Hindawi
Complexity
Volume 2021, Article ID 6698231, 9 pages
https://doi.org/10.1155/2021/6698231

Research Article

AIBPO: Combine the Intrinsic Reward and Auxiliary Task for 3D Strategy Game

Huale Li, Rui Cao, Xuan Wang, Xiaohan Hou, Tao Qian, Fengwei Jia, Jiajia Zhang, and Shuhan Qi

School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

Correspondence should be addressed to Jiajia Zhang; zhangjiajia@hit.edu.cn and Shuhan Qi; shuhanqi@cs.hitsz.edu.cn

Received 15 October 2020; Accepted 2 July 2021; Published 14 July 2021

Academic Editor: Abd E.I.-Baset Hassanien

In recent years, deep reinforcement learning (DRL) achieves great success in many fields, especially in the field of games, such as AlphaGo, AlphaZero, and AlphaStar. However, due to the reward sparsity problem, the traditional DRL-based method shows limited performance in 3D games, which contain much higher dimension of state space. To solve this problem, in this paper, we propose an intrinsic-based policy optimization (IBPO) algorithm for reward sparsity. In the IBPO, a novel intrinsic reward is integrated into the value network, which provides an additional reward in the environment with sparse reward, so as to accelerate the training. Besides, to deal with the problem of value estimation bias, we further design three types of auxiliary tasks, which can evaluate the state value and the action more accurately in 3D scenes. Finally, a framework of auxiliary intrinsic-based policy optimization (AIBPO) is proposed, which improves the performance of the IBPO. The experimental results show that the method is able to deal with the reward sparsity problem effectively. Therefore, the proposed method may be applied to real-world scenarios, such as 3-dimensional navigation and automatic driving, which can improve the sample utilization to reduce the cost of interactive sample collected by the real equipment.

1. Introduction

Machine game has always been one of the most active domains of artificial intelligence. According to whether the game states are fully observable, the machine game can be divided into two categories: perfect information game (PIG) and imperfect information game (IIG). The PIG refers to the game that participants can observe all game states, such as Go and Chess. On the contrary, the participants always hold private information in the IIG. Thus, the game state cannot be fully observed for the player, such as hole cards in Poker games. Deep reinforcement learning (DRL) has achieved great success in the field of the PIG recently [1–5]. However, due to the hidden information, the DRL-based methods have not shown satisfactory results in the IIG. The IIG is still a significant challenge in the field of machine games.

In recent years, many new testing platforms have emerged to verify methods of the IIG [6, 7]. In these platforms, the strategy game includes all the base elements of the IIG, such as StarCraft and VizDoom. Furthermore, the strategy game contains rich game scenarios and a considerable game state space [8–10]. Thus, the strategy game becomes a kind of ideal platform to verify relevant methods of the IIG. In view of the great success achieved by the DRL in the PIG, researchers have investigated intensively on solving strategy games through the DRL methods in the IIG. In 2019, DeepMind's AlphaStar, based on the DRL, became the first agent that defeated a professional human player in the StarCraft game, which has been viewed as a milestone in the advancement of artificial intelligence [11]. In addition, DeepMind designed an agent that can achieve human-level performance in a 3D multiplayer first-person video game, Quake III Arena in Capture the Flag mode, which only used pixels and game scores as the input [9].

Although the DRL-based methods have made certain achievements in the research of solving the strategy games, there are still many problems to be solved. The DRL methods highly rely on the reward to update the model during the

training process. Reasonable and instant rewards not only make the training process converge quickly but also make the learned model more robust. In the game scenario where the reward is sparse, the speed of the convergence will be slow down seriously and even more, resulted in a divergent model finally. In the 3D strategy games with a higher dimension of state space and more complex game scenarios, the reward sparsity problem will become more severe. Although there is a clear reward (e.g., the score of the game) in such game scenarios, most of the final goals need to be achieved by defining a series of subgoals in the early stage. Most of subgoals may not get rewards in time unless they are finished.

The reward sparsity problem makes the training process be much more difficult and slower because it cannot get a timely and effective reward. To solve this problem, many solutions such as reward reshaping, hierarchical DRL, and curriculum learning are proposed in recent years [12–18]. Among these methods, the intrinsic reward mechanism plays an important role. The design of the corresponding internal reward mechanism helps the agent update its policy according to the internal reward in the absence of environmental reward information. The definition of this RL model which includes intrinsic incentive reward is intrinsic incentive reinforcement learning (IIML). The general model of the IIML is shown in Figure 1. In the IIML, the external environment is the scene of the RL agent. The internal reward generation model is a fictitious environment. The agent can get the reward from external environment and intrinsic reward generation model, respectively. In this way, the agent gets more reward information than usual, which can accelerate the convergence of the agent.

However, in the process of practical application, these methods require carefully designed rewards for related problems or additional training data, which greatly limits the applicability of related methods. Therefore, these methods can not deal with the problem of sparse reward very well. To deal with the problem of reward sparsity, this paper improves the traditional strategy optimization DRL methods by introducing intrinsic reward mechanism. We proposed an intrinsic-based policy optimization (IBPO) algorithm, which improves the exploration performance of the agent in the 3D scene with the sparse reward. In this algorithm, the intrinsic reward is used to combine with the traditional strategy optimization. Moreover, by designing the various auxiliary tasks, we further proposed the auxiliary intrinsic-based policy optimization (AIBPO), which further improves the IBPO. The experimental results tested on the VizDoom show that our method has a better performance than the previous methods. We summarize our contributions as follows:

(i) We proposed a method, intrinsic-based policy optimization (IBPO) algorithm, to solve the reward sparsity problem in the 3D games with imperfect information. The IBPO effectively improves the exploration performance of the agent by combining the intrinsic reward and the traditional strategy optimization method.

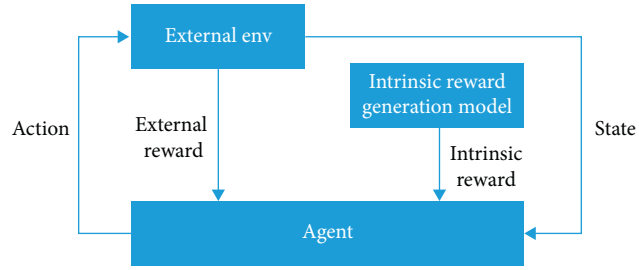


Figure 1: The model of the IIML (the agent takes the action according to the policy to obtain the external reward information fed back by the external environment; meanwhile, internal reward information will be generated in the internal environment no matter whether the current state agent gets external reward or not).

(ii) We presented the auxiliary intrinsic-based policy optimization (AIBPO), which is based on the IBPO, by integrating three kinds of auxiliary tasks: reward prediction auxiliary task, action value auxiliary task, and state value auxiliary task.

(iii) Extensive experimental results show that the performance of our method IBPO is significantly improved on VizDoom, compared with the previous methods. Moreover, the AIBPO further improves the performance of the IBPO.

The paper is organized as follows. The research method and its details are discussed in Section 2. Then, Section 3 shows the experimental results including performance comparsion with the state-of-the-art methods and ablation studies. Finally, the conclusion of this paper is given in Section 4.

2. Research Method

In this section, we seek a method of training the agent that can address the reward sparsity in the 3D game with complex environment. Intrinsic-based policy optimization (IBPO) is proposed by introducing an intrinsic reward mechanism. Moreover, combining with auxiliary tasks (reward prediction auxiliary task, action value auxiliary task, and state value auxiliary task), auxiliary intrinsic-based policy optimization (AIBPO) is presented based on the IBPO.

2.1. Intrinsic-Based Policy Optimization (IBPO). Due to the sparse reward problem in 3D strategy games, it is very difficult for the agent to carry out policy iteration and update. To overcome this problem, in this paper, we introduce the concept of intrinsic rewards, which can provide auxiliary reward information for the agent to update its policy. That is to say, here we employ an IIML (as shown in Figure 1) model. In this model, there are not only external rewards from the environment but also intrinsic rewards that are generated by a designed intrinsic rewards generation module. In addition, to adopt the intrinsic reward into the traditional policy optimization framework, we propose the intrinsic-based policy optimization (IBPO) algorithm, in

which the intrinsic reward generation module can be integrated with the policy optimization framework seamlessly.

2.1.1. The Generation of Intrinsic Reward. We describe how the intrinsic reward is designed. As mentioned above, the IIML contains the external reward and the intrinsic reward. The external reward is feedback from the environment, which cannot be obtained sometimes. When the agent sinks into a situation of lacking external reward, the intrinsic reward plays a key role in helping the agent to continue to update its policy. Especially, in the 3D game, if an external reward is sparse, the agent will easily fall into the problem of lingering in a local region. Therefore, to avoid the local region problem, the agent should increase its curiosity to explore unseen scenes. Based on this consideration, this paper constructs the intrinsic reward by estimating the novelty degree of the state. The novelty of a state means that the agent has not met this state before. A certain degree given to the new state can be viewed as the intrinsic reward, and the agent has a chance to overcome the local region problem.

Based on the above assumption, here we design the intrinsic reward generation module as follows. The intrinsic reward generation module is composed of a dual network structure as shown in Figure 2(a). In this structure, there is a target mapping network and a prediction network. Here, we set the target mapping network as a fixed network and the prediction network as a trainable network. Then, the intrinsic reward value is adopted as the similarity between the output vector of the target mapping network and prediction network. The input of these two networks is the current state of the game. That is to say, if the game state is novel, then the intrinsic reward will be a large value and vice versa. The main reason for this is to make the agent tend to explore unseen states in the environment. In the initial training phase, the agent has a smaller range of motion and there are more unfamiliar states in the game. In this case, the output vector of the two networks is less similar; thus, the calculated intrinsic reward value is larger. When the action policy is updated, the agent takes intrinsic reward as the main source of the reward.

Here, we describe the detail of constructing the target mapping network and the prediction network. A three-layer convolutional neural network is used to extract features from the input state, and finally a vector represented with a fixed dimension is output. The same network structure is adopted to reduce the error effect of different network structures on computing the similarity of the vector. The loss function L_{IR} of the intrinsic reward generation module is defined as follows:

$$L_{\mathrm{IR}} = \frac{1}{n}\sum_{i=1}^{n}(P-T)^2 + \lambda_{\mathrm{IR}}\sum_{j=1}^{k}\left\|\theta_j\right\|^2, \tag{1}$$

where P and T are the output vector of the prediction vector and the target vector, respectively, θ_j is the regularization term, λ_{IR} is the penalty factor of regular term, and n and j are the time steps.

2.1.2. Integrate the Intrinsic Reward into Policy Optimization. To train the intrinsic reward prediction network, adequate samples of the agent are necessary. Meanwhile, the agent's action in the environment is always driven by the policy optimization algorithm. In this way, it is a key to combine the intrinsic reward generation module with the policy gradient or policy optimization algorithm.

For the traditional policy optimization algorithm, the update of the policy mainly depends on the external rewards generated via the interaction with the environment. Therefore, to deal with the external reward and the internal reward is the key point of adapting the internal reward mechanism to the policy optimization algorithm. In this paper, we adopt a combination of long-term intrinsic reward and episodic external reward, as shown in Figure 2(b). The long-term reward can give the agent a goal. Meanwhile, it may lead to the reward sparse problem, which makes the strategy unable to converge or converge slow down. Therefore, the internal reward is introduced through the continuous incentive agent to explore, which will continue to support the agent to achieve its goal.

2.2. Auxiliary Intrinsic-Based Policy Optimization (AIBPO). The agent in 3D games typically requires a phased and long-term sequence to make decisions. The decision needs to be highly relevant to the current state, and the agent's action can make the state changeable. However, the state value of the DRL methods is estimated by the neural network, and the estimation is biased. From this perspective, we try to evaluate the state value and the action more accurately in 3D scenes. In this work, we designed three auxiliary tasks (reward prediction auxiliary task, action value auxiliary task, and state value auxiliary task) based on a multitask learning mechanism to perceive the information such as the reward and the connection between adjacent states, which can assist the agent in learning policy in 3D scenes.

The AIBPO with auxiliary tasks based on the IBPO is presented. Three kinds of auxiliary learning tasks are provided for the IBPO. The task can enhance agent's perception of the environmental reward to help the agent make decisions. The experience replay of the DRL is used, and the interactive sample is applied in the IBPO to train auxiliary tasks. The agent's decisions in 3D scenes are determined by learning and updating the parameters of the policy network in the DRL. The proposed auxiliary tasks can provide extra decision-support information for the agent. That is why the different auxiliary tasks can optimize the primary policy and auxiliary policy more effectively and robustly in 3D scenes.

The auxiliary task of training requires sampling the data that are related to the corresponding state, reward, and action. The IBPO agent stores these data in the experience replay memory during training process. In this paper, three kinds of auxiliary tasks (reward prediction auxiliary task, action value auxiliary task, and state value auxiliary task) are adopted by the AIBPO to assist decision-making. The auxiliary intrinsic-based policy optimization (AIBPO) can be defined as follows:

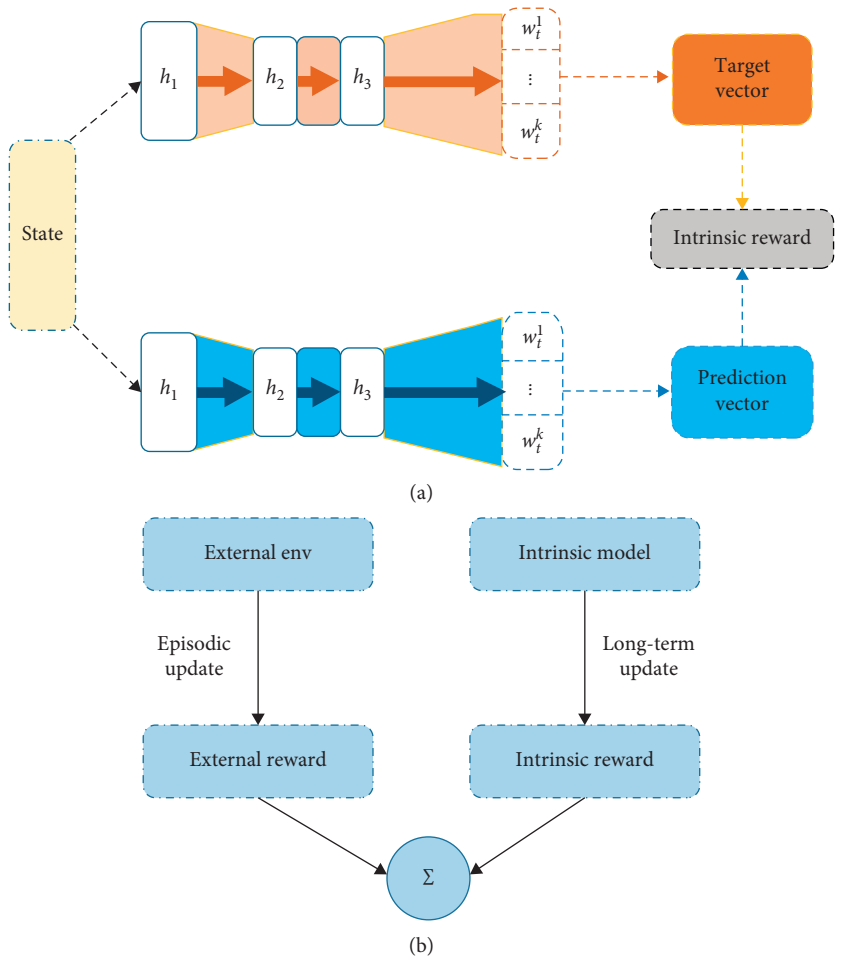


Figure 2: The structure of the IBPO (in subfigure (a), there are two channels: one is to output the target vector and the other is to output the prediction vector, which forms the intrinsic reward through the combination of them; subfigure (b) is the reward integration of the two rewards): (a) intrinsic reward generation module; (b) differentiated reward integration.

$$\mathscr{L}_{\text{AIBPO}} = \mathscr{L}_{\text{IBPO}} + \lambda_P \mathscr{L}_P + \lambda_V \mathscr{L}_V + \lambda_C \mathscr{L}_C, \tag{2}$$

where λ_P is the parameter of the reward prediction task, λ_V is the parameter of the state value task, λ_C is the parameter of the action value task, $\mathscr{L}_P$ is the loss function of the reward prediction task, $\mathscr{L}_V$ is the loss function of the state value task, $\mathscr{L}_C$ is the loss function of the action value task, and $\mathscr{L}_{\text{IBPO}}$ is the loss function of the intrinsic reward policy optimization algorithm.

Equation (2) gives the overall framework of the AIBPO. The AIBPO can learn useful information provided by different auxiliary tasks during training. This information is either related to the policy update or related to the scene perception, which improves the policy optimization in 3D scenes from different perspectives. Furthermore, the weight parameters in the auxiliary task loss function are set to determine the impact degree of the auxiliary task on the primary task. The integral model structure of the AIBPO is shown in Figure 3. The AIBPO is conducted by the whole process. The agent based on the IBPO interacts with the environment to generate more data, which are saved in the experience pool. The auxiliary tasks can be trained through sampling the experience pool.

Our goal is to add specific auxiliary tasks and finally pick up the best auxiliary tasks. To that effect, we prove that the optimal effect can be achieved by integrating the following three auxiliary tasks. The three auxiliary tasks are described in the following, respectively.

2.2.1. Reward Prediction Auxiliary Task. In the reward prediction auxiliary task, three consecutive frames are sampled as the network input. Through the convolutional layer and fully connected layer, the network outputs a classification category of rewards obtained by the agent, including positive rewards, negative rewards, and zero rewards. The label for the classification task is the one-hot encoding corresponding to the reward sampled from the experience replay memory at the next timestep. Because the multiclass cross-entropy loss function is used for classification tasks, here the loss function $\mathscr{L}_P$ in the reward prediction network can be defined as follows:

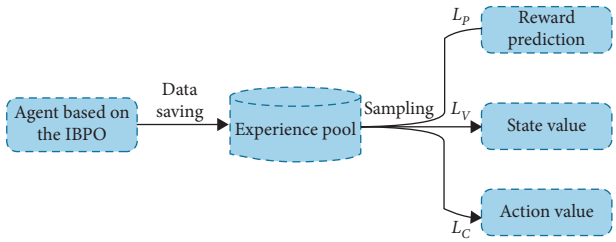


Figure 3: The model structure of the AIBPO.

$$\mathscr{L}_P = -\sum_{i=1}^{n} y_i \log\left(y_{i'}\right), \tag{3}$$

where y_i is the category of the network output and $y_{i'}$ is the reward value at the next moment.

2.2.2. State Value Auxiliary Task. The advantage estimation calculated by the method can be more accurate if the auxiliary policy can make the DRL methods tend to produce more accurate state value estimation. The auxiliary policy can make the training process of the agent more stable and the learning of the primary policy more efficient. Therefore, a state value auxiliary task whose regression prediction label is the state value corresponding to the image at the next timestep is presented. According to the mean square error loss function used for regression tasks, the loss function $\mathscr{L}_V$ in the state value network can be defined as follows:

$$\mathscr{L}_V = \frac{1}{n}\sum_{i=1}^{n}\left(v_i - v_{i'}\right)^2 + \lambda_V \sum_{j=1}^{k}\left\|\theta_j\right\|^2, \tag{4}$$

where θ_j is the regularization parameter, v_i is the state value of the network output, $v_{i'}$ is the target state value, and λ_V is the regularization penalty factor.

2.2.3. Action Value Auxiliary Task. The agent can be more prone to learn the primary policy that obtains the reward feedback efficiently if the auxiliary policy can make the selected action with greater value. Thus, an action value auxiliary task is presented here. In this task, the action value stored in the training process of the IBPO is sampled. The continuous frames and the temporal-difference action value can be taken as the data sample. According to the mean square error loss function used for regression tasks, the loss function $\mathscr{L}_C$ in the action value network can be defined as follows:t

$$\mathscr{L}_C = \frac{1}{n}\sum_{i=1}^{n}\left(Q_i - Q_{td}\right)^2 + \lambda_C \sum_{j=1}^{k}\left\|\omega_j\right\|^2, \tag{5}$$

where λ_C is the penalty factor of the regularization term, ω_j is the parameter of the regularization term, Q_i is the action value of the network output, and Q_{td} is the action value of the temporal difference.

3. Experimental Results

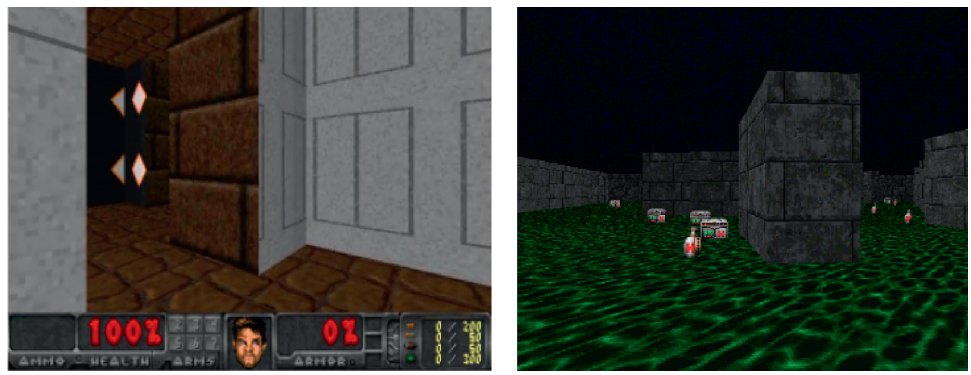
In this section, we first introduce the VizDoom. Secondly, implementation details are described. Finally, we conduct experiments to evaluate the performance of the IBPO and AIBPO. In addition, the ablation study of the AIBPO is also carried out.

3.1. Description of VizDoom. VizDoom is a first-person shooter game [8]. The agent's actions in the scene are similar to the real world, where they receive visual signals and then make decisions. As a mainstream research platform for the DRL methods, the VizDoom platform provides an interface to receive action input and reward signals and simulates the environment in a DRL model. Comprehensively, VizDoom is a platform capable of training agents to explore 3D environments. In this paper, experiments were conducted based on VizDoom's pathfinding and survival scenarios, which are described in Figure 4.

3.1.1. Pathfinding Scenario. Rewards are sparse in the pathfinding scenario, as shown in Figure 4(a). The agent can only get the reward at the target but not at any other location. The entire map is made up of several different opaque rooms with only one fixed target spot in a specific room. In this scenario, the agent can move freely but the starting position is far away from the target. The agent needs to pass through rooms with different contents to obtain rewards at the target spot.

3.1.2. Survival Scenario. There are mazes in the survival scenario, as shown in Figure 4(b), which block the agent from seeing a wide range of scenes. Moreover, the agent may continue to lose the life value during movement in this scenario. Therefore, it acquires the agent to explore and motion as efficiently as possible in the maze; otherwise, the training round would end when the life value turns 0. Besides, drug packages can restore life value appear randomly in the scenes. The agent needs to collect as many drug packages as possible to survive longer.

Different scenarios in the VizDoom platform have different environmental parameters and reward ranges. The primary evaluation indicators of the pathfinding scenario are the success rate of the agent's pathfinding and the number of required action steps. However, in a single training round, the agent cannot move on permanently. The maximum number of action steps is limited to 256 in the pathfinding scenarios. Once more than 256 steps, the current training episode ends immediately. The longer the agent spends in the pathfinding scenario, the weaker its exploration ability to be. Thus, the external environment will give the agent a penalty signal according to the time step. For the survival scenario, the primary evaluation indicators are the survival time step and the number of life value packages obtained. The agent's life value in survival scenarios decreases over time, and the decrease degree is presented by the platform.



(a) (b)

Figure 4: VizDoom's (a) pathfinding scenario and (b) survival scenario.

Table 1: The network structure.

NN1	Parameter	NN2	Parameter	NN3	Parameter
Conv	Conv (64, 3, 3)	Conv	Conv (128,3,3)	Conv	Conv (64, 3, 3)
Conv	Conv (32, 3, 3)	Conv	Conv (32,3,3)	Conv	Conv (32, 3, 3)
Active	ReLU	Active	BN	Fc	Linear (64, 3), reward prediction
Transform	Flatten	Active	ReLU	Fc	Linear (64, 3), reward prediction
Fc	Linear (288, 256)	Transform	Flatten	LSTM	Hidden state 256, state value
Fc	Linear (256, 256)	Fc	Linear (288, 256)	Fc	Linear (256, 1), state value
Actor	Linear (256, 4), a			LSTM	Hidden state 256, action value
Critic	Linear (256, 1), r			Fc	Linear (256, 1), action value
Critic	Linear (256, 1), r				

3.2. Implementation Details. The network structure of the adjusted policy optimization is shown in Table 1. The NN1 includes the actor network and the critic network [19], both of which share part of the hidden layer neurons. The critic network designed in this paper has two independent reward output heads to offer independent supervision information to the external reward and the intrinsic reward.

The intrinsic reward generation model makes up for the lack of the external reward when updating. The model includes a target mapping network and a prediction network, both of which have the same network structure. In Table 1, NN2 states the network structure of the intrinsic reward generation model.

The reward feature enhancement with auxiliary task learning enhances the agent's performance in 3D games by building their capacity on perception and state estimation. The samples for three proposed auxiliary tasks are obtained from the experience replay memory, which are stored when the agents interact with the environment. The network structures of the reward prediction, the state value, and the action value tasks are all composed of shallow convolutional neural networks and LSTM networks [20, 21]. In Table 1, NN3 shows the network structure of the auxiliary task.

In addition, the development platform of this research is Linux server with Ubuntu 16.04-5.4.0 system; the CPU is Intel (R) Xeon (R) Silver 4110 CPU @ 2.10 GHz, with 32 virtual cores; the GPU is NVIDIA Tesla P100, with 16 GB video memory; and the development language is based on Python 3.6.3.

3.3. Comparative Experiments

3.3.1. The Performance of the IBPO. The IBPO is tested in the pathfinding scenario. The agent starts to explore the entire scene from the starting position and cannot complete the task until it reaches the target position in the pathfinding scenario. This scenario is an extremely challenging training map in the VizDoom platform because there is only an external reward signal at the target. Another issue is that random agents have difficulty reaching the target position with limited action steps. Consequently, we test the IBPO algorithm in the pathfinding scenario.

In the pathfinding scenario, the evaluation indicators are the average reward value and average action steps. The average reward value is defined as the ratio of the number of the agent that reached the target position within the specified steps to the number of all trained agents currently, indicating the pathfinding success rate of the agents. The average action step is defined as the average number of action steps for 100 verifiable interactions between the environment and the agent after the algorithm converges, which indicates the stability of the agent's action policy.

We train different RL agents with IBPO, DRQN [22], and DFP [23], respectively, in the pathfinding scenario to prove the effectiveness of the IBPO by comparing their evaluation indicators. The experiments verify that the intrinsic reward can give agents assistance in updating policy

effectively without the environmental reward feedback. The DFP and the DRQN are the methods used by the second- and third-place agents in the 2017 VizDoom competition. In this competition, there is a map similar to the pathfinding scenario. The DRQN utilizes the game information provided by the simulator. The DFP owns supplementary processing modules for the state and the reward information. These two algorithms are dominant in the pathfinding scenario. Figure 5 illustrates the performance curves using the IBPO, DFP, and DRQN in the pathfinding scenario.

We can find that the IBPO reached an average reward value of 0.92, which outperforms the DRQN and the DFP, whose average reward values are 0.79 and 0.86, respectively. The closer the average reward value is to 1, the more effective the policy learned by the method is and the more likely the agent will reach the expected position. The main reason for this result is the deficiency of environmental rewards in the pathfinding scenario. The intrinsic rewards in the IBPO make up for the lack of positive reward values in experience replay memory, thus promoting the learning of exploration policy. The above comparative experiments demonstrate that the DRL agent trained by the IBPO has plentiful and efficient performance in exploring the 3D pathfinding scenarios.

Table 2 summarizes the average reward value and average action step in the pathfinding experiments. The first row corresponds to the reward in Figure 5, and the second row shows action steps required by different agents to reach the target position. We can find that the IBPO performs the best among the three methods with a minimum average action step of 61.8. It indicates the agent trained by the IBPO can find the path to the target faster with a more steady action policy.

In addition, in order to verify the effectiveness of the intrinsic reward in the IBPO, we also analyzed the trend of the intrinsic reward in the training process. As shown in Figure 6, the intrinsic reward value gradually decreased from 1 to about 0.1. In the whole training process, for the DRL agent, there are many novel states in the early stage of the training, which results in a larger intrinsic reward value. At this time, the intrinsic reward value is the main reward signal for the RL to update. As the training process goes on, the agent gradually learns the action policy to explore the environment and the internal reward value gradually decreases. At this time, the update of the RL algorithm is mainly external reward.

To sum up, the value generated by the intrinsic reward module provides auxiliary signals for the agent's policy update, which helps the agent to learn effective exploration policy in sparse reward scenarios. Through comparative experiments with the DRQN and the DFP, we conclude that the IBPO outperforms these two methods in the average reward value and average action step indicators.

3.3.2. The Performance of the AIBPO. The AIBPO is tested in the survival scenario. In VizDoom's survival scenario, the layout of each training round is different. The initial position, the number of drug packages, and the packages' locations are also random. Moreover, the maze wall limits the vision of the agent. Such random factors in the scenario would affect the agent's policy learning. The update will be unstable during training if the algorithm is unable to perceive the environmental information adequately.

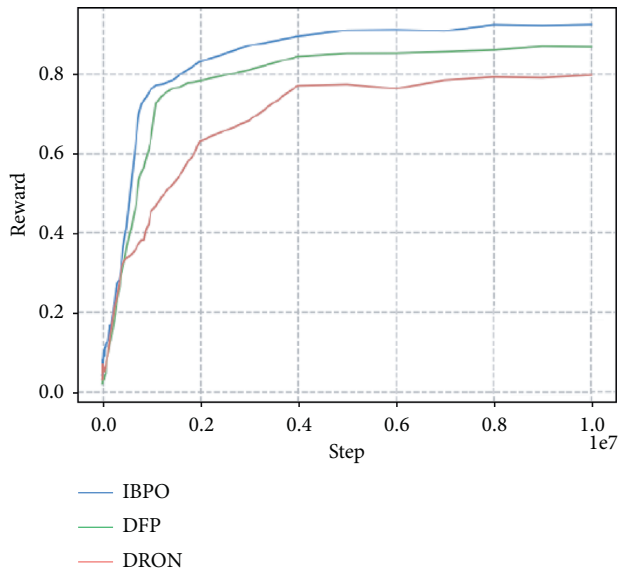


Figure 5: Experimental results of the IBPO in the pathfinding scenario (the y-axis represents the average reward value obtained by the DRL agents in the pathfinding scenario and the x-axis represents the timestep during training; the greater the reward, the better the method).

Table 2: Experimental results of the IBPO (the bold is the best).

Evaluation criteria	IBPO	DFP	DRQN
Average reward value	**0.92**	0.86	0.79
Average action steps	**61.8**	69.3	75.7

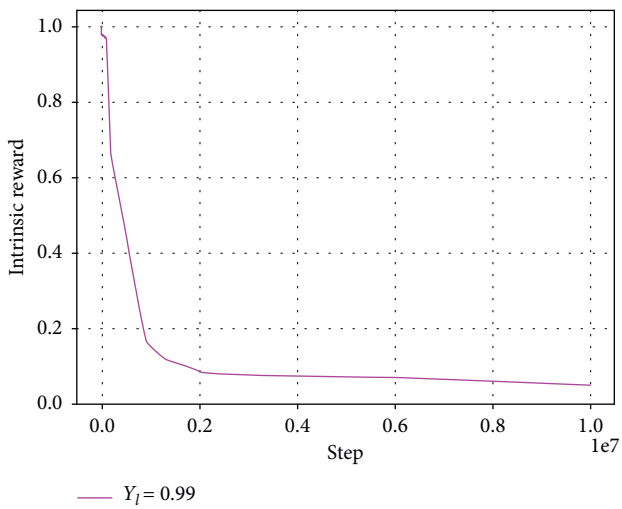


Figure 6: The trend of the intrinsic reward value (the y-axis represents the intrinsic reward value, and the x-axis represents the timestep during training).

Table 3: Experimental results of the AIBPO in the survival scenario.

Evaluation criteria	AIBPO	IBPO	DRQN	DFP	A3C	Rainbow
$\text{Surs}_{\text{step}}$	**293.7**	266.5	277.1	301.3	256.1	260.3
Pics_{hp}	**9.58**	7.83	7.41	8.81	6.85	7.32

Table 4: Ablation experiment results of auxiliary tasks.

Evaluation criteria	AIBPO	IBPO	IBPO + $\mathscr{L}_P$	IBPO + $\mathscr{L}_V$	IBPO + $\mathscr{L}_C$
$\text{Surs}_{\text{step}}$	**293.7**	266.5	273.4	268.7	282.5
Pics_{hp}	**9.58**	7.83	9.11	8.19	8.17

The AIBPO, DRQN [22], DFP [23], A3C [19], and Rainbow [24] are used to train the DRL agents for comparative experiments in the survival scenario according to the relevant evaluation indicators. The AIBPO is an improved version based on the IBPO with the auxiliary task learning mechanism. Its main body is still the IBPO; thus, an IBPO agent was also implemented in this experiment. $\text{Surs}_{\text{step}}$ and Pics_{hp} are the evaluation indicators that are equivalent to game scores in the DRL. The $\text{Surs}_{\text{step}}$ is defined as the average of the maximum action steps that the agent can survive for 100 verifiable interactions after the algorithm has converged. The Pics_{hp} is defined as the average number of drug packages picked up by the agent for 100 interactions after the algorithm has converged. Table 3 shows the experimental results in the survival scenario. The bigger $\text{Surs}_{\text{step}}$, the better Pics_{hp}.

We can find that the AIBPO is 9.58 on the Pics_{hp}, which is better than all other compared methods. They are 7.83, 7.41, 8.81, 6.85, and 7.32 for the IBPO, the DRQN, the DFP, the A3C, and the Rainbow on the Pics_{hp}, respectively. Meanwhile, the IBPO is also better than other methods except the DFP on the $\text{Surs}_{\text{step}}$. In addition, the IBPO does not perform better than the DFP and the DRQN on the $\text{Surs}_{\text{step}}$. The reason is that the IBPO principally aims at the sparse reward problem in 3D scenes, but there is sufficient reward information in survival scenarios. Fortunately, the maze features also require agents to make long-term plans based on the historical information. The sufficient reward information in this scenario can be used by auxiliary tasks of the AIBPO. Three types of auxiliary tasks play key roles in increasing the agent's reward perception and state estimation capacity. As a result, the AIBPO achieves better scores than the IBPO and the DRQN in this experiment. It is close to the DFP on the $\text{Surs}_{\text{step}}$ and superior to the DFP on the Pics_{hp}.

The proposed IBPO aims to solve the exploration problem in 3D scenes with sparse reward. The experimental result from the pathfinding scenario demonstrates that the IBPO has a certain efficiency improvement compared with the DFP in this type of scenes. The proposed AIBPO supplements the IBPO's shortcomings in perceiving environmental reward information by using an auxiliary task learning mechanism. The experimental result from survival scenarios shows that the auxiliary task learning mechanism as an auxiliary method has greatly improved the IBPO.

3.3.3. Ablation Experiment. The proposed AIBPO includes three types of auxiliary tasks: reward prediction task $\mathscr{L}_P$, state value task $\mathscr{L}_V$, and action value task $\mathscr{L}_C$. We attached the three tasks separately to the IBPO to compare their individual effects on the original policy optimization algorithm. The experimental results are shown in Table 4.

As we can see from Table 4, in survival scenarios, each of three auxiliary tasks has improved the benchmark algorithm IBPO to varying degrees, thus prolonging the survival time of the agent. The reward prediction task performed better on the Pics_{hp} than the other two tasks, while the action value task worked best on the $\text{Surs}_{\text{step}}$. Practically, they work better together than separately. The AIBPO that integrates all three tasks improved the most in the 3D circumstance.

3.4. Discussion. In many real-world scenarios, the agents require to make decisions in 3-dimensional space, for example, 3-dimensional navigation and automatic driving. Our work is able to be directly applied to these tasks, so our method shows very important application value. In addition, in some tasks based on reinforcement learning, reward sparsity is a common problem that limits the performance of the algorithm. For example, in the task of manipulator grasping, the manipulator can only get the reward after successfully grasping the target by completing a series of complex pose control. The failure of any step in the middle progress may lead to the failure to get the reward. Our approach provides additional rewards to the agent through intrinsic rewards and auxiliary tasks so as to alleviate the problem of reward sparsity effectively. Therefore, our method has a strong reference significance for these tasks in theory.

4. Conclusions

We have presented a novel approach, named IBPO, for the reward sparsity problem in 3D games. Unlike existing DRL-based methods, an agent of our approach can learn the intrinsic reward, which uses the differential fusion mechanism and the modified value network. Moreover, the AIBPO is proposed based on the IBPO by combining auxiliary tasks, which further improves the performance of the IBPO. The experimental results based on the VizDoom platform show the effectiveness of the proposed approach.

However, this method also has its limitations. First, it needs considerable expert knowledge in designing intrinsic rewards and auxiliary tasks, which limits its further application. Second, when used in more complex scenarios, computer vision, situation analysis, and other techniques are needed to make our method more robust.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by PINGAN-HITsz Intelligence Finance Research Center, Research and Development Plan of Key Fields in Guangdong Province, China (No. 2020B0101380001), National Natural Science Foundation of China (No. 61902093), and Natural Science Foundation of Guangdong (No. 2020A1515010652).

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing atari with deep reinforce-ment learning," 2013, https://arxiv.org/abs/1312.5602.

[2] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[3] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[4] D. Silver, J. Schrittwieser, K. Simonyan et al., "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[5] J. Schrittwieser, I. Antonoglou, T. Hubert et al., "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[6] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: an evaluation platform for general agents," *Journal of Artificial Intelligence Research*, vol. 47, no. 7, pp. 253–279, 2013.

[7] J. M. Font and T. Mahlmann, "Dota 2 bot competition," *IEEE Transactions on Games*, vol. 11, no. 3, pp. 285–289, 2018.

[8] M. Wydmuch, M. Kempka, and W. Jaskowski, "Vizdoom competitions: playing doom from pixels," *IEEE Transactions on Games*, vol. 11, no. 3, pp. 248–259, 2018.

[9] M. Jaderberg, W. M. Czarnecki, I. Dunning et al., "Human-level performance in 3D multiplayer games with population-based reinforcement learning," *Science*, vol. 364, no. 6443, pp. 859–865, 2019.

[10] K. Shao, Y. Zhu, and D. Zhao, "Starcraft micromanagement with reinforcement learning and curriculum transfer learning," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 1, pp. 73–84, 2018.

[11] O. Vinyals, I. Babuschkin, W. M. Czarnecki et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[12] N. Hanaki, "Action learning versus strategy learning," *Complexity*, vol. 9, no. 5, pp. 41–50, 2010.

[13] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang, "Multi-modal curriculum learning for semi-supervised image classification," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3249–3260, 2016.

[14] Z. Yang, K. Merrick, L. Jin, and H. A. Abbass, "Hierarchical deep reinforcement learning for continuous action control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5174–5184, 2018.

[15] Z. Guo, X. Wang, S. Qi, T. Qian, and J. Zhang, "Heuristic sensing: an uncertainty exploration method in imperfect information games," *Complexity*, vol. 2020, Article ID 8815770, 9 pages, 2020.

[16] T. Matiisen, A. Oliver, T. Cohen, and J. Schulman, "Teacher–student curriculum learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3732–3740, 2019.

[17] J. Li, X. Shi, J. Li, X. Zhang, and J. Wang, "Random curiosity-driven exploration in deep reinforcement learning," *Neurocomputing*, vol. 418, pp. 139–147, 2020.

[18] N. Bougie and R. Ichise, "Skill-based curiosity for intrinsically motivated reinforcement learning," *Machine Learning*, vol. 109, no. 3, pp. 493–512, 2020.

[19] V. Mnih, A. P. Badia, M. Mirza et al., "Asynchronous methods for deep reinforcement learning," in *Proceedings of the International Conference on Machine Learning*, pp. 1928–1937, New York, NY, USA, June 2016.

[20] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[21] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[22] G. Lample and D. S. Chaplot, "Playing fps games with deep reinforcement learning," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pp. 2140–2146, San Francisco, CA, USA, 2017.

[23] A. Dosovitskiy and V. Koltun, "Learning to act by predicting the future," in *Proceedings of the International Conference on Learning Representations*, pp. 637–645, Toulon, France, 2017.

[24] M. Hessel, J. Modayil, H. Van Hasselt et al., "Rainbow: combining improvements in deep reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1147–1153, New Orleans, LA, USA, 2018.