2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Behavior Cloning-Enhanced Deep Reinforcement Learning for Robot Navigation via Dynamic Reward and Adaptive Replanning

Jianning Chi[1], Fusheng Li[2], Wenjun Zhang[3] and Yongming Yang*

Abstract— **Deep reinforcement learning (DRL) is a core technology for mobile robot navigation in diverse environments, yet existing Behavior Cloning (BC)-enhanced DRL methods suffer two critical challenges: fixed imitation constraints suppress autonomous exploration in late training stages despite stabilizing early learning, and goal-obstacle avoidance task conflicts impede robust action selection during navigation. To address these issues, this paper proposes an Adaptive Strategy Deep Reinforcement Learning (ADRL) method, which reformulates BC as a progressively released transitional constraint and builds a stage-aware transition framework for robot navigation. Specifically, ADRL dynamically fuses Twin Delayed Deep Deterministic Policy Gradient (TD3) with BC via a value-driven imitation scheduling mechanism, which adaptively modulates the expert-online data mixing ratio and BC regularization strength based on critic feedback to accelerate convergence and realize a smooth shift from imitation-dominant to exploration-driven learning. A phase-aligned dynamic weight composite reward function is designed, which embeds motion constraints and stage-aware priority adjustment to mitigate reward sparsity and align learning objectives with policy maturity. Additionally, a lightweight adaptive replanning mechanism is developed as an evaluation stabilizer, which generates obstacle-avoiding waypoints by obstacle density when the robot stagnates, resolving goal-obstacle avoidance conflicts without altering the transition-centric learning objective. Multi-scenario experimental results demonstrate that ADRL outperforms state-of-the-art methods in training convergence speed, navigation success rate and robustness under identical training budgets. This method provides a principled integration strategy for imitation and reinforcement learning in robot navigation, and lays a solid foundation for building efficient and reliable autonomous navigation systems.**

I. INTRODUCTION

Navigation is a core task in autonomous mobile robot research, with learning-based methods represented by deep reinforcement learning (DRL) demonstrating superior adaptability in complex dynamic environments via end-to-end perception and decision-making [1]–[3]. Nevertheless, DRL faces inherent training inefficiency caused by random initial parameters [4], and practical deployment is further hindered by the fundamental dilemma of fixed BC-enhanced strategies: BC effectively stabilizes early-stage DRL training via expert demonstrations but excessively suppresses autonomous exploration in late stages, leading to policy solidification and limited asymptotic performance. Meanwhile, goal-oriented and obstacle avoidance task conflicts during navigation easily result in robot stagnation and local optima, degrading navigation robustness. To tackle these issues, this paper proposes an Adaptive Strategy Deep Reinforcement Learning (ADRL) method, which reinterprets BC as a progressively released transitional constraint rather than a static regularizer and constructs a stage-aware transition framework for robot navigation. The framework dynamically couples imitation scheduling, reward weighting and adaptive replanning, realizing a smooth shift from expert-guided imitation learning to autonomous reinforcement learning, while resolving navigation task conflicts and accelerating training convergence. The main contributions of this paper are as follows:

- We propose a value-driven stage-aware imitation scheduling mechanism, which adaptively modulates the expert-online data mixing ratio and BC regularization strength based on critic Q-value statistics. This mechanism enables a smooth transition from imitation-dominant early training to exploration-driven late training, effectively alleviating over-regularization and policy solidification caused by fixed BC constraints.
- We design a phase-aligned dynamic weight composite constraint reward function, which constructs a stage indicator from critic feedback to adjust sub-reward priorities adaptively. The function embeds motion constraints and state-aware guidance, mitigates reward sparsity in DRL training, and ensures consistent learning pressure by aligning reward objectives with policy maturity across different training phases.
- We develop a lightweight adaptive replanning mechanism as an evaluation stabilizer, which triggers waypoint generation based on multi-dimensional stagnation conditions and obstacle density. This mechanism resolves goal-obstacle avoidance task conflicts during navigation, ensures fair and stable evaluation of the transition-centric policy, and does not alter the core learning objective of ADRL.

II. RELATED WORKS

In recent years, deep reinforcement learning (DRL) has been increasingly applied to robot path planning to address the limitations of traditional algorithms in highly dynamic and unpredictable environments [5]–[8]. As an end-to-end

* represent corresponding author
[1]Jianning Chi and Qiang zou are with the Faculty of Robot Science and Engineering, Northeastern University, Shenyang, 110169, China chijianning@mail.neu.edu.cn
[2]Fusheng Li is with College of Information Science and Engineering, Northeastern University, Shenyang, 110169, China 2370904@stu.neu.edu.cn
[3]Wenjun Zhang is with Department of Biomedical Engineering, University of Saskatchewan,Saskatoon, S7V5A9,, Canada Chris.zhang@usask.ca
*Yongming Yang is with State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China yangyongming@sia.cn

approach, DRL-based motion planning outperforms supervised learning-based methods in unknown environments [9], [10], enabling mobile robots to learn optimal navigation strategies via trial-and-error interaction with dynamic surroundings. Representative DRL algorithms include the value-based DQN [11] and its enhanced version DDQN [12], which are limited to discrete action spaces; by contrast, Actor-Critic framework-based methods (DDPG [13], TD3 [14] et al.) excel in continuous action spaces [15], with TD3 addressing DDPG's Q-value overestimation and achieving promising navigation results [16], and DDPG performing well in mobile robot transportation and search tasks [17]. Despite these advances, DRL-based robot training suffers from slow convergence, a key barrier to practical deployment, prompting researchers to integrate Behavior Cloning (BC) with RL for efficient sample utilization [18]. A typical method is TD3-BC [19], yet it relies on fixed-scenario datasets with limited generalization [20] and is only applicable to offline RL, making it unfit for real dynamic navigation scenarios; more importantly, traditional fixed BC-enhanced DRL faces a core stabilization-exploration dilemma—BC stabilizes early training via expert demonstrations but suppresses late-stage autonomous exploration, leading to policy solidification. While recent transition-centric studies have addressed this dilemma by reinterpreting BC as a progressively released transitional constraint and adopting value-driven imitation scheduling, such works lack navigation-specific optimization. To overcome these challenges, we extend the transition perspective to mobile robot navigation and propose a novel approach that dynamically adjusts expert-online dataset proportions and BC regularization via two buffers and a selective BC regularizer, coupled with a lightweight navigation-tailored adaptive replanning mechanism as an evaluation stabilizer.

III. ADAPTIVE STRATEGY DEEP REINFORCEMENT LEARNING

A. Value-driven Stage-aware Adaptive Training

To accelerate training convergence and resolve the over-regularization issue of fixed BC constraints, an innovative value-driven stage-aware adaptive dynamic training mechanism is designed, which comprises a policy-adaptive replay buffer and a progressively released BC regularization term. This mechanism takes the critic Q-value statistic as the proxy for policy maturity, adaptively modulates the fusion ratio of online interaction data and expert demonstration data, and gradually relaxes BC constraints, thus realizing a smooth transition from imitation-dominant to exploration-driven learning.

Robot environmental interaction experience is stored as online data D_1, and expert demonstration trajectories as pre-collected expert data D_2. A mixed experience replay buffer D is constructed by adaptively fusing D_1 and D_2 based on critic feedback, defined as:

$$D = (s_t, a_t, r_t, s_{t+1}) \tag{1}$$

A critic Q-value statistic $\bar{Q}(t)$ is first calculated to characterize the training progress and policy maturity, defined as the expected value of critic outputs over the mixed buffer:

$$\bar{Q}(t) = \mathbb{E}_{(s,a)\sim D}\left[Q_\theta(s,a)\right] \tag{2}$$

where $Q_\theta(s,a)$ denotes the output of the critic network with parameters θ.

An expert reliance coefficient $\alpha(t) \in [0,1]$ is designed as a decreasing function of $\bar{Q}(t)$ to tune the fusion ratio of D_1 and D_2, implemented via the sigmoid function to ensure smooth attenuation:

$$\alpha(t) = \sigma\big(-k_\alpha \cdot \bar{Q}(t) + b_\alpha\big) \tag{3}$$

$$D = (1-\alpha(t))D_1 + \alpha(t)D_2 \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function, k_α, b_α are constant coefficients that control the attenuation speed and initial expert reliance. $\alpha(t) \to 1$ in early training (high expert data ratio for stabilization) and $\alpha(t) \to 0$ in late training (high online data ratio for autonomous exploration).

A progressively released BC regularization term is integrated into the actor network, with its weight factor $\lambda_{\text{BC}}(t)$ designed as an exponential decay function of $\bar{Q}(t)$ to realize adaptive relaxation of imitation constraints:

$$\lambda_{\text{BC}}(t) = \lambda_{\max} \exp\left(-k_q \cdot \bar{Q}(t)\right) \tag{5}$$

where $\lambda_{\max}$ is the maximum BC regularization weight (initial imitation strength), k_q is the decay coefficient. The actor objective function is then formulated as:

$$\begin{aligned}\pi_\phi = \underset{\pi_\phi}{\text{argmax}}\Big\{ & \mathbb{E}_{(s,a)\sim D}\left[Q_\theta(s,\pi_\phi(s))\right] \\ & -\lambda_{\text{BC}}(t)\cdot \mathbb{E}_{(s,a_e)\sim D_2}\left[\left\|\pi_\phi(s)-a_e\right\|_2^2\right]\Big\}\end{aligned} \tag{6}$$

where $\pi_\phi(s)$ denotes the deterministic actor network with parameters ϕ, a_e is the expert action from D_2. This formulation enforces strong BC regularization in early training to stabilize policy learning and progressively relaxes imitation constraints as $\bar{Q}(t)$ increases, thus avoiding the suppression of autonomous exploration by fixed BC constraints.

Conceptually, the training process driven by this mechanism contains three implicit phases: (i) Stabilization phase ($\bar{Q}(t) \ll 1$): high $\alpha(t)$ and $\lambda_{\text{BC}}(t)$ to yield safe initial behaviors via expert guidance; (ii) Transition phase ($\bar{Q}(t)$ increases moderately): synchronous attenuation of $\alpha(t)$ and $\lambda_{\text{BC}}(t)$, with reinforcement learning signals gradually becoming dominant; (iii) Autonomy phase ($\bar{Q}(t) \approx 1$): low $\alpha(t)$ and $\lambda_{\text{BC}}(t)$, with reinforcement learning dominating and BC serving as a weak regularizer for policy fine-tuning.

B. Phase-aligned Dynamic Reward Matrix

To address the training difficulties caused by sparse rewards and align learning objectives with policy maturity across different training phases, a phase-aligned dynamic weight composite constraint reward function is proposed. This function extracts core environmental state features and constructs a stage indicator from critic Q-value statistics to adjust sub-reward priorities adaptively, ensuring that the

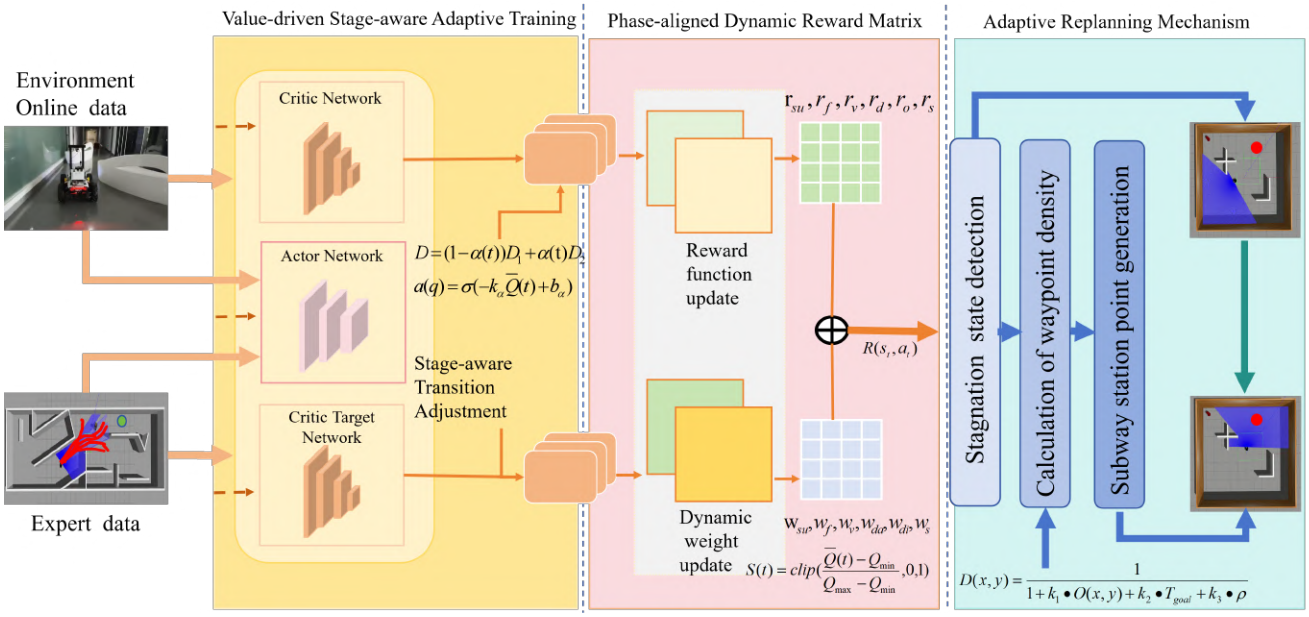


Fig. 1. ADRL implementation: Overall framework of the proposed ADRL method. It comprises three core modules: (1) Value-driven Stage-aware Adaptive Training: based on the critic Q-value statistic $\bar{Q}(t)$ (policy maturity proxy), it adaptively modulates expert data reliance $\alpha(t)$ and BC regularization strength $\lambda_{BC}(t)$ for a smooth imitation-to-reinforcement transition; (2) Phase-aligned Dynamic Reward Matrix: coupled with stage indicator $S(t)$ (derived from $\bar{Q}(t)$), it adjusts reward weights to align learning objectives with policy maturity; (3) Adaptive Replanning (Evaluation Stabilizer) : it resolves navigation conflicts with phase-adaptive thresholds, ensuring fair policy evaluation without altering the core transition objective.

reward signal responds to both scenario changes and policy maturity, and thus providing consistent learning pressure for the transition-centric training framework.

1) Composition and calculation of sub-rewards: The reward function is composed of six sub-rewards targeting task success/failure, motion state, goal approaching, safe obstacle avoidance and stagnation prevention, with optimized physical interpretability and reward/penalty gradient smoothness.

Success Reward r_{su} (positive incentive for target arrival) is calculated as:

$$r_{su} = \begin{cases} 200, & \|\boldsymbol{p}_g - \boldsymbol{p}_r\|_2 < d_g \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $\boldsymbol{p}_g = (x_g, y_g)$ and $\boldsymbol{p}_r = (x_r, y_r)$ denote the 2D coordinates of target and robot, respectively, and d_g is the target arrival threshold.

Failure Penalty r_f (negative penalty for collision/timeout) is expressed as:

$$r_f = \begin{cases} -200, & \min \|\boldsymbol{p}_r - \boldsymbol{p}_{o_i}\|_2 < d_r \vee t > t_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $\boldsymbol{p}_{o_i} = (x_{o_i}, y_{o_i})$ is the i-th obstacle coordinate, d_r is collision threshold, t is current task time, and $t_{\max}$ is timeout threshold.

Motion Smoothness Reward r_v (encourage uniform motion, suppress sharp turns) is:

$$r_v = \frac{v}{v_{\max}} - 0.8 \cdot \frac{||}{\max} \tag{9}$$

where $v, v_{\max}$ are real-time/max linear velocity, and ,max are real-time/max angular velocity of the robot.

Goal Approaching Reward r_d (associate heading adjustment with target direction) is:

$$r_d = \|\Delta \boldsymbol{p}_r\|_2 + 0.5 \cdot e^{-2.0|\Delta_{rg}|} \cdot |\Delta_r| \tag{10}$$

where $\Delta \boldsymbol{p}_r = \boldsymbol{p}_r^t - \boldsymbol{p}_r^{t-1}$ (robot step displacement), $\Delta_{rg} =_r -_{rg}$ (heading deviation to target), and $\Delta_r =_r^t -_r^{t-1}$ (step heading change).

Obstacle Avoidance Penalty r_o (gradient penalty for obstacle proximity) is:

$$r_o = \begin{cases} -\frac{5.0}{d_{\text{ob}} - d_{\text{safe}}}, & d_{\text{ob}} < d_{\text{safe}} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where d_{ob} is real-time distance to nearest obstacle, and d_{safe} is safe obstacle avoidance threshold.

Stagnation Penalty r_s (prevent long-term stagnation, anti-noise) is:

$$r_s = \begin{cases} -3.0, & \frac{1}{5}\sum_{k=0}^{4} \|\Delta \boldsymbol{p}_r^{t-k}\|_2 < d_m \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where $\Delta \boldsymbol{p}_r^{t-k} = \boldsymbol{p}_r^{t-k} - \boldsymbol{p}_r^{t-k-1}$ (step displacement at $t-k$), and d_m is stagnation displacement threshold.

2) Phase-Aware Dynamic Weight Matrix Update: Three core environmental state features are first extracted: obstacle threat degree T_{obs}, goal proximity degree T_{goal} and motion stagnation degree T_{stop}. A stage indicator $S(t)$ is then constructed from the critic Q-value statistic $\bar{Q}(t)$ to characterize the training phase of ADRL, defined as:

$$S(t) = \text{clip}\left(\frac{\bar{Q}(t) - Q_{\min}}{Q_{\max} - Q_{\min}}, 0, 1\right) \tag{13}$$

where $Q_{\min}$ and $Q_{\max}$ are the minimum and maximum critic Q-value statistics estimated from training data, $\mathrm{clip}(\cdot)$ constrains $S(t) \in [0,1]$. $S(t) \to 0$ indicates the stabilization phase, and $S(t) \to 1$ indicates the autonomy phase.

Adaptive weight update formulas for all sub-rewards are designed by coupling the environmental state features and the stage indicator $S(t)$, realizing phase-aligned reward priority adjustment. Taking the distance and angle reward weight as an example:

$$w_{di} = 0.6\,(1 - T_{obs})\,T_{goal}(1 - S(t)) + 0.1 \cdot S(t) \tag{14}$$

This formulation prioritizes goal-directed movement guidance in the early stabilization phase ($S(t) \to 0$) and appropriately enhances the flexibility of autonomous exploration in the late autonomy phase ($S(t) \to 1$). For the parking penalty weight, the stage indicator is integrated to reduce unnecessary intervention in late training:

$$w_s = 0.8 T_{stop}(1 - S(t)) + 0.05 \tag{15}$$

The initially calculated weights are normalized to avoid overflow and ensure rationality:

$$w_i' = \frac{w_i}{\sum_{j=1}^{6} w_j} \tag{16}$$

where w_i' is the final normalized weight, satisfying $0 \le w_i' \le 1$ and $\sum_{i=1}^{6} w_i' = 1$.

3) Overall Phase-Aligned Reward Function: The final reward function is the weighted sum of each sub-reward and its corresponding phase-aligned weight:

$$R\,(s_t, a_t) = \sum_{i=1}^{6} w_i' \cdot r_i \tag{17}$$

This function effectively alleviates the sparse reward problem, retains the comprehensiveness of motion constraints, and ensures that the reward objective matches the policy maturity in different training phases, thus boosting the smoothness of the imitation-to-reinforcement transition.

C. Adaptive Replanning Mechanism

Mobile robots often face stagnation in navigation due to goal-obstacle avoidance conflicts, which may bias policy evaluation by dominating metrics with pathological "stuck" episodes. To ensure fair and stable evaluation of the ADRL framework—without introducing extra policy optimization modules—we design a lightweight adaptive replanning mechanism as an evaluation stabilizer. It generates obstacle-avoiding, target-directed sub-waypoints upon valid stagnation detection, resolving navigation conflicts while preserving ADRL's core imitation-to-reinforcement transition objective. Its triggering thresholds are moderately coupled with the stage indicator $S(t)$ (from critic Q-value statistics) for phase-adaptive adjustment, reducing unnecessary intervention in late-stage autonomous exploration.

1) Multi-dimensional Adaptive Triggering Condition: A multi-dimensional condition distinguishes valid stagnation from normal low-speed navigation, activating replanning only when the robot is trapped:

$$\text{Trigger} = \begin{cases} 1, & \Delta p_{t-n:t} < d_m \cap \Delta d_{obs,t-n:t} < \delta \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where $\Delta p_{t-n:t}$ is the cumulative displacement over n steps, v_t the real-time linear velocity, $\Delta d_{obs,t-n:t}$ the nearest obstacle distance variation, and d_m, v_{min}, δ are basic thresholds. All thresholds are first adapted to the obstacle threat degree T_{obs} (taking d_m as an example):

$$d_m = d_{m0} \cdot (1 + \alpha \cdot T_{obs}) \tag{19}$$

with d_{m0} the basic displacement threshold and α the environmental adaptation coefficient. To align with ADRL's stage-aware framework, the adapted d_m is further modulated by $S(t)$:

$$d_m' = d_m \cdot (1 + \beta \cdot S(t)) \tag{20}$$

where $\beta = 0.2$ is a small phase modulation coefficient and d_m' the final threshold. This increases the stagnation threshold in the autonomy phase ($S(t) \to 1$), granting more self-adjustment space for the mature policy and preserving exploration-driven learning. Other thresholds follow the same phase-modulation logic with small coefficients to avoid overwriting environmental adaptation.

2) Kinematic-Constrained Waypoint Generation: When replanning is triggered, an improved A^* algorithm generates feasible sub-waypoints by embedding obstacle density, target proximity and robot kinematic constraints into the cost function. The waypoint density is redefined as:

$$D(x,y) = \frac{1}{1 + k_1 \cdot O(x,y) + k_2 \cdot T_{goal} + k_3 \cdot \rho} \tag{21}$$

where $O(x,y)$ is local obstacle density, T_{goal} target proximity, ρ motion curvature constraint, and k_1, k_2, k_3 normalized coefficients. Kinematic constraints (e.g., max turning angle) are embedded into the A^* cost function:

$$\text{Cost} = \text{Cost}_{dist} + \text{Cost}_{obs} + \lambda \cdot \text{Cost}_{kin} \tag{22}$$

with Cost_{dist} the distance cost, Cost_{obs} the obstacle avoidance cost, λ the kinematic weight, and Cost_{kin} the penalty for kinematically infeasible paths. After validity screening (safe obstacle distance, reasonable intervals), the optimal sub-waypoint is selected:

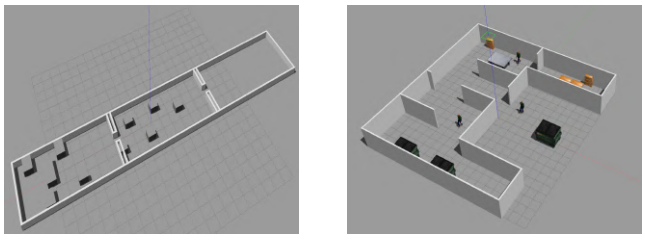
$$p_{new} = \arg\min \left(d\,(p, p_{target}) + \sum_{(x,y) \in \text{path}} D(x,y) \right) \tag{23}$$

where $d(p, p_{target})$ is the Euclidean distance from candidate p to the global target. Navigating to p_{new} resolves goal-obstacle conflicts and restores normal navigation, without altering ADRL's core transition-centric learning objective.

IV. EXPERIMENT AND RESULT

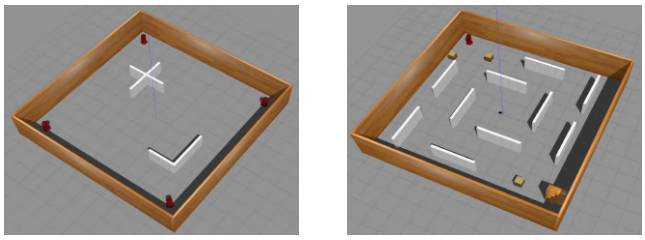
A. Experiments setup

To evaluate the performance and generalization of the robot navigation system, we built a dedicated experimental setup with a computational platform featuring an NVIDIA RTX 4060 GPU (8 GB memory) and an Intel Core i7-13650HX CPU, operating on Ubuntu 20.04 LTS. The simulation environment was developed based on the ROS framework with Gazebo 11, and two distinct simulation environments (for training and navigation phases) were designed to enrich the diversity of training data , as shown in Fig.2 and Fig.3.



(a) navigation environment 1 (b) navigation environment 2

Fig. 2: Navigation environments.



(a) Env. 1 (b) Env. 2

Fig. 3: Training environments.

B. Experiment and analysis

1) Training analysis: Fig. 4 shows the evolution of average critic Q-value statistic $\bar{Q}(t)$ (a proxy for policy maturity and transition effectiveness) for ablation variants. The Baseline (pure RL) has the lowest $\bar{Q}(t)$ and severe oscillation due to the lack of BC-guided early stabilization. RL-Reward (only dynamic reward) moderately lifts $\bar{Q}(t)$ but converges slowly, as reward shaping alone cannot resolve sparse rewards without adaptive imitation adjustment. RL-BC (only static BC) accelerates early convergence but shows unstable late-stage $\bar{Q}(t)$, a typical result of the stabilization-exploration dilemma of fixed BC—excessive imitation suppresses autonomous exploration with policy maturity. In contrast, full ADRL integrates value-driven imitation scheduling and phase-aligned dynamic rewards, realizing adaptive BC constraint relaxation and expert data mixing with rising $\bar{Q}(t)$. This transition-centric design achieves fast, stable convergence with the highest steady-state $\bar{Q}(t)$, verifying the stage-aware transition mechanism as the core of ADRL's superiority.

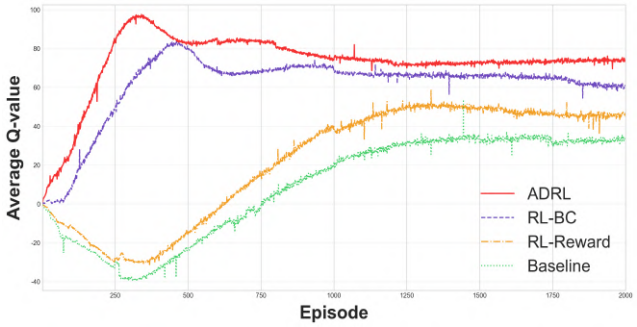
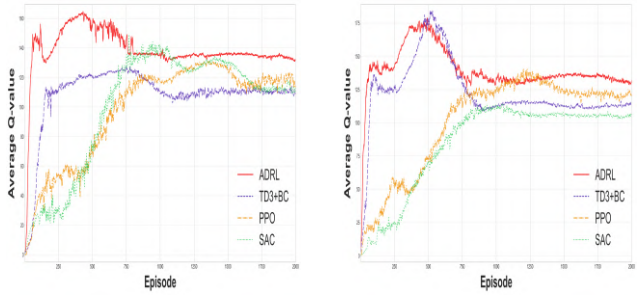


Fig. 4. Ablation experiment of average critic Q-value statistic $\bar{Q}(t)$ over episodes. ADRL resolves the fixed BC stabilization-exploration dilemma via stage-aware transition for stable fast convergence.

In this study, we considered three baseline methods for comparison: the first is TD3+BC (TD3 with behavior cloning regularization); the second is PPO (Proximal Policy Optimization); and the third is SAC (Soft Actor-Critic). All methods were tested in two customized environments to ensure the comprehensiveness and diversity of the results.

Fig. 5 compares $\bar{Q}(t)$ trends of ADRL and baselines in two scenarios, validating the generalization of ADRL's transition-centric BC fusion. SAC exhibits severe early $\bar{Q}(t)$ underestimation and persistent oscillation due to the lack of BC stabilization. PPO achieves slight early stability via clipped policy updates but suffers late-stage oscillation from unadjusted exploration-exploitation. TD3+BC (fixed BC) maintains stable early/mid $\bar{Q}(t)$ via expert initialization but stagnates in late training—static BC constraints suppress autonomous exploration, leading to policy solidification, the key dilemma addressed in this work. ADRL dynamically modulates BC strength and expert reliance by $\bar{Q}(t)$: strong BC/expert reliance stabilizes early training and accelerates $\bar{Q}(t)$ rise; gradual BC relaxation/expert attenuation releases exploration capability for continuous late-stage $\bar{Q}(t)$ growth. This smooth imitation-to-reinforcement transition enables ADRL's faster convergence, higher steady-state $\bar{Q}(t)$ and minimal oscillation in both scenarios, demonstrating the superiority of treating BC as a progressively released transitional constraint.

Fig. 6 quantifies training convergence time (episodes to steady-state $\bar{Q}(t)$), reflecting the efficiency of ADRL's transition mechanism under identical training budgets. PPO/SAC converge slowest due to random initial parameters and ineffective early exploration without BC guidance. TD3+BC shortens convergence via expert initialization but is limited by fixed BC constraints that hinder online data utilization for late-stage optimization. ADRL achieves the shortest convergence time via its value-driven stage-aware imitation scheduling: adaptive fusion of expert/online data by $\bar{Q}(t)$ leverages expert guidance for fast early convergence, and reduces expert reliance to fully exploit online data for late-stage optimization. This transition-based sample utilization maximizes efficiency, verifying that transition-centric BC fusion significantly accelerates training convergence.



(a)Env.1 (b)Env.2

Fig. 5. Average critic Q-value statistic $\bar{Q}(t)$ of different methods in two scenarios. ADRL's transition-centric design resolves fixed BC's stabilization-exploration dilemma for superior convergence and stability.

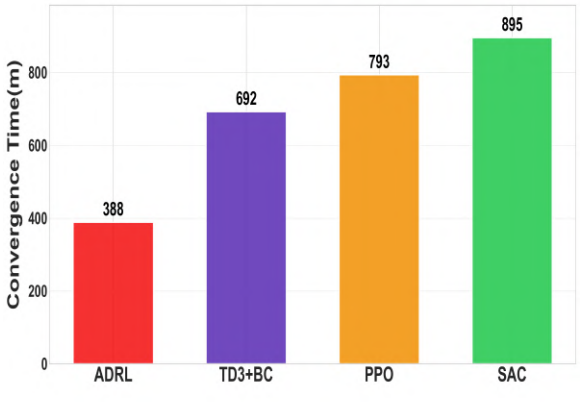


Fig. 6. Training convergence time (episodes to steady-state $\bar{Q}(t)$) of four methods. ADRL's adaptive imitation-to-reinforcement transition achieves the fastest convergence under identical budgets.

2) Navigation analysis: Table.I summarizes navigation performance metrics, verifying the downstream robustness of ADRL's transition-centric training framework. Rule-based A*+DWA has a high success rate but long navigation time due to poor complex-environment adaptability. TD3+BC shows low success rate and slow task completion in complex scenarios, as fixed BC leads to policy solidification and weak conflict handling capability. SAC/PPO yield moderate success rates via continuous optimization but sacrifice efficiency with conservative obstacle-avoidance strategies (caused by insufficient early stabilization). ADRL achieves the highest success rate and shortest navigation time with optimal path quality—this stems from its stage-aware transition mechanism: the smooth imitation-to-exploration shift equips the policy with expert-guided safety and autonomous complex-scenario decision-making. Additionally, the adaptive replanning mechanism (evaluation stabilizer) resolves goal-obstacle conflicts without altering the core transition objective, further boosting success rate and efficiency. ADRL's superior performance proves that transition-centric BC fusion synergistically improves training efficiency and downstream navigation robustness.

All experimental results validate the effectiveness of

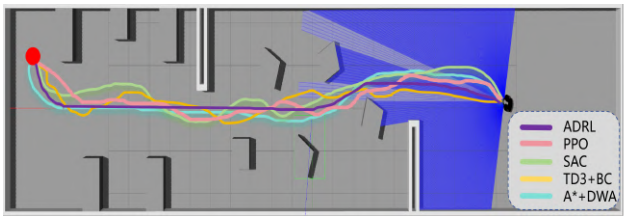


Fig. 7. Comparison of navigation routes using different methods.

TABLE I: Navigation performance of five methods in complex environments. ADRL's transition-centric framework and replanning stabilizer balance success rate and efficiency optimally.

	Av.D(m)	Av.T(s)	Succ.rate	Optimality
A*+DWA	21.42	44.33	0.92	1.02
TD3+BC	21.84	42.41	0.83	1.04
SAC	22.68	41.73	0.82	1.08
PPO	22.46	41.55	0.86	1.07
ADRL	21.41	39.62	0.97	1.02

ADRL's transition-centric BC fusion strategy, which reinterprets BC as a progressively released transitional constraint (instead of a static regularizer) for robot navigation. ADRL's superior performance—faster convergence, lower training oscillation, higher navigation success rate and robustness—stems from resolving the stabilization-exploration dilemma of fixed BC-enhanced DRL. By taking $\bar{Q}(t)$ as the policy maturity proxy, ADRL adaptively modulates BC strength and expert data reliance, realizing a smooth shift from imitation-dominant early training to exploration-driven late training. This design leverages expert demonstrations for stabilization and online data for autonomous optimization, achieving synergistic training and policy performance improvement. Phase-aligned dynamic reward weighting ensures consistent learning pressure across training phases, while the adaptive replanning evaluation stabilizer resolves navigation conflicts for fair policy assessment—both are necessary complements to the core transition mechanism. Under identical training budgets, ADRL outperforms state-of-the-art fixed BC and pure RL methods in all metrics, providing a principled integration strategy for imitation and reinforcement learning in robot navigation, and laying a solid foundation for DRL's real-world autonomous navigation applications.

V. CONCLUSIONS

This paper proposes ADRL method for mobile robot navigation, which reinterprets BC as a progressively released transitional constraint and constructs a stage-aware transition framework to overcome the stabilization-exploration dilemma of fixed BC-enhanced DRL. ADRL adopts a value-driven imitation scheduling mechanism based on critic Q-value statistics to adaptively adjust expert-online data mixing and BC regularization strength, enabling a smooth shift from imitation-dominant to exploration-driven learning, and

(a) Start point (b) ADRL 10S (c) ADRL 20S (d) ADRL 30S

(a) ADRL 40s (b) ADRL 50S (c) ADRL 60S (d) Goal point

Fig. 8. Figure presents the real-time navigation screenshot sequence of ADRL's sim-to-real transfer verification. The method achieves collision-free navigation with high movement efficiency throughout, intuitively demonstrating its dynamic navigation process in a real laboratory environment and verifying the model's reliable transfer from simulation to physical implementation.

is supplemented by a phase-aligned dynamic weight reward function and a lightweight adaptive replanning evaluation stabilizer to mitigate sparse rewards and resolve navigation conflicts. Multi-scenario experiments show that under the same training budget, ADRL significantly outperforms state-of-the-art methods in training convergence, navigation success rate and robustness, providing a principled integration strategy for imitation and reinforcement learning in robot navigation.

REFERENCES

[1] H. Sun, W. Zhang, R. Yu, and Y. Zhang, "Motion planning for mobile robots—focusing on deep reinforcement learning: A systematic review," *IEEE Access*, vol. 9, pp. 69061–69081, 2021.

[2] Y. Bi, J. Luo, J. Zhu, J. Liu, and W. Li, "Decentralized multi-robot navigation based on deep reinforcement learning and trajectory optimization," *Biomimetics*, vol. 10, no. 6, p. 366, 2025.

[3] X. Diao, Z. Sun, J. Peng, and J. Wang, "Lstp-nav: Lightweight spatiotemporal policy for map-free multi-agent navigation with lidar," *arXiv preprint arXiv:2408.16370*, 2024.

[4] Z. Xie and P. Dames, "Drl-vo: Learning to navigate through crowded dynamic scenes using velocity obstacles," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2700–2719, 2023.

[5] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. P. Chen, "Design and implementation of deep neural network-based control for automatic parking maneuver process," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1400–1413, 2020.

[6] R. Chai, H. Niu, J. Carrasco, F. Arvin, H. Yin, and B. Lennox, "Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5778–5792, 2022.

[7] R. Chai, K. Chen, L. Cui, S. Chai, G. Inalhan, and A. Tsourdos, "Advanced trajectory optimization, guidance and control strategies for aerospace vehicles," 2023.

[8] U. Orozco-Rosas, K. Picos, O. Montiel, and O. Castillo, "Environment recognition for path generation in autonomous mobile robots," *Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine*, pp. 273–288, 2020.

[9] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1088–1095, 2018.

[10] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[11] J. Chung, "Playing atari with deep reinforcement learning," *Comput. Ence*, vol. 21, pp. 351–362, 2013.

[12] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.

[13] T. Lillicrap, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[14] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, pp. 1587–1596, PMLR, 2018.

[15] L. Butyrev, T. Edelhäußer, and C. Mutschler, "Deep reinforcement learning for motion planning of mobile robots," *arXiv preprint arXiv:1912.09260*, 2019.

[16] R. Cimurs, I. H. Suh, and J. H. Lee, "Goal-driven autonomous exploration through deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 730–737, 2021.

[17] C. Wang, J. Wang, J. Wang, and X. Zhang, "Deep-reinforcement-learning-based autonomous uav navigation with sparse rewards," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6180–6190, 2020.

[18] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, *et al.*, "Reinforcement and imitation learning for diverse visuomotor skills," *arXiv preprint arXiv:1802.09564*, 2018.

[19] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," *Advances in neural information processing systems*, vol. 34, pp. 20132–20145, 2021.

[20] W. Liang, J. Xie, Z. Wang, J. Tan, and X. Ma, "Constrained behavior cloning for robotic learning," *arXiv preprint arXiv:2408.10568*, 2024.