Navigating with Less: Reinforcement Learning for UGVs Under Sparse LiDAR Inputs

Pragati Nayak[1,*], Mayank Srivastava[1,*], Harshit Agnihotri[2,*], Agniva Banerjee[2] and Arijit Sen[2]

***Abstract*— Autonomous navigation remains a fundamental challenge for unmanned ground vehicles (UGVs) operating in complex and unstructured environments. Existing learning-based solutions typically rely on computationally intensive perception pipelines such as 3D SLAM and PointNet, which are difficult to deploy on resource-constrained platforms. This paper proposes an efficient end-to-end framework using a lightweight 32-bin lidar descriptor with a simple MLP, comparing discrete-action Dueling DQN against continuous-action SAC. Simulation results show that SAC significantly outperforms DQN in terms of success rate, collision avoidance, convergence stability, and control smoothness, demonstrating that algorithmic choice can surpass perception complexity in achieving high-performance navigation on computationally limited UGV platforms.**

I. Introduction

Autonomous navigation for unmanned ground vehicles (UGVs) is a critical challenge in applications such as emergency rescue [1], planetary exploration [2], environmental monitoring [3], and agriculture [4]. These domains require systems that can operate reliably in harsh environments. Existing approaches fail in such settings due to their dependence on rich prior information [5].

Conventional SLAM-based pipelines divide the navigation task into localization, planning, and control [6]. This modular navigation design limits adaptability and forces the system to depend on maps that may be missing or become outdated in real deployments [6]. Recent DRL-based methods attempt to overcome these issues by using detailed multi-modal sensing [7]. However, they rely on heavy perception stacks that include full 3D SLAM, dense point cloud generation, and complex trajectory sampling routines [6]. These demands make such methods difficult to deploy in a resource-limited environment and create challenges for the research community [8].

In this work, we mainly focus on the gap between computational cost and navigation performance. We present a simplified framework that eliminates the need for complex 3D perception, focusing instead on the role of the learning algorithm. A lightweight 32-bin LiDAR descriptor with an MLP replaces the full SLAM and fusion pipeline used in recent systems [7]. Within this reduced setting, we examine whether the choice of algorithm can compensate for limited perceptual details and compare it with the discrete Dueling DQN [9] baseline and continuous SAC agent [10]. Simulation results using Webots2025a [11] show that SAC provides significantly stronger navigation performance than the discrete Dueling DQN.

[1]Department of Data Science and Engineering, IISER Bhopal, India, pragati23, mayanks23@iiserb.ac.in

[2]Department of Electrical Engineering and Computer Science, IISER Bhopal, India, harshit23, agniva24, ajsen@iiserb.ac.in

[*] First, second and third authors contributed equally to this research.

II. Methodology

The proposed framework (Fig. 1) extends the multimodal fusion paradigm presented in [7] through several significant architectural enhancements. Simulations are conducted using Webots2025a [11] on an RTX 5060 GPU with 8 GB of VRAM, which provides realistic physics engines with accurate sensor modelling, and configurable virtual environments. The system architecture processes multimodal sensory inputs, including pose vectors (from GPS and IMU), depth images (from the onboard camera), and LiDAR point clouds, through three modality-specific feature extractors. For computational efficiency, we replace complex 3D perception with a lightweight 32-bin min-range LiDAR descriptor processed by a simple 2-layer MLP (128 hidden units, ReLU activations). The extracted features are fused through an LSTM-based temporal integration layer and passed to the learning strategy module, enabling real-time online training where all perception and strategy modules train simultaneously from the start of the episode.

$$r = r_{dis} + r_{col} + r_{head} + r_{obs} \tag{1}$$

where each reward term is represented as,

$$r_{\text{dis}} = \mu_{r_{\text{dis}}} \left(\left\lfloor \frac{d_{\text{prev}}}{W_{r_{\text{dis}}}} \right\rfloor - \left\lfloor \frac{d}{W_{r_{\text{dis}}}} \right\rfloor \right) \quad r_{\text{obs}} = \begin{cases} -\dfrac{\exp(5-d_{\text{obs}})}{\mu_{r_{obs}}}, & d_{\text{coll}} \le d_{\text{obs}} < d_{\max}, \\ 0, & d_{\text{obs}} \ge d_{\max}, \end{cases}$$

$$r_{\text{col}} = \begin{cases} R_{\text{colison}}, & d_{\text{obs}} < d_{\text{coll}}, \\ 0, & \text{otherwise}, \end{cases} \qquad r_{\text{head}} = H\cos(\theta)$$

The reward function, defined in Eq. (1), incorporates goal proximity (r_{dis}), collision penalty (r_{col}), heading alignment (r_{head}), and obstacle proximity (r_{obs}). Additionally, we incorporate a dense segment reward of 100 for every 25-meter milestone and extend the maximum episode length to 5000 steps, providing stronger learning signals during training. This design encourages sustained forward progress and prevents premature episode termination, enabling the agent to explore more extensively and stabilize policy improvement. In reward function, d_{prev} and d denote the previous and current distances to the goal, while the discretization width is $W_{r_{dis}} = 0.2$ and the progress scale is $\mu_{r_{dis}} = 5$. The collision term uses the minimum obstacle distance d_{obs}, the collision threshold $d_{\text{coll}} = 0.8\,\text{m}$, and the penalty $R_{\text{colison}} = -200$. The heading reward depends on the alignment angle θ and the weight $H = 1$. The obstacle proximity term uses the

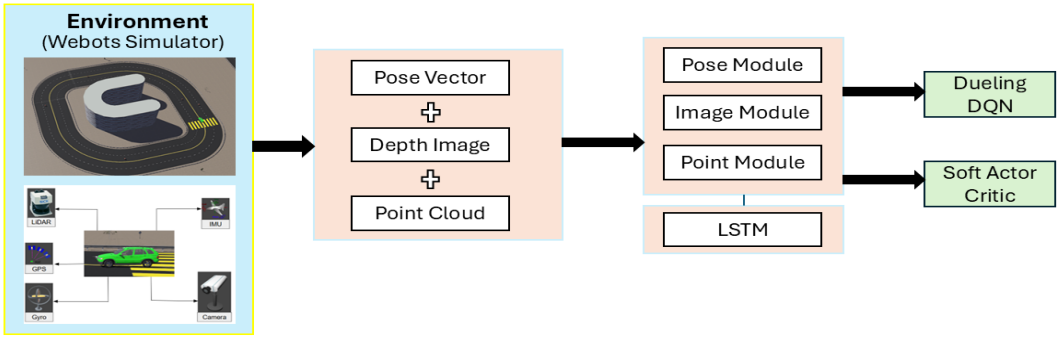


Fig. 1: System architecture for comparative DRL-based UGV navigation. The Webots simulator provides multi-modal inputs (pose vectors, depth images, point clouds) processed through modality-specific feature extractors (Pose, Image, Point modules). Features are temporally fused via LSTM and fed to either a discrete-action Dueling DQN or continuous-action Soft Actor-Critic agent.

maximum considered distance $d_{\max} = 5\,\text{m}$ and the scaling factor $\mu_{r_{obs}} = 3$.

We incorporate two deep RL architectures within this proposed framework. The baseline Dueling DQN agent splits the state representation into value and advantage streams for 36 discrete actions, using ε-greedy exploration and hard-copy target updates [8], [12]. However, the SAC agent [13] employs a Gaussian policy over a continuous 2D action space (v, w) with tanh-squashed sampling, twin-Q critics to mitigate overestimation, automatic entropy temperature tuning, and soft Polyak target updates with a reward scale of 10.

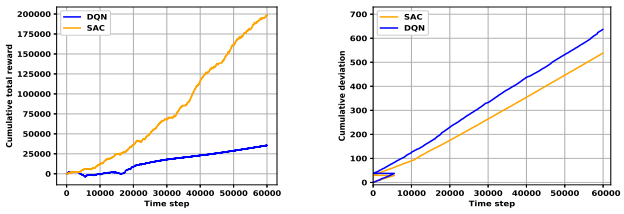
For fair comparison, we implement a hybrid protocol for SAC where continuous action (v_c, w_c) is mapped to the closest discrete action from the 36-action grid for simulator execution. However, the original continuous action is stored in the replay buffer for critic training [14]. This enables SAC to leverage smooth policy gradients while maintaining comparable discrete execution capabilities to DQN.

III. Results and Discussion

Table I compares the Dueling DQN and SAC agents integrated into the identical simplified perception framework. Navigation is tested in a $50m \times 50m$ environment with static obstacles and complex terrain (as shown in Fig. 1). SAC achieves an 82.3% success rate compared to DQN's 49%, while reducing collisions from 36% to 12.7%. The learning curves shown in Fig. 2(a), reveal that SAC exhibits significantly more stable convergence with lower variance, whereas DQN displays greater instability throughout training. Similarly, Fig. 2(b) compares the cumulative deviation of both agents, showing that SAC accumulates deviation at a slower and more consistent rate than DQN. This indicates smoother and more stable trajectory execution, whereas the steeper deviation growth of DQN reflects its tendency toward abrupt or suboptimal control actions. This performance disparity stems from fundamental algorithmic differences rather than perception capabilities; both agents utilized identical sensory inputs. DQN operates over a fixed 36-action discrete grid, limiting exploration to pre-defined combinations and restricting fine-grained motion adjustments. Conversely, SAC optimizes a Gaussian policy directly in the continuous action space (v, w) [8]. Its entropy-maximization enables systematic exploration of intermediate actions, producing smoother trajectory control. This superior exploration and policy expressiveness directly account for SAC's improved success rate, collision avoidance, and convergence stability, demonstrating that algorithmic choice can outweigh perception complexity in resource-constrained UGV navigation.

Metric	Dueling DQN	Soft Actor-Critic (SAC)
Success Rate (%)	49	82.3
Collision Rate (%)	36	12.7

TABLE I: Performance of the bassline DRL models.



(a) **(b)**

Fig. 2: Comparison of DQN and SAC performance over the training time steps, where 2a shows the cumulative total reward highlighting SAC's faster and higher reward accumulation, and 2b presents the cumulative deviation illustrating SAC's more stable deviation growth relative to DQN.

IV. Conclusion and Future Work

The study demonstrated that the multimodal perception framework with SAC algorithm performed better than the Duelling DQN baseline in terms of success rate, collision avoidance, convergence stability and control smoothness for UGV navigation in complex environments. Future work will extend this framework to real-world UGV platforms, add dynamic obstacles and explore other multi-agent RL methods. We also plan to expand this work for evaluation across diverse terrain configurations by including rough surfaces, rough terrain with obstacles, pits and scenarios involving dynamic pedestrians. This will further advance the robustness and applicability of the proposed navigation framework.

REFERENCES

[1] D. Calisi, A. Farinelli, L. Iocchi, and D. Nardi, "Autonomous navigation and exploration in a rescue environment," in *IEEE International Safety, Security and Rescue Rototics, Workshop, 2005.* IEEE, 2005, pp. 54–59.

[2] Y. Ma, C. Gu, J. Jiang, X. Wei, D. Xie, G. Wang, and J. Li, "Review of autonomous optical navigation for deep space exploration," *IEEE Transactions on Instrumentation and Measurement*, 2025.

[3] A. Vasilijević, . Na, F. Mandić, N. Mišković, and Z. Vukić, "Coordinated navigation of surface and underwater marine robotic vehicles for ocean sampling and environmental monitoring," *IEEE/ASME transactions on mechatronics*, vol. 22, no. 3, pp. 1174–1184, 2017.

[4] N. Shalal, T. Low, C. McCarthy, and N. Hancock, "A review of autonomous navigation systems in agricultural environments," *SEAg 2013: Innovative agricultural technologies for a sustainable future*, 2013.

[5] C. Wang, J. Wang, Y. Shen, and X. Zhang, "Autonomous navigation of uavs in large-scale complex environments: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2124–2136, 2019.

[6] C. Estrada, J. Neira, and J. D. Tardós, "Hierarchical slam: Real-time accurate mapping of large environments," *IEEE transactions on Robotics*, vol. 21, no. 4, pp. 588–596, 2005.

[7] Z. Han, P. Chen, B. Zhou, and G. Yu, "Real-time navigation of unmanned ground vehicles in complex terrains with enhanced perception and memory-guided strategies," *IEEE Transactions on Vehicular Technology*, 2025.

[8] N. Gholizadeh, N. Kazemi, and P. Musilek, "A comparative study of reinforcement learning algorithms for distribution network reconfiguration with deep q-learning-based action sampling," *Ieee Access*, vol. 11, pp. 13 714–13 723, 2023.

[9] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1995–2003.

[10] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870.

[11] O. Michel, "Cyberbotics ltd. webots™: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, p. 5, 2004.

[12] J. Pan, X. Wang, Y. Cheng, and Q. Yu, "Multisource transfer double dqn based on actor learning," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2227–2238, 2018.

[13] J. Duan, Y. Guan, S. E. Li, Y. Ren, Q. Sun, and B. Cheng, "Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 11, pp. 6584–6598, 2021.

[14] J. Yang, J. Zhang, M. Xi, Y. Lei, and Y. Sun, "A deep reinforcement learning algorithm suitable for autonomous vehicles: Double bootstrapped soft-actor–critic-discrete," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 15, no. 4, pp. 2041–2052, 2021.