Intrinsically-Motivated Goal-Conditioned Reinforcement Learning in Multi-Agent Environments

Elías Masquil†
Flowers Team, Inria and Ensta ParisTech
Bordeaux, France
eliasmasquil@gmail.com

Eleni Nisioti†
Flowers Team, Inria and Ensta ParisTech
Bordeaux, France
eleni.nisioti@inria.fr

Gautier Hamon†
Flowers Team, Inria and Ensta ParisTech
Bordeaux, France
gautier.hamon@inria.fr

Clément Moulin-Frier
Flowers Team, Inria and Ensta ParisTech
Bordeaux, France
clement.moulin-frier@inria.fr

ABSTRACT
How can a population of reinforcement learning agents autonomously learn a diversity of cooperative tasks in a shared environment? In the single-agent paradigm, goal-conditioned policies have been combined with intrinsic motivation mechanisms to endow agents with the ability to master a wide diversity of autonomously discovered goals. Transferring this idea to cooperative multi-agent systems (MAS) entails a challenge: intrinsically motivated agents that sample goals independently focus on a shared cooperative goal with low probability, impairing their learning performance. In this work, we propose a new learning paradigm for modeling such settings, the Decentralized Intrinsically Motivated Skill Acquisition Problem (Dec-IMSAP), and employ it to solve cooperative navigation tasks. Agents in a Dec-IMSAP are trained in a fully decentralized way, which comes in contrast to previous contributions in multi-goal MAS that consider a centralized goal-selection mechanism. Our empirical analysis indicates that a sufficient condition for efficiently learning a diversity of cooperative tasks is to ensure that a group aligns its goals, i.e., the agents pursue the same cooperative goal and learn to coordinate their actions through specialization. We introduce the Goal-coordination game, a fully-decentralized emergent communication algorithm, where goal alignment emerges from the maximization of individual rewards in multi-goal cooperative environments and show that it is able to reach equal performance to a centralized training baseline that guarantees aligned goals. To our knowledge, this is the first contribution addressing the problem of intrinsically motivated multi-agent goal exploration in a decentralized training paradigm.

KEYWORDS
Multi-agent Learning, Goal-conditioned Learning, Intrinsic Motivation, Reinforcement Learning, Emergent Communication.

1 INTRODUCTION
Many multi-agent scenarios require the cooperation of autonomous agents with rich behavioral repertoires: in games such as StarCraft [34] and Capture the Flag [15], as well as real-world scenarios such as disaster robotics, the group is rewarded according to a shared objective, the achievement of which may require a well-orchestrated response, with each agent partially contributing to the global solution. Endowing a reinforcement learning (RL) agent with intrinsic motivation (IM) is a successful paradigm for acquiring such rich behavioral repertoires: by discovering which elements in its environment are controllable an agent can acquire a wide diversity of skills in an unsupervised manner [4, 8]. The framework of Intrinsically Motivated Goal-Exploration Processes (IMGEPs) [6, 11, 25] extends IM to learning with goal-conditioned policies so that an agent can autonomously sample its own goals and learn how to achieve them. Such agents are characterized as autotelic, from the Greek auto (self) and telos (goal) [30]. When attempting to transfer IMGEPs to a multi-agent scenario we stumble upon a challenge: how can a group of agents acting in a shared environment solve goals that require cooperation if each one of them is autonomously generating its own goals? Here, we argue that the lack of a centralized process for coordinating goal selection will impede the group’s ability to achieve a large diversity of cooperative goals. We introduce a new type of problem for multi-agent RL, the Decentralized Intrinsically Motivated Skill Acquisition Problem (Dec-IMSAP), to capture such settings, propose a communication-based algorithm for tackling it and evaluate it in a cooperative navigation task.

Autonomous skill discovery has been formalized for single-agent settings as the Intrinsically Motivated Skill Acquisition Problem [7]. To meaningfully extend it to multi-agent settings we need to consider environments that contain tasks requiring cooperation, such as the one presented in Figure 1 as an illustrative example. In this case, some of the goals set by agents will be cooperative, i.e. at least one other agent needs to perform some action for the agent to achieve its goal, while others will be independent, i.e., the agent will be able to solve them by itself. To study the Dec-IMSAP we propose a new training/evaluation paradigm: during the training phase agents are autonomously setting their own goals and learning to achieve them in a fully-decentralized manner, while, during the evaluation episodes, we externally provide agents with the same cooperative or individual goal, ensuring that a wide diversity of goals is tested across evaluation episodes.

Intrinsic motivation originated in the field of cognitive science, with psychology studies focusing on human infants due to their impressive ability to efficiently learn a wide diversity of skills [25].

We provide code for reproducing the experiments presented in the paper at https://anonymous.4open.science/r/Dec-IMSAP-README.md

†Work done during internship at the Flowers team
‡Co-first author
The mechanistic explanation points to exploratory play, during which infants generate their own goals and autonomously learn how to achieve them for the mere purpose of discovering new learning situations [3, 13]. While past studies primarily consider a single human subject, we today have empirical observations with infants engaging in cooperative play that point to the existence of a developmental path-way enabling humans to cooperate with others and plan alongside them [14, 28, 35]. According to theories of human social intelligence [33], an important factor for exploring cooperatively is shared intentionality: to solve goals that require cooperation agents need to simultaneously attend to them and know that they are doing so.

Does shared intentionality play an equally important role in groups of artificial agents and, if so, how can we guarantee it in a fully-decentralized training regime? This is the main research question we aim to address with our study of the Dec-IMSAP. We measure the degree of shared intentionality as goal alignment, a metric quantifying the percentage of training time during which two agents pursue the same cooperative goal. To address this question empirically, we first artificially control for the level of alignment and observe it is highly correlated with agent performance, both in terms of achieved reward and time required for success during evaluation episodes. Then, we devise the Goal-coordination game, a fully-decentralized learning algorithm inspired from the emergent communication literature that helps the group of agents coordinate its goals during training and show that, under it, alignment can emerge through a selfish objective maximizing agents’ own individual rewards and that the population reaches equal performance with a centralized setting that guarantees alignment.

The contributions of this paper are:

1. formulation of the Dec-IMSAP, a new type of problem for studying intrinsic motivation in multi-agent systems with goal-conditioned RL agents;
2. a detailed analysis on the impact of goal alignment between agents in the Dec-IMSAP;
3. an algorithm for solving the Dec-IMSAP, the Goal-coordination game, that enables agents in a group to maximize their skill repertoires by aligning their goals.

2 BACKGROUND

We now describe the components necessary to formulate the Dec-IMSAP. First, in Section 2.1, we describe the corresponding problem in single-agent settings as an evolution from classical RL to goal-conditioned and intrinsically-motivated agents and, then, discuss multi-goal Markov games as a generalization of MARL to goal-conditioned settings with externally provided goals.

2.1 Intrinsically motivated goal-conditioned reinforcement learning

In the RL paradigm an agent sequentially interacts with an environment with the objective of maximizing the rewards it accumulates. Its most common formalization is through the use of a Markov Decision Processes (MDP): at each time step \( t \) of an episode that lasts for \( T \) time steps the agent observes state \( s_t \), performs action \( a_t \) and receives reward \( r_t \). The policy \( \pi(a_t|s_t) \), which describes the agent’s behavior by mapping states to actions, is interactively learned from experience to maximize the cumulative reward \( G_s = \sum_{t=0}^{T} \gamma^t r_t \), where \( \gamma \) is a parameter quantifying how heavily future rewards are discounted [32]. Formally, an MDP is denoted as a tuple \( (S, \mathcal{A}, \mathcal{T}, \mathcal{P}_0, R) \), where the state space \( S \) and action space \( \mathcal{A} \) denote all possible configurations for the state and action respectively, \( \mathcal{T}(s_{t+1}|s_t, a_t) \) is the transition function that controls the distribution of the next state \( s_{t+1} \) from the current state \( s_t \) when the agent takes action \( a_t \), \( \mathcal{P}_0 \) is the distribution over the initial states, and, finally, the reward function \( R(s_t, a_t) \) describes the reward that an agent receives at each time step for a given state-action pair.

As there is a single reward function, MDPs cannot handle the case where an agent may need to learn multiple tasks. If we imagine a real-world problem setting, such as a cleaning robot in a household, then the action “turn on the oven” should be rewarded if the robot’s task is to warm up the food but penalized if the task is to make sure the tenant can safely leave for a weekend trip. To expand MDPs to suit a broader class of problems in which the agent may need to perform different tasks across episodes, the goal-conditioned RL paradigm introduces the notion of a goal and conditions the reward function on it. Formally, a goal \( g \) is a tuple \( (z_g, R_g) \), where \( z_g \) is a goal embedding, and \( R_g \) denotes the goal-conditioned reward function. We can, then, define a multi-goal MDP as a set of MDPs that share the same \( (S, \mathcal{A}, \mathcal{T}, \mathcal{P}_0) \) and differ only in the reward function \( R_g \). We denote the space of possible goals as \( \mathcal{G} \) and assign a different reward function to each goal so that the multi-goal MDP can be formally denoted as a tuple \( (S, \mathcal{A}, \mathcal{T}, \mathcal{P}_0, \mathcal{G}) \).

In goal-conditioned settings we can make a distinction between externally provided and intrinsically motivated goals [7]. The former require an external supervisor for choosing the goal, while the latter emphasize the agent’s ability to autonomously set its own
goals and aim at learning to solve a wide diversity of them. Intrinsic
motivation is a mechanism that arosed to explain how an agent can
acquire a repertoire of skills without any pre-defined goal, a setting
referred to as the Intrinsically Motivated Skill Acquisition Problem
[7]. RL with intrinsically-motivated goal-conditioned agents has
been previously modeled under the framework of Intrinsically Mo-
tivated Goal Exploration Processes (IMGEPs)[7], which, contrary
to the classical RL paradigm where a reward function is part of the
environment, incorporates the reward function \( R_g \) within the agent,
alongside with a goal-conditioned policy \( \pi_g \) and a mechanism for
sampling goals from the goal space, the goal-sampling function
\( D_G \). Each agent in Figure 2 can be seen as an IMGEP.

2.2 Goal-conditioned multi-agent reinforcement learning

In multi-agent RL, \( N \) agents interact in a shared environment,
so that one’s actions affect another’s state and rewards. Markov
Games are an extension of MDPs to multi-agent systems that cap-
ture such settings. Here, an agent is indexed by \( n \) and, at every
time step \( t \), the group’s behavior is captured by the joint action
\( \vec{a}_t = (a_{1,t}, \cdots, a_{N,t}) \). After the execution of the joint action,
the environment responds with the next state \( s_{t+1} \) and a local reward
for each agent \( r_{n,t} = R(s_t, a_t) \).

To model decentralized learning in a Markov Game we often
employ the framework of decentralized partially-observable MDPs
(Dec-POMDPs). Decentralization characterizes multi-agent systems
where agents choose their actions independently based on their
local information and partial observability refers to the fact that this
local information may not be sufficient to infer the environment’s
state, which now includes the other, physically-distanced agents.

To capture partial observability POMDPs introduce the notion of
an observation \( O_n \) which maps the environmental state to a local
observation for agent \( n \). Formally, a Dec-POMDP is modeled as a
tuple \((N, S, \{A_n\}, T, \mathcal{R}, \{O_n\})\), where \( N \) sets the set of agents
and \( A_n \) and \( O_n \) are the action and observation space of a single
agent.

Multi-goal Markov Games are an extension of Markov Games
to goal-conditioned settings. They formally arise when we replace
the reward function with one conditioned on goals, i.e. extending
the goal-conditioned RL paradigm mentioned above to a Dec-
POMDP by introducing a goal-conditioned reward function \( r_{n,t} = R(s_t, a_t, g_n) \). In multi-goal Markov Games goals are externally pro-
vided by a supervisor that ensures each agent is trained on a wide
diversity of goals.

3 AUTOTELIC AGENTS IN GOAL-CONDITIONED GAMES

3.1 Motivation

How can a learning framework model a group of agents whose
objective is to learn a diversity of cooperative goals? IMGEPs well
capture autonomous skill acquisition but do not consider inter-
actions between multiple agents. Multi-goal Markov Games, on
the other hand, model interactions of co-existing goal-conditioned
agents but do not account for the fact that autotelic agents are
self-generating their own goals.

We refer to our problem under consideration as the Decentralized
Intrinsically Motivated Skill Acquisition Problem (Dec-IMSAP) and
use it to formalize problem settings in which a group of agents is
set in an open-ended environment without pre-defined goals and
needs to acquire a repertoire of skills involving both individual
and cooperative abilities, the latter requiring the coordination of
at least two agents. Evaluation in the Dec-IMSAP has the form of
multiple episodes, at the beginning of which all agents are provided
with the same goal, sampled from a wide diversity of goals, and are
evaluated on their ability to reach it. In the following, we provide a
formal definition of the Dec-IMSAP.

3.2 Formalization

A Dec-IMSAP is modeled as a tuple \((N, S, \{O^g\}, \{A^g\}, T, \{R^g\},
\{D_n(G)\})\), where \( N \) is the set of \( N \) agents, \( S \) is the state space,
denoting all the possible configurations of all \( N \) agents and the
environment, \( O^g \) is the observation space, \( A^g \) is the space of actions,
\( T(s'|s, a) \) the transition function, \( R^g_{n} \) is the goal-conditioned reward
function and \( D(G) \) is the goal-sampling distribution of agent \( n \).

A Dec-IMSAP can be interpreted as an instance of a Dec-POMDP:
at the beginning of a training episode, each agent \( n \) will sample its
own goal \( g_n \in D_n(G) \), which defines an instance of a Dec-POMDP as
detailed in the previous section. In each training episode, each
agent executes its goal-conditioned policy \( \pi_g \) and will adjust its
behavior to maximize the episodic cumulative reward. These
group-conditioned policies are learnt in a fully-decentralized fashion using
group-conditioned RL. In this work, we assume that the goal space
\( G \) which contains all the possible goals \( g \), is fixed and identical
for all agents. Therefore, we are not concerned with learning goal
representations, which we assume to be already available.
To successfully solve the Dec-IMSAP the agents need to experience and learn how to solve a wide diversity of cooperative goals during training. Since both goal-selection and training are decentralized this is not guaranteed: if agents sample their goals independently then some cooperative goals may not be pursued enough times during training for the group to learn how to achieve them. In addition, the reward feedback is noisy: even if agents have learned optimal policies for all cooperative goals, they can obtain zero reward if their sampled goals are inconsistent, a case illustrated in Figure 1. How can the agents choose goals autonomously but still manage to coordinate them to achieve a wide diversity of cooperative skills? In the following, we present the Goal-coordination game, an algorithm for coordinating goals in a decentralized way.

### 3.3 Decentralized goal coordination

To coordinate goals in a fully decentralized way we need to find an architecture that will help the agents understand which goal the others are pursuing and adapt to this new information. This process should not introduce the need for centralization of information sharing nor a pre-existing agreement between the agents in the group and should be flexible enough to deal with any behavior arising during training. To achieve this we propose an algorithm inspired from the emergent communication literature, in particular the Naming Game, a learning algorithm originally introduced to help a population of agents invent a shared lexicon that associates forms to meaning, enabling communication within shared environments [31].

Our proposed algorithm takes place as a goal coordination round right before the start of each training episode. As is common in emergent communication setups, it employs two agents and can be extended to larger groups by considering a population of agents and randomly sampling a pair of them at each training episode. Each agent is equipped with a communication matrix $C_n : |G| \times |M| \rightarrow \mathbb{R}$, where $G$ is the goal space and $M$ is a message space, where we consider that both spaces are discrete (we discuss in Section 6 an extension to continuous spaces). Each row of matrix $C_n$ corresponds to a different goal $g$ of the agent and each column to a different message $m$. Its entries answer the question: “What is the expected reward of the episode if I transmit message $m$ when I have goal $g$?” and are all initialized with zeros. Communication is asymmetric: at the beginning of the goal coordination round one agent is randomly chosen to be the leader and the other the follower, therefore ensuring that each agent takes both roles across episodes. In what follows we employ underscore $l$ to denote properties of the leader and underscore $f$ to denote properties of the follower. The leader first samples a goal $g_l$ according to its own goal sampling strategy, $D_l(G)$, and then transmits the message $m_l$ that has the highest entry chosen by a softmax function for the corresponding row in matrix $C_l$. The follower receives message $m_f = m_l$ and chooses as its own goal $g_f$ as the row using a softmax function for the corresponding column in its matrix $C_f$. After playing several episodes every agent updates their matrices based on their respective rewards obtained at the end of each episodes. The matrix update is performed by updating the current values of the estimate of the expected reward for every goal/message association, by the average reward for that specific goal/message association computed on the batch of episodes collected. Note that during an episode, the leader and follower may be pursuing different goals, so although they experience the same episode their rewards may differ. To ensure that the matrix updates are not too quick for an agent to adapt its policy, we employ an exponential moving average update function with update rate $\alpha$. We refer to our algorithm as the Goal-coordination game and present its pseudocode in Algorithm 1, where first rollouts are collected for a batch of episodes (lines 6-14) and then processed to update the matrices of all agents in the population (lines 16-33). We also illustrate a single round in Figure 3.

An interesting property of the Goal-coordination game is that successful communication emerges from the only maximization of the individual reward, in a fully decentralized fashion (i.e. agents can receive different rewards according to their sampled goal and do not have access to the reward of each other). This is comes in contrast to the original naming game algorithm [31], where updates of the communication matrix are instead based on an explicit communication objective and where the outcome (communication success or failure) is the same for both agent. In the next section, we will see that the Goal-coordination game enable the emergence of shared intentionality in the agent population.

### 4 EMPIRICAL RESULTS

For simulating the Dec-IMSAP, we propose the Cooperative landmarks environment, that we instantiate using Simple Playgrounds [12]. This 2-D environment, illustrated in Figure 4, consists of a room with a number of $L$ landmarks on its walls and two agents that receive continuous-valued observations about the distance and angle to all landmarks and other agents and perform discrete-valued actions that control their angular velocity and longitudinal force. We consider navigation tasks in which agents need to go to different landmarks. To model such navigation tasks we define goals as vectors of dimension $L$ that indicate the desired landmark. Goals may have the form of one-hot or two-hot vectors, the former corresponding to individual goals and the latter to cooperative ones, i.e. $g = [x_1, \ldots, x_L], \sum_j [g]_j \in [1, 2]$ where $I_m = 1$ indicates that landmark $l$ needs to be reached by at least one agent. Thus, for a goal $g \in G$, the reward function at the end of an episode is positive, i.e $r_{g,a,T} = 1$, if the agents cover all landmarks ($\sum_j [g]_j = 1$) at the end of the episode. For example, if in the 3-landmarks environment the goal of an agent is $g = [1,10]$, then it receives a positive reward only...
Algorithm 1 Goal-coordination game

1. **Input:** Population: \( \mathcal{P} = \{n_1, \cdots, N\} \), matrix update rate \( \alpha \), message space size \( M \), goal space size \( G \), batch size \( B \).
2. \textbf{for} agent \( n \in N \) \textbf{do} \quad \triangleright \text{Initialize matrices}
3. \textbf{end for}
4. \textbf{while} not converged \textbf{do}
5. \textbf{for} rollout \( \in [1, \cdots, B] \) \textbf{do} \quad \triangleright \text{Collect a batch of episodes}
6. \textbf{end for}
7. \textbf{end while}

if one of them is touching the first landmark (red rectangle) and the other the second landmark (blue rectangle). We also introduce a scaling factor \( \beta \) for controlling the relative importance between independent and cooperative goals. In particular, we divide rewards for individual goals by \( \beta \), keep \( \beta = 2 \) in all results discussed in the main paper to incentivize all methods to focus on cooperative goals and study the effect of this hyper-parameter in Appendix 7.3.3. The episode finishes for an agent once it receives a reward or a time limit is reached. Then, the agent waits for the others to also complete their episode before a new one starts for the whole group. We experiment with environments where \( L = 3 \) and \( L = 6 \) and refer to them as the 3-landmarks environment, which entails three independent and three cooperative goals, and the 6-landmarks environment, which contains six individual and fifteen cooperative goals. We provide the complete list of goal encodings for the two settings in Appendix 7.1, alongside a formal definition of the observation and action space. In all simulations, we normalize rewards in the \([0, 1]\) by considering the maximum achievable reward for the centralized setting and, when applicable, present 95% confidence intervals based on 5 independent trials.

Each agent learns a goal-conditioned policy using PPO and has a uniform goal-sampling distribution \( \mathcal{D}(\mathcal{G}) \). We provide the values of all hyper-parameters characterizing an agent in Appendix 7.2. Our empirical investigation is structured as follows: a) in Section 4.1 we evaluate the role of goal alignment by designing baseline goal sampling strategies for different levels of alignment. For a given \( x\% \) desired level of alignment, each agent samples its own goal using \( \mathcal{D}(\mathcal{G}) \), but in \( x\% \) of the trials we interfere in the sampling procedure to force agents to sample the same goal. We evaluate 0%-aligned (also referred to as independent), 0%-aligned, 75%-aligned and 100%-aligned (also referred to as centralized); b) in Section 4.2 we evaluate the ability of the Goal-coordination game to reach the performance of the centralized baseline and provide insights into the co-adaptation dynamics of emergent alignment and evaluation performance.

In the following experiments we analyze a problem setting with \( L = 6 \) and defer results for \( L = 3 \) in Appendix 7.3.1 present the corresponding results for \( L = 3 \), which allows us to see that increasing the complexity of environment enables us to observer larger differences between methods. We also performed an additional robustness analysis to examine the effect of the hyper-parameters of our set-up: in Appendix 7.3.2 we study the effect of the number of message in the Goal-coordination game and observe that performance does not vary significantly with messages of intermediate length performing best, while in Appendix 7.3.3 we study the effect of the reward scaling factor \( \beta \) and observe that its value is correlated with the alignment achieved by Goal-coordination game during training but evaluation performance is insensitive to it. In addition in Appendix 7.3.4, we train and evaluate with cooperative goals only to totally remove the individual goals component and observe similar conclusions as in the current setup but with more visible gap between evaluation performances of the different methods.
4.1 The role of alignment

We have hypothesized that agents that do not coordinate their goals during training will find cooperative goals challenging. We now examine this hypothesis by comparing the performance during evaluation trials between groups of centralized, independent and 50%-aligned agents. We present the performance during evaluation and training episodes in Figure 5 for the 6-landmarks environment. In addition to the collected rewards we also monitor alignment during training trials and the length of the episode during evaluation trials, where shorter episodes indicate that the group solved the tasks quicker. By inspecting the evaluation results on the left of Figure 5 we observe that ensuring alignment during training, which in these experiments is obtained by the centralized sampling strategy, proves to have an impact in the performance during evaluation. Both in terms of rewards and speed of completing the tasks, agents who were trained using the centralized sampling strategy outperform the independent baselines and 50%-aligned baselines, with the latter achieving an intermediate level of performance. To understand why this behavior arises we turn to the training performance on the right of Figure 5: we observe that centralized agents get the most informative training signal, in which correct actions are always reinforced by the rewards they get. On the other hand, independent agents get the lowest training reward as they may receive 0 reward for every possible course of action they might take due to choosing different cooperative goals. This "noisy" rollout hinders the learning process of the policy.

We should note that alignment is a priori neither necessary nor sufficient for acting optimally in this environment. First, it is not necessary as, even with the independent baseline, the agents could denoise the training signal by observing others and inferring their goals and strategies, taking advantage of the randomly aligned episodes to solve cooperative tasks. As we see here, however, this is rather challenging: the independent baseline does not reach perfect performance and it requires longer episodes. Second, it is not sufficient as, even if both agents choose the same cooperative goal they still need to coordinate on who goes where. How can they do so with perfect success rate? We hypothesize that the agents will find it challenging to adapt to the other’s behavior due to the high level of partial observability in the environment: without a recurrent policy and without receiving an observation of the direction an agent is moving to, inferring the sub-goal pursued by the other is difficult. Instead, a specialization strategy where the two agents reach an a-priori agreement during training on who goes where (e.g. agent 0 always goes to the left-most landmark and agent 1 to the rightmost) requires less learning effort. To detect this behavior, we search for specialization, i.e., policies that, when assigned with a cooperative goal during evaluation, are biased to one of its landmarks. We define the specialization score for quantitatively measuring this behavior as the ratio of the episodes in which the agent went to its preferred landmark when following a cooperative policy and without receiving an observation of the direction an agent is moving to, inferring the sub-goal pursued by the other is difficult. Instead, a specialization strategy where the two agents reach an a-priori agreement during training on who goes where (e.g. agent 0 always goes to the left-most landmark and agent 1 to the rightmost) requires less learning effort. To detect this behavior, we search for specialization, i.e., policies that, when assigned with a cooperative goal during evaluation, are biased to one of its landmarks. We define the specialization score for quantitatively measuring this behavior as the ratio of the episodes in which the agent went to its preferred landmark when following a cooperative goal. For example, if for goal [101000] an agent went 7 times to [100000] and 3 times to [001000] this score would be 0.7. Measuring this across conditions in Figure 6 we observe that specialization is correlated to the level of alignment: the low specialization of the independent baseline explains why this method requires longer episodes.

4.2 Learning to align goals

We have established that alignment is an efficient strategy for solving a Dec-IMSAP in our proposed Collaborative landmark environment. However, enforcing goal alignment cannot be considered as a satisfactory solution: in a Dec-IMSAP, agents must learn in a fully decentralized fashion. To achieve this, we now turn to the evaluation of our proposed method for autonomously coordinating goals through communication, the Goal-coordination game. In Figure 5 we compare the performance of the Goal-coordination game to...
the other baselines and observe that it performs on par with the centralized baseline. We also observe that the Goal-coordination game does not reach the alignment of the centralized baseline: this is because in some cases the follower may adopt the risky behavior of choosing a cooperative goal when the leader communicates that it has chosen an individual goal. As the scaling factor of rewards $\beta$ equals two, cooperative goals are twice as rewarding as individual ones, incentivizing the follower to try this risky behavior. When it comes to specialization, the Goal-coordination game surpasses the centralized baseline. Although this difference is not statistically significant we hypothesize that it is due to the fact that, under the Goal-coordination game, the two agents experience a skewed distribution towards cooperative goals. This phenomenon becomes apparent when we examine different values for $\beta$ and observe that higher values lead to lower alignment during training albeit equally good evaluation performance (see Figure 11 in Appendix 7.3.3).

To get a clearer picture of the dynamics of the Goal-coordination game we visualize in Figure 7 the matrices for the simulation corresponding to Figure 5. In particular, we take a snapshot of the two matrices early in training ($t_{\text{train}} = 1000$), in the middle of training ($t_{\text{train}} = 15000$) and at the end of training ($t_{\text{train}} = 30000$).

Rows and columns of the matrices correspond to goals and the columns correspond to messages. We make the convention here of plotting the individual goals first, so, for the 6-landmarks environment studied in Figure 7, the first 6 rows correspond to individual and the following 15 to cooperative goals. We have set the message size to a slightly higher value than the number of goals, i.e., $M = 30$, which as we show in the analysis in Appendix 7.3.2 facilitates training by decreasing the probability that the matrix updating will get stuck. Rows and columns where we can find a single cell with higher intensity than others indicate a converged goal-message association while we can also detect alignment by tracing if the goal-message associations of the two agents agree. We observe that early in training the agents have low confidence for all but a couple of associations and alignment has not been achieved. By the middle of training, however, the two matrices are almost identical, an indication of high alignment. Note that this only applies to cooperative goals: alignment for individual goals is not necessary and therefore confidence in the values of the first 6 rows is low. At the end of training, the two agents have perfectly aligned their matrices.

A challenging feature of the Goal-coordination game is that the matrices and policies are updated simultaneously. This can lead to a chicken-and-egg problem: the matrix updates may fail even if the goal-message association is correct because the policy has not managed to solve a goal, leading to uninformative rewards, or the policy may struggle to solve goals because of bad goal-message associations that lead to episodes unfeasible to solve (such as one agent having goal $[100001]$ and the other $[010001]$). In Figure 8, we observe that the co-evolution of alignment and rewards during training for a random subset of the goals, we observe that our introduction of the matrix update rate $\alpha$, with value $0.1$, helps avoid this problem: rewards and alignment are highly correlated for cooperative goals, where improvements in one drive improvements in the other. Instead we observed that too high values for $\alpha (> 0.5)$ lead to failure of the Goal-coordination game. We also observe that for individual goals, rewards are maximized without requiring alignment as we expect.

5 RELATED WORKS

In goal-conditioned MARL, goals are commonly provided extrinsically in a supervised process and training is centralized. Under
Figure 8: Co-evolution of alignment and rewards during training using the Goal-coordination game.

these conditions, it has been shown that a compositional communication protocol can emerge for solving collective navigation tasks [21]. In another work, a multi-agent curriculum combined with a mechanism for credit assignment assignment was proven useful [36]. These works further differ from ours as they consider a multi-goal setting, i.e. each agent is assigned with a sub-goal while during evaluation we provide a group-level goal and allow agents to self-organize to an assignment to sub-goals.

Intrinsic motivation in MARL has been shown to facilitate cooperation in groups of single-goal RL agents [8, 16]. Here, rewards encouraging social influence [16] and rewards that disentangle an individual’s contributions from group behavior [8] can lead to behaviors on par with algorithms requiring centralized training [10, 18, 27]. In contrast, our proposed algorithm, Goal-coordination game, does not require intrinsic rewards and considers multiple goals.

Although some contributions have studied the interaction between autotelic agents and social partners, the latter usually take the role of a teacher. For example, the SGIM architecture studied how an IMGEP agent can autonomously decide how to interact with a teacher agent [23, 24]. The IMAGINE architecture [6] couples an language-based IMGEP agent with a social partner that provides linguistic descriptions of trajectories. The agent is then able to re-interpret these descriptions as linguistic goals and to imagine new goals by leveraging the compositionality of language. Our work differs from these approaches as we consider a population of agents with no prior knowledge and skills who jointly learn how to achieve a diversity of goals in the absence of a teacher.

6 DISCUSSION

In this work we presented a new problem for formulating intrinsically-motivated multi-agent goal exploration in a decentralized training paradigm, Dec-IMSAP, and proposed an algorithm for solving it, the Goal-coordination game. We empirically observed that shared intentionality, which we measured as alignment of cooperative goals during training, plays an important role in the ability of groups to solve a wide diversity of cooperative tasks. Aligned agents do not only get the highest rewards but also find the quickest policies for solving tasks. We also showed that, under the Goal-coordination game, shared intentionality emerges without being explicitly rewarded and groups reach equal performance to a centralized setting with perfect alignment.

To understand how alignment acquires such an important role in our setting, we inspected the policies learned by groups controlled for different levels of it and observed that groups with higher alignment solve the tasks by specializing: instead of monitoring and adapting to others, which, as has been observed in previous works in MAS is a behavior challenging to emerge unless explicitly rewarded [22], agents using the Goal-coordination game prefer to align their goals and coordinate through specialization. Importantly, the Goal-coordination game does not optimize for alignment, which rather emerges as the agents maximize their individual rewards.

Our study of the Goal-coordination game has been limited to populations of two agents and discrete message and goal spaces. Extending it to groups with more than two agents is important for scaling up its applicability. We hypothesize that in such settings specialization will no longer lead to optimal performance and that the goal-conditioned policy will need to be extended by conditioning it on messages and introducing recurrency to equip agents with memory [9]. To extend the Goal-coordination game to continuous message and goal spaces, we can adopt approaches based on energy-based models employed in previous works [19].

The results we presented in this work are consistent between different experiments, and across two instantiations of a cooperative navigation task with different number of landmarks. An interesting extension of this work would be to test our approach in a more complex, multi-agent environment like Grafter [2]. Another interesting research direction would be to study how the Goal-coordination game behaves when combined with other goal-sampling strategies besides uniform sampling, such as learning progress [5]. In addition, we could consider combining this algorithm with other IMGEP modules (e.g learning the goal representations, goal conditioned rewards, or the support of the goal space) [7].

We believe that our proposed problem, the Dec-IMSAP, can potentially result in novel real-world applications for MARL. It allows to consider a population of goal-conditioned RL agents that will learn how to achieve a wide diversity of collaborative tasks in a fully autonomous manner. This way, a user could place agents (simulated or robotics) in some environment and let them interact without any supervision for a period of time. At the end of this training phase, the agent population will have autonomously learned how to achieve diverse individual and collaborative goals without any
supervision and a human user will be able to benefit from these acquired skills. We believe this can be of interest in real-world scenarios such as robotics for disaster rescue or extraterrestrial exploration.

ACKNOWLEDGMENTS

This research was partially funded by the Inria Exploratory action ORIGINS (https://www.inria.fr/en/origins) as well as the French National Research Agency (https://anr.fr/). Project ECOCURL, Grant ANR-20-CE23-0006. This work also benefited from access to the HPC resources of IDRIS under the allocation 2020-A0091011996 made by GENCI.

REFERENCES

[1] Marc Andrychowicz, Anton Raichuk, Piotr Stachniak, Manu Orsini, Sertan Girgin, Raphael Marinier, Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, et al. 2020. What matters for on-policy deep actor-critic methods? A large-scale study. In International conference on learning representations.

[2] Chris Bamford. 2022. Grafter. https://github.com/GriddlyAI/grafter.

[3] Daniel E Berlyne. 1966. Curiosity and Exploration: Animals spend much of their time seeking stimuli whose significance raises problems for psychology. Science 151, 3731 (1966), 25–33.

[4] Nuttapong Chentanez, Andrew Barto, and Satinder Singh. 2004. Intrinsically Motivated Reinforcement Learning. In Advances in Neural Information Processing Systems 17 (2004), 1159–1199.

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

[6] Yon Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. 2016. Fast reinforcement learning via slow reinforcement learning. arXiv preprint arXiv:1611.02779 (2016).

[7] Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip HS Torr, Pushmeet Kohli, and Shimon Whiteson. 2017. Stabilizing experience replay for deep multi-agent reinforcement learning. In International conference on machine learning, PMLR, 1146–1155.

[8] Sébastien Forestier, Rémy Portelas, Yoan Mollard, and Pierre-Yves Oudeyer. 2022. CURIOS: Intrinsically Motivated Modal Multi-Goal Reinforcement Learning. (2018). https://doi.org/10.48550/ARXIV.1810.06284

[9] Marc Girgin, Raphaël Marinier, Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, et al. 2020. What matters for on-policy deep actor-critic methods? A large-scale study. In International conference on learning representations.

[10] Cédric Colas, Tristan Karch, Nicolas Laur, Jean-Michel Dussoux, Clément Moulin-Frier, Peter Dominey, and Pierre-Yves Oudeyer. 2020. Language as a cognitive tool to imagine goals in curiosity driven exploration. Advances in Neural Information Processing Systems 33 (2020), 3761–3774.

[11] Igor Mordatch and Pieter Abbeel. 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In International conference on machine learning, PMLR, 4295–4304.

[12] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. 2019. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In International conference on machine learning, PMLR, 3040–3049.

[13] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic opti-
misation. arXiv preprint arXiv:1412.6980 (2014).

[14] Marc Lanctot, Vinicius Zambaldi, Audranus Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. 2017. A unified game-theoretic approach to multiagent reinforcement learning. Advances in neural information processing systems 30 (2017).

[15] Yoann Lemede, Tristan Karch, Romain Laroche, Clément Moulin-Frier, and Pierre-Yves Oudeyer. 2022. Emergence of Shared Sensory-motor Graphical Language from Visual Input. https://doi.org/10.48550/ARXIV.2210.06468

[16] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Jon Stoica. 2018. RLLib: Abstractions for Distributed Reinforcement Learning. In Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80), Jennifer Dy and Andreas Krause (Eds.). PMLR, 3053–3062. https://proceedings.mlr.press/v80/liang18b.html

[17] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. 2019. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In International conference on machine learning, PMLR, 3040–3049.

[18] Marc Lanctot, Vinicius Zambaldi, Audranus Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. 2017. A unified game-theoretic approach to multiagent reinforcement learning. Advances in neural information processing systems 30 (2017).

[19] Yoann Lemede, Tristan Karch, Romain Laroche, Clément Moulin-Frier, and Pierre-Yves Oudeyer. 2022. Emergence of Shared Sensory-motor Graphical Language from Visual Input. https://doi.org/10.48550/ARXIV.2210.06468

[20] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Jon Stoica. 2018. RLLib: Abstractions for Distributed Reinforcement Learning. In Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80), Jennifer Dy and Andreas Krause (Eds.). PMLR, 3053–3062. https://proceedings.mlr.press/v80/liang18b.html

[21] Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32.

[22] Kamal Ndoussé, Douglas Eck, Sergey Levine, and Natasha Jaques. 2021. Emergent Social Learning via Multi-Agent Reinforcement Learning. In ICML.

[23] Sao Mai Nguyen and Pierre-Yves Oudeyer. 2012. Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner. Paladyn 3, 3 (2012), 136–146.

[24] Sao Mai Nguyen and Pierre-Yves Oudeyer. 2012. Socially guided intrinsic motivation for robot learning, of motor skills. Autonomous Robots 36, 3 (2014), 273–294.

[25] Pierre-Yves Oudeyer and Frederic Kaplan. 2009. What is intrinsic motivation? A typology of computational approaches. Frontiers in neurobotics (2009), 6.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[27] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In International conference on machine learning, PMLR, 4295–4304.

[28] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438 (2015).

[29] Luc Steels. 2004. The autotelic principle. In Embodied artificial intelligence. Springer, 231–242.

[30] Luc L. Steels. 2015. The Talking Heads experiment. Number 1 in Computational Models of Language Evolution. Language Science Press, Berlin. https://doi.org/10.17169/FUDOCs_document_000000022455

[31] Richard S Sutton and Andrew G Barto. 2018. Reinforcement learning: An introduction, MIT press.

[32] Michael Tomasello and Malinda Carpenter. 2007. Shared intentionality. Developmental science 10, 1 (2007), 121–125.

[33] Oriol Vinyals, Timo Epplers, Sergey Bartunov, Petko Georgiev, Alexander Sashka Verzhevatsev, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Sig Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy Lillicrap, Kevin Crowley, Paul Keet, Anthony Brunosso, David Lawrence, Anders Ekerom, Jacob Repp, and Rodney Tenng. 2017. StarCraft II: A New Challenge for Reinforcement Learning. https://doi.org/10.48550/ARXIV.1708.04782

[34] Felix Warneken, Jasmin Steinwender, Katharina Hamann, and Michael Tomasello. 2014. Young children’s planning in a collaborative problem-solving task. Cognitive Development 31 (2014), 48–58.

[35] Jiachen Yang, Alireza Nakhaei, David Ielee, Kikuo Fujimura, and Hongyun Zha. 2018. Cm3: Cooperative multi-goal multi-stage multi-agent reinforcement learning. arXiv preprint arXiv:1809.05188 (2018).
7 APPENDIX

7.1 Environment details

The environment is implemented in Python using Simple Playgrounds [12]. As a learning algorithm for the goal-conditioned policies we use RLlib’s [20] PPO implementation and its multi-agent API with the PyTorch backend [26].

For the 3-landmarks environment the set of individual goals is \{[001], [010], [100]\} and the set of cooperative goals is \{[101], [011], [110]\}. For the 6-landmarks environment the set of individual goals is \{[000001], [000000], [000010], [000100], [010000], [100000]\} and the set of cooperative goals is \{[110000], [101000], [100100], [100010], [100001], [0110000], [010100], [010010], [010001], [001100], [000110], [000011], [100001]\}.

Observation space. Agents are able to see each other and all the objects of the environment. We use object-centric representations, the observation vector contains the distance and the angle to each of the physical entities in the room (i.e walls, other agent, and landmarks). The order of the coordinates in the observation vector is preserved, e.g the first two coordinates are the distance and angle to the left wall. To make the navigation policy a goal-conditioned one, we concatenate the goal representation at the end of the observation vector to build the input to the networks. Observations are normalized between 0 and 1.

Action space. We consider a discrete action space. Each agent is controlled by two actions: longitudinal force, and angular velocity. These actuators can take three different values: -1, 0, or 1.

Rewards and episodes. Rewards are given independently to each agent conditioned on the agent’s own goal. At each time step, if the goal is not fulfilled, the reward is 0, and 1 otherwise. All interactions with the environment are fully decentralized, each agent only has access to its own reward, and cannot see the reward of the others.

Once an agent gets a positive reward, the episode ends for them, i.e they cannot perform any other action but remain physically present in the environment. Episodes end either when both agents obtained their rewards or if a time limit is reached. At the beginning of an episode each agent is randomly placed inside the room, without touching any of the landmarks. The time limit in the environment was set to 250 and 500 time steps, for the 3 and 6 landmark instances respectively.

7.2 Hyperparameters

Both the design choices and hyperparameters are the same for all the goal exploration processes we tried (independent agents, centralized sampling, communication for goal alignment, etc).

PPO. We base most of our design choices in the recommendations from [1]. We use:

- PPO policy loss with 0.3 clipping threshold.
- tanh as activation function for the networks. We don’t use shared layers for the value and policy networks.
- Generalized Advantage Estimation (GAE) [29] with \(\lambda = 0.9\)
- A discount factor of \(\gamma = 0.99\)
- Adam optimizer [17] with a learning rate of 0.0003

From the many hyperparameters we can tune, we found that the batch size was the most relevant. After some test experiments, benchmarking results with the centralized uniform sampling baseline, we set this value to 16500 and 60000 time steps for the 3 and 6 landmarks experiments. We observed that usually a higher batch size was beneficial. For most of the hyperparameters we found that the defaults provided by RLlib were safe choices.

Goal-coordination game. We use a softmax of temperature \(T = \frac{1}{30}\) to sample messages \(m_i\) and goal \(g_i\) from the matrix. The update of the matrix is made with \(\alpha = 0.1\) to dampen the changes of estimates of expected reward for each goal/message couple.

7.3 Additional results

7.3.1 3-landmarks environment. Figure 9 contains the evaluation performance, on the left, and training performance, on the right for the 3-landmarks environment. We observe that, compared to the 6-landmarks environment, the population requires significantly less training time (about one order of magnitude smaller) and that differences across methods during evaluation are not as pronounced.

During training, we observe that alignment is correlated with performance with the independent baselines collecting the least rewards. Thus, we conclude that our empirical conclusions generalize to simpler problem settings and that studying problems with increased task complexity is important for evaluating methods on the Dec-IMSAP.

7.3.2 Effect of message size. In Figure 10 we study the effect of message size on the Goal-coordination game in the 6-landmarks environment by setting it to the smallest possible value (\(M = 21\) is equal to the number of goals), a medium value (\(M = 30\)) and a high value (\(M = 40\)). We observe that evaluation performance does not vary significantly with message size except for the fact that small message size leads to slower convergence to the optimal policy. During training, we observer that small message size cannot reach perfect alignment and amasses slightly lower rewards. Thus, we conclude that the message size should be said to a value relatively higher than the number of goals but no further benefits are gained when it increases beyond that.

7.3.3 Effect of scaling factor \(\beta\). As we described in Section 4 the scaling factor \(\beta\) controls the relative importance of individual versus cooperative goals: increasing the value of \(\beta\) indicates a proportional decrease in the importance of solving independent goals. To examine the effect of \(\beta\) we present the performance of the Goal-coordination game for different values (\(\beta \in \{1, 2, 4, 8\}\)) in Figure 11 and compare across methods for \(\beta = 4\) in Figure 12 to contrast with the setting with \(\beta = 2\) studied in the main paper. We observe that, for the Goal-coordination game, higher value of \(\beta\) lead to lower alignment: as cooperative goals are very rewarding in this case agents with the role of follower prefer them over individual ones even when the leader communicates about a cooperative goal. At the same time, low values of \(\beta\) lead to slower convergence to the optimal solution, as agents with the role of follower are not incentivized enough to choose cooperative goals, as they still receive rewards when they choose individual goals regardless of the leader’s follower. Thus, there is an intermediate value of \(\beta = 2\)
that performs best. Finally, contrasting the behavior of the Goal-coordination game to the other baseline in Figure 12 shows that by increasing $\beta$ the Goal-coordination game can amass more rewards during training that the centralized baseline. This is not surprising: as the agents learn this risky behavior or aligning cooperative with individual goals, they experience more rewarding episodes.

7.3.4 Cooperative goals only. In this experiment, individual goals are removed both in training and evaluation. Meaning that the leader can only sample cooperative goals and the follower can only choose cooperative goals from its matrix. We observe the same conclusion as in the all goals experiments but with bigger gap between methods both for the 3 landmarks and the 6 landmarks cases. Also in this setup we see that the Goal-coordination game converges to 100% alignment during training and converges to the same performances as the 100% alignment method both in term of reward and episode length.
Figure 11: Effect of scaling factor $\beta$ on the Goal-coordination game in the 6-landmarks environment during evaluation (left) and training (right) episodes.

Figure 12: Comparing different methods for $\beta = 4$ during evaluation (left) and training (right) episodes.
Figure 13: Performance for the 3-landmarks environment with only cooperative goals during evaluation (left) and training (right) episodes for baselines exhibiting different levels of alignment and the Goal-coordination game.

Figure 14: Performance for the 6-landmarks environment with only cooperative goals during evaluation (left) and training (right) episodes for baselines exhibiting different levels of alignment and the Goal-coordination game.