents inter-seed standard deviation).

b Safety violation rate learning curves of all algorithms (shaded area represents inter-seed standard deviation).

Figure 2: Learning curves of all algorithms under the baseline environment.

Safety performance analysis. Under comparable conditions where SR is close to 1.0, the VR of DS-AHP-SAC is 0.574, which is 20.0% lower than that of SAC (0.717). This performance gain originates from two aspects: first, the safety stream fits the long-term expectation of cumulative cost in an independent parameter space, providing the Actor with a more stable gradient signal than the single-stream Lagrangian modified reward; second, the curriculum-based scheduling increases the safety constraint weight linearly from 0.1 to 0.7, and the increase in the later training phase amplifies the influence of the safety stream in the fused Q-value, thereby achieving fine-grained adjustment of safety behavior on top of the converged navigation policy. The VR of SAC-Lagrangian is 0.447, but considering that its SR is only 0.143, this low VR constitutes spurious safety due to policy degradation and is not a valid reference for algorithm comparison. The VR of CPO is 0.529, lower than that of DS-AHP-SAC (0.574); the trust-region constraint mechanism strictly

limits the policy update step length and yields a higher degree of constraint satisfaction, but its SR (0.951) is also slightly lower than that of DS-AHP-SAC (0.991). The VR of CVPO is 0.556, close to that of DS-AHP-SAC (0.574), but its SR (0.829) is significantly lower than that of DS-AHP-SAC (0.991), indicating that the synergistic design of the dual-stream Q-network and dynamic AHP sampling offers a greater advantage in the combined safety and navigation performance.

The convergence results in the five-seed main comparison further show that DS-AHP-SAC first reaches SR $\geq$ 0.90 at an average of 56 episodes, while SAC, CPO, and CVPO require 79, 90, and 103 episodes, respectively. This result suggests that the full design reaches a usable navigation policy earlier than the selected baselines under the same training budget. The early low safety penalty makes the dominant signal of the dual-stream optimization close to unconstrained SAC, while the replay design increases the chance of revisiting safety-urgent samples. Since this comparison contains several coupled modules, the specific contribution of AHP is interpreted separately in the ablation study rather than being treated as an independent final-performance gain.

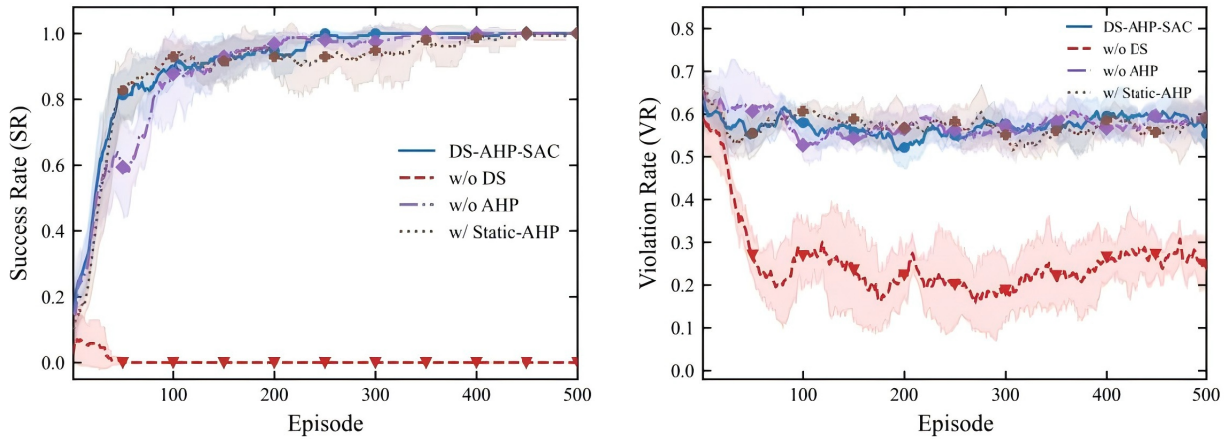
5.2 Ablation Study Analysis

Table 4 presents the quantitative results of the ablation experiments (5 random seeds each, 500 episodes), covering the independent contribution verification of the three innovations: dual-stream structure, AHP tiered sampling, and dynamic weight interpolation.

Table 4: Ablation study results.

Variant	SR (↑)	VR (↓)	Conv. Eps.
w/o DS	0.000	0.231 ± 0.043	—
w/o AHP	0.990 ± 0.007	0.575 ± 0.017	88
w/Static-AHP	0.960 ± 0.037	0.565 ± 0.024	57

Fig. 3a,b shows the success rate and violation rate learning curves for the ablation experiments.



a Ablation study: success rate learning curves.

b Ablation study: violation rate learning curves.

Figure 3: Ablation study learning curves.

Necessity of the dual-stream structure: After removing the dual-stream Q-network (w/o DS), the SR of the algorithm drops rapidly to 0 and the policy degenerates completely. Even with the AHP tiered

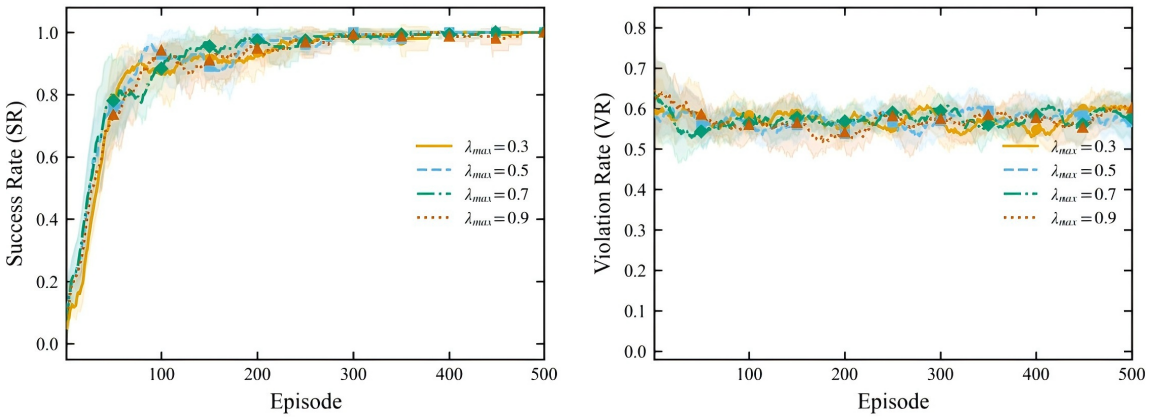
sampling mechanism retained, the gradient conflict between the navigation reward and safety cost in the same parameter space of the single-stream Lagrangian structure still causes training to collapse. This result experimentally verifies that dual-stream decoupling is a prerequisite for the algorithm to work effectively. In addition, the VR of w/o DS is 0.231, which appears lower than that of the full model, but considering SR = 0, this low VR originates from the spurious safety of the agent stopping moving and is not a valid reference.

The AHP tiered replay result should be interpreted cautiously. After replacing AHP weighting with uniform tiered sampling, the SR is 0.990 and the VR is 0.575 in the original five-seed ablation table, both within the inter-seed standard-deviation overlap of the full model. To check whether this conclusion is affected by the small seed number, we additionally ran seeds 5 to 9 for DS-AHP-SAC and w/o AHP under the same 500-episode setting, and combined them with the original seeds 0 to 4. Over the last 100 episodes, the 10-seed result of DS-AHP-SAC is SR = 0.960 ± 0.018 and VR = 0.567 ± 0.039, while w/o AHP gives SR = 0.962 ± 0.037 and VR = 0.558 ± 0.029. Welch tests still do not support a statistically significant final performance advantage of AHP (p = 0.879 for SR and p = 0.545 for VR). With the same sliding-window criterion, DS-AHP-SAC reaches SR ≥ 0.90 at 65.9 episodes on average, whereas w/o AHP requires 75.1 episodes. This trend suggests earlier entrance into a usable policy region, but the difference is also not statistically significant under the current 10-seed scale (p = 0.568). Therefore, the role of AHP in this setting is not claimed as improving the final steady-state policy. It is more reasonable to regard it as a training organization mechanism that biases early replay toward safety-urgent and informative samples, which may help reduce ineffective interactions in a limited training budget.

The comparison with w/Static-AHP also shows the value of changing the sampling focus over training. w/Static-AHP always uses the fixed criterion weights from the early training phase. Its VR is 0.565, slightly lower than that of the full model, but its SR drops from 0.991 to 0.960, and the SR standard deviation of 0.037 is far larger than that of the full model. This result indicates that fixed weights overemphasize safety urgency in the later training phase, leading to insufficient sampling diversity and navigation policy degradation in some seeds. Dynamic AHP transitions the criterion weights from safety urgency to information value with training progress, maintaining SR stability while preserving safety performance, reflecting the necessity of phase adaptive adjustment.

5.3 Sensitivity Analysis

Fig. 4a,b shows the success rate and violation rate learning curves under different $\lambda_{\max}$ settings, and Table 5 reports the corresponding metrics.



a Sensitivity analysis: success rate learning curves. **b** Sensitivity analysis: violation rate learning curves.

Figure 4: Sensitivity analysis learning curves under different $\lambda_{\max}$.

Table 5: Sensitivity analysis results for the safety penalty weight upper bound $\lambda_{\max}$ (5 seeds, evaluated over episodes 201–500; note that this evaluation window differs from the last-100-episode window used in Table 3, so absolute values are not directly comparable).

$\lambda_{\max}$	SR (↑)	VR (↓)
0.3	0.992 ± 0.012	0.572 ± 0.014
0.5	0.992 ± 0.005	0.576 ± 0.025
0.7	0.993 ± 0.009	0.580 ± 0.012
0.9	0.988 ± 0.007	0.574 ± 0.018

We conducted a sensitivity analysis on the upper bound of the safety penalty weight. As shown in Fig. 4 and Table 5, evaluated over 5 independent seeds in the second half of training (episodes 201–500): (1) In terms of SR, the differences among the four values are very small (range about 0.5%); SR drops slightly to 0.988 at $\lambda_{\max} = 0.9$, while the others are all above 0.992, indicating that the algorithm is strongly robust to the choice of $\lambda_{\max}$. (2) In terms of VR, VR exhibits a trend of increasing up to $\lambda_{\max} = 0.7$ and then slightly decreasing: VR = 0.580 at $\lambda_{\max} = 0.7$ and VR = 0.572 at $\lambda_{\max} = 0.3$, with a difference of only 0.008. A larger safety weight does not significantly reduce the navigation success rate, indicating that the dual-stream Q-network can effectively absorb the safety penalty signal without harming task performance. Overall, $\lambda_{\max} = 0.7$ achieves the best balance between SR and VR, and the algorithm remains stable over a very small perturbation range, verifying that the proposed method is insensitive to the core hyperparameter.

To verify the robustness of the AHP method to the choice of matrix parameters, we designed three matrix variants while keeping the priority ordering $D > C \gg A > B$ unchanged. The baseline matrix is the default matrix with CR = 0.025, the mild variant has CR = 0.012, and the strengthened variant has CR = 0.056. Each group was trained with 3 independent seeds for 500 episodes. The experimental results, summarized in Table 6, are as follows. The baseline matrix obtains SR = 0.967 and VR = 0.578; the mild variant obtains SR = 0.940 and VR = 0.583; the strengthened variant obtains SR = 0.953 and VR = 0.561.

Table 6: Robustness of the AHP comparison matrix (3 seeds, episodes 201–500; all variants satisfy CR < 0.1 and keep the priority ordering $D > C \gg A > B$).

Matrix Variant	SR (↑)	VR (↓)
Baseline (CR = 0.025, default)	0.967	0.578
Mild variant (CR = 0.012)	0.940	0.583
Strengthened variant (CR = 0.056)	0.953	0.561

The SR range of the three variants is 0.027 (about 3%) and the VR range is 0.022 (about 4%); compared with the full model (DS-AHP-SAC, SR = 0.991, VR = 0.574), the maximum SR gap is about 5% and the VR gap does not exceed 2%. Constrained by the sample size of 3 seeds, the confidence intervals of each group are relatively wide, but the direction of the results is consistent. Under the premise that the priority ordering ($D > C \gg A > B$) remains unchanged, reasonable perturbations of the matrix element values have a limited impact on the final performance, and the AHP method exhibits a certain robustness to specific matrix parameters.

With the upper bound of the safety constraint weight fixed at $\lambda_{\max} = 0.7$, we further ran each of $\lambda_{\min} \in \{0.0, 0.05, 0.1, 0.2\}$ over 3 seeds for 500 episodes. The experimental results are summarized in Table 7. When

λ_{min} = 0.0, SR = 0.980 and VR = 0.564. When λ_{min} = 0.05, SR = 0.940 and VR = 0.555. The default value λ_{min} = 0.1 obtains SR = 0.967 and VR = 0.578. When λ_{min} = 0.2, SR = 0.873 and VR = 0.490.

Table 7: Sensitivity to λ_{min} (λ_{max} = 0.7 fixed, 3 seeds, episodes 201–500).

λ_{min}	SR (↑)	VR (↓)
0.0	0.980	0.564
0.05	0.940	0.555
0.1 (default)	0.967	0.578
0.2	0.873	0.490

When λ_{min} is in the range of 0.0 to 0.1, SR stays between 0.940 and 0.980 (range about 4%) and VR between 0.555 and 0.578 (range < 3%), with stable performance. When λ_{min} is increased to 0.2, the initial strength of the safety constraint increases, exhibiting a more conservative trade-off: VR drops to 0.490 (improved safety), but SR drops to 0.873; constrained by the sample size of 3 seeds, the SR confidence interval is relatively wide (±0.174), with limited statistical reliability. The default value λ_{min} = 0.1 achieves a good balance between task success rate and safety. *Note: these small-sample (3-seed) experiments yield absolute values that fluctuate from the main 5-seed results in Table 3; the conclusions here are drawn from the trends rather than the absolute values.*

5.4 Generalization and Safety-Performance Tradeoff Analysis

To preliminarily verify the behavior of the algorithm in more complex environments, we conducted supplementary experiments in a high-difficulty navigation scenario (20 × 20 m map, 20 obstacles, randomly generated targets). Four core algorithms (DS-AHP-SAC, SAC, PPO-Lagrangian, SAC-Lagrangian) are evaluated under the same hyperparameters, trained from scratch, independently assessing performance under more challenging conditions. The main results, summarized in Table 8, are: the SR of DS-AHP-SAC is 0.644, lower than that of unconstrained SAC (SR = 0.892); the SR of PPO-Lagrangian is 0.768 and that of SAC-Lagrangian is 0.680, both higher than DS-AHP-SAC, whereas the VR of DS-AHP-SAC (0.352) is lower than all comparison methods (SAC: 0.498, SAC-Lagrangian: 0.448, PPO-Lagrangian: 0.381).

Table 8: Generalization in the high-difficulty scenario (20 × 20 m, 20 obstacles, 3 seeds; std shown where available). Bold marks the best SR (highest) and best VR (lowest).

Algorithm	SR (↑)	VR (↓)
DS-AHP-SAC	0.644 ± 0.109	**0.352**
SAC	**0.892** ± 0.060	0.498
PPO-Lagrangian	0.768	0.381
SAC-Lagrangian	0.680	0.448

This result indicates that the safety constraint remains effective in the high-difficulty scenario, with the violation rate maintained at the lowest among all methods. However, as the environmental complexity increases, the constraint imposed by the safety restriction on the task completion rate becomes more pronounced: SR drops by about 24.8 percentage points compared with unconstrained SAC, and the method exhibits a safety-conservative characteristic. This is the inherent cost of safety constraints as task difficulty

increases; in safety-priority application scenarios, this tradeoff is acceptable. How to simultaneously maintain the success rate in more complex scenarios is left for future improvement.

We also attempted a short external check on SafetyPointGoal1-v0 from Safety-Gymnasium, using cumulative reward and per-step cost rate as metrics because this benchmark does not share the same terminal-success definition as the custom navigation task. The installed benchmark stack required a local compatibility setting for Gymnasium 0.28.1 and a MuJoCo group field shape correction before the environment could be initialized, so the experiment is reported only as a preliminary benchmark check. Under a 120-episode, 3-seed, 300-step budget, we compared DS-AHP-SAC with unconstrained SAC, and the results are summarized in Table 9.

Table 9: SafetyPointGoal1-v0 preliminary check (3 seeds, last 30 episodes). Lower cost rate indicates fewer unsafe steps.

Method	Return (↑)	Cost Rate (↓)	Cost Per Episode (↓)
DS-AHP-SAC	−0.404 ± 0.333	0.0526 ± 0.0030	15.77 ± 0.90
SAC	1.046 ± 0.357	0.0743 ± 0.0346	22.29 ± 10.39

In this short-budget setting, SAC obtains a higher cumulative reward, while DS-AHP-SAC gives a lower cost rate and lower per-episode cost. This result is consistent with the safety-performance tradeoff observed in the custom environment. The SafetyPointGoal1-v0 check is still not used to claim benchmark level superiority. Its role is to test whether the proposed safety cost stream can be executed outside the hand built two-dimensional environment and whether it can reduce unsafe interaction relative to an unconstrained SAC reference under the same small budget. Because the observation dimension, reward scale, termination logic, and cost source differ from the custom environment, a fair full benchmark comparison against more constrained RL baselines such as CPO, CVPO, and Lagrangian variants would require environment specific hyperparameter tuning, longer training budgets, and more random seeds. This remains an important limitation of the present study.

Fig. 5 shows the final performance bar chart for the four algorithms in the high-difficulty scenario.

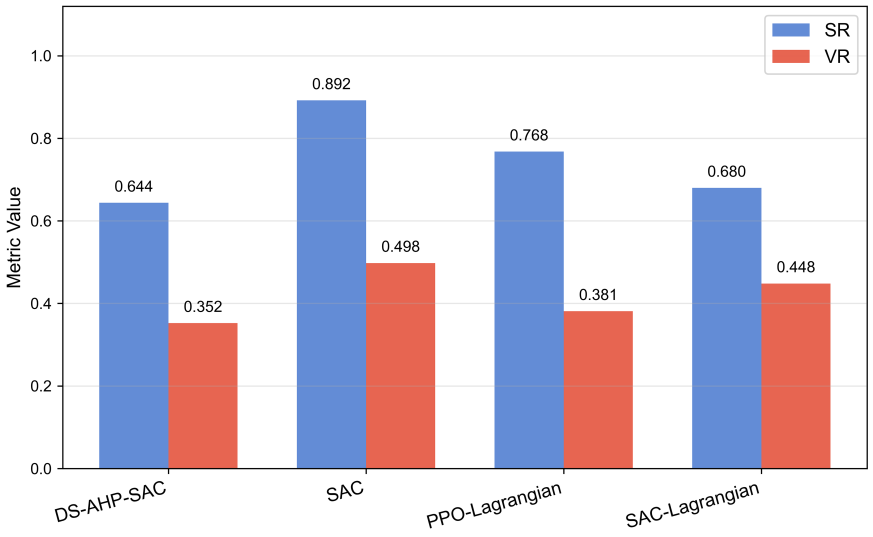


Figure 5: Generalization validation: bar chart comparison in the high-difficulty scenario.

6 Conclusion

This paper addresses the conflict between navigation performance and safety constraints in safe reinforcement learning by proposing the DS-AHP-SAC algorithm. The algorithm eliminates gradient conflicts at the Critic level and mitigates gradient competition at the Actor level through the dual-stream Q-network, accelerates convergence through dynamic AHP tiered experience replay, and achieves a smooth transition from exploration to safety through curriculum-based scheduling. Experimental results demonstrate that DS-AHP-SAC significantly reduces violation rates compared to unconstrained baselines without compromising navigation success rates, achieving a good balance between navigation performance and safety. Ablation experiments verify the necessity of the dual-stream Q-network and show that AHP mainly improves early learning efficiency rather than final steady-state metrics. The method is still at the simulation validation stage. For realistic autonomous driving use, several conditions must be satisfied before deployment. The cost function should include vehicle speed, braking distance, perception uncertainty, and dynamic obstacles; the lidar observations used here should be replaced or fused with noisy camera, radar, and lidar measurements; the policy should be tested under actuator delay and model mismatch; and the safety layer should be connected with rule based emergency braking or backup controllers based on control barrier functions. In terms of scalability, the Actor keeps the same single forward pass at inference time, but training memory and sampling cost grow with replay buffer size and with the number of safety categories. Larger maps, moving obstacles, and multi-vehicle interactions would therefore require parallel rollout collection and more careful replay management. Future work will carry out full Safety-Gymnasium benchmark experiments with tuned hyperparameters and more random seeds, and will explore learnable cost models and recovery controllers to improve the safety and success rate tradeoff in high difficulty scenarios.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, Shengxuan Dong and Xiongwei Li; methodology, Shengxuan Dong; software, Shengxuan Dong; validation, Shengxuan Dong and Xiongwei Li; formal analysis, Shengxuan Dong; investigation, Shengxuan Dong; writing—original draft preparation, Shengxuan Dong; writing—review and editing, Xiongwei Li; visualization, Shengxuan Dong; supervision, Xiongwei Li. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the Corresponding Author, Xiongwei Li, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature. 2015;518(7540):529–33. doi:10.1038/nature14236.
2. Yang L, Bi J, Yuan H. Dynamic path planning for mobile robots with deep reinforcement learning. IFAC-PapersOnLine. 2022;55(11):19–24. doi:10.1016/j.ifacol.2022.08.042.
3. Zhang Z, Fu H, Yang J, Lin Y. Deep reinforcement learning for path planning of autonomous mobile robots in complicated environments. Complex Intell Syst. 2025;11(6):277. doi:10.1007/s40747-025-01906-9.
4. Wachi A, Shen X, Sui Y. A survey of constraint formulations in safe reinforcement learning. In: Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI-24); 2024 Aug 3–9; Jeju, Republic of Korea. p. 8262–71. doi:10.24963/ijcai.2024/913.

5. Stooke A, Achiam J, Abbeel P. Responsive safety in reinforcement learning by PID Lagrangian methods. In: Proceedings of the 37th International Conference on Machine Learning (ICML); 2020 Jul 13–18; Virtual. p. 9133–43.
6. Zhang Y, Vuong Q, Ross KW. First order constrained optimization in policy space. In: Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS); 2020 Dec 6–12; Virtual.
7. Achiam J, Held D, Tamar A, Abbeel P. Constrained policy optimization. In: Proceedings of the 34th International Conference on Machine Learning (ICML); 2017 Aug 6–11; Sydney, Australia. p. 22–31.
8. Liu Z, Cen Z, Isenbaev V, Liu W, Wu S, Li B, et al. Constrained variational policy optimization for safe reinforcement learning. In: Proceedings of the 39th International Conference on Machine Learning (ICML); 2022 Jul 17–23; Baltimore, MD, USA. p. 13644–68.
9. Cheng R, Orosz G, Murray RM, Burdick JW. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. Proc AAAI Conf Artif Intell. 2019;33(1):3387–95. doi:10.1609/aaai.v33i01.33013387.
10. Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. In: Proceedings of the 4th International Conference on Learning Representations (ICLR); 2016 May 2–4; San Juan, Puerto Rico.
11. Saaty TL. The analytic hierarchy process: planning, priority setting, resource allocation. New York, NY, USA: McGraw-Hill; 1980.
12. Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proceedings of the 35th International Conference on Machine Learning (ICML); 2018 Jul 10–15; Stockholm, Sweden. p. 1861–70.
13. Zhao T, Wang M, Zhao Q, Zheng X, Gao H. A path-planning method based on improved soft actor-critic algorithm for mobile robots. Biomimetics. 2023;8(6):481. doi:10.3390/biomimetics8060481.
14. Fedus W, Ramachandran P, Agarwal R, Bengio Y, Larochelle H, Rowland M, et al. Revisiting fundamentals of experience replay. In: Proceedings of the 37th International Conference on Machine Learning (ICML); 2020 Jul 13–18; Virtual. p. 3061–71.
15. 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 (ICML); 2018 Jul 10–15; Stockholm, Sweden. p. 1587–96.
16. Bengio Y, Louradour J, Collobert R, Weston J. Curriculum learning. In: Proceedings of the 26th International Conference on Machine Learning (ICML); 2009 Jun 14–18; Montreal, Canada. p. 41–8. doi:10.1145/1553374.1553380.