Multimodal Feature Fusion for Deep Reinforcement Learning-based Mobile Robot Navigation

Zhiyuan Nan
Hanyang University
Department of Electrical and Electronic Engineering
Ansan, Korea
namjw@hanyang.ac.kr

Haewoon Nam
Hanyang University
Department of Electrical and Electronic Engineering
Ansan, Korea
hnam@hanyang.ac.kr

***Abstract*—Robust navigation in complex and dynamic environments poses a significant challenge for reinforcement learning (RL) algorithms. Traditional RL methods often rely on single-modality inputs, such as LiDAR or camera data, which limits their perception and decision-making capabilities in real-world applications. In this paper, we propose a multimodal feature extraction network integrated with a Soft Actor-Critic (SAC) framework to address these limitations. Our method combines Bird's-Eye View (BEV) images, distance data, and target-related information to generate a unified feature representation. The network incorporates convolutional layers and Layer Normalization to effectively fuse multimodal features while mitigating gradient instability. Furthermore, we employ a Gated Recurrent Unit (GRU) to model temporal dependencies, enabling the agent to adapt to dynamic environmental changes. To evaluate the proposed approach, we conducted extensive experiments on a navigation task in a complex environment. Results demonstrate that our method significantly outperforms baseline methods in terms of training efficiency, policy robustness, and reward convergence. Notably, the inclusion of temporal modeling via GRU enhances the agent's ability to capture dynamic environmental features, leading to superior performance compared to methods without temporal modeling. These findings highlight the potential of multimodal reinforcement learning frameworks for real-world robotic applications.**

***Index Terms*—Deep Reinforcement Learning, Multimodal Feature Fusion, Mobile Robot Navigation, Obstacle Avoidance**

I. Introduction

In recent years, Deep Reinforcement Learning (DRL) has made significant advancements in fields such as robotic navigation, autonomous driving, and gaming. However, in the robotics domain, traditional DRL algorithms often rely on single-modality inputs, such as 2D LiDAR or RGB-D cameras. This reliance can lead to insufficient perception capabilities in complex environments, thereby limiting the performance of learned policies.

To achieve more robust perception and decision-making, DRL approaches that incorporate multimodal information have garnered increasing attention. Particularly in robotic navigation tasks, multimodal data, such as Bird's-Eye View (BEV) images, LiDAR information, and millimeter-wave radar, can provide rich spatial and environmental awareness. However, efficiently fusing multimodal features and fully exploiting the dynamic changes in temporal sequences remain pressing challenges that need to be addressed.

Current research has demonstrated significant potential in multimodal fusion and temporal sequence modeling; however, there remain challenges in achieving effective feature integration. Most existing approaches simply concatenate multimodal features, failing to fully exploit the complementary information between different modalities.

To address these limitations, this paper proposes a reinforcement learning network framework based on multimodal features. The framework is designed to improve the efficiency and robustness of policy learning through advanced multimodal feature fusion and temporal sequence modeling.

The main contributions of this paper are as follows:

1. Novel Multimodal Fusion Network: We design an innovative multimodal fusion network that effectively integrates features from BEV images, distance data, and target information, enabling efficient training for subsequent reinforcement learning tasks.

2. Temporal Sequence Modeling with GRU: By introducing Gated Recurrent Units (GRU) [1], the framework performs temporal sequence modeling on the fused multimodal features, significantly enhancing the navigation strategy's performance in complex environments.

3. Performance Comparison with Single-Modality Methods: Experimental results demonstrate that the proposed method achieves higher training efficiency, better convergence, and improved robustness compared to single-modality approaches.

The remainder of this paper is organized as follows: Section II introduces the deep reinforcement learning algorithm utilized in this study. Section III provides a detailed description of the proposed network architecture and reinforcement learning framework. Section IV presents the experimental setup and analyzes the results. Section V concludes the paper with a summary of the findings.

II. Soft Actor-Critic

Soft Actor-Critic (SAC) [2] is a policy-based deep reinforcement learning algorithm within the maximum entropy reinforcement learning framework, designed to achieve efficient policy optimization. The primary objective of SAC is to simultaneously maximize the cumulative reward and the entropy of the policy, promoting exploration during learning while enhancing the algorithm's robustness and stability. By

employing techniques such as experience replay and target networks, SAC effectively utilizes data and mitigates the risk of suboptimal solutions caused by premature convergence through the maximum entropy objective. SAC is particularly well-suited for tasks such as robotic control and navigation. The algorithm consists of two main components: the Actor network and the Critic network. The Critic network uses two action-value Q - function networks to improve stability and performance.

The loss function for the actor network in the SAC algorithm is defined as

$$\mathcal{L}_{\text{actor}} = \mathbb{E}_{s_t \sim \mathcal{D}} \Big[\mathbb{E}_{a_t \sim \pi_\phi} \Big[\alpha \log \pi_\phi(a_t|s_t) - Q_{\min}(s_t, a_t) \Big] \Big] \tag{1}$$

The loss function for the critic network in the SAC algorithm is defined as

$$\mathcal{L}_{\text{critic}} = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}} \Big[\frac{1}{2} \big(Q_{\theta_i}(s_t, a_t) - y_t \big)^2 \Big] \tag{2}$$

The target value update is defined as

$$y_t = r_t + \gamma \mathbb{E}_{a_{t+1} \sim \pi_\phi} \Big[Q_{\min}(s_{t+1}, a_{t+1}) - \alpha \log \pi_\phi(a_{t+1}|s_{t+1}) \Big] \tag{3}$$

III. Proposed Method

In this paper, we propose the BEV-GRU-SAC algorithm for robot obstacle-avoidance navigation, which leverages BEV images derived from 3D LiDAR data and distance information computed from the same 3D LiDAR data. By efficiently extracting features from multimodal information, the proposed method enhances the performance of deep reinforcement learning algorithms, thereby improving the robot's obstacle-avoidance navigation capabilities.

A. Problem Definition

We model the robot navigation problem using a Markov Decision Process (MDP) [3], formally defined as a five-tuple M=(S, A, R, P, γ), S represents the set of states, A represents the set of available actions within the decision-making framework, R specifies the immediate reward associated with each state-action pair, P defines the state transition probability matrix, γ is the discount factor, which governs the trade-off between immediate rewards and long-term rewards, with $\gamma \in$ [0, 1].

To model the robot navigation problem using an MDP framework, we define the state space, action space, and reward function as follows.

1) State space S: In this paper, we utilize a 3D LiDAR sensor for obstacle-avoidance navigation tasks. However, directly using 3D point cloud data for training neural networks poses significant challenges due to its high computational complexity. To address this, we mathematically transform the 3D point cloud data into two-dimensional BEV representations and one-dimensional distance information between the robot and obstacles. In addition, we include the robot's previous action, its orientation angle relative to the target, and the distance to the target as part of the state representation for training the robot. The state space is defined as

$$S_t = \{bev_t,\ l_t,\ d_t,\ \theta_t,\ v_{t-1},\ \omega_{t-1}\}, \tag{4}$$

where bev_t represents the BEV generated from the 3D point cloud data, l_t denotes the distance information between the robot and obstacles derived from the 3D point cloud, d_t is the distance between the robot and the target point, and θ_t represents the orientation angle of the robot relative to the target point. Additionally, v_{t-1} and ω_{t-1} correspond to the robot's actions taken in the previous step.

2) Action space A: The robot has two actions: linear velocity and angular velocity. We define the continuous action space as

$$A_t = \{(v_t,\ \omega_t) \mid v_t \in [0,\ 1],\ \omega_t \in [-1,\ 1]\}, \tag{5}$$

where v_t represents the linear velocity, constrained within the range $[0, 1]$, and ω_t represents the angular velocity, constrained within the range $[-1, 1]$.

3) Reward function R: We define a piecewise reward function as

$$R(s,a) = \begin{cases} r_{goal}, & if\ d_t < d_\beta, \\ r_{collision}, & if\ \ collision, \\ \alpha * v - \lambda * |\omega| - \delta * d_o, & otherwise. \end{cases} \tag{6}$$

The robot is considered to have reached the target point when the distance between the robot and the target, d_t, is less than the threshold d_β, at which point a positive reward of $+100$ is assigned. In the event of a collision, a negative reward of -100 is given. For obstacle avoidance, let d_o represent the shortest distance between the robot and the nearest obstacle. If $d_o < 1$, the reward is calculated as $1 - d_o$; otherwise, the reward for d_o is set to 0. In this paper, the hyperparameters α, λ and δ are set to 0.5.

B. Network architecture

As shown in Fig. 1, we propose a novel network architecture designed to handle multimodal input data. This includes utilizing BEV images, distance data, and target-related information to generate effective features suitable for deep reinforcement learning.

We process the robot's orientation and distance to the target point, as well as the previous step's action information, through a fully connected layer to extract initial features. These features are subsequently normalized using Layer Normalization (LN) [4] and passed through a second fully connected layer to generate a compressed feature representation. The distance data, derived from 3D point cloud data captured by the 3D LiDAR sensor, represents the robot's surrounding environmental distances. This data is processed through three layers of 1D convolution to iteratively extract spatial features, followed by a fully connected layer with LN to produce a low-dimensional feature representation. For BEV images, we

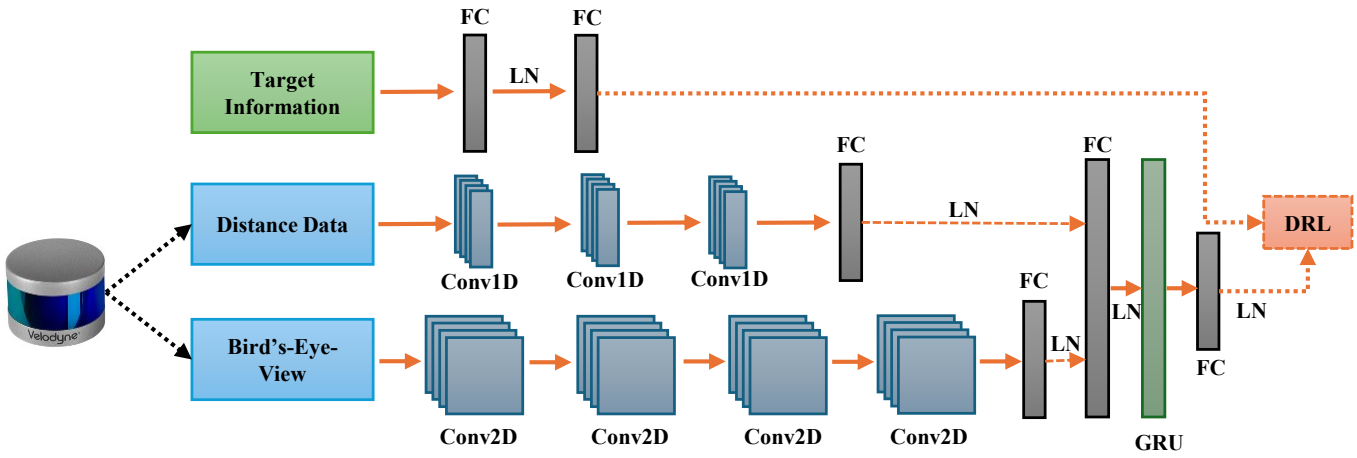

Fig. 1. Feature extraction network architecture.

employ four layers of 2D convolution to extract visual features, capturing obstacle distributions and global spatial information. The resulting features are then compressed through a fully connected layer with LN to generate a compact feature representation. The normalized and compressed distance data features are concatenated with the compact visual features extracted from BEV images. This concatenated representation is processed through a fully connected network designed for feature fusion, resulting in a unified multimodal feature representation. This fusion approach leverages the complementary nature of different modalities, enhancing the network's ability to understand spatial relationships. The fused features are further processed through a GRU to model temporal dynamics. The GRU's output is normalized using LN, improving the network's ability to adapt to dynamic environmental changes. Finally, the fused features of distance data, BEV images, and target-related information are fed into the deep reinforcement learning algorithm as input to optimize navigation strategies and decision-making. The proposed network architecture, with its multimodal design and temporal modeling capabilities, enables efficient feature extraction and decision-making in complex, dynamic environments, providing robust support for robot navigation tasks.

LN is incorporated into the feature extraction process for target information, distance data, and BEV features, effectively mitigating issues related to gradient vanishing or explosion while enhancing the stability of network training. Additionally, LN is applied between the feature fusion stage and the output layer to ensure a more uniform distribution of features, further improving the model's overall performance and robustness.

As shown in Fig. 2, in the deep reinforcement learning network, the fused features extracted from BEV images and distance data are concatenated with target-related features to serve as input to both the actor and critic networks. For the critic, the concatenated features are further combined with the action corresponding to the current state, forming the complete input to the critic network. In the actor network, the concatenated features are passed through a fully connected layer, followed by LN and a ReLU activation function to generate nonlinear features. These features are further processed by a second fully connected layer to refine the feature representation. Finally, the output layer, implemented as another fully connected layer, predicts the action values. In the critic network, the concatenated state features and action input are processed through a fully connected layer with LN and a ReLU activation function to generate nonlinear features. These features are further refined through an additional fully connected layer, and the final output layer computes the Q -values, representing the value of the given state-action pair.

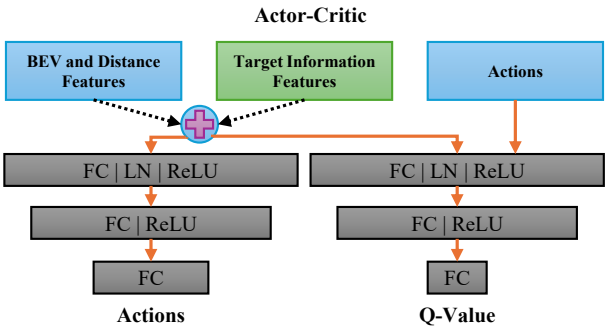

Fig. 2. Actor-Critic network architecture.

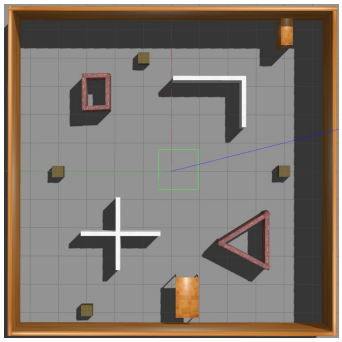

Fig. 3. Training environment.

IV. Experimental Results

A. Experimental Environment

In this paper, the proposed algorithm's effectiveness was validated through obstacle-avoidance navigation experiments conducted in the GAZEBO [5] simulation environment. Three sets of experiments were performed: (1) a baseline navigation algorithm using Twin Delayed Deep Deterministic policy gradient (TD3) [6] based solely on 3D LiDAR-derived distance information and target point information, (2) the proposed BEV-SAC algorithm leveraging BEV, distance data, and target point information, and (3) the BEV-GRU-SAC algorithm incorporating temporal modeling. In the experiments, the TD3 baseline algorithm used distance information with a maximum range of 10 meters as input, while our proposed algorithms (BEV-SAC and BEV-GRU-SAC) used distance information with a maximum range of 5 meters as input. The parameters used in the three algorithms are listed in Table I. Notably, the parameters for the two proposed algorithms (BEV-SAC and BEV-GRU-SAC) are identical.

The experimental setup utilized an Intel i9-12900K CPU, NVIDIA GeForce RTX 3090 GPU, the Robot Operating System (ROS: Noetic) [7], and the GAZEBO simulator. Both training and testing of the robot were carried out in the GAZEBO simulator, using the Pioneer 3-DX robot. The experimental scenarios are illustrated in Fig. 3.

TABLE I
PARAMETERS

	TD3	BEV-GRU-SAC
learning rate	1e-3	3e-4
gamma	0.99999	0.99
tau	5e-3	5e-3
log std min	-	-20
log std max	-	2
buffer size	100000	100000
batch size	40	128

The robot utilizes a 3D LiDAR sensor to capture the current environment's 3D point cloud data, which is mathematically processed and transformed into BEV images and distance information. In this environment, the robot relies on sensor data to identify obstacles and uses its orientation and distance to the target to learn how to navigate successfully to the target point without collisions.

During training, if the robot collides with an obstacle, the environment resets. This reset includes randomizing the robot's position, orientation, the positions of obstacle boxes, and the target point's location within the environment.

As more data is collected and the number of training iterations increases, the network gradually learns from the data how to avoid obstacles and navigate successfully to the target point.

B. Experimental Result

In the training setup, each episode has a maximum of 500 steps. The environment is reset if a collision occurs or if the robot exceeds 500 steps without reaching the target or colliding.

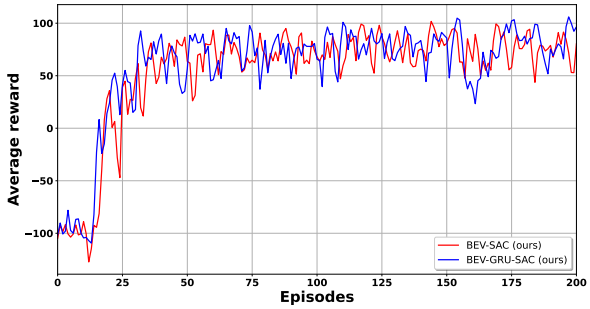

Fig. 4. Train results.

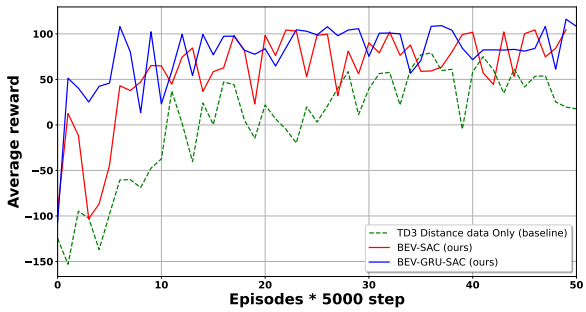

Fig. 5. Test results.

As shown in Fig. 4, we compared the performance of the two proposed algorithms based on BEV, distance information, and target point information. During the first 25 episodes, both methods exhibited relatively low rewards. However, BEV-GRU-SAC demonstrated a noticeably faster convergence rate compared to BEV-SAC in the early stages. Starting from episode 25, the average reward increased rapidly, with both methods stabilizing around episode 50. Throughout the training process, BEV-GRU-SAC achieved slightly higher rewards than BEV-SAC, indicating that the introduction of the GRU module enhances performance during the training phase.

As shown in Fig. 5, we compared the performance of the two proposed algorithms (BEV-SAC and BEV-GRU-SAC) with the baseline TD3 algorithm. The rewards achieved by TD3 were consistently lower and more volatile throughout the testing phase, indicating that relying solely on distance information is insufficient for efficient navigation decisions.

Both BEV-SAC and BEV-GRU-SAC exhibited rapid improvement during the early stages (first 10 episodes), with BEV-GRU-SAC achieving significantly higher rewards. As testing progressed, BEV-GRU-SAC maintained consistently higher average rewards and outperformed BEV-SAC in most cases. While BEV-SAC demonstrated better reward stability compared to the baseline method, its rewards were slightly lower than those of BEV-GRU-SAC.

V. Conclusion

In this paper, we propose a novel multimodal feature extraction network integrated with the Soft Actor-Critic (SAC) deep reinforcement learning algorithm for obstacle-avoidance navigation tasks. The network processes different features in separate branches and leverages the complementary nature of multimodal features for effective feature fusion. Input and output layers are normalized to ensure the extraction of high-accuracy multimodal fused features.

The proposed network's multimodal design and temporal modeling capabilities enable efficient feature extraction and decision-making in complex, dynamic environments, providing robust support for robotic navigation tasks. Experimental results in the GAZEBO simulation environment demonstrate that the proposed method outperforms the TD3 baseline algorithm and effectively accomplishes obstacle-avoidance navigation tasks.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A2C1011862).

References

[1] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014, arXiv:1412.3555 [cs]. [Online]. Available: http://arxiv.org/abs/1412.3555

[2] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft Actor-Critic Algorithms and Applications," Jan. 2019, arXiv:1812.05905 [cs]. [Online]. Available: http://arxiv.org/abs/1812.05905

[3] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Aug. 2014, google-Books-ID: VvBjBAAAQBAJ.

[4] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," Jul. 2016, arXiv:1607.06450 [stat]. [Online]. Available: http://arxiv.org/abs/1607.06450

[5] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sep. 2004, pp. 2149–2154 vol.3. [Online]. Available: https://ieeexplore.ieee.org/document/1389727

[6] S. Fujimoto, H. v. Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," Oct. 2018, arXiv:1802.09477 [cs]. [Online]. Available: http://arxiv.org/abs/1802.09477

[7] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System."