Leveraging Coordination with Joint Intrinsic Motivation in Multi-Agent Deep Reinforcement Learning

Paper ID 1219

Abstract. Multi-agent deep reinforcement learning (MADRL) problems often encounter the challenge of sparse rewards. This challenge becomes even more pronounced when coordination among agents is necessary. As performance depends not only on one agent's behavior but rather on the joint behavior of multiple agents, finding an adequate solution becomes significantly harder. In this context, a group of agents can benefit from actively exploring different joint strategies in order to determine the most efficient one. In this paper, we propose an approach for rewarding strategies where agents collectively exhibit novel behaviors. We present JIM (Joint Intrinsic Motivation), a multi-agent intrinsic motivation method that follows the centralized learning with decentralized execution paradigm. JIM rewards joint trajectories based on a centralized measure of novelty designed to function in continuous environments. We demonstrate the strengths of this approach both in a synthetic environment designed to reveal shortcomings of state-of-the-art MADRL methods, and in simulated robotic tasks. Results show that joint exploration is crucial for solving tasks where the optimal strategy requires a high level of coordination.

1 Introduction

One crucial aspect of human intelligence is its ability to act coincidentally with other human beings, to either cooperate or compete in a given task. This has led researchers to study reinforcement learning (RL) in the context of multi-agent systems (MAS), where multiple artificial agents interact with their environment and each other while concurrently learning to perform a task [28, 11]. However, having multiple agents in the environment makes the RL process significantly more difficult for several reasons [33]. In particular, the global reward depends on the actions of several independent agents, which makes the search for the optimal joint policy more complicated.

Recently, multi-agent deep reinforcement learning (MADRL) approaches have combined advancements in RL and deep learning to tackle long-standing problems in MAS such as credit assignment or partial observability [11, 22, 6]. These techniques are able to solve very complex multi-agent tasks such as autonomous driving [25] or real-time strategy video games [17]. However, major issues still remain with these approaches, such as the problem of relative overgeneralization [32, 30] where agents struggle to find the optimal joint policy because local policies are attracted towards suboptimal areas of the search space. This makes most algorithms inefficient in tasks where the optimal strategy requires strong coordination among agents. Relative overgeneralization can be described as a problem of exploration of the joint-state space: as the success of the MAS depends on the coordination of multiple agents, exploring the joint-observation space is required to discover optimal joint behaviors. In this paper, we address the question of how to explore the joint-state space to efficiently discover superior coordinated strategies for solving the task at hand.

In single-agent RL, the problem of exploration has been studied to solve hard exploration tasks where positive reward signals are very sparse. One solution is to use intrinsic motivation [24, 18, 10] to incite agents to explore unknown parts of the environment. In addition to the environment reward, agents are given an auxiliary reward related to the novelty of encountered states. Maximizing this intrinsic reward leads agents to visit previously unexplored regions of the environment, ultimately discovering new solutions to the task. These methods have shown great success in helping RL agents solve hard exploration tasks [19, 2].

In the multi-agent setting, intrinsic objectives have also been studied to induce different kinds of behaviors in agents such as coordinated exploration [8], social influence [9, 29] or alignment with other agents' expectations [13]. However, previous works have only used local observations to generate intrinsic rewards. With partial observability, local observations often lack crucial information to fully understand the current configuration of the environment. In the context of exploration, an intrinsic reward based only on local observations will lead to each agent exploring their own observation space, paying no attention to the current state of other agents. This can result in inefficient exploration in cooperative tasks where the success of the MAS depends on the coordination of all agents.

In this paper, we introduce a novel multi-agent exploration approach called Joint Intrinsic Motivation (JIM) which can be combined with any MADRL algorithm that follows the centralized training with decentralized execution paradigm (CTDE). JIM exploits centralized information to motivate agents to explore new coordinated behaviors. In order to compute joint novelty, JIM combines two previous state-of-the-art approaches: NovelD [35] for exploring unknown parts of the environment, and E3B [7] for having more diverse trajectories. Adding this auxiliary reward to the agents' objective incites them to diversify their collective behavior until they have a fair knowledge of the environment and can focus on the main task at hand.

To demonstrate the advantages of our approach, we first design a simple test environment to showcase a clear example of relative overgeneralization. We show that the state-of-the-art algorithm QMIX [22] struggles in this scenario and that motivating the exploration of coordinated behavior helps solve the task. Next, we validate these results in a continuous virtual environment, showing that coordination tasks benefit from joint exploration. Finally, we present an ablation study to demonstrate the strength of our intrinsic reward definition.

2 Related Works

In recent years, deep reinforcement learning techniques have been used in the context of MAS to tackle long-standing issues in multi-agent learning. Successful single-agent RL approaches have been adapted to the CTDE framework [11, 34], using a centralized value function to guide the training of decentralized policies. Recent studies have investigated the problem of credit assignment [6] in MADRL, i.e., distributing the global reward among agents based on their participation. Value factorization methods also do this implicitly [27], combining the output of local value functions into a centralized one that predicts the current value of the system. In particular, QMIX [22] uses a separate network to predict the Q-value of the joint action, given the output of local Q-values and the global state of the environment. QMIX has established itself as a long-standing state-of-the-art approach, despite its inherent limitations that several works have tried to surpass [26, 21]. However, MADRL algorithms have been shown to suffer from the problem of relative overgeneralization [32, 31]. So far, few works have addressed this problem: Wei et al. [31] propose maximum entropy RL to explore the joint-action space, and MAVEN [14] augments QMIX using a hierarchical policy to guide the exploration of joint behaviors.

A promising approach to overcome relative overgeneralization is to intrinsically motivate agents to explore their environment, ultimately discovering the optimal reward signals. In single-agent RL, curiosity has been defined to help agents solve hard exploration tasks [24, 18, 10] by rewarding the visitation of states considered as novel. For measuring novelty, several methods have used the error of trainable prediction models. The Intrinsic Curiosity Module (ICM) [19] trains a model of environment dynamics and uses the prediction error as a measure of novelty. Random Network Distillation (RND) [3] uses a target network that produces a random encoding of the state and trains a predictor network to generate the same encoding, the prediction error being the measure of novelty. The idea behind these two approaches is that the prediction models will yield low novelty for states similar to what they have trained on while producing high novelty for unknown parts of the environment. RIDE [20] and NovelD [35] use respectively ICM and RND to compute a reward from the difference of novelty between the next state and the current state, pushing the agents to always seek novel states. Similarly, NGU [2] and E3B [7] use clustering techniques to reward states that are distant from previous states. Finally, a similar approach is proposed by AGAC [5] which trains an adversarial policy to predict the main policy's output, the latter being rewarded with the former's prediction error.

In MADRL, recent works have demonstrated the effectiveness of intrinsic rewards in promoting desirable behaviors in groups of agents. One example is social influence [9, 29] that rewards agents for performing actions that have a significant impact on other agents. Ma et al. [13] propose an intrinsic reward based on the average alignment with other agents' expectations, promoting more predictable behaviors in agents. Lupu et al. [12] propose to reward policies that perform diverse trajectories in comparison to a population of agents, which is shown to help train agents to be more versatile. Du et al. [4] use intrinsic objectives as a credit assignment technique. Finally, Iqbal and Sha [8] propose an approach for coordinated exploration using several metrics for estimating the novelty of observations that depend on all agents' past experiences. However, their model is computationally expensive and does not address the exploration of the joint-observation space, which can be problematic for hard exploration tasks where relative overgeneralization can occur.

In this paper, we address the challenge of relative overgeneralization by rewarding agents for exploring the joint-observation space. In the following sections, we will present the necessary formal background and an overview of the proposed algorithm that implements joint intrinsic motivation.

3 Background

3.1 Dec-POMDP

To describe cooperative multi-agent tasks, we use the definition of decentralized partially-observable Markov decision process (Dec-POMDP) [16], defined as a tuple $\langle \mathbf{S}, \mathbf{A}, T, \mathbf{O}, O, R, n, \gamma \rangle$ with n being the number of agents. $\mathbf{S}$ describes the set of global states s of the environment. $\mathbf{O}$ is the set of joint observations, with one joint observation $\mathbf{o} = \{o_1, ..., o_n\} \in \mathbf{O}$, and $\mathbf{A}$ the set of joint actions, with one joint action $\mathbf{a} = \{a_1, ..., a_n\} \in \mathbf{A}$. T is the transition function defining the probability $P(s'|s, \mathbf{a})$ to transition from state s to next state s' with the joint action $\mathbf{a}$. O is the observation function defining the probability $P(\mathbf{o}|\mathbf{a}, s')$ to observe the joint observation $\mathbf{o}$ after taking joint action $\mathbf{a}$ and ending up in s'. $R : \mathbf{O} \times \mathbf{A} \rightarrow R$ is the reward function producing at each time step the reward shared by all agents. Finally, $\gamma \in [0, 1)$ is the discount factor controlling the importance of immediate rewards against future gains.

3.2 Intrinsic rewards

In Section 2, we introduced intrinsic motivation as a way to incite agents to actively explore their environment. To this end, at each time step t, agents receive an augmented reward $r_t = r_t^e + \beta r_t^{int}$, where r_t^e is the extrinsic reward given by the environment, r_t^{int} is the intrinsic reward and β is a hyperparameter controlling the weight of the intrinsic reward in the agents' objective.

In this Section, we describe three methods of intrinsic rewards from the literature that we will use later in Section 4.2.

Random Network Distillation (RND). In RND, Burda et al. [3] compute novelty using two neural networks with the same architecture: a target network ϕ and a predictor network ϕ'. The target's parameters are initialized randomly and fixed. It takes as input the state s_t and produces a random embedding $\phi(s_t)$. The predictor is trained to output the same embedding, minimizing the Euclidean distance:

$$RND_t(s_t) = \|\phi(s_t) - \phi'(s_t)\|_2 \tag{1}$$

This distance is used as a measure of the novelty of state s_t and is given as an intrinsic reward to agents.

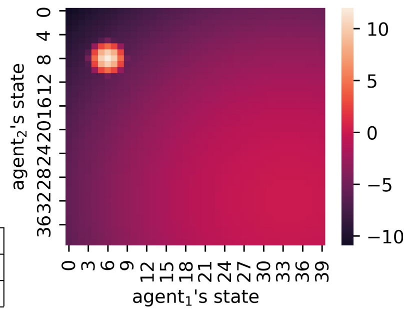
Novelty Difference (NovelD). Zhang et al. [35] build upon RND to devise a novelty criterion termed NovelD. It is defined as follows:

$$N(s_t, s_{t+1}) = \max[RND(s_{t+1}) - \alpha RND(s_t), 0] \times \\ \times \mathbb{1}\{N_e(s_{t+1}) = 1\} \tag{2}$$

with α a scaling factor and N_e an episodic count of visited states. The first part is the core of the novelty criterion. It uses RND to reward agents for positive gains in novelty between the current and the next states. The second part is an episodic restriction that ensures the reward is given only when state s_{t+1} is observed for the first time in this episode. This restriction limits the use of NovelD to discrete state spaces as it relies on an explicit count of visited states.

	A	B	C
A	10	-5	-5
B	-5	7	7
C	-5	7	7

(a)



(b)

Figure 1: Two examples of relative overgeneralization: (a) payoff matrix of a social dilemma game, (b) heat-map of the reward function of the `rel_overgen` environment for $D = 40$ and $\delta = 30$.

Exploration via Elliptical Episodic Bonuses (E3B). With E3B, Henaff et al. [7] propose an episodic bonus based on the position of the observed state with respect to an ellipse that fits all states previously encountered in the current episode. Formally, it is computed as follows:

$$b(s_t) = \psi(s_t)^\top C_{t-1}^{-1} \psi(s_t), \tag{3}$$

with

$$C_{t-1} = \sum_{i=1}^{t-1} \psi(s_i)\psi(s_i)^\top + \lambda I, \tag{4}$$

where I is the identity matrix and λ a scalar coefficient. ψ is an embedding network trained using an inverse dynamics model [19]: embeddings of following states $\psi(s_t)$ and $\psi(s_{t+1})$ are used by a separate neural network trained to predict the action a_t taken between these states. As a result of this training process, ψ encodes parts of the observation that are controllable by the agents (please refer to [7] for details). Intuitively, b can be understood as a generalization of a count-based episodic bonus for a continuous state space. States that are close to previously encountered states in the current episode will yield low bonuses, whereas states that are very different will produce high bonuses.

4 Algorithm

In this Section, we introduce the Joint Intrinsic Motivation (JIM) exploration criterion for coordinated multi-agent exploration. Firstly, we describe the motivation behind our approach by providing a detailed description of the problem of relative overgeneralization. Then, we define the intrinsic reward used for motivating agents to explore a continuous state-space environment in a coordinated fashion. Finally, we explain how this reward is used in a multi-agent setting with JIM.

4.1 The challenge of coordinated actions

Addressing hard exploration environments is challenging because of the few positive reward signals that exist to guide the agent's learning process. This becomes even worse with MAS as the completion of a task depends on the actions of multiple independent agents. When strong coordination is needed, agents will struggle to find the optimal strategy and settle for an easier suboptimal joint strategy, which is a problem known as relative overgeneralization [32, 30]. Figure 1a provides an example of a social dilemma game where relative overgeneralization occurs. The optimal strategy requires both agents to choose action A. But if only one agent chooses action A, the payoff is very bad. Therefore, agents will independently prefer to take actions B or C, as action A most often leads to sub-optimal outcomes.

In MAS, this can be seen as a problem of ill-coordinated exploration. As success depends on coordinated behaviors, exploration of joint policies is required in order to discover which ones lead to optimal returns. In the example of Figure 1a, exploring independent strategies will lead to ultimately choosing suboptimal actions as they individually may yield better expected returns. On the other hand, we argue that uniformly exploring joint actions would enable agents to choose optimal joint strategies more often and consequently learn more efficient individual behaviors. The approach described in the following two sections implements an algorithm that efficiently rewards agents for exploring the joint-observation space, in order to consistently find optimal strategies.

4.2 Double-timescale Intrinsic Reward

Similarly to previous works on single-agent intrinsic motivation [2], we define a novelty metric that combines two exploration criteria working at different timescales:

- A **life-long exploration criterion (LLEC)** that captures how novel is the current observation with respect to all observations since the beginning of training.
- An **episodic exploration criterion (EEC)** that captures the difference between the current observation and all previous observations in the current episode.

Intuitively, the life-long reward motivates agents to search for never-experienced parts of the environment. Meanwhile, the episodic bonus induces more diverse trajectories. These two elements will feed each other and reinforce agents to efficiently explore their environment.

Concretely, for each transition from state s_t to the next state s_{t+1}, we define the double-timescale intrinsic reward as follows:

$$r_t(s_t, a_t, s_{t+1}) = N_{LLEC}(s_t, s_{t+1}) \times N_{EEC}(s_{t+1}), \tag{5}$$

with the life-long novelty N_{LLEC} inspired from NovelD [35] (see Eq. (2)):

$$N_{LLEC}(s_t, s_{t+1}) = \max[RND(s_{t+1}) - \alpha RND(s_t), 0], \tag{6}$$

with α a scaling factor and RND the novelty measure (see Eq. (1)). Further, the episodic novelty uses the bonus from E3B [7] (see Eq. (3)):

$$N_{EEC}(s_{t+1}) = \sqrt{2b(s_{t+1})}. \tag{7}$$

We remove the episodic restriction of NovelD as it relies on an episodic count of visited states. This makes it impractical in a continuous state space, as one state is very unlikely to be visited twice. Instead, we scale the life-long novelty using the elliptical episodic bonus b from E3B [7]. This bonus acts as an episodic restriction by scaling N_{LLEC} up or down, depending on the novelty of the current state compared to what has been observed during the current episode. As b provides very large bonuses and decreases very fast, we use $\sqrt{2b(s_{t+1})}$ to both smooth out large values and increase small ones.

Combining these two rewards makes it possible to take the benefits of both. N_{LLEC} pushes agents to explore regions of the state space that are not well-known to agents. Meanwhile, N_{EEC} favors diverse trajectories, inciting agents to always seek new observations during a single episode. As the agents explore their environment, the prediction error of RND (see Eq. (1)) slowly decreases. Thus, N_{LLEC} decreases as well, tending toward zero, allowing agents to progressively focus on the extrinsic reward. Finally, as the episodic restriction does not rely on any explicit count of visited states, it can be used in continuous state spaces.

4.3 *The Joint Intrinsic Motivation algorithm*

Building from the intrinsic reward introduced previously, we propose the Joint Intrinsic Motivation (JIM) algorithm to incite MADRL agents to explore the joint-observation space. At each time step, all agents receive the same global reward $r_t = r_t^e + \beta r_t^{JIM}$, where r_t^e is the extrinsic reward given by the environment, r_t^{JIM} is our joint exploration criterion, and β is a hyper-parameter controlling the weight of the intrinsic reward. The exploration criterion in JIM uses the double-timescale intrinsic reward defined earlier to compute the novelty of the joint observation:

$$r_t^{JIM}(\mathbf{o}_t, \mathbf{a}_t, \mathbf{o}_{t+1}) = N_{LLEC}(\mathbf{o}_t, \mathbf{o}_{t+1}) \times N_{EEC}(\mathbf{o}_{t+1}), \quad (8)$$

where $\mathbf{o}_t = \{o_t^i\}_{0\leq i\leq N}$, i.e., the concatenation of all local observations. Figure 2 shows the architecture for JIM, compared to an approach of local intrinsic motivation. The local method has one intrinsic motivation module for each agent, computing the novelty of local observations. Meanwhile, JIM computes only one intrinsic reward, making for a lighter architecture. Agents being rewarded by the novelty of the joint observation, they will learn to search for new combinations of observations with other agents of the system, rather than only exploring their local-observation space.

As JIM uses joint observations for computing the intrinsic reward, it can be associated with any MADRL algorithm that fits in the CTDE paradigm. These algorithms usually employ a centralized value function [11, 22, 34] that looks at the joint observation to predict the value of the agents' actions. Such centralized value functions will be able to associate rewards provided by JIM to new configurations in the joint observation space, thus inducing agents to actively search for these configurations.

One could note that the joint observation has two notable drawbacks: the number of dimensions grows exponentially with the number of agents and there is a risk of capturing redundant information. These issues are both alleviated with JIM as we use dimensionality reduction techniques. N_{LLEC} and N_{EEC} use respectively ϕ and ψ (see Section 3.2) as embedding networks to encode the joint observation into a more condensed representation that contains only the required information.

5 Implementation details

As previously said, JIM can be used to augment any MADRL approach that fits in the CTDE paradigm. In the experiments presented in the next Section, we use JIM with QMIX [22]. We use the default QMIX architecture and hyperparameters, along with prioritized experience replay [23]. In all experiments, we compare three algorithms:

- **QMIX+JIM**, augmenting QMIX with joint exploration, as shown in Figure 2, and described in Section 4.2.
- **QMIX+LIM**, a degraded version of QMIX+JIM where local (rather than global) intrinsic motivation is used. Each agent generates its own intrinsic reward based solely on its local observation, using the same reward definition as JIM (see Section 4.2). The LIM (Local Intrinsic Motivation) architecture is described in the Technical Appendix.
- The original state-of-the-art **QMIX** algorithm [22] with no intrinsic motivation, used as a baseline.

Note that the training and execution of the agents are identical in these three algorithms. The only thing that changes is whether (and how) an intrinsic reward is computed at each time step.

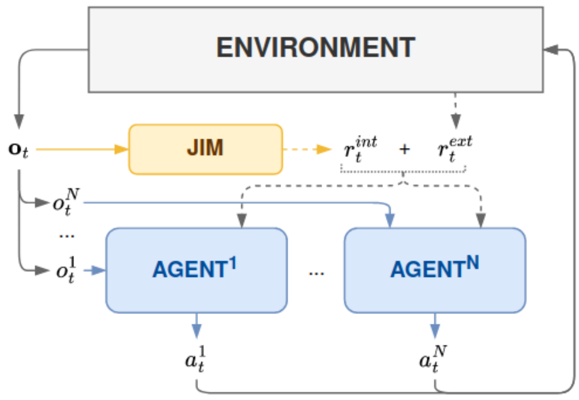


Figure 2: Architecture for the Joint Intrinsic Motivation (JIM) algorithm. JIM has only one intrinsic motivation module for the whole multi-agent system, computing novelty of the joint observation $\mathbf{o}_t$. However, agents only use their local observation to choose their action.

To ensure a fair comparison between JIM and LIM, we use different sets of hyperparameters for the two versions in order for them to have a similar number of learnable parameters. All hyperparameters used in our experiments are listed in the Technical Appendix. Our code is available in the Supplementary Materials.

6 Experiments

In this Section, we present a set of experiments to evaluate the exploration criterion of JIM when used along the state-of-the-art QMIX algorithm [22]. First, we show the results in a synthetic discrete environment where the problem of relative overgeneralization can be artificially tuned and observe that JIM helps alleviate this issue. Then, we test our approach on pseudo-realistic robotic tasks in a continuous environment and show that exploring the joint-observation space helps solve cooperative tasks. Finally, we present an ablation study of our approach by comparing JIM with two simpler versions that each lack one of the two exploration criteria described in Section 4.2, showing the advantage of combining the two.

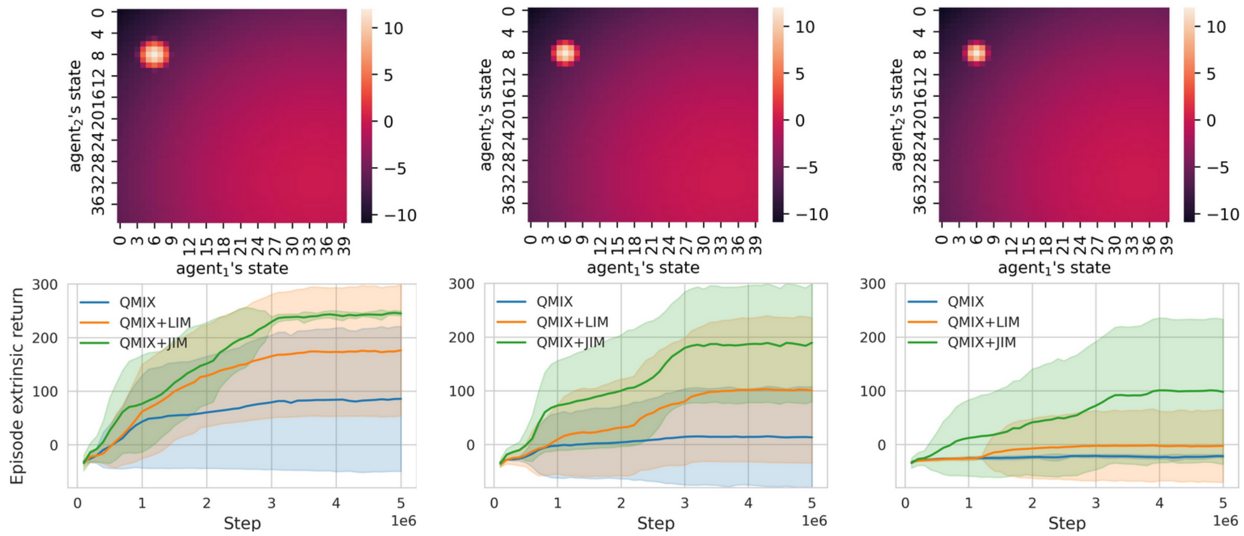
6.1 *Addressing relative overgeneralization*

6.1.1 *Environment definition*

To demonstrate how joint exploration helps solve the problem of relative overgeneralization, we design a simple test environment that expands the example shown in Figure 1a. In this environment called `rel_overgen`, two agents can move on a discrete one-dimensional axis with D possible positions. The two agents are denoted by their position, namely x and y. At each time step, agents observe their position as a one-hot vector $o_t^{\mathrm{x}} = \{o_t^{\mathrm{x},j} = 1 \text{ if } \mathrm{x} = j,\ 0 \text{ otherwise}\}_{0\leq j<D}$ and can choose between three actions: move in one direction or the other or stay in position. They receive a reward corresponding to their combined position:

$$r_t^e(x, y; \delta) = \max\Big(R^+ - \frac{\delta}{D}\left[(x - r_x^+)^2 + (y - r_y^+)^2\right], \quad R^- - \frac{1}{8D}\left[(x - r_x^-)^2 + (y - r_y^-)^2\right]\Big) \quad (9)$$

The result of this formula is displayed in Figure 1b. The reward combines two hyperboles in opposite corners: one narrow that culminates



(a) easy ($\delta = 30$) (b) hard ($\delta = 40$) (c) very hard ($\delta = 50$)

Figure 3: Performance of variants of QMIX in the `rel_overgen` environment, with three levels of difficulty. On top, we show the heat maps representing the reward function in each environment version, where the difficulty is dictated by the width coefficient of the optimal reward spike δ (as defined in Eq. (9)). Increasing δ leads to a slightly narrower optimal reward spike. Below is shown the performance during training of QMIX with no intrinsic reward (QMIX), local intrinsic motivation (QMIX+LIM), and joint intrinsic motivation (QMIX+JIM) (mean and standard deviation shown for 15 runs each). We see that a slight decrease in the size of the optimal reward spike results in a considerable increase in the difficulty of the task.

at R^+ at position (r_x^+, r_y^+), and another much wider that plateaus at R^- at position (r_x^-, r_y^-). We set the optimal reward R^+ to 12 and the suboptimal R^- to 0. The width of the optimal reward spike is controlled by the parameter δ: a higher δ value yields a narrower spike.

The goal of the agents is to find where to go to maximize their aggregated rewards. The wide suboptimal hyperbole will probably attract agents to minimize their loss. The optimal reward spike is difficult to find because it covers a small portion of the state space, but it guarantees much greater returns. We can vary the difficulty of the task by changing the width of this optimal reward spike: the narrower the spike, the harder it is to find.

In this environment, we expect MADRL methods to struggle to find the optimal reward spike. Exploring local states could help but would not be sufficient to consistently solve the task. As the dimension D of the local-state space is fairly small, novelty rewards will quickly vanish and will not help agents to find the optimal reward spike. Exploring the joint-observation space adequately is required in order to consistently find optimal rewards. As JIM will reward exploration until all combined positions (x, y) are visited several times, agents will visit the optimal reward spike more often, thus helping them to learn the optimal coordinated strategy.

6.1.2 *Results*

The results shown in Figure 3 confirm the hypotheses formulated in the previous Section. We show the performance of QMIX, QMIX+LIM, and QMIX+JIM across 15 independent runs each. Further, we present results in three difficulty levels dictated by the width of the optimal reward spike. The results clearly demonstrate the importance of exploring the joint-state space. QMIX alone manages to get a positive reward on the easy scenario, but its performance is both lower and with a larger standard deviation compared to the two other algorithms. In the harder scenarios, QMIX's performance degrades strongly, never finding any positive reward in the hardest case. JIM clearly improves the performance. In the easy scenario, QMIX+JIM consistently goes for the optimal reward spike. In the harder settings, it still performs well on average, even in the "very hard" scenario where the optimal reward spike covers only 0.013% of all combined positions. The results of QMIX+LIM show that exploring the local-observation space helps agents find the optimal reward spike more often. However, it performs worse than JIM as it does not ensure that all combined positions are sufficiently explored. This shows that exploring the joint-observation space is crucial to allow agents to discover optimal coordinated behaviors.

6.2 *Coordination tasks in a continuous environment*

6.2.1 *Environment definition and setups*

Next, we study how JIM scales to more realistic continuous environments and more complex tasks. We use the multi-agent particle environment[1] (MPE) [15, 11] to simulate cooperative robotic tasks that require a high degree of coordination. The state space of MPE is continuous: in our setups, agents receive as observation a vector with their position in the two-dimensional space and, for all the other entities in the environment, their relative position and velocity. QMIX being a value-based approach, it is restricted to discrete action selection. Thus, agents can choose between five actions: move in any four cardinal directions or stay in place. Agents can navigate in a closed two by two meters area and are represented by circles of four centimeters.

The first task we define is a box-pushing task that requires agents to push an object and place it on top of a landmark. Figure 4a shows a screenshot of this scenario. At the start of each episode, the landmark is randomly placed in any one of the four corners. The initial positions of the agents and the object are randomly set. If the agents man-

[1] https://github.com/openai/multiagent-particle-envs

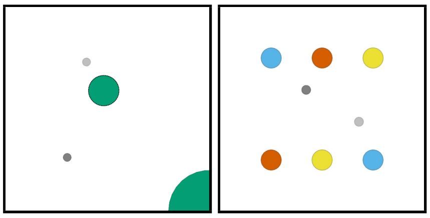

(a) (b)

Figure 4: Screenshots of our tasks in the MPE, agents are the small grey circles. (a) shows the cooperative box pushing scenario, where the green circle in the middle is an object to deliver to the landmark in the bottom right corner. (b) shows the coordinated placement task, the colored circles representing landmarks agents have to navigate on to gain rewards.

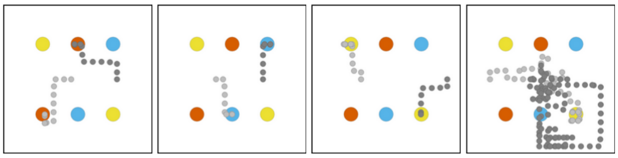

(a) (b) (c) (d)

Figure 5: Examples of trajectories in the coordinated placement task. They present different levels of strategy, with (a) > (b) > (c) > (d): (a) optimal strategy with both agents on red, (b) both on blue, (c) both on yellow, (d) one on blue/yellow.

age to push the object and place it on the landmark, the episode ends and they receive a reward of +100. Agents also get a small penalty of -1 at each time step to reward faster strategies. This reward is purposefully defined to be very sparse, in order to study how exploration helps in this kind of scenario.

Lastly, we study the cooperative placement task where agents must place themselves over landmarks in order to maximize their reward. As shown in Figure 4b, there are two sets of three landmarks of three different colors. The reward given at each time step depends on the placement of the agents on the landmarks. The optimal state is having both agents placed on the red landmarks, yielding a reward of +10 at each time step. The blue and yellow landmarks act as deceiving rewards, yielding much smaller rewards (+2 for blue, +1 for yellow). To increase the deceiving aspect of the blue and yellow landmarks, we also reward agents collectively by +0.5 if only one of them stands on one of these two colors. This leads to a need for coordination between agents, as they will locally find that going on blue or yellow landmarks systematically leads to a small reward. Only if agents explore their environment well, will they discover that they need to be both on red to get the optimal reward signal. Importantly, this scenario features partial observability, with agents only having information about entities close to them (less than sixty centimeters from them), the others being masked with neutral values. This means that agents do not see which landmark the other agent goes to. [2]

6.2.2 Results

Results of training QMIX, QMIX+LIM, and QMIX+JIM in the cooperative box pushing scenario are shown in Figure 6 (median and confidence interval shown for 11 runs each). First, we observe that

[2] See the Technical Appendix for a detailed description of these two tasks.

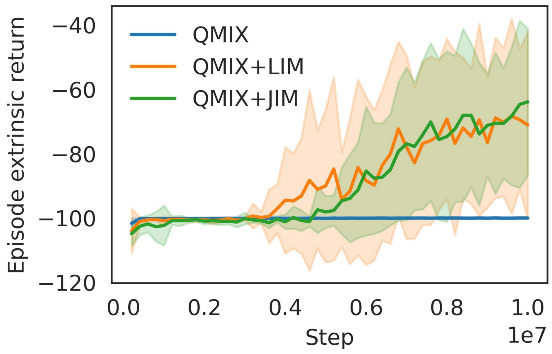
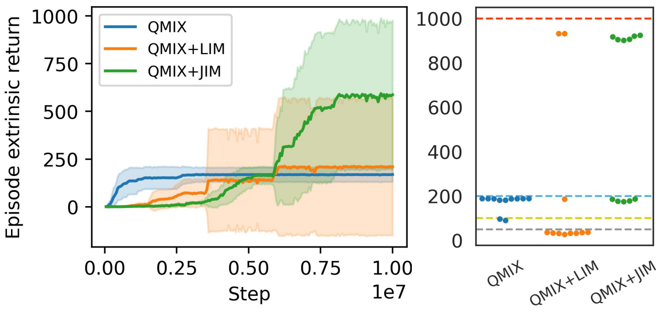


Figure 6: Training curves of the three variants of QMIX in the cooperative box pushing task, with the mean and standard deviation across 11 runs each.

QMIX alone performs very poorly as it is unable to find the solution to the task. The most likely cause of this is the high sparsity of the reward function, making it impossible for agents to discover the objective with random exploration of the environment. Second, we see that JIM and LIM achieve similar levels of performance. While coordination can help agents perform well, it is actually not a requirement for this task. In fact, one agent alone is able to push the object and place it on the landmark. Thus, exploring the space of joint configurations is not helpful in this scenario. This shows however that actively exploring the environment is crucial in tasks where the reward function is very sparse.

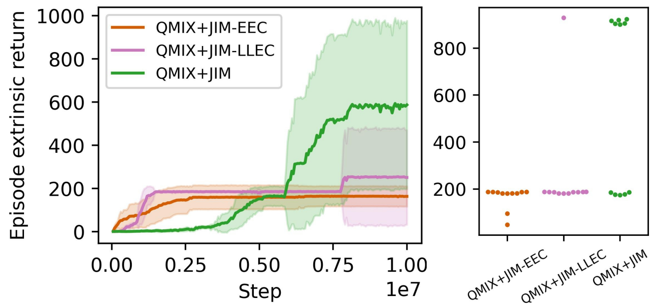
Conversely, experiments in the coordinated placement task demonstrate well the importance of exploring jointly. In Figure 7a, we display the training curves of QMIX, QMIX+LIM, and QMIX+JIM, with 11 independent runs each. Figure 7b shows the performance of each run at the last iteration of training. The colored dashed lines give an insight into the level of strategy learned by each run. These levels of strategy can be visualized with example trajectories displayed in Figure 5. We see that QMIX alone almost always goes for the blue landmarks, while sometimes settling for the yellow ones. This indicates that without actively exploring the environment, QMIX gets attracted to deceptive rewards and is unable to find the optimal strategy. While QMIX+LIM seems slightly better than QMIX on the training curves, the individual run performance shown in Figure 7b shows that LIM arguably performs worse. Two runs manage to find the optimal strategy, but LIM often performs poorly with only one agent on a blue or yellow landmark. This demonstrates that exploring the space of local observations can be helpful, as it pushes agents to explore the environment. But exploring local observations can also be misleading as they do not contain all the information about the state of the environment. With JIM however, exploring the joint-observation space clearly improves the quality of the chosen strategies. More than half of the time, QMIX+JIM finds the optimal reward signal and develops an effective strategy to go on red landmarks, showing that JIM allows for more efficient exploration of coordinated behaviors. When agents do not find the optimal strategy, they stick with the best sub-optimal strategy to go both on blue. This shows that agents benefit from exploring the space of joint observations as they are directly linked to the obtained reward, whereas local observations lack crucial information to understand the global reward.

Another advantage of JIM is the simplicity of its architecture. While LIM and similar approaches in previous works [8, 4, 29] require computing one intrinsic reward for each agent, JIM only com-



(a) (b)

Figure 7: Performance of the three variants of QMIX in the coordinated placement task. (a) shows the training curves with the mean and standard deviation across 11 independent runs each, while (b) displays the performance of each independent run at the last iteration of training. Dashed lines on (b) indicate the optimal (unattainable) level of return obtained with different strategies: red, blue, and yellow lines represent the return obtained if both agents are on landmarks of the related color during 100 steps (the duration of an episode), the grey line corresponds similarly describes only one agent being either on blue or yellow.



(a) (b)

Figure 8: Ablation study of JIM in the coordinated placement task. (a) shows the training curves with the mean and standard deviation across 11 independent runs each, while (b) displays the performance of each independent run at the last iteration of training. The two ablated versions only feature one of the two exploration criteria defined in Section 4.2: JIM-EEC for N_{EEC} and JIM-LLEC for N_{LLEC}. The results show the importance of combining the two criteria.

putes one intrinsic reward for the whole group of agents. This makes JIM significantly more efficient to run, with LIM being approximately 24% slower than JIM to train (see Technical Appendix for details).

6.3 *Ablation study*

In this ablation study, JIM is compared with two ablated versions of the reward: one with only the episodic exploration criterion N_{EEC} (JIM-EEC) and one with only the life-long exploration criterion N_{LLEC} (JIM-LLEC). Note that JIM-LLEC is actually equivalent to NovelD [35] in this environment as the episodic restriction of NovelD (see Section 3.2) would be ineffective in a continuous environment such as MPE. To compare these three versions properly, we scale the intrinsic rewards of the two ablated models to be at a similar magnitude as the intrinsic reward generated by JIM. Figure 8 shows the results of training these versions in the coordinated placement task, with 11 independent runs each. Both ablated algorithms perform significantly worse than JIM. First, the episodic bonus of JIM-EEC alone lacks the motivation for discovering unseen configurations. Thus, it explores less and is not able to find the optimal solution to the task. Meanwhile, without the episodic restriction, JIM-LLEC develops less efficient exploration strategies. This confirms that, as shown in a recent study [1], the episodic restriction implemented in NovelD and other intrinsic rewards [20, 2] is crucial for developing efficient exploration strategies. Overall, this proves the importance of combining the two stages of exploration defined in N_{LLEC} and N_{EEC}.

7 Conclusion

In this paper, we present a simple yet effective algorithm for joint intrinsic motivation (JIM), which rewards teams of cooperating agents for exploring the space of joint observations. JIM can be integrated to enhance any Multi-Agent Deep Reinforcement Learning algorithm that uses centralized training with decentralized execution, and can be applied to problems with continuous state-action spaces. By combining JIM with the state-of-the-art QMIX algorithm, we demonstrate its efficiency. Our results show that QMIX with JIM outperforms both the original QMIX algorithm and QMIX with single-agent intrinsic rewards. We show that active exploration is a key component for multi-agent learning in environments with sparse rewards. Moreover, joint exploration enables the discovery of optimal coordinated behaviors that would be hard to find otherwise as they necessitate a high level of coordination between agents. Ultimately, JIM helps the agents overcome the problem of relative overgeneralization.

This shows the importance of using joint observations in the process of computing an intrinsic reward. In fact, this joint observation is the best estimate of the global state of the environment available for the agents. Using it allows more efficient learning of multi-agent joint behaviors, in addition to, as we have shown, making the algorithm simpler and thus less computationally expensive. In future works, it would be interesting to study how the joint observations could be used in other kinds of intrinsic rewards to shape the agents' behavior further.

References

[1] Alain Andres, Esther Villar-Rodriguez, and Javier Del Ser, 'An evaluation study of intrinsic motivation techniques applied to reinforcement learning over hard exploration environments', in *arXiv:2205.11184*, (2022).

[2] Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martin Arjovsky, Alexander Pritzel, Andrew Bolt, and Charles Blundell, 'Never give up: Learning directed exploration strategies', in *8th International Conference on Learning Representations*, (2020).

[3] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov, 'Exploration by random network distillation', in *7th International Conference on Learning Representations*, (2019).

[4] Yali Du, Lei Han, Meng Fang, Tianhong Dai, Ji Liu, and Dacheng Tao, 'LIIR: Learning individual intrinsic reward in multi-agent reinforcement learning', in *Advances in Neural Information Processing Systems*, volume 32, (2019).

[5] Yannis Flet-Berliac, Johan Ferret, Olivier Pietquin, Philippe Preux, and Matthieu Geist, 'Adversarially guided actor-critic', in *9th International Conference on Learning Representations*, (2021).

[6] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson, 'Counterfactual multi-agent policy

gradients', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, (2018).

[7] Mikael Henaff, Roberta Raileanu, Minqi Jiang, and Tim Rocktäschel, 'Exploration via elliptical episodic bonuses', in *Advances in Neural Information Processing Systems*, volume 35, pp. 37631–37646, (2022).

[8] Shariq Iqbal and Fei Sha, 'Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning', in *arXiv:1905.12127*, (2019).

[9] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A. Ortega, DJ Strouse, Joel Z. Leibo, and Nando de Freitas, 'Social influence as intrinsic motivation for multi-agent deep reinforcement learning', in *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3040–3049, (2019).

[10] Joel Lehman and Kenneth O Stanley, 'Abandoning objectives: Evolution through the search for novelty alone', in *Evolutionary computation*, volume 19, pp. 189–223, (2011).

[11] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch, 'Multi-agent actor-critic for mixed cooperative-competitive environments', in *Advances in Neural Information Processing Systems*, volume 30, (2017).

[12] Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster, 'Trajectory diversity for zero-shot coordination', in *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 7204–7213, (2021).

[13] Zixian Ma, Rose E Wang, Li Fei-Fei, Michael S. Bernstein, and Ranjay Krishna, 'ELIGN: Expectation alignment as a multi-agent intrinsic reward', in *Advances in Neural Information Processing Systems*, volume 35, pp. 8304–8317, (2022).

[14] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson, 'MAVEN: Multi-agent variational exploration', in *Advances in Neural Information Processing Systems*, volume 32, (2019).

[15] Igor Mordatch and Pieter Abbeel, 'Emergence of grounded compositional language in multi-agent populations', in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2018).

[16] Frans A Oliehoek and Christopher Amato, *A concise introduction to decentralized POMDPs*, Springer, 2016.

[17] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d. O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang, 'Dota 2 with large scale deep reinforcement learning', in *arXiv:1912.06680*, (2019).

[18] Pierre-Yves Oudeyer and Frederic Kaplan, 'What is intrinsic motivation? a typology of computational approaches', in *Frontiers in neurorobotics*, volume 1, p. 6, (2007).

[19] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell, 'Curiosity-driven exploration by self-supervised prediction', in *Proceedings of the 34th International Conference on Machine Learning*, volume PMLR 70, pp. 2778–2787, (2017).

[20] Roberta Raileanu and Tim Rocktäschel, 'RIDE: Rewarding impact-driven exploration for procedurally-generated environments', in *9th International Conference on Learning Representations*, (2020).

[21] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson, 'Weighted QMIX: Expanding monotonic value function factorisation for deep multiagent reinforcement learning', in *Advances in Neural Information Processing Systems 33*, pp. 10199–10210, (2020).

[22] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson, 'QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning', in *Proceedings of the 35th International Conference on Machine Learning*, volume PMLR 80, pp. 4295–4304, (2018).

[23] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver, 'Prioritized experience replay', in *4th International Conference on Learning Representations*, (2016).

[24] Jürgen Schmidhuber, 'A possibility for implementing curiosity and boredom in model-building neural controllers', in *Proceedings of the international conference on simulation of adaptive behavior: From animals to animats*, pp. 222–227, (1991).

[25] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua, 'Safe, multi-agent, reinforcement learning for autonomous driving', in *arXiv:1610.03295*, (2016).

[26] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi, 'QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning', in *Proceedings of the 36th International Conference on Machine Learning*, volume PMLR 97, pp. 5887–5896, (2019).

[27] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel, 'Value-decomposition networks for cooperative multi-agent learning based on team reward', in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, p. 2085–2087, (2018).

[28] Ming Tan, 'Multi-agent reinforcement learning: Independent versus cooperative agents', in *Proceedings of the Tenth International Conference on Machine Learning*, p. 330–337, (1993).

[29] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang, 'Influence-based multi-agent exploration', in *8th International Conference on Learning Representations*, (2020).

[30] Ermo Wei and Sean Luke, 'Lenient learning in independent-learner stochastic cooperative games', in *The Journal of Machine Learning Research*, volume 17, pp. 2914–2955, (2016).

[31] Ermo Wei, Drew Wicke, David Freelan, and Sean Luke, 'Multiagent soft q-learning', in *arXiv:1804.09817*, (2018).

[32] Rudolf Paul Wiegand, *An Analysis of Cooperative Coevolutionary Algorithms*, Ph.D. dissertation, George Mason University, 2003.

[33] Michael Wooldridge, *An introduction to multiagent systems*, John wiley & sons, 2009.

[34] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu, 'The surprising effectiveness of ppo in cooperative, multi-agent games', in *arXiv:2103.01955*, (2021).

[35] Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E Gonzalez, and Yuandong Tian, 'NovelD: A simple yet effective exploration criterion', in *Advances in Neural Information Processing Systems*, volume 34, pp. 25217–25230, (2021).