Adaptive Inner-reward Shaping in Sparse Reward Games

Dong Yang
HPCL College of Computer
National University of Defense Technology
Chang Sha, China
yangdong14@nudt.edu.cn

Yuhua Tang
HPCL College of Computer
National University of Defense Technology
Chang Sha, China
yhtang@nudt.edu.cn

***Abstract*—Reinforcement learning focuses on goal-directed learning from interaction and the success of its applications strongly depends on how well the reward signal frames the problem and how well it assesses progress in solving it. But in many real-world scenarios, the agent is supplied with extremely sparse or even no rewards which makes learning fail and fall into ineffective exploration. In psychology, shaping is a method of animal training by reinforcing successive approximations of rewards to finally achieve the desired complex behavior. Inspired by this phenomenon of animal learning and reward as a signal in neuroscience, in this paper we solve the sparse reward problem by constructing a reward generator to generate inner-rewards and guide the agent learning control policies with deep neural networks. The proposed learning-based reward shaping does not require specific domain knowledge, but rather enable the agent to learn how to generate inner rewards to guide itself in any scenarios online jointly with the actual reinforcement learning process. To validate the performance in complex sparse reward problems, the proposed approach is evaluated in a challenging scenario, *Football Academy* in *Google Research Football Environment*, a newly released reinforcement learning environment with physics-based 3D simulator, instead of maze environments or grid world that are commonly used in research which are not sufficiently challenging. We compare the performance of our inner-rewards approach with two reinforcement algorithms (*PPO* and *ICM + PPO*). Experimental results show that our method improves the learning performance in terms of speed and quality, and also enables the agent to learn generalized skills applied to novel scenarios.**

***Index Terms*—Reinforcement learning; Football game; Reward shaping; Inner-rewards**

I. Introduction

Reinforcement learning (RL) is learning what to do, how to map situations to actions, so as to maximize a numerical reward signal [28]. RL has a wide range of applications in the real world, including autonomous driving [2, 3], robotics [8, 11], intelligent transportation systems [6, 29], etc. Especially, games provide challenging environments where new algorithms and ideas can be quickly tested in a safe and reproducible manner [12], which are often used as benchmarks and promote great progress in reinforcement learning. Commonly used gaming platforms include the iconic Atari console games [15, 16], computer go [25], and video games like *Football* [12] or *Dota 2* (OpenAI-Five). In some scenarios, rewards extrinsic to the agent are continuous, such as a running "score" in an Atari game [16]. In these problems, we can usually define the reward function by "whether the problem is solved" or "whether the game score is improved". But in more complex game scenarios as well as in real-world scenarios, goals can not be easily translated into reward functions and the rewards are extremely sparse or absent, e.g. *Super Mario Bros* [19], which means that it is difficult for agents to learn useful knowledge in such an environment.

For example, in a football game, only the reward for the goal is determined and it is difficult to evaluate a fixed reward for an intermediate process. Therefore designing an adaptive reward function is difficult because it strongly depends on how the reward signal frames the problem and how it assesses progress in solving it. In most scenarios, the reward signal is usually the only learning signal therefore is the primary basis for altering the policy. Therefore, delayed and sparse rewards make the agent get caught in "mesa phenomena[1]" and can significantly slow down learning through long and uninformed exploration trajectories. Another challenge that arise in sparse rewards environment, is the trade-off between exploration and exploitation which has been intensively studied by mathematicians for many decades, yet remains unresolved [28]. To obtain the sparse reward, the agent must prefer actions that it has tried in the past and found to be effective in producing reward. But to discover such actions, it has to try actions that it has not selected before.

As human agents, we can learn and accomplish tasks very well with sparse rewards or no rewards. For example, while doing research, researchers are ready to work for decades or even a whole lifetime without explicit rewards provided by the environment. But they still gradually push this work forward through the guidance of their inner-rewards generated by reward shaping [18], e.g. motivation [21] and curiosity [26] have been used to explain the need to explore the environment and discover novel states. In fact, the inner-reward guides agents on how to explore the environment and learning policy. Sutton & Barto [28] think of the reward as a signal in neuroscience rather than an object or event in the agents environment, which is a signal internal to the brain, like the activity of neurons that influences decision making

[1]Mesa Phenomenon is a concept in which a small change in a parameter usually leads to either no change in performance or to a large change in performance [13].

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

(a) Run to score with keeper (b) Pass and shoot with keeper (c) Run,pass and shoot with keeper

Fig. 1. A snapshot of the football scenarios investigated in the paper. Our inner-reward shaping function can be learned online in parallel with the process of the agent learning to play football. These results suggest that the proposed method accelerate learning by bootstrapping an agent with additional information. Moreover, results suggest that the proposed method enables the agent to learn generalized skills, and perform better in scenarios that have never been experienced.

and learning. The parts of brain that produce these signals have evolved for millions of years, indicating that designing a reward signal is a difficult task, and it is challenging to learn a shaped inner-reward function.

In this paper, we propose to reinforce successive approximations of rewards to finally achieve the desired complex behavior by using neural networks to simulate neuroscience of reward signals generation. In order to simulate the process of reward signals generated by neuroscience, the neural network takes state and action as input, outputs the inner reward of the current state, and uses the reward of the environmental feedback to guide. In other words, we use our inner reward as the primary basis for altering the control policy, and use realistic feedback reward to constrain our inner reward. The proposed approach is evaluated in a challenging scenario, *Football Academy* in *Google Research Football Environment* (shown in Figure 1). We choose two very challenging baselines: proximal policy optimization algorithms (PPO [24]) and PPO with intrinsic curiosity module (ICM). Experiment results (Section V-A) confirmes that our proposed method enables the agent explore in the early stage and exploit later, significantly improves the learning performance of the agent and outperforms the baseline methods in the environment with sparse reward. Further experimental results prove that our method enables the agent to learn generalized skills applied to novel scenarios.

In our work, we seek to improve the current state of designing a reward signal by introducing a learning-based reward shaping that attains the better speed and quality performance without altering the optimal policies, which we call inner-reward shaping(IRS). The contributions of this work are presented as follows:

1) We present a novel adaptive approach IRS which does not require specific domain knowledge, instead it makes the agent learn how to generate inner-rewards to guide itself in any scenarios online jointly with the actual reinforcement learning process.
2) We provide solution for exploration-exploitation dilemma in the environment with sparse reward, which enable agents try a variety of actions and progressively favor those that appear to be best.
3) Furthermore, experiment results indicate that our approach improves the generalized ability of PPO to adapt to novel scenario.

We believe that future applications can benefit from a better understanding of how reward signals affect learning and from improved methods for designing them.

II. Related Work

Tempting to address the sparse reward problem, some more sophisticated ways to find good intrinsic reward signals have been proposed, but the subject has interesting and relatively unexplored dimensions requiring further studies.

A very effective way of dealing with the problem of sparse reward is curiosity [4, 19, 22], which can serve as an intrinsic reward signal to enable the agent to explore its environment and learn skills that might be useful later in its life. Curiosity is related to what one already knows about the world. One gets curious as soon as one believes that there is something that one does not know. [22] introduces a framework for 'curious neural controllers' which measures the Euclidian distance between reality and prediction of the model network. The activation of the curiosity unit is a function of this distance. Its desired value is a positive number corresponding to the ideal mismatch between belief and reality. However, there are currently no known computationally feasible mechanisms for measuring learning progress. [4] performs a large-scale study of curiosity-driven exploration across a variety of environments and extensively investigates different feature spaces for learning the dynamics-based curiosity. But if the transitions in the environment are random, then even with a perfect dynamics model, the expected reward will be the entropy of the transition, and the agent will seek out transitions with the highest entropy. That is, when the environment has randomness that is not related to the agent, the agent will be stuck in the corresponding position because it cannot predict the next state. An excellent method is used in the [19], only predicting those changes in the environment that could possibly be due to the actions of our agent or affect the agent, and ignore the rest. Predicting the agents action (a_t) given its current(s_t) and next(s_{t+1})

states by learning an inverse dynamics model. The forward model is another neural network that takes action a_t and a feature vector $\phi(s_t)$ as inputs and predicts the feature encoding $\hat{\phi}(s_{t+1})$ which is the estimate of $\phi(s_{t+1})$ of the next state. The prediction error is computed as curiosity reward r_t^i. The results of [19] show that a curiosity-driven intrinsic reward is crucial in accomplishing these tasks, such as *VizDoom* and *Super Mario Bros*.

Another approach to finding a good reward signal is based on shaping [27], in which reward contingencies are progressively altered to train an animal to successively approximate a desired behavior. Shaping is not only indispensable for animal training, but also an effective tool for training reinforcement learning agents. Reward shaping has proven to be a powerful method for speeding up many RL tasks. Randløv and Alstrøm [20] describe a system that learns to drive a bicycle using reinforcement learning and shaping. However, these policies of providing positive rewards whenever the agent made progress towards the goal are clearly not optimal for the original task. Intuitively, policy invariance is important in shaping. Ng et al. [17] shows that potential-based reward shaping(PBRS) model satisfying the requirement of the final policy is equivalent to the original one. A number of interesting related works have appeared after recognizing that policy invariance is important in shaping. [1]demonstrates a concrete way of using shaping functions with model-based learning algorithms and relates it to model-free shaping and the Rmax algorithm. [7] argues that how to compute potential in the absence of knowledge, and the potential function can be learned online in parallel with the actual reinforcement learning process. [5] focuses on reward shaping with partial observability, and shows that landmarks can be used to shape the rewards in reinforcement learning with hidden states.

III. The proposed Method

In this work, we focus on how to guide the agent learn inner rewards policy in parallel with making the control policy performance faster and better. By changing the rewards, the agent can know what is good in a short time and can affect the value function, so that the agent knows what is good in the long run.

Then we present a novel approach IRS to shape rewards using neural networks by simulating neuroscience to generate reward signals. Based on a transformed markov decision process(MDP) satisfying Equation (2), IRS accelerates the exploration and exploitation of agents without changing the original optimal policy while learning online jointly with the actual reinforcement learning process. Our method is composed of two subsystems: a shaping-reward policy and a control policy that outputs a sequence of actions to maximize that reward signal. See Figure 2 for illustration of our method in the transformed Markov decision progress. In general, our method can seamlessly adapt to any reinforcement learning algorithm through empirical learning values or action value functions.

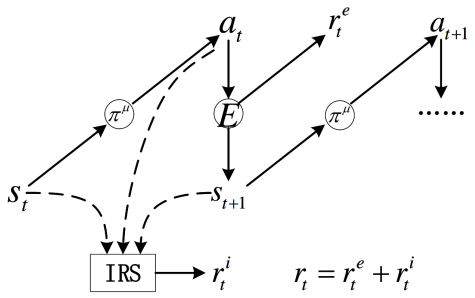


Fig. 2. Our method in the transformed Markov decision progress. The agent in state s_t interacts with the environment by executing an action a_t sampled from its current policy π^μ and ends up in the state s_{t+1}. The policy π^μ is trained to optimize the sum of the extrinsic reward (r_t^e) provided by the environment E and the curiosity based inner reward signal (r_t^i) generated by the policy π^θ in our proposed IRS model.

In order to guarantee the policy invariance in reward shaping, our method is based on a new transformed markov decision process. A markov decision process is defined as a tuple (S, A, T, R, γ), where S is a finite set of states, A is a finite set of the actions, $T = \{P_{sa}(\cdot)|s \in S, a \in A\}$ is the transition probability, R is the reward function and $\gamma \in (0, 1]$ is the discount factor. Rather than running reinforcement learning algorithm on

$$M = (S, A, T, R, \gamma), \tag{1}$$

we running it on a new transformed MDP

$$M' = (S, A, T, R', \gamma), \tag{2}$$

where $R' = R + F$ is the reward function in the transformed MDP, and $F : S \times A \times S \to \mathbb{R}$ is a bounded real-valued function called the shaping reward function [17]. Next, we will explain how to construct a function F in combination with deep reinforcement learning algorithm leaving optimal policy unchanged.

We strive to make our method seamlessly integrate with any deep reinforcement learning algorithm through empirical learning values or action value functions. Here we use the policy gradient method which work by computing an estimator of the policy gradient and plugging it into a stochastic gradient ascent algorithm. Implementations that use automatic differentiation software work by constructing an objective function whose gradient is the policy gradient estimator. The most commonly used gradient estimator by differentiating the objective is

$$L^{PG}(\mu) = \hat{\mathbb{E}}_t[log\pi_\mu(a_t|s_t)\hat{A}_t], \tag{3}$$

where π_μ is astochastic policy and $\hat{A}_t$ is an estimator of the advantage function at timestep t. Our proposed method works by acting on the advantage function

$$\hat{A}_t = -V(s_t) + \cdots + \gamma^{T-t-1} r_{T-1} + \gamma^{T-t} V(s_T). \tag{4}$$

The surrogate objective we use is maximized as following:

$$L^{CLIP}(\mu) = \hat{\mathbb{E}}_t[min(h_t(\mu)\hat{A}_t, clip(h_t(\mu), 1-\epsilon, 1+\epsilon)\hat{A}_t)], \tag{5}$$

where $h_t(\mu)$ denote the probability ratio $h_t(\mu) = \frac{\pi_\mu(a_t|s_t)}{\pi_{\mu old}(a_t|s_t)}$ in PPO. The first term inside the min is $L^{CPI} = h_t(\mu)\hat{A}_t$ in Trust Regin Policy Optimization, which refers to its conservative policy iteration[9]. The second term, $clip(h_t(\mu), 1-\epsilon, 1+\epsilon)\hat{A}_t)$, modifies the surrogate objective by clipping the probability ratio, which removes the incentive for moving r_t outside of the interval $[1-\epsilon, 1+\epsilon]$. Finally, we takes the minimum of the clipped and unclipped objective, so the final objective is a lower bound on the unclipped objective [24].

We propose to reinforce successive approximations of rewards to finally achieve the desired complex behavior by using neural networks with parameters θ to simulate neuroscience of reward signals generation. Combined with the rewards generated by IRS, the control policy $\pi^{\mu}_{M'}$ with parameters μ is better optimized in MDP Equation (1). Using IRS method is a sufficient condition for the original MDP and the MDP with reward shaping to have consistent optimal policies. In addition to inner-reward, the agent may also receive some extrinsic reward from the environment. Let the extrinsic reward be r^e_t and the inner-reward generated by the agent at time t be $r^i_t = F(s_t, a_t, s_{t+1}, \theta)$. In the original MDP M Equation(1) we would have received reward $R(s, a, s')$ for transition from s to s' on action a, instead in the transformed MDP M' Equation(2) we would receive reward $R'(s, a, s') = R(s, a, s') + F(s, a, s')$. The control policy $\pi_{M'}$ is trained to maximize the sum of these two rewards

$$\begin{aligned} r_t &= R'(s, a, s') \\ &= R(s, a, s') + F(s, a, s') \\ &= r^i_t + r^e_t. \end{aligned} \tag{6}$$

Our inner-reward shaping model can potentially be used with a range of policy learning methods. Combining the above work, we maximizes the following objective:

$$\begin{aligned} &L^{CLIP+VF+S}(\mu, \theta) \\ &= \hat{\mathbb{E}}_t[L^{CLIP}(\mu, \theta) - c_1 L^{VF}_t(\mu) + c_2 S[\pi^{\mu}_{M'}](s_t)], \end{aligned} \tag{7}$$

where c_1 and c_2 are coefficients. And the second term (L^{VF}_1) inside the $\hat{\mathbb{E}}_t$ is a squared-error loss $(V_\mu(s_t) - V^{targ}_t)^2$ [14], where $V(s)$ is a learned state-value function $V(s)$. Because of parameters sharing between the policy and value function, the third term is an entropy bonus used to improve exploration by limiting the premature convergence to suboptimal policy. In order to combine inner rewards and extrinsic rewards from the environment, we use a truncated version of generalized advantage estimation [23], which significantly reduce the variance while maintaining a tolerable level of bias,

$$\hat{A}^{GAE(\gamma,\lambda)}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots\cdots + (\gamma\lambda)^{T-t-1}\delta_{T-1}, \tag{8}$$

where the generalized advantage estimator for $0 < \lambda < 1$ makes a compromise between bias and variance, controlled by parameter λ. And t specifies the time index in $[0, T]$ within a given T-length trajectory segment. In each iteration, we calculate equation(8) by δ_t and each agent collect T timesteps of data,

$$\begin{aligned} \delta_t &= r_t + \gamma V(s_{t+1}) - V(s_t) \\ &= r^i_t + r^e_t + \gamma V(s_{t+1}) - V(s_t) \\ &= F(s_t, a_t, s_{t+1}, \theta) + r^e_t + \gamma V(s_{t+1}) - V(s_t). \end{aligned} \tag{9}$$

Then we construct the surrogate loss with reference to parameter μ on these data, and optimize it with Adam[10]. The neural network parameters θ are optimized by minimizing the loss function L' that measures the discrepancy between the inner reward and extrinsic reward:

$$\begin{aligned} &L'(r^i_t, \eta_t) \\ &\text{where} \quad \eta_t = v_1 r^e_t + v_2 r^e_{t+2}, \ldots, v_V r^e_{t+V}. \end{aligned} \tag{10}$$

In the control task, the actions taken by the agent often affect not only the immediate rewards, but also the next scenario, thus affecting the subsequent rewards. So in equation(10), rewards is weighted by coefficient v and V specifies the time index.

To illustrate how IRS can augment existing RL algorithms, PPO with IRS is shown in algorithm 1. In algorithm 1, after iterating U times, r_t learned online in parallel with the actual RL task gradually approaches r^e_t which is extremely sparse . Therefore, IRS running on a transformed MDP satisfying Equation (2), accelerates the exploration without changing the original optimal policy.

Algorithm 1: PPO algorithm with inner-reward

```
Initialise λ, γ ∈ [0,1] ; U, T, N, K ∈ ℕ
Initialise θ, μ
for iteration = 1, 2,...,U do
    for actor = 1, 2,...,N do
        for t = 1, 2,...,T do
            Choose a_t from A using policy π^{μ_old}_{M'} from s_t
            Take action a_t. Get next state s_{t+1} and extrinsic reward r^e_t
            Run policy π^θ from {s_t, a_t, s_{t+1}}. Get inner-reward r^i_t
            Collecting {s_t, a_t, r^i_t, r^e_t }
        end
        Compute Â^{GAE(γ,λ)}_1, ..., Â^{GAE(γ,λ)}_t
        Compute η_1, ..., η_T
    end
    Optimize surrogate L wrt μ and L' wrt θ, with K epochs and minibatch size M ≤ NT
    μ_old ← μ
end
```

IV. Experimental setup

To evaluate our IRS module on its ability to accelerate learning, improve the learning performance and learn generalized skills applied to novel scenarios, we test in the scenarios in *Football Academy* in *Google Research Football Environment*. This section describes the details of the environments and the experimental setup.

A. Environments

Our method was evaluated empirically on the football tasks that are shown in Figure 1, and one of the tasks is explained in Figure 3. The environment provides a physics-based 3D football simulation where agents have to control their players, learn how to pass in between them and how to overcome their opponent's defense in order to score goals. This provides a challenging reinforcement learning problem as football requires a natural balance between short-term control, learned concepts such as passing, and high level strategy [12].

In addition to the scenario presented in Figure 3, the scenario of *Run to Score with Keeper*(as show in subfigure 1(a)) consists of eight players, including the goalkeepers on both sides. And another player of ours starts in the middle of the field with the ball, and needs to score against a keeper while five opponent players chase ours from behind. In scenarios of *Run, Pass and Shoot with Keeper*(as show in subfigure 1(c)), two of our players try to score from the edge of the box. The unmarked one is on the side with the ball and the other is at the center next to a defender facing the opponent keeper.

In these tasks, the agent has to interact with a fixed environment and maximize its episodic reward by sequentially choosing suitable actions based on observations of the environment. The action space of the agent consists of 21 actions [12], including standard move actions(up, down, left, right etc.) and different ways to kick the ball(shot, pass etc.). The goal of these tasks is to win the game against the opponent players.

Reward Our experiment evaluates on two types of environments with different rewards: *very sparse reward* (only scoring) and *sparse reward* (scoring and checkpoint reward). The environments with *very sparse reward*which corresponds to the most intuitive reward in these tasks give feedback a *+1* reward to our agent when our team scores a goal, and a *-1* reward when conceding one to the opposing team. The agent is only provided with a sparse terminal reward of *+1* or *-1* if a player successfully scores and zero otherwise. The *very sparse reward* can be hard to observe during the initial stages of training, as it requires a long sequence of consecutive events: overcoming the defense of a potentially strong opponent, and scoring against a keeper. Due to the sparsity of scoring, we also use the checkpoint reward from the environment. In environments with *sparse reward* which corresponds to the additional auxiliary reward, the agent obtains an additional auxiliary reward of *+0.1* the first time an agent moves the ball close to the goal and crosses each of the checkpoint regions.

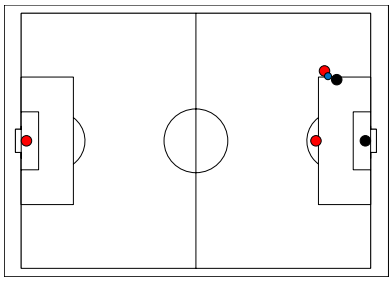


Fig. 3. Map for scenarios in *Football Academy*. The scene on the field is *Pass and Shoot with Keeper*(as show in subfigure 1(b)). Red dots denote our players. Two of our players besides the keeper try to score from the edge of the box. Blue dots denote the ball and black dots refer to two opponent players.

B. Comparison of baseline methods

In this paper, we compare the *PPO with IRS*, *PPO* and *PPO with ICM*. We now provide more details about the algorithms.

PPO We use the convolutional neural network architecture which has three convolution layers. The input state s_t is passed through the first convolution layer with 32 filters each, kernel size of 8×8, stride of 4 and padding of 1. An Rectified linear unit(Relu) is used after each convolution layer. The output of the last convolution layer is fed into a fully connected layers with 512 units. Two separate fully-connected layers are used to predict the value function and the action from the previous fully connected layer feature representation. We rely on the Adam optimizer for training.

ICM The forward model is constructed by two fully connected layers. Pass $\phi(s_t)$ and a_t into a sequence of two fully connected layers with 256 and 288 units respectively. The inverse model first maps the input state into a feature vector $\phi(s_t)$ using a series of four convolution layers, each with 32 filters, kernel size 3x3, stride of 2 and padding of 1. An exponential linear unit(ELU) is used after each convolution layer. $\phi(s_t)$ and $\phi(s_{t+1})$ are concatenated into a single feature vector and passed as inputs into a fully connected layer of 256 units. The output fully connected layer with 21 units to predict one of the 21 possible actions.

IRS The input state s_t is passed through a sequence of three convolution layers and three fully connected layers. The first convolution layers with32 filters, kernel size of 8×8, stride of 4. The second convolution layers with 64 filters, kernel size of 4×4, stride of 2. The last convolution layers with 64 filters, kernel size of 3×3, stride of 1. An Rectified linear unit(Relu) is used after each convolution layer. And then three fully connected layers are used to predict the inner-reward. Also, we rely on the Adam optimizer for training.

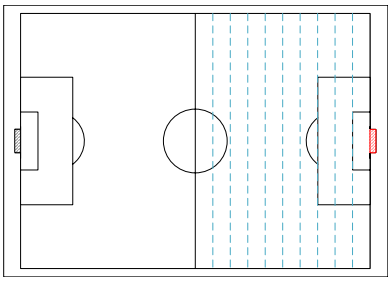
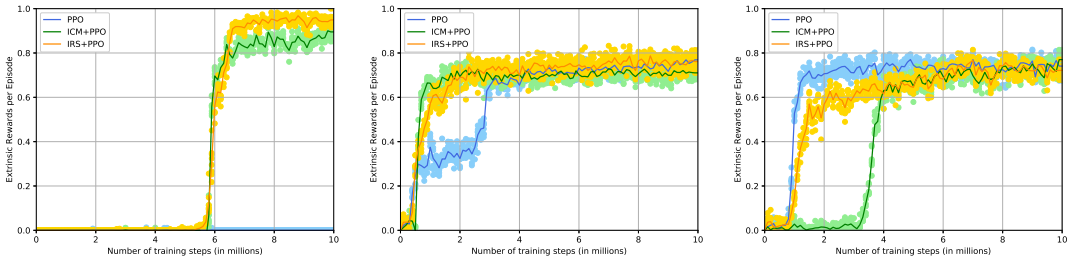


Fig. 4. Reward settings in the environment. 1) In environment with *very sparse reward* (only scoring), our team obtains a +1 reward when taking the ball into the red shaded area or -1 reward for black shaded area. 2) Opponents field is divided in n = 10 regions according to the Euclidean distance to the opponent goal. In environment with *sparse reward* (scoring and checkpoint reward), the agent obtains +0.1 reward when taking the ball across the blue boundary line.



(a) Run to score with keeper (b) Pass and shoot with keeper (c) Run pass and shoot with keeper

Fig. 5. In the environment with very sparse reward, compare the performance of the PPO agent (blue), the curiosity-driven *ICM* agent (green) and the proposed inner-reward shaping *IRS* agent (orange) as the complexity of the task increased from left to right. Running three independent runs of each algorithm, darker line represents mean and the scattered dots represent different samples.

C. Training details

We use Super Mini Map (SMM) representation which consists of four 96×72 matrices encoding information about the home team, the away team, the ball and the active player respectively. The encoding is binary, representing whether there is a player, ball, or active player in the corresponding coordinate or not. In order to model temporal dependencies, the state representation of the environment is constructed by concatenating the current frame with the previous frames. All agents in this work are trained using SMM inputs with eight environments to run in parallel. *IRS + PPO* denotes our full algorithm which combines inner-reward shaping model with proximal policy optimization algorithms. Across different scenarios, we compare our approach with two baselines. First is the original *PPO* algorithm. Second is the *ICM + PPO* algorithm consisting of the forward and the inverse model.

In order to speed up learning, and to reduce useless exploration, the episodes are terminated either when some events happen (including score, agent loses the ball, and game stops) or the agent exceeds a maximum of 400 time steps. Judging from previous experience, we consider testing the generalization of the proposed method. For generalization experiments, we pre-train a control policy using *IRS + PPO* model in *Pass and Shoot with Keeper* scenario with *very sparse reward* and then evaluate the result control policy on the *Run to Score with Keeper* scenario. It takes approximately 60 steps for an optimal policy to score in *Run to Score with Keeper* scenario , 50 steps in *Pass and Shoot with Keeper* scenario, 40 steps in *Run, Pass and Shoot with Keeper* scenario. Details of the experiment results elaborated in the next section.

V. Experimental Result

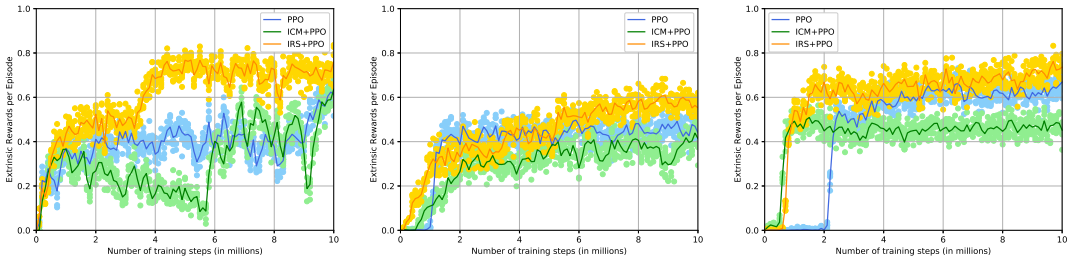
Although our experiment evaluates on two types of environments with different extrinsic rewards, we use scoring reward to measure the performance of algorithm. First, we evaluated the performance of *IRS+PPO* and two baseline methods under different settings with different extrinsic rewards in different scenarios (Section V-A). After that, generalization is evaluated on the map with novel scenarios and the same goal.

A. Reward Setting and Result

We compare the experimental results by considering two different setups with *sparse* and *very sparse* rewards. In section IV-A, we detailed the extrinsic reward settings for the experimental scenarios. Regardless of the scenarios, the sparse extrinsic rewards are only provided to the agent when it takes the goal to a specific location. And in the environment, the distance between the agent from the goal and the state of whether to face the opponent are different. The closer the agent is to the goal, or the less defense the opponent set, the more likely it is to score.

Very sparse reward A good intrinsic reward generation policy is one that guides the agent to discover the target and enables the agent to accomplish the task as much as possible. In the case of playing a game, we expect the agent to win the game even with only a goal and without any reward in the process of the game. In order to test whether IRS can learn a good inner reward generation policy, we trained it on three football scenarios only with scoring reward from the environment. The results are shown in Figure 5.

First of all, the influence of the difficulty of different scenarios can be clearly seen from the figure 5. Harder tasks take significantly more steps to make some progress. The results shown in Figure 5 indicates that the *IRS+PPO* agent is superior in all cases, although it also takes a long time to learn to score with very sparse reward. In the first scenario, the agent needs to run a long distance and get closer to the goal to get a chance to score with keeper. Results shown in Figure 5(a) indicate that the performance of *IRS+PPO* agent is better than that of other baseline methods, especially PPO agent. After 10 million training steps, PPO agent with *very sparse reward* has not been able to score goals in the scenario with the longest distance between the agent from the goal, *Run to Score with Keeper*. In the second scenario, the agent needs to pass the ball to another player who is not defended by the opponent, and then scores with keeper. The final performance of the three methods is extremely similar, but *ICM+PPO* and *IRS+PPO* learn faster. In the third scenario, it is very difficult for the agent to take the ball over from the opponent and then scores with keeper, so the agent must learn to pass the ball



(a) Run to score with keeper (b) Pass and shoot with keeper (c) Run pass and shoot with keeper

Fig. 6. Evaluating the performance of three methods in the environment with sparse reward.

with the teammate or run away from the opponent to get a chance to score. All the algorithms do enable the agent learn to rely on teammates to complete the task, and end up well after 10 millions training steps.

In the environment with *very sparse reward*, the distance between the agent from the goal makes it difficult for the agent to learn to score in the early stage. So in the first scenario with the longest distance, the agent did not perform well in the beginning. But once the agent explores the goal, the learning speed will be extremely fast. In the hardest scenario *Run, Pass and Shoot with Keeper*, It is difficult for an agent to learn to cooperate with teammates to deal with complex tasks.

Sparse reward Due to the sparsity of scoring, we also use the checkpoint reward from the environment. The idea behind this reward is to encode our domain knowledge that scoring requires the ball to be close to the opponent's goal [12]. The experimental results with *sparse reward* are shown in Figure 6.

In the first scenario, all methods can quickly learn useful skills to improve performance. After 10 million training steps, the *IRS+PPO* agent scored more than 0.7 on average when the average score of the other two methods is around 0.6. *ICM+PPO* agent is very unstable and reduces learning performance over time. In the second scenario, the performance of all methods is very similar, but the dots representing the *IRS+PPO* method are more concentrated which means the robustness of it is better. In the third scenario, compared to PPO, *IRS+PPO* and *ICM+PPO* methods learn the way to consistently score earlier, but the *IRS+PPO* method performs best.

Overall, the *IRS+PPO* agent learns much faster than the baseline agents in the *sparse reward* case. Compare to the environment with *very sparse reward*, the checkpoint reward reduces the final performance of the agent in some scenarios. Although we know that scoring requires the ball to be close to the opponents goal, the rewards provided by the environment will also make the agent ignore the final goal to some extent. Regrettably, our proposed method does not solve the bad influence from checkpoint reward, but it is well coupled to various extrinsic reward settings regardless of any scenarios.

VI. CONCLUSION

In this paper, we present a novel adaptive IRS approach which does not require specific domain knowledge, but rather enables the agent to learn how to generate inner rewards to guide itself in any scenarios online jointly with the actual reinforcement learning process. Experiments results confirmes that our method accelerate the learning speed of the agent and improve the learning performance which performs best in all scenarios. Compare to the baseline methods, the results demonstrate that our IRS method is significantly superior to the baseline PPO in the task, better than the ICM method in some scenarios with *very sparse reward*. Although ICM is good enough to explore the environment, it is too weak to enable the agent to learn complex skills with *very sparse reward*. Because IRS enable agents try a variety of actions and progressively favor those that appear to be best and provides a good solution to the trade-off between exploration and exploitation, our proposed method is better than the ICM in exploring the environment, but it is stronger in exploitation that guiding the agent to learn useful skills. Whats more, the results suggest that the proposed method enables an agent to learn generalizable skills applied to novel scenarios. What deserve to be ment is that, our method integrates seamlessly with any reinforcement learning algorithm that learns a value or action-value function through experience in any scenarios with sparse reward.

VII. FUTURE WORK

Facing the strong opponents, the agent can not solve the problem perfectly and there exists much space for improvement with a limited number of training steps. Our method solves the difficulties brought by the fixed environment well, but it has a limited effect due to the strength and uncertainty of the opponents. And our proposed method does not help the agent learn to work with teammates and score with the action of teammates. Next, we will further study how to solve this problem, and enable the agent to cooperate with the robot teammates. And we also want to further study how to use reward shaping on multi-agents tasks to make the agents cooperate better. We believe that future applications can benefit from a better understanding of how reward signals affect learning and from improved methods for designing them.

References

[1] J. Asmuth, M. L. Littman, and R. Zinkov. Potential-based shaping in model-based reinforcement learning. In *AAAI*, pages 604–609, 2008.

[2] M. Bansal, A. Krizhevsky, and A. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

[3] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[4] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

[5] A. Demir, E. Çilden, and F. Polat. Landmark based reward shaping in reinforcement learning with hidden states. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1922–1924. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[6] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): methodology and large-scale application on downtown toronto. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1140–1150, 2013.

[7] M. Grzes and D. Kudenko. Learning shaping rewards in model-based reinforcement learning. In *Proc. AAMAS 2009 Workshop on Adaptive Learning Agents*, volume 115, 2009.

[8] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[9] S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274, 2002.

[10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[12] K. Kurach, A. Raichuk, P. Stańczyk, M. Zając, O. Bachem, L. Espeholt, C. Riquelme, D. Vincent, M. Michalski, O. Bousquet, et al. Google research football: A novel reinforcement learning environment. *arXiv preprint arXiv:1907.11180*, 2019.

[13] M. Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

[14] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[16] 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*, 518(7540):529, 2015.

[17] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

[18] A. Y. Ng and M. I. Jordan. *Shaping and policy search in reinforcement learning*. PhD thesis, University of California, Berkeley Berkeley, 2003.

[19] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

[20] J. Randløv and P. Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *ICML*, volume 98, pages 463–471, 1998.

[21] R. M. Ryan and E. L. Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

[22] J. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pages 222–227, 1991.

[23] 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.

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

[25] 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*, 529(7587):484, 2016.

[26] P. J. Silvia. Curiosity and motivation. *The Oxford handbook of human motivation*, pages 157–166, 2012.

[27] B. F. Skinner. Reinforcement today. *American Psychologist*, 13(3):94, 1958.

[28] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[29] X. Tang, Z. T. Qin, F. Zhang, Z. Wang, Z. Xu, Y. Ma, H. Zhu, and J. Ye. A deep value-network based approach for multi-driver order dispatching. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1780–1790. ACM, 2019.