D2AH-PPO: Playing ViZDoom With Object-Aware Hierarchical Reinforcement Learning

Longyu Niu
MAIS, CASIA, Beijing, China
School of Artificial Intelligence, UCAS, Beijing, China
niulongyu2021@ia.ac.cn

Jun Wan
MAIS, CASIA, Beijing, China
School of Artificial Intelligence, UCAS, Beijing, China
jun.wan@ia.ac.cn

***Abstract*—Deep reinforcement learning (DRL) has achieved superhuman performance on Atari games using only raw pixels. However, when applied to complex 3D first-person shooter (FPS) environments, it often faces compound challenges of inefficient exploration, partial observability, and sparse rewards. To address this, we propose the *D*epth-*D*etection *A*ugmented *H*ierarchical *P*roximal *P*olicy *O*ptimization (D2AH-PPO) method. Specifically, our framework utilizes a two-level hierarchy where the higher-level controller handles option control learning, while the lower-level workers focus on mastering sub-tasks. To boost the learning of sub-tasks, D2AH-PPO involves a combination technique, which includes 1) object-aware representation learning that extracts high-dimensional information representation of crucial components, and 2) a rule-based action mask for safer and more purposeful exploration. We assessed the efficacy of our framework in the 3D FPS game 'ViZDoom'. Extensive experiments indicate that D2AH-PPO significantly enhances exploration and outperforms several baselines.**

***Index Terms*—deep reinforcement learning, ViZDoom, representation learning, FPS**

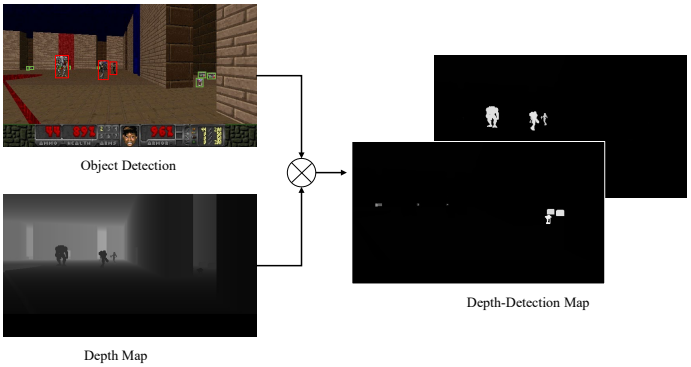


Fig. 1. Motivation: The depth-detection map provides essential information about crucial components' location and depth distance. According to the rules of VDAIC, direct access to this high-dimensional abstraction is restricted. Therefore, we utilize depth-detection masks to learn this representation.

I. Introduction

Deep reinforcement learning (DRL) has achieved significant progress in various game environments, such as Go [1], Arcade Games [2], and StarCraft II [3]. It is worth noting that the environments in which DRL has excelled are primarily two-dimensional. In this case, learning control policies directly from raw input pixels is feasible, as raw pixels typically contain almost all the state information of the game. Additionally, 2D games usually have lower-dimensional action-state spaces, implying more manageable exploration and a higher likelihood of learning consistent features. However, the landscape changes significantly when transitioning to 3D games [4]. Firstly, the extra spatial dimensions pose challenges to partial observability. Moreover, dynamic viewpoints introduce variations in scene component sizes and rotations, complicating learning an adequate representation. Most notably, the expansive exploration space characteristic of 3D environments exacerbates the issue of sparse environment feedback. Agents are often confronted with sparse reward problems that can overwhelm those with low exploration efficiency, directly impeding their ability to learn effectively.

ViZDoom is a classic 3D FPS game research platform that allows agents to play Doom games using screen buffers and game variables. Since its release, substantial efforts [5] [6] have been dedicated to investigating optimal strategies for guiding AI agents through visual information. Existing methods [7] simply introduce game scalar features as anchors for input augmentation. Bhatti [4] has achieved good results by utilizing 3D-scene reconstruction and on-the-fly object detection to extract semantic abstractions of scene components. However, this highly engineered architecture has led to a lack of generalization. Song [8] and Huang [9] face similar issues. The former involves designing a divided action space and structured intrinsic rewards, while the latter utilizes depth prediction and enemy detection to guide combo-action. However, both methods heavily rely on the quality and accuracy of human prior knowledge. Any mismatch between a unique situation and existing knowledge can disrupt the effectiveness of intrinsic rewards or auxiliary networks, limiting their capacity to adapt to different scenarios. Current RL algorithms struggle to outperform human performance in 3D FPS games due to the numerous challenges mentioned.

This paper introduces the D2AH-PPO method, which employs a hierarchical structure to enhance learning in 3D visual environments like ViZDoom. It involves a two-level hierarchy. The high-level controller utilizes depth-detection masks to extract crucial components' depth and positional information (as shown in Fig. 1) automatically. This information is transmitted to low-level workers to enhance input following option decisions. These workers are specialized in mastering specific

sub-tasks. Drawing inspiration from autonomous driving [10] and Tencent Solo [11], we introduce the rule-based action mask to establish correlations between output policies and actual actions. This approach effectively reduces ineffective exploration and improves training efficiency. Importantly, to ensure the algorithm's generality and robustness, the rules embedded in the masks are based on universal logic, such as preventing collisions. This design ensures scalability and applicability across various scenarios.

II. Related Work

A. Games AIs.

As the most classic FPS benchmark, ViZDoom [12] features lightweight, fast, and highly customizable characteristics. Three Visual Doom AI Competitions (VDAIC) [13] have been organized to promote the application of RL in shooter games [7] [8]. One standout example is the F1 [14] agent, which successfully implemented a modified version of the A3C algorithm and incorporated curriculum learning. This approach allowed the agent to progressively challenge itself against increasingly difficult opponents, ultimately winning the championship on track 1 in 2016. On the other hand, Clyde [5] attained particular results through reward shaping, while Curiosity [15] utilized intrinsic rewards to hasten the network's convergence by introducing human-designed features and other signals. Moreover, DRL has found extensive applications in various gaming environments. For instance, the Deep Q-Network (DQN) algorithm [16] has demonstrated remarkable success in mastering Atari 2600 games [2] and outperforming world-class Go players [1]. Juewu-mc's [17] proposal suggests that the implementation of A2RL and DSIL technologies can significantly enhance the performance and learning efficiency of sub-policies, resulting in successful diamond mining in Minecraft. Tencent Solo [11] incorporated innovative techniques like control dependency decoupling and action masks into its approach. Additionally, they leveraged a large-scale training system to achieve victory over top professional human players in the most popular MOBA game, Honor of Kings. Pearce [18] employed large-scale behavioral cloning (BC) [19] to play CS:GO.

B. Deep Reinforcement Learning.

Hierarchical reinforcement learning (HRL), which is relevant to our research, enhances learning efficiency through the creation and utilization of a hierarchical framework for cognitive and decision-making processes. Building upon the work of Dayan [20], Vezhnevets [21] introduced a feudal network that can autonomously identify subgoals. Drawing from the concept of options [22], the Option-Critic [23] (OC) model expanded the policy gradient theory to include options, enabling the learning of scalable options applicable to extensive domains. the MaxQ architecture [24] decomposed tasks by decomposing value functions, whereas H-DQN [25] focused on acquiring hierarchical work values across various time scales. Barto [26] believed that HRL offers a natural framework for incorporating principles of intrinsic motivation.

Our work is also related to representation learning, a crucial aspect for enhancing sample efficiency in RL. Previous research by Wu [27] and Lin [17], for instance, leveraged attention-aware masks and action-aware masks, respectively, to highlight crucial information within states, facilitating comprehensive representations of high-dimensional unlabeled data. This approach holds promise for various tasks in a self-supervised manner. Similarly, Srinivas [28] utilized contrastive learning to derive representations with similarity constraints from well-organized datasets comprising both similar and dissimilar pairs. These representations were then utilized in off-policy control learning. Our work proposes a new self-supervised representation learning method to generate depth-detection feature representations of key components within self-supervised learning scenarios. This method contributes to the broader endeavor of enhancing representation learning techniques for RL applications.

III. Preliminaries

Below we briefly introduce the theoretical support for the hierarchical framework and the PPO algorithm [29].

A. Hierarchical Reinforcement Learning

In the context of modeling the action decision process, a standard Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ is considered. Here, $\mathcal{S}$ denotes the space of feasible states and $\mathcal{A}$ represents the space of feasible actions. The function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ serves as the reward function, and $\mathcal{P}(s^{'}|s, a)$ represent the transition probability. Additionally, $\gamma \in (0, 1]$ is referred to as the discount factor.

The Semi-Markov decision process [22] (Semi-MDP) provides a theoretical framework for implementing a hierarchical approach based on options, where the duration between actions (options) is uncertain. Formally, the decision process for the controller can be represented as a semi-MDP denoted as $M_c = (\mathcal{S}, \mathcal{O}, \mathcal{P}_c, \mathcal{R}_E, \gamma, \mathcal{F})$. This is based on multiple worker-based MDP processes denoted as $M_i = (\mathcal{S}, \mathcal{A}, \mathcal{P}_i, \mathcal{R}_E + \mathcal{R}_i, \gamma)$, where $\mathcal{F}(t|s, o)$ is the termination condition which represents the probability that the transition time is t when option o is executed in state s. The controller selects workers i according to its policy p_c and assigns intrinsic rewards R_I^i accordingly.

B. Proximal Policy Optimization (PPO)

The RL agent aims to find an optimal strategy π_o to maximize the cumulative expected return, i.e. the objective $\mathop{\mathbb{E}}_{(s,a)\sim\pi}[\sum_t^{\infty} \gamma^t r(s_t, a_t)]$. The PPO [29] algorithm incorporates a direct clipping mechanism into the objective function used for policy gradients. This approach results in a more conservative update strategy and simplifies the computational process. Let $r_t(\theta)$ denote the probability ratio $\prod_{i=0}^{M-1} \frac{\pi_\theta(a^{(i)}|s)}{\pi_{\theta_{old}}(a^{(i)}|s)}$. The standard PPO algorithm uses a ratio clip function as follows to discipline extreme changes to the policy:

$$\mathcal{J}_\pi^{CLIP}(\theta) = \mathop{\mathbb{E}}_{(s,a)\in\mathcal{B}_\pi}\left[\min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \tag{1}$$

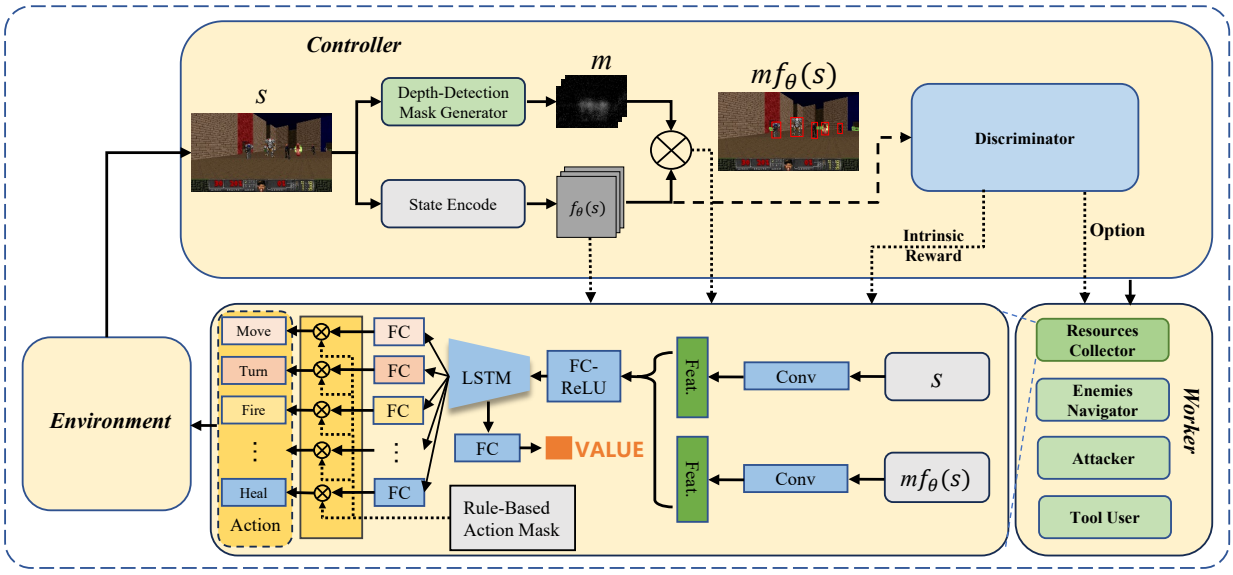


Fig. 2. Our proposed D2AH-PPO architecture. combines a depth-detection mask with input embedding to form the representation of object awareness. Subsequently, 'options' are made based on this input, and intrinsic rewards are computed. The selected worker executes a series of orthogonal actions modified by rule-based action masks, engaging with the environment in alignment with their strategies.

where $\hat{A}_t(s,a) = \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}^V$ is the advantage value computed by generalized advantage estimation (GAE) [30], and $\delta_t^V = R_t + \gamma V(s_{t+1}) - V(s_t)$. ϵ is a truncation factor used to limit the magnitude of updates, ensuring that the difference between old and new policies is not too large.

IV. Algorithm Design

In consideration of the phases of the shooter games, we design a hierarchical network comprised of a controller and workers. Within this framework, we define n sub-tasks based on human cognitive patterns and structure demonstrations, which are used to pre-train the controller ($\mathcal{B}_D \rightarrow \{\mathcal{B}_{D_0}, \mathcal{B}_{D_1}, \ldots, \mathcal{B}_{D_{n-1}}\}$) [31]. Fig. 2 illustrates the architecture. Specifically, the controller determines abstract sub-tasks, or options, at a lower level of time resolution. Meanwhile, the worker assigned to the option performs fundamental game actions at a higher time resolution to complete the sub-tasks provided. Following human habits, the action space is divided to allow for the simultaneous execution of multiple actions. Further, environmental depth-detection information is introduced to facilitate exploration, and intrinsic reward is designed for each worker. Last, we introduce a rule-based action mask embedded with basic logical rules, making it suitable for application in any RL environment. The details of our network are provided in the remaining paragraphs.

A. Hierarchical Framework

In a two-level hierarchical network, the controller serves as a high-level option-maker. Its role involves learning a policy that effectively translates the continuous state space $\mathcal{S}$ into a discrete options space $\mathcal{O} = (0, 1, \ldots, n-1)$. On the other hand, the low-level workers are responsible for learning policies that can accomplish the objectives associated with each option. They interact with the environment, continually observe it, and generate actions until the termination conditions are met.

Negative log-likelihood loss was used in pre-training to help the controller accurately select workers at an early stage:

$$\min_\theta \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{(s,o^i)\in\mathcal{B}_{D_i}} [-\log \pi_\theta^c(o^i|s)], \tag{2}$$

where (s, o^i) is state-option pair sampled from the demonstrations $\mathcal{B}_{D_i}$ and θ represents parameters of the controller's policy π^c. We design four distinct types of workers. Each shares the same network structure and outputs the original game action after decoupling. Details will be presented later on.

Decoupling control dependencies can greatly improve agent actions' flexibility while reducing the action space's complexity. Referring to Song [8], to address the independence of action space, we partition the entire action space $\mathcal{A}$ into $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_M$, where $\mathcal{A}_i$ represents a distinct action subspace where actions are mutually exclusive (e.g., moving forward and backward), the actions within each $\mathcal{A}_i$ are orthogonal (e.g., movement and turning), and M is the number of subspace partitions. Different from Ye [11], let us consider each action $a = (a^0, \ldots, a^{M-1})$. Then the workers' PPO objective without clipping after decoupling is:

$$\max_\theta \mathop{\mathbb{E}}_{(s,a)\in\mathcal{B}_\pi} \left[\left(\prod_{i=0}^{M-1} \frac{\pi_\theta(a^{(i)}|s)}{\pi_{\theta_{old}}(a^{(i)}|s)} \right) \hat{A}_t(s,a) \right] \tag{3}$$

B. Depth-Detection Augmentation

Inspired by previous research [32] [9], we combine ViZDoom's component identification and depth prediction capabilities to augment worker performance **during the training phase**. Unlike 2D scenes, exploration in 3D environments requires a sense of distance, so depth information is necessary

to guide the agents. Various components are segmented from the depth map to create a depth-detection map $D(s)$, where s is the current state of the environment.

The depth-detection map is a valuable tool for workers to extract and comprehend high-level information effectively. Nonetheless, it is important to highlight that the competition guidelines allow only the use of the original pixel input for evaluation. We suggest acquiring this representation to replace the additional input to address this limitation. Specifically, our proposal involves training a mask network on the feature maps of each critical component. This approach aims to capture components' position and distance information within the current input state. Denote the feature maps as $f_\theta(s) \in \mathbb{R}^{H\times W}$ and the mask network as $m_\phi(s) \in [0,1]^{H\times W}$, where θ and ϕ correspond to the parameters of the neural network for the policy and the mask network, respectively. Given the state s, the objective function for training the mask is as follows:

$$\mathcal{J}_m^1(\phi) = \mathop{\mathbb{E}}_{s\in\mathcal{B}_\pi}[-\|g_\psi(m_\phi(s)\odot f_\theta(s)) - f_\theta(D(s)\cdot s)\|_2], \quad (4)$$

$$\mathcal{J}_m^2(\phi) = \|m_\phi(s)\|_1, \quad (5)$$

$$\mathcal{J}_m(\phi) = \mathcal{J}_m^1(\phi) + \lambda_m \mathcal{J}_m^2(\phi). \quad (6)$$

Eq 4 is designed to maximize the preservation of crucial components in the embedded information after masking, while Eq 5 serves as a regularization term that prevents the mask from excessively covering multiple regions. The hyper-parameter λ_m plays a critical role in balancing these dual goals of retaining important information and avoiding data saturation caused by the mask. g_ψ is a projection function trained using the following objective function:

$$\mathcal{J}_g(\psi) = \mathop{\mathbb{E}}_{s\in\mathcal{B}_\pi}[\|g_\psi(f_\theta(s)) - f_\theta(D(s)\cdot s)\|_2]. \quad (7)$$

This objective function learns to evaluate the information content of crucial components in the embedding. g_ψ is pre-trained and fine-tuned when optimizing the mask network.

The learned representation can replace the depth-detection map to enhance any RL algorithm. For example, within policy-based PPO methods, integrating the learned representation into the policy $\pi_\theta(a|(1+m_\theta(s,a))\odot f_\theta(s))$ can facilitate more effective back-propagation and updates of policy gradients.

C. Intrinsic Reward Shaping

Combining intrinsic reward based on workers' utility with a hierarchical structure is a successful approach for addressing issues related to reward delay and sparse reward [8]. The distance information can be obtained from the depth map, and the intrinsic reward for each worker is designed as follows:

Attacker is responsible for aiming and shooting. Consequently, its intrinsic rewards consist of a positive reward for the proximity of the crosshair to the enemy and a negative reward for the enemy disappearing from view:

$$R_I^{att}(s,a,s') = \begin{cases} 0, & if\ n_e = 0\ or\ n_e' = 0 \\ 1, & if\ \max\limits_{i=0}^{n_e} d_{att,i}(s) > \max\limits_{j=0}^{n_{e'}} d_{att,j}(s') \\ -1, & otherwise \end{cases}, \quad (8)$$

where n_e is the number of enemies on the screen, and $d_{att} = (d_{att,1}, \ldots, d_{att,n_e})$ is the distance between enemies and crosshair.

Resources Collector collects resources such as guns, ammo, and medicine by walking around the map. So its intrinsic rewards should include a positive reward for finding and getting close to the item and a negative reward for staying away from the resources:

$$R_I^{res}(s,a,s') = \begin{cases} 0, & if\ n_r = 0\ or\ n_r' = 0 \\ 1, & if\ \max\limits_{i=0}^{n_r} d_{res,i}(s) > \max\limits_{j=0}^{n_{r'}} d_{res,j}(s') \\ -1, & otherwise \end{cases}, \quad (9)$$

where n_r is the number of resources on the screen, and $d_{res} = (d_{res,1}, \ldots, d_{res,n_r})$ is the distance to the resources.

Enemies Navigator is responsible for finding enemies, so a positive reward for finding an enemy:

$$R_I^{ene}(s,a,s') = \begin{cases} 1, & n_e' - n_e \geq 1 \\ 0, & otherwise \end{cases}. \quad (10)$$

Tools User is exclusively authorized to utilize tools (such as medicine), so it receives an additional positive reward for successful tool usage and a negative reward for being attacked while using the tool.

$$R_I^{too}(s,a,s') = \begin{cases} 1, & use\ successful \\ -1, & injured \\ 0, & else \end{cases}. \quad (11)$$

For worker i, its reward function is $R_t^i = R_{I,t}^i + R_{E,t}$. As the controller runs at a slower time scale, its reward function is calculated as the sum of extrinsic rewards accumulated during the worker's period: $R_t^c = \sum\limits_{t'=t}^{t+N} R_{E,t'}$.

D. Rule-Based Action Masks

Based on universal logic in FPS games and the basic prior knowledge of human players, we propose a rule-based action mask to eliminate the discrepancies between workers' final strategies and the actual situation. The judgment method for situations refers to Song [8]. For example, our masks obey the rules including but not limited to the following aspects: 1) avoid collisions with prohibited walls or obstacles directly in your path; 2) refrain from firing weapons without discretion when there are no visible enemies; 3) ensure appropriate utilization of items by avoiding the unnecessary use of healing supplies when health levels are optimal; 4) take care not to discard equipment or items that could potentially lead to self-weakening. Experiments show that our action mask helps reduce exploration in RL and speeds up training efficiency.

V. Experiments

A. Experiments Setup

In our experimental setup, we utilize the ViZDoom platform for conducting our research. Specifically, we focus on **the full deathmatch** scenario, which involves unknown maps adapted

TABLE I
PERFORMANCE OF THE AGENTS IN THE TEST MAPS.

Player	Frags	F/D ratio	Kills	Suicides	Deaths
F1	55.0	3.24	56.7	**1.7**	17.0
Arnold	51.0	**4.78**	53.3	2.3	**10.7**
CLYDE	45.7	2.63	47.7	2.0	17.3
Human	39.0	2.17	45.7	6.7	18.0
D2AH-PPO	**58.0**	4.05	**62.7**	4.7	14.3

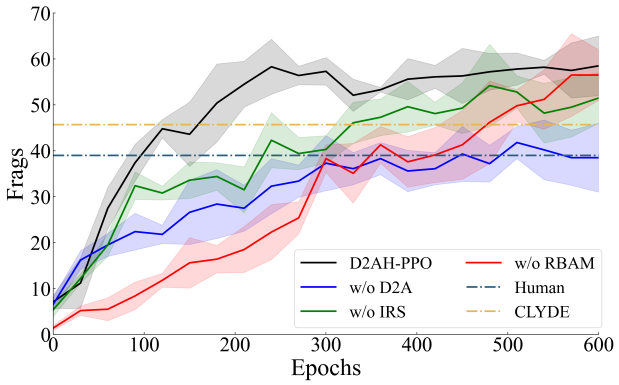


Fig. 3. Ablation study of our proposal. Each experiment was conducted with three different random seeds, and we presented their means and variances.

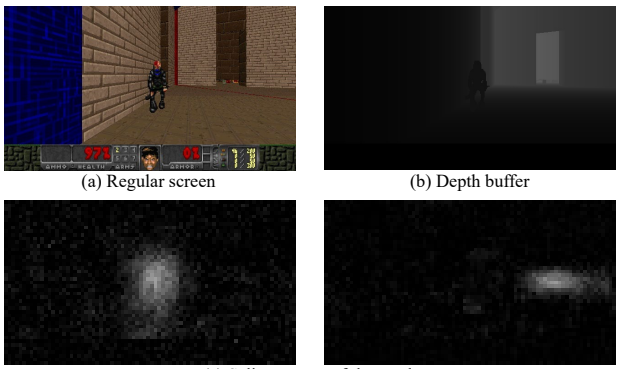
from VDAIC 2016. During the experiments, agents are trained and evaluated on various maps. At the beginning of each scenario, agents are equipped with a pistol and can acquire different weapons and items, including ammunition, medical kits, and armor. To ensure consistency and comparability in our experiments, we employed PyOblige [12] to create a set of seven maps for training purposes and three maps for testing. It is noteworthy that all generated maps share similar levels of difficulty and textures, providing a standardized environment for our evaluations.

To ensure a fair comparison, we utilize the same resource configuration and fixed hyperparameters for all experiments. All algorithms are run on the same machine, equipped with 24 Intel(R) 4310 (2.1GHz) CPU cores and 1 3090 GPU card. The initial learning rate for the Adam optimizer is set to 1e-4, and the reward discount factor γ is set as 0.99. Additionally, we set $\lambda = 0.95$ in the GAE.

B. Performance Evaluation

Following the VDAIC guidelines, we employed Frags, calculated as the difference between kills and suicides, as a **evaluation metric** for evaluation. Additionally, we reported the number of kills, suicides, deaths, and Frag to death (F/D) ratios to further analyze the agent's abilities. Our agent was benchmarked against F1 [14], Arnold [7], CLYDE [5], and human players for comparative analysis.

Table I shows the agents' performance in the test maps, each map was tested for 15 minutes, and the final performance was averaged. D2AH-PPO achieved the highest Frags score, 5.5% higher than the second place, 48.7% higher than humans, and exhibited a higher rate of both suicides and deaths while also



(a) Regular screen (b) Depth buffer

(c) Saliency map of the mask

Fig. 4. Saliency map for the learned mask. (a) regular screen buffer; (b) depth buffer; (c) saliency map of the learned mask.

achieving a higher number of kills. This may be attributed to the intrinsic reward shaping being more offensive.

The experiment findings indicate a significant enhancement in our approach regarding sample efficiency, stability, and overall performance. Upon analyzing the experimental replay records, a notable observation was made regarding certain baselines, like Arnold, which displayed proficiency in resource location but struggled with more intricate tasks like precise aiming and effective shooting. Specifically, when confronted with adversaries possessing limited hit ranges, these agents exhibited persistence in their attacks but demonstrated poor accuracy in hitting their targets. Consequently, this inadequacy in target accuracy often led to their defeat by the enemies.

C. Ablation Study

To evaluate the efficacy of the proposed techniques, we analyzed three variations of our approach: 1) without depth-detection Augmentation (w/o D2A), 2) without intrinsic reward shaping (w/o IRS), and 3) without rule-based action masks (w/o RBAM). The training curves for each variant are illustrated in Fig. 3. We found that each introduced technique has substantially contributed to the overall performance. Among them, D2A has the greatest impact on the final performance of the agents because it directly enhances the input, significantly improving the decision-making and control abilities of the agents in later stages, such as more precise shooting. We also observed that RBAM and IRS have significantly improved the convergence speed of agent training. Particularly, RBAM demonstrates a notable impact in accelerating the early learning process. This is due to its ability to guide agents towards exploring more rational action trajectories in the initial stages of training. It is worth mentioning that regardless of the variant, the final performance has reached human levels, and two variants outperform the CLYDE, further demonstrating the effectiveness of our approach.

D. Visualization

We also employ a comprehensive visual analysis to gain a deeper understanding of our methods' effectiveness. Initially, we analyze the learned mask within the D2AH-PPO

framework by generating saliency maps [33] for the depth-detection mask generator. To be more precise, to illustrate the most significant parts of the images as perceived by the mask network, we calculate the absolute value of the Jacobian $|\nabla_s m_\phi(s)|$. The visualizations depicting these saliency maps for a particular state are presented in Figure 4.

We found that the trained mask can effectively capture positional and distance information of distinct component types. This insight sheds light on the attention mechanism within the model, highlighting its emphasis on crucial components. More specifically, the mask network demonstrates an ability to prioritize crucial component information within the current state, thereby enhancing the input for the workers' network. The learned mask can facilitate a deeper understanding of the 3D environment for the agent, ultimately leading to improved learning and control over actions.

VI. Conclusion

This paper proposes D2AH-PPO, a hierarchical RL framework designed for playing ViZDoom. Our approach consists of a high-level controller and several low-level workers that employ representation learning to acquire high-dimensional abstractions of the game environment components. Additionally, the introduction of intrinsic rewards and rule-based action masks serves to enhance the performance and learning efficiency of the sub-policies. Experiments demonstrate that our framework outperforms all solutions in the past VDAIC competitions. In future work, we aim to apply D2AH-PPO to other FPS games, as well as other 3D open-world games.

VII. Acknowledge

This work was supported by the National Key Research and Development Program of China under Grant 2021YFE0205700, Beijing Natural Science Foundation JQ23016, the External cooperation key project of Chinese Academy Sciences 173211KYSB20200002, the Science and Technology Development Fund of Macau Project 0123/2022/A3, and 0070/2020/AMJ.

References

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[3] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[4] S. Bhatti, A. Desmaison, O. Miksik, N. Nardelli, N. Siddharth, and P. H. Torr, "Playing doom with slam-augmented deep reinforcement learning," *arXiv preprint arXiv:1612.00380*, 2016.

[5] D. Ratcliffe, S. Devlin, U. Kruschwitz, and L. Citi, "Clyde: A deep reinforcement learning doom playing agent," 2017.

[6] A. Dosovitskiy and V. Koltun, "Learning to act by predicting the future," *arXiv preprint arXiv:1611.01779*, 2016.

[7] G. Lample and D. S. Chaplot, "Playing fps games with deep reinforcement learning," in *AAAI*, vol. 31, no. 1, 2017.

[8] S. Song, J. Weng, H. Su, D. Yan, H. Zou, and J. Zhu, "Playing fps games with environment-aware hierarchical reinforcement learning." in *IJCAI*, 2019, pp. 3475–3482.

[9] S. Huang, H. Su, J. Zhu, and T. Chen, "Combo-action: Training agent for fps game with auxiliary tasks," in *AAAI*, vol. 33, no. 01, 2019, pp. 954–961.

[10] E. Medvet, A. Bartoli, and J. Talamini, "Road traffic rules synthesis using grammatical evolution," in *EvoApplications 2017*. Springer, 2017, pp. 173–188.

[11] D. Ye, Z. Liu, M. Sun, B. Shi, P. Zhao, H. Wu, H. Yu, S. Yang, X. Wu, Q. Guo *et al.*, "Mastering complex control in moba games with deep reinforcement learning," in *AAAI*, vol. 34, no. 04, 2020, pp. 6672–6679.

[12] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaśkowski, "Vizdoom: A doom-based ai research platform for visual reinforcement learning," *IEEE*, 2016.

[13] M. Wydmuch, M. Kempka, and W. Jaśkowski, "Vizdoom competitions: Playing doom from pixels," *IEEE ToG*, vol. 11, no. 3, pp. 248–259, 2018.

[14] Y. Wu and Y. Tian, "Training agent for first-person shooter game with actor-critic curriculum learning," in *ICLR*, 2016.

[15] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *ICML*. PMLR, 2017, pp. 2778–2787.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *Computer Science*, 2013.

[17] Z. Lin, J. Li, J. Shi, D. Ye, Q. Fu, and W. Yang, "Juewu-mc: Playing minecraft with sample-efficient hierarchical reinforcement learning," *arXiv preprint arXiv:2112.04907*, 2021.

[18] T. Pearce and J. Zhu, "Counter-strike deathmatch with large-scale behavioural cloning," in *CoG*. IEEE, 2022, pp. 104–111.

[19] M. Bain and C. Sammut, "A framework for behavioural cloning." in *Machine Intelligence 15*, 1995, pp. 103–129.

[20] P. Dayan and G. E. Hinton, "Feudal reinforcement learning," *NIPS*, vol. 5, 1992.

[21] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, "Feudal networks for hierarchical reinforcement learning," in *ICML*. PMLR, 2017, pp. 3540–3549.

[22] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[23] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," in *AAAI*, vol. 31, no. 1, 2017.

[24] T. G. Dietterich, "Hierarchical reinforcement learning with the maxq value function decomposition," *Journal of artificial intelligence research*, vol. 13, pp. 227–303, 2000.

[25] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," *NIPS*, vol. 29, 2016.

[26] A. G. Barto, "Intrinsic motivation and reinforcement learning," *Intrinsically motivated learning in natural and artificial systems*, pp. 17–47, 2013.

[27] H. Wu, K. Khetarpal, and D. Precup, "Self-supervised attention-aware reinforcement learning," in *AAAI*, vol. 35, no. 12, 2021, pp. 10 311–10 319.

[28] M. Laskin, A. Srinivas, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," in *ICML*. PMLR, 2020, pp. 5639–5650.

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[30] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[31] H. Le, N. Jiang, A. Agarwal, M. Dudík, Y. Yue, and H. Daumé III, "Hierarchical imitation and reinforcement learning," in *ICML*. PMLR, 2018, pp. 2917–2926.

[32] F. G. Glavin and M. G. Madden, "Adaptive shooting for bots in first person shooter games using reinforcement learning," *IEEE Trans. Comp. Intell. AI Games*, vol. 7, no. 2, pp. 180–192, 2014.

[33] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.