DOI 10.15622/ia.25.4.9

Y. TANG, Y. GERMAN

ENHANCED Q-LEARNING FOR GRID NAVIGATION: DIRECTION-SENSITIVE PRIORS, ANNEALED EXPLORATION, AND POTENTIAL-BASED REWARD OPTIMIZATION

Tang Y., German Y. **Enhanced Q-Learning for Grid Navigation: Direction-Sensitive Priors, Annealed Exploration, and Potential-Based Reward Optimization.**

Abstract. This paper presents an enhanced Q-learning framework specifically designed for grid navigation in environments with obstacles. Building upon the foundation of standard tabular Q-learning, we propose three pivotal improvements: a direction-sensitive Q-table initialization method that achieves goal alignment without requiring prior knowledge of obstacles; an annealed Boltzmann exploration strategy augmented with upper confidence bound terms to facilitate balanced and adaptive sampling; and a potential-based continuous reward shaping function to deliver denser feedback and accelerate the learning process. These enhancements address common challenges in sparse-reward settings, such as inefficient exploration and sluggish value propagation. Experimental evaluations conducted on randomly generated grids demonstrate that, compared to baseline methods such as standard Q-learning and its variants, our approach achieves higher success rates, shorter optimal path lengths, and faster convergence speeds. By preserving the model-free generality of Q-learning while enhancing sample efficiency, this framework proves well-suited for practical applications in robotics and path planning domains.

Keywords: reinforcement learning, Q-table initialization, path planning, exploration-exploitation trade-off, continuous reward, grid map.

1. Introduction. Path planning constitutes a fundamental challenge within the domains of robotics and autonomous systems, with the primary objective of computing collision-free trajectories from an initial position to a designated target while adhering to environmental constraints (such as obstacle avoidance) [1]. Representative applications include navigation of mobile robots in warehouse settings and traversal of complex terrains by unmanned aerial vehicles (UAVs). Within discrete grid-world environments, this task is typically reduced to identifying the shortest feasible path across a cellular grid [2].

Traditional heuristic methods have dominated the early development of path planning techniques. Dijkstra's algorithm, a graph search approach, systematically explores nodes in ascending order of distance from the starting point to ensure the shortest path in graphs with non-negative weights. However, its time complexity of $O(V^2)$ renders it computationally inefficient for large-scale grids due to substantial memory requirements. The A* algorithm and its variants extend this methodology by incorporating admissible heuristics, such as Manhattan or Euclidean distances, to optimize the search space and achieve optimality in static maps while reducing computational overhead. Sampling-based methods, including the Rapidly-exploring Random Tree (RRT) and its derivatives, offer probabilistic

completeness in high-dimensional spaces [3]. Nevertheless, these approaches rely heavily on complete prior knowledge of the environment and necessitate frequent replanning in dynamic scenarios. Moreover, their inability to learn adaptive strategies from experience contributes to inefficiencies in real-time robotic tasks.

Reinforcement Learning (RL) has emerged as a compelling alternative, wherein an agent learns optimal behaviors through iterative interactions with its environment, guided by rewards or penalties to maximize cumulative returns. Key components include the agent-environment interaction loop, formalized through a Markov Decision Process (MDP) that encapsulates states, actions, transition probabilities, and reward functions, alongside value or policy functions for decision-making [4]. Unlike supervised learning, which depends on labeled datasets, RL emphasizes trial-and-error learning, rendering it particularly suitable for path planning in partially observable or dynamically changing environments. Presently, Q-learning has established itself as the predominant RL method for path planning, marking a paradigmatic shift from search-based to learning-based approaches.

Despite the notable successes of Q-learning and its variants, their application to practical, especially large-scale, grid navigation tasks reveals three significant bottlenecks stemming from their core mechanisms. The initial challenge confronting standard Q-learning lies in the "cold start" and blind exploration problem, wherein the Q-table is typically initialized with zero or random values, resulting in a lack of any prior guidance for the agent at the outset of the learning process [5]. Within a large map, this necessitates prolonged periods of random wandering until the agent accidentally encounters the target or receives a non-zero reward, a process that may demand thousands of episodes in sparse reward settings, thereby leading to a delayed start and significantly reduced sample efficiency during the initial training phase. The second issue pertains to the inefficiency and high variance inherent in the exploration strategy, with the commonly adopted ε-greedy approach proving both noisy and undirected [6]. It randomly selects among all actions, including those known to be suboptimal, during exploration phases, which introduces substantial variance without the ability to distinguish between high-potential unknown actions and evidently poor ones. The third critical bottleneck is the credit assignment problem arising from sparse rewards, where the most natural reward structure in path planning provides positive feedback only upon the agent reaching the target. This engenders a severe challenge, as the value signal must propagate backward from the goal state to the initial state across potentially thousands of steps, with the agent receiving feedback of minimal informational value [7], such as small

movement penalties, for the majority of the time, thereby severely impeding the convergence of the value function.

In response to the aforementioned challenges, we propose an enhanced Q-learning framework that systematically addresses these issues, with the following specific contributions:

To mitigate the "cold start" and blind exploration problem, we introduce a direction-sensitive Q-table initialization approach, which integrates normalized Manhattan distance and action-goal alignment to provide mild prior guidance for Q-values. This method, independent of obstacle information, breaks the action symmetry in equidistant states by favoring actions oriented toward the target, thereby significantly reducing ineffective lateral movements and early random wandering.

To address the inefficiency and high variance of the exploration strategy, we devise a hybrid exploration strategy that combines an annealed Boltzmann distribution with Upper Confidence Bound (UCB) terms. The annealing temperature decreases with the progression of episodes, enabling a smooth transition from broad exploration to exploitation via softmax, while the UCB component encourages the selection of under-explored actions, reducing noise and optimizing the exploration-exploitation trade-off [8].

To overcome the credit assignment difficulty induced by sparse rewards, we propose a potential-based continuous reward function that leverages distance progress and potential differences to offer dense feedback. This approach guides the agent toward the target through a potential field and imposes penalties upon collisions, ensuring policy invariance while markedly enhancing learning efficiency in long-sequence tasks through accelerated convergence.

2. Related Works. The conventional Q-learning algorithm relies on temporal-difference updates to estimate the action-value function, achieving convergence to the optimal policy in finite MDPs. However, it encounters pronounced difficulties in complex grid environments, manifesting as protracted convergence and susceptibility to local optima [9]. Contemporary research has predominantly pursued enhancements along three principal axes.

Q-table initialization constitutes the first avenue of improvement. Zhou et al. [10] introduced the O-QL algorithm, employing distance-based heuristics and action-goal alignment to initialize the Q-table, thereby providing a "warm start" that mitigates early blind exploration. In a similar vein, Ben-Akka et al. [11] implemented tailored rewards penalizing state revisitation to promote novel path discovery, coupled with cumulative reward-driven dynamic adjustment of the ϵ-greedy decay, facilitating a seamless transition from exploration to exploitation; nonetheless, their initialization inadequately accounts for directional alignment, potentially

engendering action symmetry in equidistant states. Ma et al. [12] partitioned the environment into contiguous local segments and leveraged prior knowledge for Q-table initialization; while this elevates local efficiency, the global initialization lacks direction sensitivity, often overlooking goal-oriented biases.

The second dimension encompasses innovations in exploration strategies. Huang et al. [13] integrated heuristic search, annealing mechanisms, and reactive navigation within a Dyna-Q variant to equilibrate global optimality with real-time adaptability, outperforming classical Q-learning in unknown dynamic environments; however, its ϵ -greedy component may introduce noise, resulting in insufficient late-stage exploration. Hidayat et al. [14] incorporated an achievement motivation model into Q-learning to generate path variants, thereby reducing collision risks in multi-robot systems and yielding 2-4 optimal trajectories; nevertheless, the tuning of motivation parameters remains empirically driven, potentially inducing unstable paths in uncertain settings. Wang et al. [15] proposed the ETQ-learning algorithm, which integrates a "static assignment plus dynamic adjustment" reward mechanism with an ε -increasing greedy strategy to enhance global optimality and convergence rates. This method also introduces an expansion distance to establish collision buffers for safer navigation; however, its reliance on empirically fitted formulas to determine iteration thresholds may restrict the algorithm's adaptability when applied to environments with highly irregular or structurally distinct topologies without recalibration. Additionally, Falloh et al. [16] proposed a dynamic path planning algorithm that modifies Q-learning exploration through integration with a novel dynamic reward, enhancing adaptability in unknown dynamic environments; despite diminishing ineffective iterations, this strategy retains dependence on a fixed decay rate, predisposing it to premature greediness.

The third axis focuses on reward function optimization. The O-QL framework [10] devised a continuous reward function in conjunction with RMSprop-based adaptive learning rate adjustment for dynamic tuning to expedite convergence; yet, its reward design omits potential differences, risking policy bias. Ben-Akka et al. [11] penalized state revisitation within the same episode to avert looping and augment exploration efficiency; although efficacious, the reward remains sparse, exacerbating credit assignment challenges in extended sequences. Furthermore, Wang et al. [17] fused heuristic search with continuous reward shaping to accelerate convergence in complex grids and demonstrated robustness under high-density obstacles; however, its reward mechanism is static, lacking adaptive modulation to accommodate dynamic variations.

Despite these advancements in conventional Q-learning, its variants continue to grapple with persistent challenges, including delayed guidance from random initialization, variance induced by noisy exploration, and impeded credit assignment due to sparse rewards. Investigations reveal that traditional reinforcement learning necessitates thousands of episodes for convergence in large or obstacle-dense spaces, with success rates declining markedly in high-density obstacle configurations. To address these bottlenecks while maintaining computational feasibility, the proposed method is strategically positioned as a model-free learning approach that bridges the gap between classical heuristic search and high-complexity deep learning architectures. Specifically, it distinguishes itself from search-based algorithms like A* or Dijkstra by focusing on an agent's ability to learn environmental structures without a predefined model. Notably, while Dijkstra is utilized here as a 'ground truth' for global optimality (given that A* is theoretically consistent with Dijkstra in path quality under admissible heuristics), our framework emphasizes adaptive interaction. Furthermore, unlike the high-resource dependency and 'black-box' nature of Deep RL, or specialized variants such as MQL [14] that prioritize 'path variations' for multi-robot deconfliction, our method is optimized for single-agent global optimality and extreme computational efficiency. By resolving 'cold start' and convergence issues via direction-sensitive initialization and stable exploration mechanisms, this framework provides a lightweight yet robust alternative.

3. Problem Formulation. We formulate the grid navigation task as an MDP defined by the tuple (S, A, P, R, γ) where S is the state space consisting of grid positions $\{(i,j)|0 \le i \le H, 0 \le j \le W\}$ in an $H \times W$ grid; $A = \{up, down, left, right\}$ represents the four directional actions, as shown in Figure 1, the four-directional action set enables efficient navigation in grid-based environments; $P: S \times A \to S$ is the deterministic transition function, subject to obstacle collisions (where the agent remains in place upon hitting an obstacle); $R: S \times A \times S \to \mathbb{R}$ is the reward function; and $\gamma \in [0,1)$ is the discount factor. The objective is to learn a policy $\pi: S \to A$ that maximizes the expected cumulative discounted reward, starting from an initial state s_0 and reaching a goal state g. Obstacles are randomly placed with density $\xi \in [0,1)$, ensuring s_0 and g remain unobstructed.

The environment is modeled using grid-based discretization, a standard technique in Q-learning-based mobile robot path planning. This approach partitions the continuous workspace into a uniform lattice of cells, each classified as either obstacle $o \in O$ or free space $e \in E$, yielding a compact representation (O, E). The robot is treated as a point mass, and the outcome of any action a_t from state s_t is deterministically resolved via the mapping:

$$L(s_t, a_t) = l_t, \ \ l_t \in \{E, O\}. \tag{1}$$

Discretizing the state space in this manner confers multiple benefits. It transforms potentially infinite continuous states into a finite set, rendering the Q-table tractable and mitigating excessive memory demands or convergence delays. Each cell uniquely defines a state, and transitions under discrete actions are unambiguous, streamlining Q-value updates [18].

Standard Q-learning estimates the action-value function $Q(s, a)$ via temporal-difference updates calculated as:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a} Q(s_{t+1}, a') - Q(s_t, a_t)], \tag{2}$$

where α is the learning rate. However, when applied to large or obstacle-dense grids, standard Q-learning exhibits significant limitations. First, random or zero initialization coupled with ε-greedy exploration leads to inefficient early-stage wandering, resulting in slow convergence.

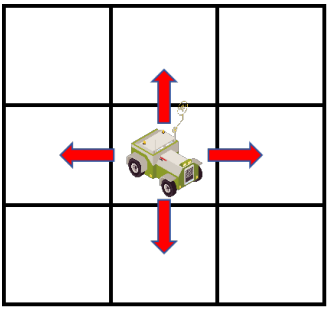


Fig. 1. Discrete action selection

4. Overview of the Proposed Enhanced Q-learning Algorithm. We present an enhanced Q-learning framework that integrates three core improvements: direction-sensitive Q-table initialization, UCB-augmented annealed Boltzmann exploration, and potential-based continuous reward function. These components systematically address the aforementioned bottlenecks by respectively injecting lightweight navigation priors, facilitating a smooth balance between exploration and exploitation, and delivering dense feedback. Furthermore, this framework preserves the model-free nature of Q-learning, thereby ensuring its applicability across a broad range of scenarios.

4.1. Direction-Sensitive Q-Table Initialization. A significant limitation of standard Q-learning stems from its dependence on zero or

random initialization, a method that overlooks inherent navigation priors such as goal proximity and directional orientation. Recent research, such as that by Zhou et al. [10], has proposed a heuristic initialization approach to enhance early exploration. However, their method leads to symmetric treatment of actions in equidistant states, resulting in prolonged and ineffective lateral movements during the initial exploration phase. Such a symmetry-breaking failure exacerbates sample inefficiency, particularly in obstacle-rich environments, where agents squander computational updates on non-progressive paths.

To mitigate this, we introduce a direction-sensitive initialization that provides a "warm start" by biasing Q-values toward goal-aligned actions without requiring obstacle knowledge [19], thus accelerating value propagation while maintaining generality. This approach integrates the normalized Manhattan distance and an action-goal alignment score κ to guide the agent effectively. In the resulting Q-value distribution, actions consistent with the target direction (e.g., up, down, left, right) are assigned higher initial Q-values, disrupting action symmetry in equidistant states and prioritizing goal-oriented movements. The gradient-like color distribution across the grid, as visualized, illustrates a spatial value progression from the starting point to the target, significantly mitigating ineffective lateral shifts and early random wandering, with pronounced benefits in large or obstacle-dense grid environments.

Specifically, for each state (i, j) and action a, the initial Q-value is computed as:

$$Q(i,j,a) = \beta_0 + \beta_1 \left(1 - \hat{d}\big(i,j;g_i,g_j\big)\right) + \beta_2 \kappa(i,j,a), \tag{3}$$

where β_0 is a constant bias, β_1 weights the distance term, and β_2 emphasizes directional alignment. The normalized Manhattan distance $\hat{d}(i,j;g_i,g_j)$ promotes current state $s(i,j)$ closer to the goal (g_i,g_j):

$$\hat{d}(i,j;g_i,g_j) = \frac{|i-g_i|+|j-g_j|}{(H-1)+(W-1)+\varepsilon_0}, \tag{4}$$

with $\varepsilon_0 = 1e-8$to prevent division by zero. The alignment score $\kappa(i,j,a)$ quantifies how well action a points toward the goal:

$$\vec{u}_g(i,j) = \left(\frac{g_i - i}{\sqrt{(g_i-i)^2+(g_j-j)^2+\varepsilon_0}}, \frac{g_j-j}{\sqrt{(g_i-i)^2+(g_j-j)^2+\varepsilon_0}} \right), \tag{5}$$

$$\kappa(i,j,a) = max(0, \vec{u}_g(i,j) \cdot \vec{u}_a), \quad (6)$$

where $\vec{u}_a$ is the unit vector for action a (e.g., (1,0) for right). This favors actions aligned with the goal vector, assigns zero to opposites, and intermediate values to perpendiculars, effectively reducing invalid traversals.

This initialization is vectorizable for low overhead and integrates with visit counts for dynamic adjustment during training. The Q-table following initialization is depicted in Figure 2, where the generated Q-value distribution is visualized as a binary grid map.

The color intensity directly corresponds to the magnitude of the initial Q-values: darker shades indicate higher Q-values, while lighter shades denote lower ones. The gradient-like color distribution across the grid, as visualized, illustrates a spatial value progression from the starting point to the target, significantly mitigating ineffective lateral shifts and early random wandering, with pronounced benefits in large or obstacle-dense grid environments.

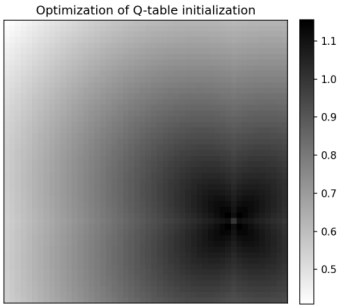


Fig. 2. Optimization of Q-table initialization

4.2. Annealed Boltzmann Exploration with UCB. Standar ε-greedy exploration suffers from high noise in early phases (uniform random actions) and insufficient diversity in later stages (premature greediness), often trapping agents in suboptimal paths amid obstacles and increasing variance in learning. This binary switch from exploration to exploitation hinders smooth adaptation, particularly in dynamic or uncertain grids where rare detours must be discovered.

To overcome these issues, we propose an annealed Boltzmann exploration augmented with UCB terms, enabling probabilistic sampling that starts broad and gradually sharpens while optimistically favoring under-explored actions. The temperature T_e decays exponentially per episode e:

$$T_e = max(T_{min}, T_0 \times \rho^e), \quad (7)$$

where T_0 represents the initial temperature, $T_{\min}$ denotes the minimum temperature, ρ is the decay rate ($0 < \rho < 1$). Action probabilities are computed using the softmax function:

$$P(a|s) = \frac{exp(Q^{'}(s,a)/T_e)}{\sum_{a^{'}\in A} exp(Q^{'}(s,a^{'})/T_e)}, \quad (8)$$

where $Q'(s,a) = Q(s,a) + c\sqrt{\frac{ln(t+1)}{N(s,a)+\varepsilon_0}}$ incorporates UCB, with $c = 0.1$ as the exploration constant, t denotes the total number of steps, and $N(s,a)$ is the state-action visit count. Visit counts are tracked via arrays for states and actions, incrementing per step. This hybrid mechanism encourages deep exploration in high-uncertainty areas (e.g., near obstacles) through UCB optimism. It mitigates the risk of local optima by maintaining calibrated uncertainty awareness, even in the later stages of training. The specific algorithm flow is shown in Figure 3.

Algorithm 1 UCB-Annealed Boltzmann Action Selection

Require: Current state s, Q-values $Q(s,\cdot)$, visit counters $N(s)$ and $N(s,a)$, UCB constant c, temperature schedule τ_e, small $\epsilon > 0$
Ensure: Selected action a
$N(s) \leftarrow N(s) + 1$ ▷ Increment state visit count
// Compute UCB-augmented action values
for all $a \in \mathcal{A} = \{\text{up, down, left, right}\}$ **do**
  $U(s,a) = Q(s,a) + c\sqrt{\frac{\ln N(s)}{N(s,a)+\epsilon}}$ ▷ *Eq. (6)*
end for
// Softmax sampling with annealed temperature
Compute probabilities:
  $P(a|s) = \frac{\exp(U(s,a)/\tau_e)}{\sum_{a'}\exp(U(s,a')/\tau_e)}$ ▷ *Eq. (7)*
Sample action $a \sim P(\cdot|s)$
$N(s,a) \leftarrow N(s,a) + 1$ ▷ Update action visit count
return a

Fig. 3. Algorithm flow of improved exploration strategy

4.3. Potential-Based Continuous Reward Function. Sparse rewards in standard Q-learning present a significant credit assignment challenge, as agents receive feedback solely at episode termini or upon encountering collisions. This sparsity creates “dead zones” in long-horizon tasks, where the absence of intermediate feedback hampers the propagation of value signals, consequently diminishing success rates in environments characterized by dense obstacles due to the resulting sparse gradients [20]. Such a reward structure impedes the agent’s ability to make meaningful progress,

particularly over extended sequences where timely reinforcement is critical.

To address this limitation, we adopt a potential-based reward methodology, drawing inspiration from the Potential Field Method (PFM) framework [21]. The core concept reimagines the grid environment as a virtual potential field, wherein the goal state represents the global minimum of the potential, and each state transition corresponds to a descent along this field's gradient. This approach transforms the sparse reward landscape into a continuous signal, guiding Q-value updates toward the target without altering the optimal policy. The reward function is defined in Equation (9):

$$r = (\hat{d}(s,g) - \hat{d}(s',g)) - p_{step} + \gamma \cdot \Phi(s') - \Phi(s) - c_{collision}\,, \qquad (9)$$

where $\Phi(s) = -\hat{d}(s,g)$ is the potential function, engineered to decrease as the agent approaches the goal, thus mimicking an attractive potential field. The step penalty $p_{step} = 0.05$ is applied universally, while an additional collision penalty $c_{collision} = 0.25$ is subtracted in the event of a collision. The specific hyperparameters were determined through empirical tuning to establish an optimal balance between exploration efficiency and safety constraints. Specifically, the step penalty p_{step} functions as a regularization term for path optimality. Empirically, we observed that an excessively high value (e.g., > 0.1) tends to overwhelm the potential gradient, leading to excessive risk aversion or premature episode termination, whereas a negligible value results in circuitous paths. Simultaneously, the collision penalty $c_{collision}$ was set to a moderate value to enforce strict obstacle avoidance. This value was chosen to deliver a sharp negative feedback signal upon collision without inducing "freezing" behavior, which is a common failure mode observed with higher penalties where agents cease exploration to avoid high-variance negative rewards in dense obstacle regions.

The continuous feedback, driven by potential differences, effectively alleviates the "dead zone" effect, enabling efficient value propagation across the state space. Empirical evaluations conducted on randomly generated grids demonstrate that this approach accelerates convergence, surpassing traditional sparse reward configurations in both sparse-reward and obstacle-rich environments. The enhanced learning efficiency and robustness are particularly pronounced in high-density obstacle scenarios, where the agent navigates complex grids with greater stability and precision.

5. Experiments

5.1. Datasets and Experimental Devices. To evaluate the universality and robustness of the enhanced Q-learning algorithm across different physical environments, we designed experimental scenarios using randomly generated grid maps with varying obstacle densities. Specifically, we generated a

comprehensive map pool consisting of 20 distinct random grid configurations for each of the obstacle density $\xi \in \{10\%, 20\%, 25\%, 30\%, 40\%\}$. The starting point and the ending point are (0,0) and (49,49) respectively. We have provided the code for generating the maps and the corresponding JSON files for the maps on GitHub (https://github.com/tyjcbzd/Enhanced-Q-DAP). For all randomly generated maps, we used the 'Breadth-First Search' algorithm to verify the reachability of the paths. The pseudocode for generating grid maps is shown below in Figure 4. All simulations were performed on the Windows 10 operating system, equipped with a 13th Gen Intel Core i5-13490F CPU and 32GB of RAM.

Algorithm 2 Random Grid Generation with Reachability Guarantee

Require: Map height H, width W, obstacle density ξ, start node s_{start}, goal node s_{goal}
Ensure: A valid grid map M where s_{goal} is reachable from s_{start}

function GENERATEGRID($H, W, \xi, s_{start}, s_{goal}$)
  for $attempt \leftarrow 1$ **to** $MaxTries$ **do**
    Initialize grid M of size $H \times W$ with zeros (free space)
    ▷ Randomly place obstacles
    for each cell (i, j) in M **do**
      $r \sim Uniform(0, 1)$
      if $r < \xi$ **then**
        $M[i][j] \leftarrow 1$ ▷ Mark as Obstacle
      end if
    end for
    ▷ Ensure endpoints are free
    $M[s_{start}] \leftarrow 0$
    $M[s_{goal}] \leftarrow 0$
    ▷ Verify reachability using BFS
    if ISREACHABLE(M, s_{start}, s_{goal}) **then**
      return M
    end if
  end for
  Error: "Failed to generate valid map within max tries"
end function

function ISREACHABLE($M, start, goal$)
  Initialize queue $Q \leftarrow \{start\}$
  Initialize $visited$ set $\leftarrow \{start\}$
  while Q is not empty **do**
    $current \leftarrow Q.pop()$
    if $current == goal$ **then**
      return True
    end if
    for each neighbor n of $current$ **do**
      if n is within bounds **and** $M[n] == 0$ **and** $n \notin visited$ **then**
        $visited.add(n)$
        $Q.push(n)$
      end if
    end for
  end while
  return False
end function

Fig. 4. Map generation algorithm with reachability verification

5.2. Parameter Setups. The algorithm operates within an $H \times W$ grid environment, where each state is represented by its coordinates (i, j) and the action space $A = \{up, down, left, right\}$ corresponds to displacement vectors $\Delta = [(-1,0), (1,0), (0,-1), (0,1)]$.

The values of Q-table $Q \in \mathbb{R}^{H \times W \times 4}$ maintain the expected discounted return for every state-action pair. Training is conducted in an episodic manner, comprising a total of N episodes, with each episode initialized at the starting state s_0 and constrained to a maximum of M steps. The learning rate α undergoes linear decay as a function of the episode index e, while the discount factor γ modulates the weighting of future rewards. To expedite convergence within finite time horizons, a secondary update is executed immediately following the standard temporal-difference update, leveraging the refreshed Q-values to propagate the value function more rapidly.

To ensure adaptability and convergence stability throughout the learning process, the learning rate is governed by a linear annealing schedule, defined as in Equation (10):

$$\alpha_e = max(\alpha_{min}, \alpha_0 \times (1 - \frac{e}{N})), , \tag{10}$$

where α_e denotes the learning rate at episode e, α_0 is the initial learning rate, α_{min} establishes a lower bound to prevent stagnation. This linear decay mechanism guarantees elevated learning rates during the early training phases to facilitate rapid initial adjustments to the Q-table.

Environment interaction is facilitated through a step function: given a current state s and action a, the subsequent state s' is computed. If the transition results in out-of-bounds movement or collision with an obstacle, the agent remains in the current state and incurs a penalty. Dijkstra's algorithm is utilized as a secondary validation step to confirm reachability before the training phase begins.

The experimental parameter configurations are detailed in Table 1. These specific values were determined through a hybrid approach: initial ranges were established according to previous studies, followed by empirical fine-tuning to ensure optimal performance within our experimental environment.

Table 1. Experimental parameters

Description	Symbol	Value
Number of episodes	N	1000
Max steps for each episode	M	10000
Discount rate	γ	0.95
Initial learning rate	α_0	0.625

Continuation of Table 1. Experimental parameters

Minimum learning rate	α_{min}	0.005
constant bias	β_0	0.1
The weight of distance	β_1	0.8
The weight of direction	β_2	0.7
Initial temperature	T_0	1.5
Minimum temperature	T_{min}	0.001
Decay rate	ρ	0.998

5.3. Evaluation Metrics. To provide a comprehensive assessment of the proposed algorithm's performance relative to baseline methods, the evaluation is conducted using three key metrics: Success Rate (SR), Average Path Length (APL) and Average Planning Time (APT). All quantitative results are computed over M independent experimental trials (where $M = 20$ for each density scenario, as detailed in Section 5.1). The detailed definitions and calculation formulas for these indicators are formally defined as follows:

Success Rate (SR): It evaluates the robustness and adaptability of the algorithm. It is calculated as the ratio of trials in which the agent successfully finds a collision-free path to the goal within the maximum allowable steps, relative to the total number of trials.

$$\mathrm{SR} = \frac{N_{success}}{M} \times 100\%, \tag{11}$$

where $N_{success}$ is the count of trials where a valid path was successfully generated.

Average Path Length (APL): APL quantifies the optimality of the generated path, indicating route efficiency. It is calculated as the mean sum of Euclidean distances between consecutive waypoints in the generated paths across all successful trials:

$$\mathrm{APL} = \frac{1}{N_{success}} \sum_{k=1}^{N_{success}} \left(\sum_{t=1}^{L_k - 1} \| n_{t+1}^{(k)} - n_t^{(k)} \|_2 \right), \tag{12}$$

where L_k is the total number of nodes in the k-th path, and $n_t^{(k)}$ represents the coordinate of the t-th grid node visited in the path. Here, $\|\cdot\|_2$ denotes the L_2 norm, and $\| n_{t+1}^{(k)} - n_t^{(k)} \|_2$ calculates the standard Euclidean distance between consecutive waypoints.

Average Planning Time (APT): APT measures the computational efficiency and real-time performance of the algorithm. It calculates the average wall-clock time consumed to generate a complete path:

$$\mathrm{APT} = \frac{1}{M}\sum_{k=1}^{M} T_k, \tag{13}$$

where T_k represents the total computation time (in seconds) for the k-th trial.

5.4. Quantitative Analysis. We conducted a comprehensive evaluation of our proposed method against the classical Q-learning algorithm [22], the optimized Q-learning algorithm (Optimized-Q) [10] and the recently introduced tailored reward and epsilon-greedy decay Q-learning (TRE-Q) algorithm [11].

Table 2 provides a comprehensive quantitative comparison of the proposed Enhanced-Q algorithm against Optimized Q-learning (O-QL), Classic Q-learning, and TRE-Q across obstacle densities ranging from 10% to 40%. In terms of robustness and adaptability, the performance of Classic Q-learning degrades significantly as environmental complexity increases, with its success rate dropping to 70% at a 40% obstacle density. This decline is primarily attributed to its "blind" exploration and initialization with uniform or random values, leading to slow convergence and susceptibility to local optima.

Table 2. Quantitative comparison of algorithm performance metrics under different obstacle densities

Method	Metrics	Obstacle density				
		10%	20%	25%	30%	40%
Classic-Q	SR	100%	100%	95%	85%	70%
	APT	2.884	4.478	5.275	6.072	9.996
	APL	103.2	110.8	122.2	138.6	175.6
TRE-Q	SR	100%	100%	100%	100%	100%
	APT	1.614	3.530	4.488	5.446	8.207
	APL	110.1	118.4	130.2	157.0	201.3
Optimized-Q	SR	100%	100%	100%	100%	100%
	APT	1.269	1.536	2.309	3.376	4.667
	APL	108.0	121.5	134.2	155.3	202.7
Enhanced-Q	SR	100%	100%	100%	100%	100%
	APT	1.549	2.228	2.918	3.607	5.793
	APL	100.4	108.6	118.3	135.7	170.5

In contrast, Optimized-Q, TRE-Q, and our proposed Enhanced-Q maintain a 100% success rate across all density levels. For Optimized-Q, this stability is achieved through a novel Q-table initialization method and Root-Mean-Square-Propagation-based learning rate adjustment, allowing the agent to adaptively tune the learning rate based on gradient changes. While Optimized-Q demonstrates high reliability, our Enhanced-Q performs even better by employing a potential-based reward function that provides continuous, gradient-like feedback, ensuring the agent consistently progresses toward the goal even when encountering deep obstacle traps.

Regarding computational efficiency, the execution time of our Enhanced-Q is slightly slower than that of Optimized-Q but remains significantly superior to TRE-Q and Classic Q-learning. In the most complex scenarios, Enhanced-Q outperforms the TRE-Q algorithm by approximately 29.4% and Classic Q-learning by about 42.0% in terms of Average Planning Time. This efficiency gain is directly attributable to the direction-sensitive Q-table initialization, which provides a "warm start" that substantially reduces the initial random wandering phase inherent in the "cold start" problem of Classic Q-learning. Although the TRE-Q algorithm improves upon the baseline via tailored rewards, its mechanism for penalizing state revisits introduces additional computational overhead during the search process, rendering it slightly slower than our streamlined approach.

Finally, the analysis of path optimality highlights a critical trade-off between exploration and exploitation among the baseline methods. Although the Optimized-Q algorithm achieves the fastest planning speed, it produces the longest path lengths in dense environments at 40% density, which is nearly identical to the poor path quality exhibited by the TRE-Q algorithm. This phenomenon suggests that while Optimized-Q's initialization improves efficiency, its precision significantly impacts final convergence, often leading the agent to follow sub-optimal paths dictated by Euclidean gradients rather than exploring shorter shortcuts. Similarly, the aggressive epsilon decay and revisit penalties in TRE-Q lead to a premature greedy tendency, where the agent settles for the first feasible but non-optimal path discovered. Conversely, Enhanced-Q achieves the shortest average path length across all test cases, outperforming Optimized-Q by approximately 16%. This demonstrates that our annealed Boltzmann exploration combined with UCB successfully balances global exploration and local optimization, ensuring that the agent maintains sufficient curiosity to refine its policy and discover global optimal shortcuts rather than stagnating in local sub-optimal solutions.

5.5. Comparison of Training Process and Visualization. In this section, we selected the 25% and 40% density scenarios as representative case

studies. The starting point and the ending point are selected randomly. These specific densities were chosen to vividly contrast the algorithm's learning behavior in a typical complex environment versus an extreme, near-limit environment, without cluttering the figures with intermediate curves.

Figures 5 and 6 illustrate the training dynamics in grid maps with obstacle densities of 25% and 40%, respectively. Subfigure (a) shows how the cumulative reward changes over the episode, and subfigure (b) shows how the number of steps changes over the episode. Figure 7 compares the final paths generated by the three algorithms side by side, using the true optimal path obtained by Dijkstra's algorithm as a reference.

It can be observed from both figures that during the early stages of training, our method requires a larger number of steps per episode compared to other algorithms, resulting in relatively lower cumulative rewards. This phenomenon arises from the adoption of a UCB-enhanced annealing Boltzmann exploration strategy during the exploration phase. The primary motivation behind this strategy is to prioritize extensive environmental sampling in order to maximize the discovery of novel state–action transitions. By doing so, the algorithm proactively broadens its exploration coverage in the early stages, thereby achieving a more comprehensive traversal of the state space and effectively mitigating the random wandering behavior commonly encountered in conventional Q-learning.

As the training progresses, our method exhibits a significant acceleration in convergence followed by a stable performance phase, reflecting a smooth transition from exploration to the exploitation of accumulated knowledge. This behavior is clearly manifested in the smooth evolution of both the reward and step curves. The underlying reason lies in the potential-based continuous reward shaping mechanism introduced in our framework, which provides dense feedback gradients that efficiently propagate value signals. Consequently, it alleviates the "dead zone" problem that is intrinsic to sparse reward settings, ensuring stable and efficient learning dynamics.

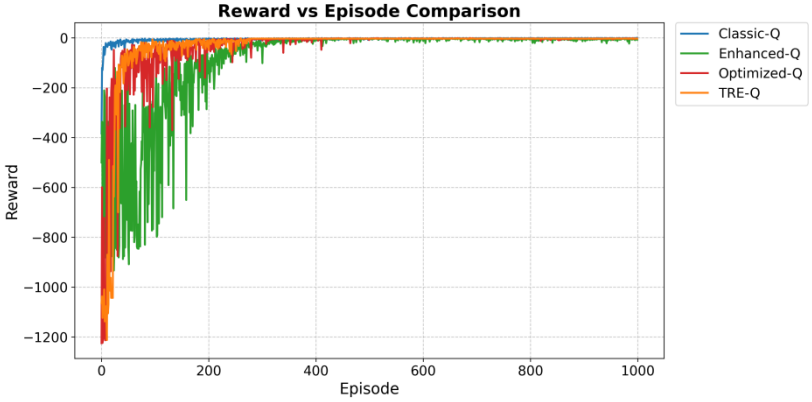
In contrast, while the TRE-Q algorithm achieves faster reward convergence through its aggressive ε-greedy decay strategy and state revisit penalty rewards, the path length of the TRE-Q algorithm deviates significantly from the optimal one. This limitation stems from its premature entry into the exploitation phase: once the algorithm triggers the pre-set exploitation mechanism, the learning process largely stagnates, overly relying on the incomplete environmental understanding formed in the early stages, thus "freezing" the policy updates. Similarly, the Optimized-Q algorithm achieves rapid initial convergence by utilizing a Q-table initialization strategy based on a simplified Euclidean distance heuristic to

the goal. While this prior knowledge provides strong directional guidance and accelerates the discovery of an initial path, it is fundamentally agnostic to the complex spatial distribution of obstacles. Consequently, the agent is biased toward following a "straight-line" gradient, which frequently leads to concave obstacle traps or circuitous routes that appear locally efficient but are globally sub-optimal. The sudden fluctuations observed mid-training coincide with the algorithm's transition from Boltzmann exploration to a more exploitative ε-greedy strategy upon reaching a fixed episode threshold. At this juncture, the agent often fails to rectify sub-optimal policy commitments formed during the early stages, as the reduced exploration prevents the discovery of shorter shortcuts. As a result, the paths generated by TRE-Q and Optimized-Q often remain longer, a point particularly evident in the extended step count and path deviation shown in Figure 7.

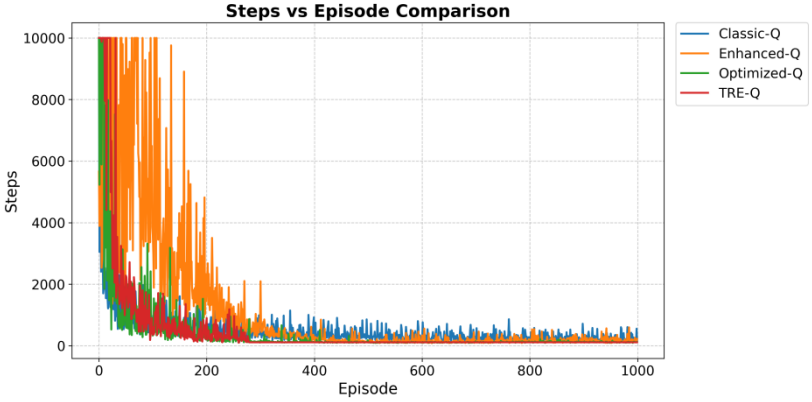
Meanwhile, while the classic Q-learning algorithm can gradually achieve a near-optimal policy after training, its convergence rate is significantly slower. In environments with high obstacle density, limited by its randomized, uniform exploration mechanism, the algorithm experiences a large number of invalid state transitions in the early stages, resulting in inefficient sample utilization. Moreover, in the later stages, affected by epsilon-greedy sampling, the step count curve still fluctuates, indicating a certain degree of policy instability.

Furthermore, as shown in Figure 7, our proposed method consistently generates trajectories of equal length to the Dijkstra optimal path in environments with varying obstacle densities, demonstrating its stable performance in achieving global optimality. In contrast, TRE-Q and Optimized-Q generate significantly longer paths, further demonstrating its limitations in global optimization capabilities, despite their faster reward convergence.

Overall, our method's high exploration intensity in the early stages results in a high number of steps and low rewards, but this is precisely the key to its efficient learning through sufficient exploration. In the later stages, the algorithm gradually shifts to leveraging acquired knowledge, and the training curve stabilizes, demonstrating a good balance between exploration and exploitation and efficient convergence.



a)



b)

Fig. 5. Training process in grid map with 25% obstacles: a) Variation of QL's reward across episodes; b) Variation of QL's required steps across episodes

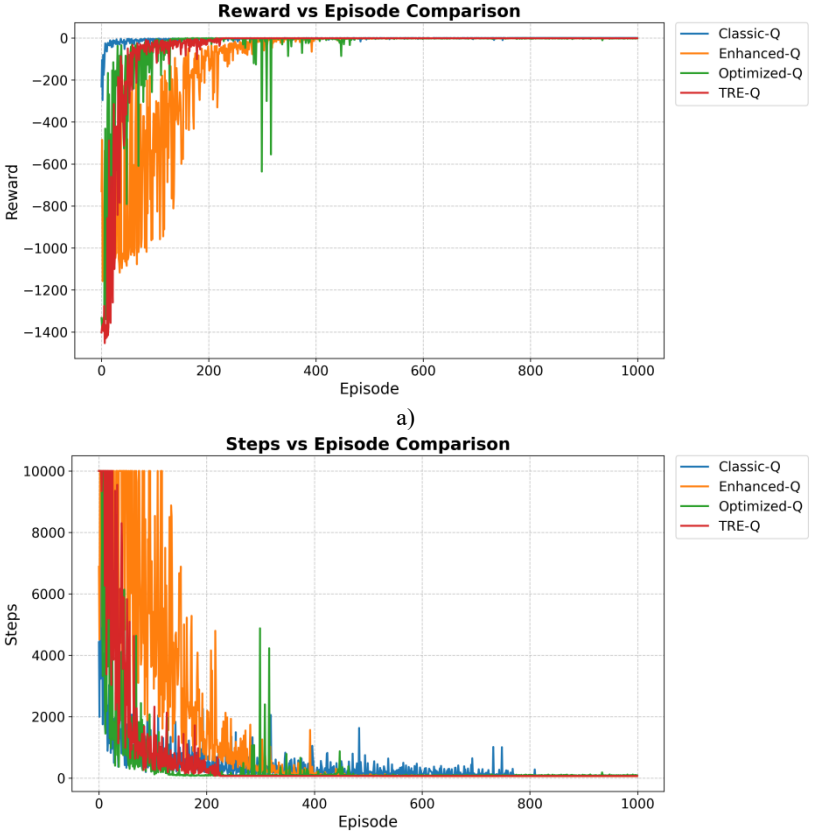
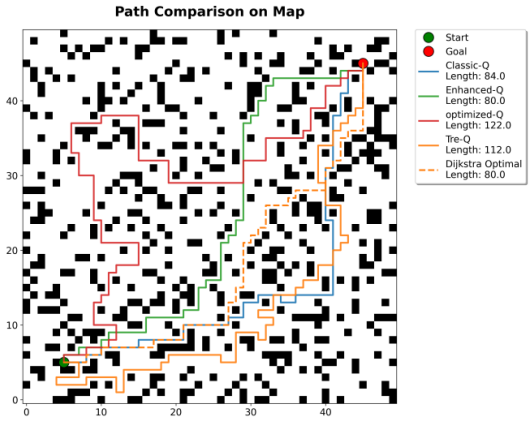
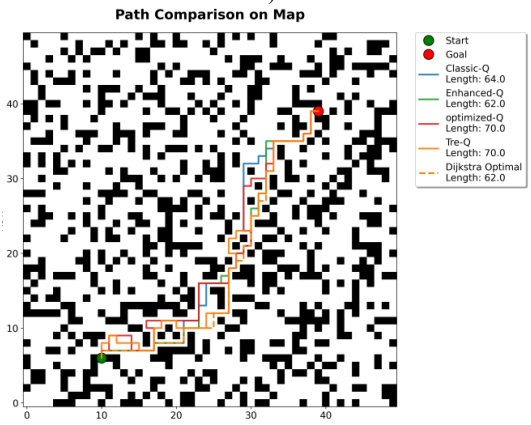


Fig. 6. Training process in grid map with 40% obstacles: a) Variation of QL's reward across episodes; b) Variation of QL's required steps across episodes



a)



b)

Fig. 7. Path comparison on grid map with different density of obstacles: a) planned path on grid map with 25% density of obstacles; b) planned path on grid map with 40% density of obstacles

6. Conclusion. This paper addresses the fundamental inefficiencies of traditional Q-learning in complex grid navigation, specifically the "cold start" problem, the lack of directional guidance in sparse-reward environments, and the instability of exploration strategies. To resolve these issues, we proposed an enhanced framework that integrates three targeted mechanisms: a direction-sensitive Q-table initialization to inject prior knowledge, an annealed Boltzmann exploration augmented with UCB to optimize the exploration-exploitation trade-off, and a potential-based

continuous reward shaping function to ensure dense feedback.

We validated the proposed framework through systematic experiments in grid environments with obstacle densities ranging from 10% to 40%, benchmarking against Classic Q-learning, Optimized-Q and TRE-Q algorithm. Specifically, by mitigating early-stage blind searching through direction-sensitive initialization, our method significantly accelerates learning, reducing the APT by approximately 42.0% compared to the baseline in the most complex scenarios; The potential-based reward mechanism effectively eliminates "dead zones" in dense obstacle fields, ensuring robust convergence. This results in a 100% SR even at 40% obstacle density, whereas Classic Q-learning fails to converge in 30% of trials due to gradient sparsity; the proposed UCB-augmented exploration ensures thorough state space coverage, generating paths that match the global optimality of Dijkstra's algorithm.

Future research will focus on extending this framework to continuous state spaces, developing adaptive exploration strategies for dynamic environments, and applying the algorithm to multi-agent collaborative tasks to enhance its scalability and generalization in real-world robotic applications.

References

1. Yang L., Li P., Qian S., Quan H., Miao J., Liu M., et al. Path Planning Technique for Mobile Robots: A Review. Machines. 2023. vol. 11. no. 10. 980 p. DOI: 10.3390/machines11100980.
2. Xu X., Zeng J., Zhao Y., Lu X. Research on global path planning algorithm for mobile robots based on improved A*. Expert Systems with Applications. 2024. vol. 243. 122922 p. DOI: 10.1016/j.eswa.2023.122922.
3. Reda M., Onsy A., Haikal A.Y., Ghanbari A. Path planning algorithms in the autonomous driving system: A comprehensive review. Robotics and Autonomous Systems. 2024. vol. 174. 104630 p. DOI: 10.1016/j.robot.2024.104630.
4. Zhang Y., Zhao W., Wang J., Yuan Y. Recent progress, challenges and future prospects of applied deep reinforcement learning: A practical perspective in path planning. Neurocomputing. 2024. vol. 608. 128423 p. DOI: 10.1016/j.neucom.2024.128423.
5. Zhao Y., Zhang Y., Wang S. A Review of Mobile Robot Path Planning Based on Deep Reinforcement Learning Algorithm. Journal of Physics: Conference Series. 2021. vol. 2138. 012011 p. DOI: 10.1088/1742-6596/2138/1/012011.
6. Khlif N., Nahla K., Safya B. Reinforcement learning with modified exploration strategy for mobile robot path planning. Robotica. 2023. vol. 41. no. 9. pp. 2688–2702. DOI: 10.1017/S0263574723000607.
7. Jaramillo-Martinez R., Chavero-Navarrete E., Ibarra-Perez T. Reinforcement-Learning-Based Path Planning: A Reward Function Strategy. Applied Sciences. 2024. vol. 14. no. 17. 7654 p. DOI: 10.3390/app14177654.
8. Zhang Y., Cai P., Pan C., Zhang S. Multi-agent deep reinforcement learning-based cooperative spectrum sensing with upper confidence bound exploration. IEEE Access. 2019. vol. 7. pp. 118898–118906. DOI: 10.1109/ACCESS.2019.2936147.
9. Gao P., Liu Z., Wu Z., Wang D. A global path planning algorithm for robots using reinforcement learning. Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO). 2019. pp. 1693–1698.

10. Zhou Q., Lian Y., Wu J., Zhu M., Wang H., Cao J. An optimized Q-Learning algorithm for mobile robot local path planning. Knowledge-Based Systems. 2024. vol. 286. pp. 111400. DOI: 10.1016/j.knosys.2024.111400.
11. Ben-Akka M., Tanougast C., Diou C. Novel design of reward and epsilon-greedy decay strategy tailored for Q-learning in optimizing local mobile robot path planning. Knowledge-Based Systems. 2025. vol. 324. 113836 p. DOI: 10.1016/j.knosys.2024.113836.
12. Ma T., Lyu J., Yang J., Xi R., Li Y., An J., Li C. CLSQL: Improved Q-Learning Algorithm Based on Continuous Local Search Policy for Mobile Robot Path Planning. Sensors. 2022. vol. 22. no. 15. 5910 p. DOI: 10.3390/s22155910.
13. Huang J., Zhang Z., Ruan X. An Improved Dyna-Q Algorithm Inspired by the Forward Prediction Mechanism in the Rat Brain for Mobile Robot Path Planning. Biomimetics. 2024. vol. 9. no. 6. 315 p. DOI: 10.3390/biomimetics9060315.
14. Hidayat H., Buono A., Priandana K., Wahjuni S. Modified Q-Learning Algorithm for Mobile Robot Path Planning Variation using Motivation Model. Journal of Robotics and Control (JRC). 2023. vol. 4. no. 5. pp. 696–707. DOI: 10.18196/jrc.v4i5.18777.
15. Wang H., Jing J., Wang Q., He H., Qi X., Lou R. ETQ-learning: an improved Q-learning algorithm for path planning. Intelligent Service Robotics. 2024. vol. 17. no. 4. pp. 915–929. DOI: 10.1007/s11370-024-00544-3.
16. Fallooh N., Sadiq A., Abbas E., Hashim I. Robot path planning using enhanced Q-learning algorithm based on single parameter. Engineering and Technology Journal. 2025. vol. 43. no. 2. pp. 1–15. DOI: 10.30684/etj.2024.154230.1831.
17. Wang Y., Xie Y., Xu D., Shi J., Fang S., Gui W. Heuristic dense reward shaping for learning-based map-free navigation of industrial automatic mobile robots. ISA Transactions. 2025. vol. 156. pp. 579–596. DOI: 10.1016/j.isatra.2024.10.026.
18. Zhang T., Mo H. Reinforcement learning for robot research: A comprehensive review and open issues. International Journal of Advanced Robotic Systems. 2021. vol. 18. no. 3. pp. 1–22. DOI: 10.1177/17298814211007305.
19. Shi Z., Wang K., Zhang J. Improved reinforcement learning path planning algorithm integrating prior knowledge. PLoS ONE. 2023. vol. 18. no. 5. e0285129 p. DOI: 10.1371/journal.pone.0284942.
20. Miranda V., Neto A.A., Freitas G., Mozelli L.A. Generalization in Deep Reinforcement Learning for Robotic Navigation by Reward Shaping. IEEE Transactions on Industrial Electronics. 2023. vol. 71. no. 6. pp. 6013–6020. DOI: 10.1109/TIE.2023.3290244.
21. Lv Q., Hao G., Huang Z., Li B., Fu D., Zhao H., et al. Localized Path Planning for Mobile Robots Based on a Subarea-Artificial Potential Field Model. Sensors. 2024. vol. 24. no. 11. 3604 p. DOI: 10.3390/s24113604.
22. Clifton J., Laber E. Q-Learning: Theory and Applications. Annual Review of Statistics and Its Application. 2020. vol. 7. pp. 279–301. DOI: 10.1146/annurev-statistics-031219-041220.

Tang Yi — Ph.D., Student, Department of Information Technology of Automated Systems (ITAS), Belarusian State University of Informatics and Radioelectronics (BSUIR). Research interests: object detection and tracking, path planning. The number of publications — 7. tangyijcb@163.com; 16, Petrusya Brovki St., 220013, Minsk, Belarus; office phone: +375(25)764-4191.

German Yuliya — Ph.D., Associate professor, Department of Information Technology of Automated Systems (ITAS), Belarusian State University of Informatics and Radioelectronics (BSUIR). Research interests: algorithm design, combinatorial optimization, decision making. The number of publications — 50. jgerman@bsuir.by; 6, Petrusya Brovki St., 220013, Minsk, Belarus; office phone: +375(17)293-8823.

УДК 004.021 **DOI** 10.15622/ia.25.4.9

И. Тан, Ю.О. Герман

УЛУЧШЕННОЕ Q-ОБУЧЕНИЕ ДЛЯ НАВИГАЦИИ ПО СЕТКЕ, ИСПОЛЬЗУЮЩЕЕ АПРИОРНЫЕ ПРИОРИТЕТЫ НАПРАВЛЕНИЙ, ВЫБОР МАРШРУТОВ МЕТОДОМ ОТЖИГА И ОПТИМИЗАЦИЮ ВОЗНАГРАЖДЕНИЙ НА ОСНОВЕ ПОТЕНЦИАЛЬНОЙ ДОСТИЖИМОСТИ ЦЕЛИ

Тан И., Герман Ю.О. **Улучшенное Q-обучение для навигации по сетке, использующее априорные приоритеты направлений, выбор маршрутов методом отжига и оптимизацию вознаграждений на основе потенциальной достижимости цели.**

Аннотация. В статье представлен улучшенный вариант Q-обучения для навигации по сетке при наличии препятствий. Основываясь на стандартном табличном Q-обучении, предложены три ключевых улучшения: инициализация Q-таблицы с учетом направления к цели, дающее согласование с целью без необходимости предварительного знания о препятствиях; модифицированная стратегия отжига Больцмана, расширенная включением верхнего доверительного предела «энергии» системы для более сбалансированного и адаптивного отбора направления движения; функция расчета вознаграждения на основе потенциала, дающая более тесную обратную связь для ускорения процесса обучения. Отмеченные улучшения повышают эффективность Q-обучения в условиях относительно редких случаев вознаграждения при достижении цели, что связано с неэффективным исследованием области поиска и медленным формированием значений Q-функции. Экспериментальная проверка на случайно сгенерированных сетках, показала, что предложенный в статье подход обеспечивает более успешные конечные результаты, связанные с отысканием более коротких путей к цели и более быстрой сходимостью в сравнении с известными базовыми методами, такими как стандартное Q-обучение и его вариантами. Описанный в статье подход обеспечивает общую модельно-независимую природу Q-обучения и его достаточно высокую эффективность, что важно для практических приложений в робототехнике и планировании маршрутов движения.

Ключевые слова: обучение с подкреплением, инициализация таблицы Q, планирование маршрута, компромисс между разведкой и эксплуатацией, постоянное вознаграждение, сетчатая карта.

Литература

1. Yang L., Li P., Qian S., Quan H., Miao J., Liu M., et al. Path Planning Technique for Mobile Robots: A Review // Machines. 2023. vol. 11. no. 10. 980 p. DOI: 10.3390/machines11100980.
2. Xu X., Zeng J., Zhao Y., Lu X. Research on global path planning algorithm for mobile robots based on improved A* // Expert Systems with Applications. 2024. vol. 243. 122922 p. DOI: 10.1016/j.eswa.2023.122922.
3. Reda M., Onsy A., Haikal A.Y., Ghanbari A. Path planning algorithms in the autonomous driving system: A comprehensive review // Robotics and Autonomous Systems. 2024. vol. 174. 104630 p. DOI: 10.1016/j.robot.2024.104630.
4. Zhang Y., Zhao W., Wang J., Yuan Y. Recent progress, challenges and future prospects of applied deep reinforcement learning: A practical perspective in path planning // Neurocomputing. 2024. vol. 608. 128423 p. DOI: 10.1016/j.neucom.2024.128423.

5. Zhao Y., Zhang Y., Wang S. A Review of Mobile Robot Path Planning Based on Deep Reinforcement Learning Algorithm // Journal of Physics: Conference Series. 2021. vol. 2138. 012011 p. DOI: 10.1088/1742-6596/2138/1/012011.
6. Khlif N., Nahla K., Safya B. Reinforcement learning with modified exploration strategy for mobile robot path planning // Robotica. 2023. vol. 41. no. 9. pp. 2688–2702. DOI: 10.1017/S0263574723000607.
7. Jaramillo-Martinez R., Chavero-Navarrete E., Ibarra-Perez T. Reinforcement-Learning-Based Path Planning: A Reward Function Strategy // Applied Sciences. 2024. vol. 14. no. 17. 7654 p. DOI: 10.3390/app14177654.
8. Zhang Y., Cai P., Pan C., Zhang S. Multi-agent deep reinforcement learning-based cooperative spectrum sensing with upper confidence bound exploration // IEEE Access. 2019. vol. 7. pp. 118898–118906. DOI: 10.1109/ACCESS.2019.2936147.
9. Gao P., Liu Z., Wu Z., Wang D. A global path planning algorithm for robots using reinforcement learning // Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO). 2019. pp. 1693–1698.
10. Zhou Q., Lian Y., Wu J., Zhu M., Wang H., Cao J. An optimized Q-Learning algorithm for mobile robot local path planning // Knowledge-Based Systems. 2024. vol. 286. pp. 111400. DOI: 10.1016/j.knosys.2024.111400.
11. Ben-Akka M., Tanougast C., Diou C. Novel design of reward and epsilon-greedy decay strategy tailored for Q-learning in optimizing local mobile robot path planning // Knowledge-Based Systems. 2025. vol. 324. 113836 p. DOI: 10.1016/j.knosys.2024.113836.
12. Ma T., Lyu J., Yang J., Xi R., Li Y., An J., Li C. CLSQL: Improved Q-Learning Algorithm Based on Continuous Local Search Policy for Mobile Robot Path Planning // Sensors. 2022. vol. 22. no. 15. 5910 p. DOI: 10.3390/s22155910.
13. Huang J., Zhang Z., Ruan X. An Improved Dyna-Q Algorithm Inspired by the Forward Prediction Mechanism in the Rat Brain for Mobile Robot Path Planning // Biomimetics. 2024. vol. 9. no. 6. 315 p. DOI: 10.3390/biomimetics9060315.
14. Hidayat H., Buono A., Priandana K., Wahjuni S. Modified Q-Learning Algorithm for Mobile Robot Path Planning Variation using Motivation Model // Journal of Robotics and Control (JRC). 2023. vol. 4. no. 5. pp. 696–707. DOI: 10.18196/jrc.v4i5.18777.
15. Wang H., Jing J., Wang Q., He H., Qi X., Lou R. ETQ-learning: an improved Q-learning algorithm for path planning // Intelligent Service Robotics. 2024. vol. 17. no. 4. pp. 915–929. DOI: 10.1007/s11370-024-00544-3.
16. Fallooh N., Sadiq A., Abbas E., Hashim I. Robot path planning using enhanced Q-learning algorithm based on single parameter // Engineering and Technology Journal. 2025. vol. 43. no. 2. pp. 1–15. DOI: 10.30684/etj.2024.154230.1831.
17. Wang Y., Xie Y., Xu D., Shi J., Fang S., Gui W. Heuristic dense reward shaping for learning-based map-free navigation of industrial automatic mobile robots // ISA Transactions. 2025. vol. 156. pp. 579–596. DOI: 10.1016/j.isatra.2024.10.026.
18. Zhang T., Mo H. Reinforcement learning for robot research: A comprehensive review and open issues // International Journal of Advanced Robotic Systems. 2021. vol. 18. no. 3. pp. 1–22. DOI: 10.1177/17298814211007305.
19. Shi Z., Wang K., Zhang J. Improved reinforcement learning path planning algorithm integrating prior knowledge // PLoS ONE. 2023. vol. 18. no. 5. e0285129 p. DOI: 10.1371/journal.pone.0284942.
20. Miranda V., Neto A.A., Freitas G., Mozelli L.A. Generalization in Deep Reinforcement Learning for Robotic Navigation by Reward Shaping // IEEE Transactions on Industrial Electronics. 2023. vol. 71. no. 6. pp. 6013–6020. DOI: 10.1109/TIE.2023.3290244.

21. Lv Q., Hao G., Huang Z., Li B., Fu D., Zhao H., et al. Localized Path Planning for Mobile Robots Based on a Subarea-Artificial Potential Field Model // Sensors. 2024. vol. 24. no. 11. 3604 p. DOI: 10.3390/s24113604.
22. Clifton J., Laber E. Q-Learning: Theory and Applications // Annual Review of Statistics and Its Application. 2020. vol. 7. pp. 279–301. DOI: 10.1146/annurev-statistics-031219-041220.

Тан И — канд. техн. наук, студент, кафедра информационных технологий автоматизированных систем (ИТАС), Белорусский государственный университет информатики и радиоэлектроники (БГУИР). Область научных интересов: обнаружение и отслеживание объектов, планирование траектории движения. Число научных публикаций — 7. tangyijcb@163.com; Улица Петруся Бровки, 16, 220013, Минск, Беларусь; р.т.: +375(25)764-4191.

Герман Юлия Олеговна — канд. техн. наук, доцент, кафедра информационных технологий автоматизированных систем (ИТАС), Белорусский государственный университет информатики и радиоэлектроники (БГУИР). Область научных интересов: разработка алгоритмов, комбинаторная оптимизация, принятие решений. Число научных публикаций — 50. jgerman@bsuir.by; Улица Петруся Бровки, 6, 220013, Минск, Беларусь; р.т.: +375(17)293-8823.