ELIGN: Expectation Alignment as a Multi-Agent Intrinsic Reward

Zixian Ma[1], **Rose E. Wang**[1], **Li Fei-Fei**[1], **Michael Bernstein**[1], **Ranjay Krishna**[1,2]
Stanford University[1], University of Washington[2]
{zixianma,rewang,feifeili,msb,ranjaykrishna}@cs.stanford.edu

Abstract

Modern multi-agent reinforcement learning frameworks rely on centralized training and reward shaping to perform well. However, centralized training and dense rewards are not readily available in the real world. Current multi-agent algorithms struggle to learn in the alternative setup of decentralized training or sparse rewards. To address these issues, we propose a self-supervised intrinsic reward *ELIGN - expectation alignment* - inspired by the self-organization principle in Zoology. Similar to how animals collaborate in a decentralized manner with those in their vicinity, agents trained with expectation alignment learn behaviors that match their neighbors' expectations. This allows the agents to learn collaborative behaviors without any external reward or centralized training. We demonstrate the efficacy of our approach across 6 tasks in the multi-agent particle and the complex Google Research football environments, comparing ELIGN to sparse and curiosity-based intrinsic rewards. When the number of agents increases, ELIGN scales well in all multi-agent tasks except for one where agents have different capabilities. We show that agent coordination improves through expectation alignment because agents learn to divide tasks amongst themselves, break coordination symmetries, and confuse adversaries. These results identify tasks where expectation alignment is a more useful strategy than curiosity-driven exploration for multi-agent coordination, enabling agents to do zero-shot coordination.

1 Introduction

Many real world AI applications can be formulated as multi-agent systems, including autonomous vehicles (Cao et al., 2012), resource management (Ying & Dayong, 2005), traffic control (Sunehag et al., 2017), robot swarms (Swamy et al., 2020), and multi-player video games (Berner et al., 2019). Agents must adapt their behaviors to each other in order to coordinate successfully in these systems. However, adaptive coordination algorithms are challenging to develop because each agent is not privy to other agents' intentions and their future behaviors (Foerster et al., 2017).

These challenges are more acute in decentralized training under partial observability than centralized training or full observability. In the real world, agents act under partial observability and learn in a decentralized manner: they do not learn collaborative behaviors with a single centralized algorithm with a complete knowledge of the environment (Iqbal & Sha, 2019; Liu et al., 2020). Unfortunately, the most successful multi-agent algorithms train agents with a centralized critic, assuming access to all agents' observations and actions (Foerster et al., 2018; Rashid et al., 2018; Sunehag et al., 2017; Lowe et al., 2017). The most successful multi-agent algorithms for decentralized training and partial observability assume task-specific reward shaping (Jain et al., 2020; Iqbal & Sha, 2019), which is expensive to generate. These algorithms struggle to learn with sparse reward structure.

Consider a cooperative navigation task, where N agents aim to simultaneously occupy N goal locations. A centralized algorithm with full observability is capable of optimally assigning the nearest

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

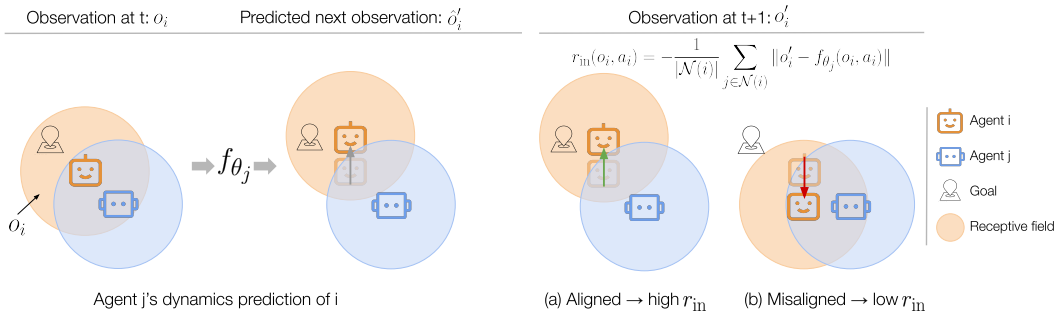

Figure 1: We introduce ELIGN, i.e, expectation alignment, a task-agnostic intrinsic reward to improve multi-agent systems. Intuitively, ELIGN encourages agents to become more predictable to their neighbors. An agent (e.g., agent i here) learns to behave in ways that match its neighbors' (e.g., agent j's) predictions of its next observation. Here, agent j expects agent i to move up instead of down, moving closer to a point of interest above it. Agent i attains (a) a higher reward when its action (e.g., upward) aligns with this expectation or (b) a lower reward when its action (e.g., downward) is misaligned.

goal location to each respective agent. However, with partial observability, agents can see only a handful of goal locations and other agents. They are unaware of others' observations, actions, and intentions with decentralized training. We observe that agents simultaneously occupy the same goal; they fail to collaborate because they do not predict which goal each agent is expected to occupy. To overcome instances of miscoordination, decentralized algorithms have adapted single-agent curiosity-based intrinsic rewards (Pathak et al., 2017; Stadie et al., 2015). Multi-agent curiosity-based rewards incentivize agents to explore novel states (Iqbal & Sha, 2020). Although curiosity helps agents discover new goal locations, it doesn't solve the challenge of coordination, such as assigning goals to each agent. Only a few attempts explore other forms of multi-agent intrinsic rewards (Iqbal & Sha, 2020; Böhmer et al., 2019; Schafer, 2019).

In this work, we propose ELIGN as a novel multi-agent self-supervised intrinsic reward, enabling decentralized training under partial observability. Intuitively, expectation alignment encourages agents to elicit behaviors that decrease future uncertainty for their team: it encourages each agent to choose actions that match their teammates' expectations. Going back to the cooperative navigation task, expectation alignment encourages each agent to move to goals others expect it to occupy, like goals that are either closest to the agent or goals that other agents aren't moving towards (Figure 1). We take inspiration from the self-organization principle in Zoology (Couzin, 2007). This principle hypothesizes that collective animal intelligence emerges because groups synchronize their behaviors using only their local environment; they do not rely on complete information about other agents and can coordinate successfully by predicting the dynamics of agents within their field-of-view (Collett et al., 1998; Theraulaz & Bonabeau, 1995; Ben-Jacob et al., 1994; Buhl et al., 2006). Similarly, expectation alignment as an intrinsic reward is calculated based on the agent's local observations and its approximation of neighboring agents' expectations. It does not require a centralized controller nor full observability. ELIGN is task-agnostic and we apply it to both collaborative and competitive multi-agent tasks.

We demonstrate the efficacy of our approach in the multi-agent particle and Google Research football environments, two popular benchmarks for multi-agent reinforcement learning (Lowe et al., 2017; Kurach et al., 2019). We evaluate ELIGN under partial and full observability, with decentralized and centralized training, and in terms of scalability. We observe that expectation alignment outperforms sparse and curiosity-based intrinsic rewards (Ndousse et al., 2021; Stadie et al., 2015; Iqbal & Sha, 2020), especially under partial observability with decentralized training. We additionally test expectation alignment as a way to perform zero-shot coordination with new agent partners, and investigate why ELIGN improves coordination. We show that agent coordination improves through expectation alignment because agents learn to divide tasks amongst themselves and break coordination symmetries (Hu et al., 2020).

2 Related Work

Our formulation of expectation alignment, a task-agnostic intrinsic reward for multi-agent training, draws inspiration from the self-organization principle in Zoology, which posits that synchronized group behavior is mediated by local behavioral rules (Couzin, 2007) and not by a centralized controller (Camazine et al., 2020). Group cohesion emerges by predicting and adjusting one's behavior to that of near neighbors (Buhl et al., 2006). This principle underlies the coordination found in multi-cellular organisms (Camazine et al., 2020), the migration of wingless locusts (Collett et al., 1998), the collective swarms of bacteria (Ben-Jacob et al., 1994), the construction of bridge structures by ants (Theraulaz & Bonabeau, 1995), and some human navigation behaviors (Couzin, 2007).

Intrinsic motivation for single agents. Although we draw inspiration from Zoology for formalizing expectation alignment as an intrinsic reward, there is a rich body of work on intrinsic rewards within the single-agent reinforcement learning community. To incentivize exploration, even when non-optimal successful trajectories are uncovered first, scholars have argued for the use of intrinsic motivation (Schmidhuber, 1991). Single-agent intrinsic motivation has focused on exploring previously unencountered states (Pathak et al., 2017; Burda et al., 2018a), which works particularly well in discrete domains. In continuous domains, identifying unseen states requires keeping track of an intractable number of visited states; instead, literature has recommended learning a forward dynamics model to predict future states and identify novel states using the uncertainty of this model (Achiam & Sastry, 2017). Other formulations encourage re-visiting states where the dynamics model's prediction of future states errs (Stadie et al., 2015; Pathak et al., 2017). Follow up papers have improved how uncertainty (Kim et al., 2020) and model errors (Burda et al., 2018b; Sekar et al., 2020) are calculated.

Intrinsic motivation for multiple agents. Most multi-agent intrinsic rewards have been adapted from single-agent curiosity-based incentives (**?**Böhmer et al., 2019; Schafer, 2019) and have primarily focused on cooperative tasks. They propose intrinsic rewards to improve either coordination, collaboration, or deception: These rewards either maximize information conveyed by an agent's actions (**?**Chitnis et al., 2020; Wang et al., 2019), shape the influence of an agent (Jaques et al., 2019; Foerster et al., 2017), incentivize agents to hide intentions (Strouse et al., 2018), build accurate models of other agents' policies (Hernandez-Leal et al., 2019; Jaques et al., 2019), or break extrinsic rewards for better credit assignment (Du et al., 2019).

Several multi-agent intrinsic rewards (Hernandez-Leal et al., 2019; Jaques et al., 2019), including ours, rely on the ability to model others' dynamics in a shared environment. This ability is a key component to coordination, closely related to Theory of Mind (Tomasello et al., 2005). Our work can be interpreted as using a Theory of Mind model of others' behaviors to calculate an intrinsic motivation loss. Unlike existing Theory of Mind methods that learn a model per collaborator (Roy et al., 2020), we learn a single dynamics model, allowing our method to scale as the number of agents increase. Our proposal is related to model-based reinforcement learning (Jaderberg et al., 2016; Wang et al., 2020a); however, instead of learning a dynamics model for control, we learn a dynamics model as a source of reward. Our work is closely related to a recently proposed auxiliary loss on predicting an agent's own future states (Ndousse et al., 2021). However, there are three key differences. First, their work predicts ego-agent observations, whereas our work additionally predicts future observations from the other agents' point of view. Second, their loss optimizes state embeddings while ours optimizes agents' policies. Third, their work focuses on cooperative tasks whereas ours applies to both cooperative and competitive domains.

Multi-agent reinforcement learning algorithms. Today, the predominant deep multi-agent framework uses actor-critic methods with a centralized critic and decentralized execution (Lowe et al., 2017; Foerster et al., 2018; Iqbal & Sha, 2019; Liu et al., 2020; Rashid et al., 2018). This framework allows a critic to access the observations and actions of all agents to ease training. However, there are several situations where centralized training may not be desirable or possible. Examples include low bandwidth communication restrictions or human-robot tasks where observations cannot be easily shared between agents (Ying & Dayong, 2005; Cao et al., 2012; Huang et al., 2015). Decentralized training is therefore the most practical training paradigm but it suffers from unstable training: the environment is nonstationary from a single-agent's perspective (Lowe et al., 2017). Our work uses a decentralized training framework and tackles the nonstationarity challenge with an intrinsic reward designed to improve an agent's ability to model others. We also apply expectation alignment to centralized training and observe that it still aids cooperative and some competitive tasks.

3 Background

We formulate our setting as a partially observable Markov game $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, r_{\text{ex}}, N)$ (Littman, 1994). A Markov game for N agents is defined by a state space $\mathcal{S}$ describing the possible configurations of the environment. The observation space for agents is $\mathcal{O} = (\mathcal{O}_1, \ldots, \mathcal{O}_N)$ and the action space is $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_N)$. Each agent i observes $\mathbf{o}_i \in \mathcal{O}_i$, a private partial view of the state, and performs actions $a_i \in \mathcal{A}_i$. Using the observation, each agent uses a stochastic policy $\pi_{\theta_i} : \mathcal{O}_i \times \mathcal{A}_i \to [0, 1]$, where θ_i parameterizes the policy. The environment changes according to the state transition function which transitions to the next state using the current state and each agent's actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$. The team of agents obtains a shared extrinsic reward as a function of the environment state, $r_{\text{ex}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The team's goal is to maximize the total expected return: $R = \sum_{t=0}^{T} \gamma^t r_{\text{ex}}^t$ where $0 \leq \gamma \leq 1$ is the discount factor, t is the time step, and T is the time horizon. The environment may also contain adversarial agents who have their own reward structure.

4 Expectation Alignment

To understand expectation alignment intuitively, let's revisit the cooperative navigation task, where N agents are rewarded for simultaneously occupying as many goal locations as possible. In Figure 1, agent i has a dynamics model trained on its past experiences. It predicts how future states will evolve from the point of view of agent j, who is within i's view. In this example, j will expect i to move towards the goal since i is closer to it. ELIGN encourages i to pursue the action that j expects (Figure 1(a)). In turn, j can now assume that the observed goal location will eventually be occupied by i and should therefore explore to find another goal. By aligning shared expectations, agent behaviors become more predictable. Conversely, when neighbors behave opposite to an agent's predictions, the agent can infer about the environment outside of its own receptive field (Krause et al., 2002). For example, in Figure 1 (b), if agent j observes i running away from a goal, this surprising behavior might indicate the existence of an adversary outside j's receptive field.

Our training algorithm consists of three interwoven phases of learning a dynamics model, calculating the ELIGN reward, and optimizing the agent's policy (Algorithm 1).

4.1 Training the dynamics model

Similar to prior work (Wang et al., 2018; Kidambi et al., 2020), each agent i learns a dynamics model f_{θ_i} to predict the next observation $\hat{o}'_i$ given its current observation and action o_i, a_i, i.e,

$$\hat{o}'_i = f_{\theta_i}(o_i, a_i).$$

We use a three-layer Multi-Layer Perceptron with ReLU non-linearities as the dynamics model. We minimize the mean squared error between its prediction and ground truth next observation o'_i.

4.2 Calculating intrinsic reward

The intrinsic reward captures how well agent i aligns to its neighbors' (e.g., agent j's) expectations on its next state. Calculating this reward requires j to accurately predict i's behavior, simulating a Theory of Mind (Tomasello et al., 2005). As suggested by the self-organization principle, i must learn to align to j's predictions. Ideally, the ELIGN intrinsic reward is calculated as:

$$r_{\text{in}}(o_i, a_i) = -\frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \|o'_i - f_{\theta_j}(o_i, a_i)\|$$

where $\mathcal{N}(i)$ is the set of neighbors within i's receptive field, including i itself. The ELIGN reward is high when the average L_2 loss is small, i.e, when i's actual next observation is close to agent j's predicted observation of i for all j in its neighbors. In that case, i has chosen an action that aligns with j's expectations of how i should act.

In a decentralized training setup, however, i doesn't have access to j's dynamics model f_{θ_j}, so i approximates j's dynamic model with a proxy: its own dynamics model f_{θ_i} and the knowledge of agent j's observation radius. Such an approximation is ecologically valid since we often approximate others' behaviors using a second-order cognitive Theory of Mind (Morin, 2006). Additionally, i

doesn't have access to j's entire observation; so, we restrict the future prediction from j's point of view by using the portion of j's observation i can see: $o_{i \cap j} = o_i \odot o_j$. Agent i's decentralized intrinsic reward then becomes:

$$r_{\text{in}}(o_i, a_i) = -\frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \|o'_{i \cap j} - f_{\theta_i}(o_{i \cap j}, a_i)\|$$

We found that the approximation of f_{θ_j} using f_{θ_i} works well empirically. Dynamics model losses for all agents quickly decrease within 5-10 training epochs. we validate its applicability in small-scale heterogeneous multi-agent tasks where agents have variable capabilities, although we find the methods perform similarly when more heterogeneous agents are added.

4.3 Policy learning

Once the ELIGN rewards are calculated, the total rewards at each step for each agent i is: $r_i = r_{\text{ex}} + \beta r_{\text{in}}(o_i, a_i)$ where r_{ex} is the extrinsic reward provided by the environment and β is a hyperparameter for weighing the intrinsic reward in the agent's overall reward calculation. In practice, we set β to be $\frac{1}{|\mathcal{O}_i|}$ where $|\mathcal{O}_i|$ is the observation dimension; we find this scale generalizes well across tasks. Since our contribution is agnostic to any particular multi-agent training algorithm, the team of agents can now be trained using any multi-agent training algorithm to maximize returns $R = \sum_{t=0}^{T} \gamma^t r$.

Both centralized and decentralized training algorithms can make use of these rewards. We primarily use the multi-agent decentralized variant of the soft-actor critic algorithm in our experiments (Haarnoja et al., 2018; Iqbal & Sha, 2019). Compared to centralized joint-action training, whose action space grows exponentially in N agents, our decentralized method has linear space complexity. Further, decentralized training can parallelize training time to be less than linear with respect to N. Although we present results with one centralized training framework, studying the impact of expectation alignment with all the centralized critic frameworks is out of scope for this paper.

Algorithm 1 ELIGN: Expectation Alignment

```
Initialize replay buffer D and D'
Initialize N agents with random θ_i: i ∈ [1, N]
while not converged do
    for b = 1 ... B do
        Populate buffer D with episode using policies (π_θ1, ..., π_θN)
    end for
    // TRAIN DYNAMICS MODEL
    for agent i = 1 ... N do
        Sample transitions: {(o_i, a_i, r_ex, o'_i)} ~ D_i
        Predict ô'_i = f_θi(o_i, a_i)
        Update dynamics θ_i using o'_i.
    end for
    // CALCULATE ELIGN REWARD
    for agent i = 1 ... N do
        Sample B transitions: {(o_i, a_i, r_ex, o'_i)} ~ D_i
        Compute intrinsic rewards r_in(o_i, a_i)
        Add {(o_i, a_i, r_ex + β r_in, o'_i)} to D'_i
    end for
    // POLICY LEARNING
    Update all θ_i s using transitions from D'
end while
```

4.4 Extending expectation alignment to competitive tasks

We extend the ELIGN formulation to competitive tasks where a team of agents compete against adversaries. In this case, agents are encouraged to *misalign* with their adversaries' expectations, i.e, agents are incentivized to be unpredictable to their adversaries within its receptive field ($\mathcal{N}_{\text{adv}}(i)$):

$$r_{\text{in}} = \frac{1}{|\mathcal{N}_{\text{adv}}(i)|} \sum_{k \in \mathcal{N}_{\text{adv}}(i)} \|o'_{i \cap k} - f_{\theta_i}(o_{i \cap k}, a_i))\|$$

5 Experiments

Our experiments explore the utility of using expectation alignment as an intrinsic reward compared to sparse and curiosity-based intrinsic rewards. We primarily focus on decentralized training under partial observability. However, we also demonstrate that ELIGN can easily augment centralized methods and assist in fully observable tasks. We vary the number of agents in the multi-agent particle tasks to test scalability. We end by investigating how and why ELIGN improves coordination by designing three evaluation conditions. First, does expectation alignment improve coordination by

Table 1: We report the mean test episode extrinsic rewards and standard errors of *decentralized training under partial observability* in multi-agent particle and Google Research football environments. $\text{ELIGN}_{\text{self/team}}$ outperform SPARSE and both curiosity-based intrinsic rewards. $\text{ELIGN}_{\text{adv}}$ achieves the best performance among all competitive tasks except for *Physical deception*, where $\text{ELIGN}_{\text{team}}$ is the best. These results demonstrate the benefit of using alignment as intrinsic reward to train better decentralized policies under partial observability.

	Cooperative		**Competitive**			
Task (Agt# v Adv#)	**Coop nav. (3v0)**	**Hetero nav. (4v0)**	**Phy decep. (2v1)**	**Pred-prey (2v2)**	**Keep-away (2v2)**	**3v1 w/ keeper (3v2)**
SPARSE[1]	139.07 ± 13.63	284.42 ± 12.83	93.60 ± 8.61	-4.72 ± 2.4	4.58 ± 3.27	0.020 ± 0.001
$\text{CURIO}^2_{\text{self}}$	133.93 ± 7.66	286.22 ± 9.97	68.80 ± 7.93	-6.50 ± 2.18	11.88 ± 2.88	0.024 ± 0.004
$\text{CURIO}^3_{\text{team}}$	125.42 ± 11.95	262.28 ± 22.59	85.31 ± 11.93	-3.57 ± 1.75	9.54 ± 5.04	0.021 ± 0.002
$\text{ELIGN}_{\text{self}}$	$\mathbf{155.88 \pm 5.11}$	292.34 ± 9.24	69.91 ± 4.51	-7.58 ± 2.55	12.84 ± 4.29	0.003 ± 0.018
$\text{ELIGN}_{\text{team}}$	141.04 ± 8.04	$\mathbf{311.67 \pm 10.88}$	$\mathbf{101.72 \pm 6.31}$	-7.69 ± 2.69	2.96 ± 4.03	0.022 ± 0.001
$\text{ELIGN}_{\text{adv}}$	—	—	92.20 ± 4.23	$\mathbf{-2.51 \pm 1.70}$	$\mathbf{19.46 \pm 5.05}$	$\mathbf{0.025 \pm 0.001}$
HAND-CRAFTED[1]	75.56 ± 18.90	228.48 ± 18.88	94.25 ± 14.75	$\mathbf{-0.77 \pm 0.17}$	$\mathbf{52.14 \pm 3.11}$	–

[1] Lowe et al. (2017); Kurach et al. (2019),[2] Stadie et al. (2015),[3] Iqbal & Sha (2020)

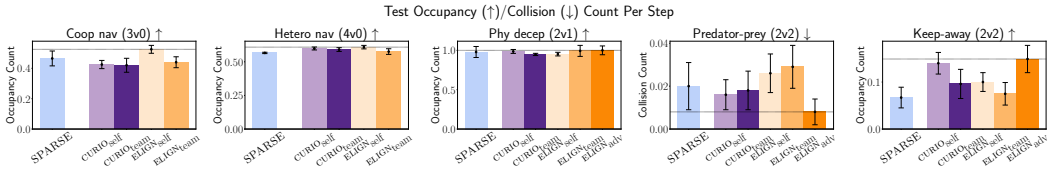

Figure 2: We plot the average test occupancy/collision count per step of decentralized algorithms under partially observable multi-agent particle tasks. On these metrics, $\text{ELIGN}_{\text{self}}$ and $\text{ELIGN}_{\text{adv}}$ perform the best on cooperative and competitive tasks respectively.

breaking symmetries (Hu et al., 2020; Wang et al., 2020b)? Second, does ELIGN enable zero-shot generalization to new partners? Lastly, is the dynamics model critical in aligning agent behaviors?

5.1 Environments

We evaluate ELIGN across both cooperative and competitive tasks in the multi-agent particle environment (Mordatch & Abbeel, 2017; Lowe et al., 2017) and the Google Research football environment (Kurach et al., 2019).

State and action space The multi-agent particle environment is a two-dimensional world. The Google Research football environment is a three-dimensional world. Both environments have continuous state spaces and discrete action spaces. Particle agents observe all agents' positions and velocities. They can "stay" or change their velocity in one of the four cardinal directions. Each football agent controls one player. Players observe the ball, other players' positions and directions. They can apply one of ten actions from "top_left", "top", "top_right", "right", "bottom_right", "bottom", "bottom_left", "sprint", and "dribble".

Observability The original environments assume full observability, where each agent can observe the position $p = (x, y)$ and velocity $v = (\Delta x, \Delta y)$ of all agents; each agent's observation vector is thus $\mathbf{o}_{i,\text{full}} = [p_1, \ldots, p_N, v_1, \ldots, v_N]$. We extend these environments to be partially observable, where agent i observes only the portion within its receptive field; like prior work with partial observability (Corder et al., 2019), we hide the position and velocity information of any agent j outside of agent i's receptive field; i.e, if the Euclidean distance between agent i and j surpasses a vicinity threshold τ, then p_j and v_j are 0 in $\mathbf{o}_{i,\text{partial}}$. We set $\tau = 0.5$ for partially observable and ∞ in the original fully observable case, where the world's width and height are 2.0 in the multi-agent particle environment and $0.84 : 2.00$ in the Google Research football environment. Both environments also contain objects such as obstacles, goals and a ball; they are similarly hidden if out of sight. Partial observability is a more ecologically valid training condition since most agents in real-world tasks can only observe a small portion of their environment at a given time.

5.2 Tasks

Multi-agent particle environment We use the following tasks from the multi-agent particle environment (Lowe et al., 2017; Liu et al., 2020). We choose N based on prior work.

Table 2: We report the mean test episode extrinsic rewards and standard errors as the number of agents is increased and trained using of *decentralized* algorithms. When the number of agents increases, one of ELIGN still performs the best in all tasks except for *Heterogenous navigation*.

	Cooperative		Competitive		
Task (Agt # vs. Adv #)	**Coop nav. (5v0)**	**Hetero nav. (6v0)**	**Phy decep. (4v2)**	**Pred-prey (4v4)**	**Keep-away (4v4)**
SPARSE[1]	459.92 ± 22.44	616.62 ± 25.30	166.89 ± 27.72	-28.75 ± 7.3	0.75 ± 1.82
CURIO$^2_{\text{self}}$	458.45 ± 19.79	$\mathbf{702.73 \pm 18.57}$	146.55 ± 29.05	-25.35 ± 6.16	10.52 ± 5.48
CURIO$^3_{\text{team}}$	497.15 ± 11.47	695.38 ± 12.22	84.66 ± 16.94	-17.21 ± 8.23	1.40 ± 2.06
ELIGN$_{\text{self}}$	$\mathbf{498.24 \pm 9.77}$	646.70 ± 23.25	137.38 ± 30.00	$\mathbf{-9.14 \pm 5.57}$	9.83 ± 11.22
ELIGN$_{\text{team}}$	488.83 ± 20.82	638.74 ± 28.93	$\mathbf{186.83 \pm 21.92}$	-20.4 ± 5.93	2.07 ± 4.55
ELIGN$_{\text{adv}}$	—	—	182.61 ± 17.63	-21.37 ± 7.02	$\mathbf{11.29 \pm 9.02}$

[1] Lowe et al. (2017),[2] Stadie et al. (2015),[3] Iqbal & Sha (2020)

Cooperative navigation: N agents must cooperate to reach a set of N goal locations. Agents are collectively rewarded based on the occupancy of any agent on any goal location.

Heterogeneous navigation: N agents must reach N goals but they differ in speeds and sizes. $\frac{N}{2}$ agents are slow and big, and the other $\frac{N}{2}$ agents are fast and small.

Physical deception: N agents cooperate to reach a single goal location and are rewarded if any one occupies the goal. However, they are penalized when any of $\frac{N}{2}$ adversaries occupies the goal and gets rewarded. The adversaries do not know which landmark is the goal and must infer it from the agents' behavior. The agents should learn to deceive the adversaries by covering all the landmarks.

Keep-away: There are N landmarks, one of which is the goal and known to N agents. Agents are rewarded for occupying it and preventing M adversaries from reaching it. Adversaries are rewarded for pushing the agents away from the goal, but they can only infer which landmark is the goal.

Predator-prey: N slow adversaries chase and capture N fast cooperating agents around a randomly generated obstacle-filled environment. Each time an adversary catches an agent, the agent is penalized and the adversary is rewarded.

Google Research football We use the *Academy 3vs1 with Keeper* competitive task from the Google Research football environment (Kurach et al., 2019). In this task, three agents try to score from the edge of the penalty box, one on each side, and the other at the center. This task is initialized with the centered agent having the ball and facing the defender. There is an adversary who plays the keeper.

5.3 Training and evaluation

We train all algorithms with 5 random seeds. All the hyperparameters used in the training can be found in the Appendix. For the Multi-agent particle environment, each experiment uses one Tesla K40 GPU to train until convergence, i.e. the best evaluation episode reward hasn't changed for 100 epochs. Each epoch equates to $4K$ episodes of 25 timesteps. We evaluate the algorithms by running $1K$ test episodes of 25 timesteps and mainly report the mean average test episode reward and standard error across the random seeds. We also evaluate on task-specific metrics, including agent-goal occupancy/agent-adversary collision count, and agent-goal/agent-adversary distance. For Google Research football, each experiment trains for $5M$ timesteps. We evaluate on and report the mean average episode rewards and the standard errors across the seeds.

5.4 Baselines

All algorithms are trained using the same agent architecture and optimization algorithm. They vary in task-specific extrinsic rewards and intrinsic rewards. We use two versions of the soft actor-critic algorithm Haarnoja et al. (2018): a decentralized one that trains each agent individually without access to other agents' observations and actions (ie. the original soft-actor critic algorithm) and a centralized one with a critic that has access to other agents' observations and actions (Iqbal & Sha, 2019). Note, our intrinsic reward can also be added to non-actor-critic methods, such as COMA (Foerster et al., 2018) and VDN (Sunehag et al., 2017). We leave this to future work to avoid conflating the effects of expectation alignment with COMA's counterfactual reasoning and VDN's value decomposition.

We use SPARSE (Lowe et al., 2017; Kurach et al., 2019), CURIO$_{\text{self}}$ (Stadie et al., 2015), CURIO$_{\text{team}}$ (Iqbal & Sha, 2020), and variations of our ELIGN rewards. SPARSE rewards agents

only when they reach a goal state. CURIO$_{team}$ is a curiosity-based multi-agent intrinsic reward which maximizes the average L_2 loss (instead of minimizing it in ELIGN). It rewards agents for exploring novel states (Iqbal & Sha, 2020). CURIO$_{self}$ also maximizes the L_2 loss but only using agent i's own observation (Stadie et al., 2015). We experiment with three variants of ELIGN: ELIGN$_{self}$, incentivizing alignment to one's own expectation ELIGN$_{team}$ incentivizing agents to align to their team, and ELIGN$_{adv}$ incentivizing misalignment to adversaries' expectations. Note that ELIGN$_{self}$ is similar to the auxiliary loss in Ndousse et al. (2021) but we use it for policy optimization, rather than for training a state encoder. We also add hand-crafted dense rewards to provide oracle performance for all tasks.

5.5 Results in partially observable environments with decentralized training

ELIGN outperforms baselines across cooperative and competitive tasks in the multi-agent particle environment. Table 1 demonstrates that both ELIGN$_{self}$ and ELIGN$_{team}$ outperform all SPARSE and CURIO$_{self/team}$ baseline rewards in cooperative tasks. While not all ELIGN variants surpass the baselines in competitive tasks, we find that ELIGN$_{team}$ achieves the highest reward in *Physical deception*, and ELIGN$_{adv}$ performs the best in *Predator-prey* and *Keep-away*. Similarly, Figure 2 shows that ELIGN$_{self}$ achieves the highest per-step occupancy count in both cooperative tasks, and ELIGN$_{adv}$ does the best in all competitive tasks.

ELIGN outperforms baselines in the complex Google Research football environment. As shown in Table 1, ELIGN$_{adv}$ achieves the best mean average episode reward in the competitive *Academy 3vs1 with keeper* task. Collectively, these results provide empirical evidence that the self-organizing principle improves coordination under partial information, a setting that is most realistic to real world multi-agent systems.

In competitive tasks, agents benefit more from being misaligned to adversaries than being aligned to their team members. Among the four competitive tasks in the multi-agent particle and football environments, we find that ELIGN$_{adv}$ outperforms all SPARSE and CURIO$_{self/team}$ baselines and other variants of ELIGN in *Predator-prey*, *Keep-away* and *Academy 3vs1 with keeper*. This suggests that being misaligned to adversaries, ie taking surprising actions that conflict with the adversary's expectations, might be a more useful strategy in multi-agent competitive tasks.

When the number of agents increases, ELIGN scales well in all multi-agent particle tasks except for *Heterogenous navigation*. Table 2 shows that our ELIGN intrinsic reward still largely achieves the best performance when more agents are added to cooperative and competitive tasks. The only exception is the *Heterogenous navigation* task, where both ELIGN$_{self}$ and ELIGN$_{team}$ outperform SPARSE but not CURIO$_{self/team}$. We hypothesize that it is more difficult for agents to predict their neighbors' future states accurately when there are more agents with different sizes and speeds, and errors in dynamics prediction could lead to misleading alignment signals. Further, we see a consistent increase in ELIGN$_{team}$'s performance compared against SPARSE even when the number of agents scales to ten in *Cooperative navigation* (Figure 3).

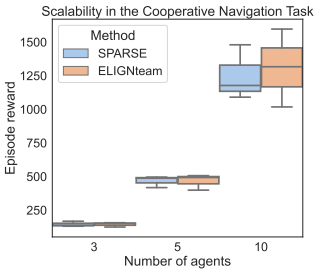

Figure 3: ELIGN$_{team}$ achieves consistent gains compared against SPARSE when the number of agents increases in the *Cooperative navigation* task.

5.6 Results with full observability and centralized ELIGN

We further test the utility of decentralized ELIGN in fully observable environments and centralized ELIGN under partial observability. We find that decentralized expectation alignment helps in fully observable *Cooperative navigation*, *Heterogenous navigation*, and *Predator-prey*, tasks where expectation alignment has been observed in nature. Similarly, centralized ELIGN also improves agents' performance compared against SPARSE and CURIO$_{self/team}$ rewards in the same tasks with partial observability. These results can be found in Tables 3 and 4 in the Appendix. As full observability and centralized training are our main focus, we leave it to future work to investigate why expectation alignment benefits these tasks but not others.

5.7 Investigating how the ELIGN reward helps

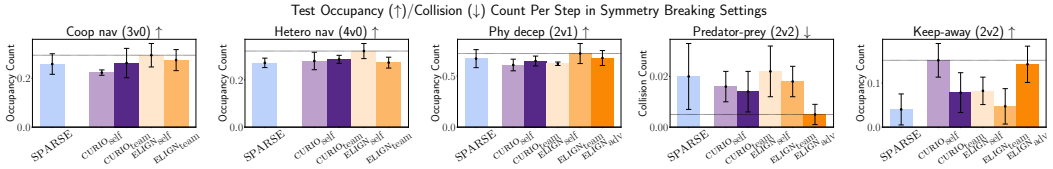

Figure 5: We plot the test occupancy/collision count per step of decentralized algorithms in symmetry-breaking settings under partial observability. ELIGN$_{self}$ performs the best in both cooperative tasks. ELIGN$_{team}$ and ELIGN$_{adv}$ are the best strategies in *Physical deception* and *Predator-prey*.

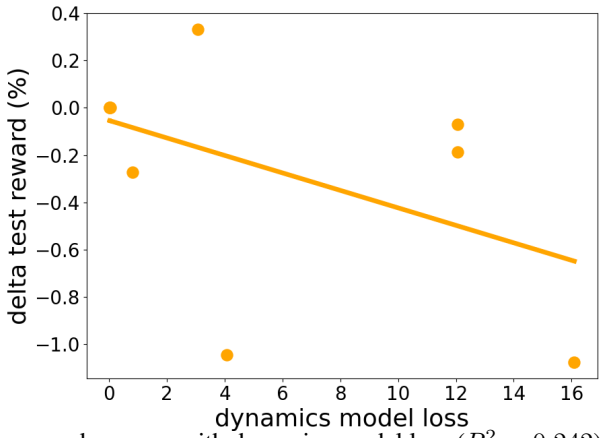

Figure 6: Test performance decreases with dynamics model loss ($R^2 = 0.242$), implying that ELIGN requires an accurate dynamics model.

We further investigate how expectation alignment improves coordination through three evaluation setups.

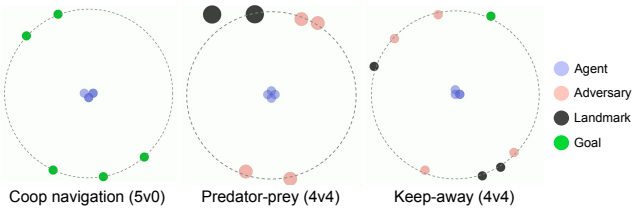

Figure 4: We visualize the symmetry-breaking setups in three example tasks. More details can be found in the Appendix.

ELIGN helps agents divide sub-tasks. A core challenge in multi-agent collaboration is efficient task division (Wang et al., 2020b). Here, we test whether expectation alignment improves sub-task allocation. We initialize agents in states without an optimal sub-task allocation, necessitating symmetry-breaking (Hu et al., 2020). Figure 4 illustrates the symmetry-breaking setups: in cooperative navigation, when agents are initialized equidistant to all the goal locations, there isn't an optimal allocation of agents to goals. We find that ELIGN$_{self}$ achieves the best performance in both cooperative tasks, while ELIGN$_{team}$ and ELIGN$_{adv}$ are the best strategies in *Physical deception* and *Predator-prey* respectively (Figure 5). Upon a qualitative evaluation of *Cooperative navigation*, we observe that agents with expectation alignment are able to predict which goals will be covered by their collaborators and move towards their allocated one. Without expectation alignment, agents move towards the same goal.

ELIGN helps agents generalize to new partners. Another core challenge in multi-agent collaboration is zero-shot coordination, where agents are tested to collaborate with new partners they haven't been trained with. We study whether expectation alignment enables better zero-shot coordination. New partners are sampled from other training runs with different seeds and the team is evaluated using the same metrics as before. We observe that ELIGN strategies enable better performance than SPARSE on average, and one of ELIGN$_{self,team,adv}$ performs the best in *Heterogenous navigation*, *Physical deception* and *Keep away*. (Figure 7). These results suggest that ELIGN results in better zero-shot coordination with new partners sampled from separate training runs.

Accuracy of the dynamics model affects ELIGN. We investigate the accuracy of the dynamics model in calculating useful intrinsic rewards. Since ELIGN uses a dynamics model to calculate

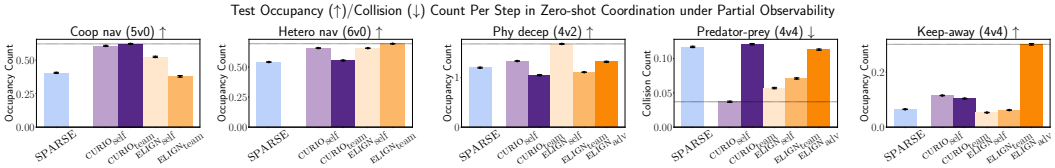

Figure 7: We plot the test occupancy/collision count per step of decentralized algorithms in zero-shot evaluation with new partners. We find that one of ELIGN$_{\text{self,team,adv}}$ achieves the best performance in *Heterogenous navigation*, *Physical deception* and *Keep away*.

rewards, we test whether an inaccurate model misleads agents towards unaligned behaviors. We train agents on noisy dynamics models by adding Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma)$ to the predictions made by the dynamics model. We run experiments with multiple σ values to study how performance changes as the dynamics model becomes more noisy: $\sigma \in [0.5, 1.0, 2.0]$. Our experiments cover one cooperative and one competitive task. Figure 6 plots the *final* dynamics loss against the reward change from the noiseless run. As the dynamics model degrades, we observe that the task performance also drops. This study identifies the importance of an accurate dynamics model, suggesting that expectation alignment should used in environments where an accurate dynamics model can be learned.

6 Discussion

Limitations and future work. While curiosity has proven useful for exploration in single-agent tasks, we find that expectation alignment—which mathematically encourages agents to be more predictable instead of finding novelty—outperforms curiosity in numerous multi-agent tasks. We hypothesize that our results arise because today's multi-agent task state space requires significantly less exploration than those used for single-agent (e.g. Atari games). Our findings are limited to the multi-agent particle and Google Research football environments, which have a smaller action space than most ecologically valid scenarios.

Language, motion, and human gesture are all combinatorially vast; in such action spaces, expectation alignment might develop social dynamics that hinder non-optimal multi-agent behaviors. Similarly, photorealistic environments have a larger state space, where teams perform common household activities (e.g., cooking, cleaning, etc.) or drive together in crowded cities (Srivastava et al., 2021). Future work should develop new multi-agent environments that demand exploration complexity and where both curiosity and expectation alignment would be necessary for collaboration. For example, in a search and rescue task where a single agent is unable to carry the injured, curiosity would encourage "search" while ELIGN would speed up "rescue". In the end, we envision that both these forms of rewards would be necessary for successful collaboration. However, choosing when to encourage curiosity versus expectation alignment is an open research problem.

Additionally, enabling stable multi-agent training without centralized training could open up future opportunities for legible (Dragan et al., 2013) agents in human environments. Agents with interpretable actions can induce more faithful human mental models, improving human-AI interaction; however, predictability does not imply legibility. Future work could explore the role of legibility in designing intrinsic rewards.

Future work should also explore the use of expectation alignment in massive collaboration settings with hundreds of agents. Drawing on Zoology research Couzin (2007) expectation alignment should scale in such settings if agents align their behaviors only to their nearest neighbors and not to the entire cohort.

Conclusion. Inspired by the self-organizing principle in Zoology, we introduce ELIGN, i.e, expectation alignment, a simple, task-agnostic, and self-supervised intrinsic reward for multi-agent systems. ELIGN rewards agents when they act predictably to their teammates and unpredictably to their adversaries. ELIGN improves multi-agent performance across six cooperative and competitive tasks in the multi-agent particle and Google Research football environments, especially for decentralized training under partial observability. It also scales well, helps agents break symmetries, and generalize to new partners.

References

Achiam, J. and Sastry, S. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

Ben-Jacob, E., Schochet, O., Tenenbaum, A., Cohen, I., Czirok, A., and Vicsek, T. Generic modelling of cooperative growth patterns in bacterial colonies. *Nature*, 368(6466):46–49, 1994.

Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Böhmer, W., Rashid, T., and Whiteson, S. Exploration with unreliable intrinsic reward in multi-agent reinforcement learning. *arXiv preprint arXiv:1906.02138*, 2019.

Buhl, J., Sumpter, D. J., Couzin, I. D., Hale, J. J., Despland, E., Miller, E. R., and Simpson, S. J. From disorder to order in marching locusts. *Science*, 312(5778):1402–1406, 2006.

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

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018b.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraula, G., and Bonabeau, E. *Self-organization in biological systems*. Princeton university press, 2020.

Cao, Y., Yu, W., Ren, W., and Chen, G. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1):427–438, 2012.

Chitnis, R., Tulsiani, S., Gupta, S., and Gupta, A. Intrinsic motivation for encouraging synergistic behavior. *arXiv preprint arXiv:2002.05189*, 2020.

Collett, M., Despland, E., Simpson, S. J., and Krakauer, D. C. Spatial scales of desert locust gregarization. *Proceedings of the National Academy of Sciences*, 95(22):13052–13055, 1998.

Corder, K., Vindiola, M. M., and Decker, K. Decentralized multi-agent actor-critic with generative inference. *arXiv preprint arXiv:1910.03058*, 2019.

Couzin, I. Collective minds. *Nature*, 445(7129):715–715, 2007.

Dragan, A. D., Lee, K. C., and Srinivasa, S. S. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 301–308. IEEE, 2013.

Du, Y., Han, L., Fang, M., Dai, T., Liu, J., and Tao, D. Liir: learning individual intrinsic reward in multi-agent reinforcement learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 4403–4414, 2019.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Foerster, J. N., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 2018.

Hernandez-Leal, P., Kartal, B., and Taylor, M. E. Agent modeling as auxiliary task for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pp. 31–37, 2019.

Hu, H., Lerer, A., Peysakhovich, A., and Foerster, J. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.

Huang, C.-M., Cakmak, M., and Mutlu, B. Adaptive coordination strategies for human-robot handovers. In *Robotics: science and systems*, volume 11. Rome, Italy, 2015.

Iqbal, S. and Sha, F. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 2961–2970. PMLR, 2019.

Iqbal, S. and Sha, F. Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning, 2020. URL `https://openreview.net/forum?id=rkltE0VKwH`.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Jain, U., Weihs, L., Kolve, E., Farhadi, A., Lazebnik, S., Kembhavi, A., and Schwing, A. A cordial sync: Going beyond marginal policies for multi-agent embodied tasks. In *European Conference on Computer Vision*, pp. 471–490. Springer, 2020.

Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P., Strouse, D., Leibo, J. Z., and De Freitas, N. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pp. 3040–3049. PMLR, 2019.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*, 2020.

Kim, K., Sano, M., De Freitas, J., Haber, N., and Yamins, D. Active world model learning with progress curiosity. In *International conference on machine learning*, pp. 5306–5315. PMLR, 2020.

Krause, J., Ruxton, G. D., Ruxton, G., Ruxton, I. G., et al. *Living in groups*. Oxford University Press, 2002.

Kurach, K., Raichuk, A., Stanczyk, P., Zajac, M., Bachem, O., Espeholt, L., Riquelme, C., Vincent, D., Michalski, M., Bousquet, O., and Gelly, S. Google research football: A novel reinforcement learning environment. *CoRR*, abs/1907.11180, 2019. URL `http://arxiv.org/abs/1907.11180`.

Liang, E., Liaw, R., Moritz, P., Nishihara, R., Fox, R., Goldberg, K., Gonzalez, J. E., Jordan, M. I., and Stoica, I. URL `https://arxiv.org/abs/1712.09381`.

Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Liu, I.-J., Yeh, R. A., and Schwing, A. G. Pic: permutation invariant critic for multi-agent deep reinforcement learning. In *Conference on Robot Learning*, pp. 590–602. PMLR, 2020.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, 2017.

Mordatch, I. and Abbeel, P. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908*, 2017.

Morin, A. Levels of consciousness and self-awareness: A comparison and integration of various neurocognitive views. *Consciousness and cognition*, 15(2):358–371, 2006.

Ndousse, K., Eck, D., Levine, S., and Jaques, N. Emergent social learning via multi-agent reinforcement learning, 2021.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning, 2018.

Roy, J., Barde, P., Harvey, F., Nowrouzezahrai, D., and Pal, C. Promoting coordination through policy regularization in multi-agent deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:15774–15785, 2020.

Schafer, L. *Curiosity in multi-agent reinforcement learning*. PhD thesis, Master's thesis, The University of Edinburgh, 2019.

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

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pp. 8583–8592. PMLR, 2020.

Srivastava, S., Li, C., Lingelbach, M., Martín-Martín, R., Xia, F., Vainio, K., Lian, Z., Gokmen, C., Buch, S., Liu, C. K., et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. *arXiv preprint arXiv:2108.03332*, 2021.

Stadie, B. C., Levine, S., and Abbeel, P. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Strouse, D., Kleiman-Weiner, M., Tenenbaum, J., Botvinick, M., and Schwab, D. J. Learning to share and hide intentions using information regularization. *Advances in Neural Information Processing Systems*, 31:10249–10259, 2018.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Swamy, G., Reddy, S., Levine, S., and Dragan, A. D. Scaled autonomy: Enabling human operators to control robot fleets. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5942–5948. IEEE, 2020.

Theraulaz, G. and Bonabeau, E. Coordination in distributed building. *Science*, 269(5224):686–688, 1995.

Tomasello, M., Carpenter, M., Call, J., Behne, T., and Moll, H. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and brain sciences*, 28(5):675–691, 2005.

Wang, R. E., Kew, J. C., Lee, D., Lee, T.-W. E., Zhang, T., Ichter, B., Tan, J., and Faust, A. Model-based reinforcement learning for decentralized multiagent rendezvous, 2020a.

Wang, R. E., Wu, S. A., Evans, J. A., Tenenbaum, J. B., Parkes, D. C., and Kleiman-Weiner, M. Too many cooks: Bayesian inference for coordinating multi-agent collaboration, 2020b.

Wang, T., Wang, J., Wu, Y., and Zhang, C. Influence-based multi-agent exploration. *arXiv preprint arXiv:1910.05512*, 2019.

Wang, X., Xiong, W., Wang, H., and Wang, W. Y. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 37–53, 2018.

Weng, J., Chen, H., Yan, D., You, K., Duburcq, A., Zhang, M., Su, H., and Zhu, J. Tianshou: A highly modularized deep reinforcement learning library. *arXiv preprint arXiv:2107.14171*, 2021.

Ying, W. and Dayong, S. Multi-agent framework for third party logistics in e-commerce. *Expert Systems with Applications*, 29(2):431–436, 2005.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 5 for the results mentioned in the abstract.
   (b) Did you describe the limitations of your work? [Yes] See Section 6 for a discussion on the limitations.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] See the Appendix

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have read the ethics review guidelines.

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [N/A]

   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the Appendix

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the Experiments 5 and Appendix sections.

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See the Experiments 5 section.

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the Experiments 5 and Appendix sections.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes] See the Alignment 4 and Experiments 5 sections.

   (b) Did you mention the license of the assets? [Yes] See the Appendix

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See the Appendix

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We did not collect any data.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We did not collect any data.

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not crowdsource any data.

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] There were no human subjects experiments in our work.

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No human subjects were hired.

6.1 Code

We upload our code for training and evaluating agents with and without alignment in both the multi-agent particle and Google Research football environments here: `https://anonymous.4open.science/r/alignment-9F11/`.

A Appendix

A.1 Code

We upload our code for training and evaluating agents with and without expectation alignment in both the multi-agent particle and Google Research football environments here: `https://github.com/StanfordVL/alignment`.

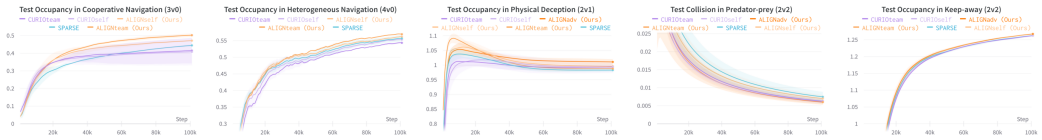

Figure 8: Learning curves of test occupancy/collision in all five tasks in the multi-agent particle environment. On average, it takes the best ELIGN variant 65 epochs to reach the maximum score of the best CURIO method at 100 epochs, which means ELIGN requires on average 35% fewer samples than CURIO.

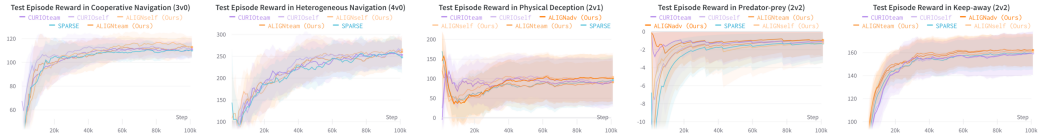

Figure 9: Learning curves of test episode rewards in all five tasks in the multi-agent particle environment.

A.2 Symmetry-breaking initializations

We create a symmetry-breaking version of each task for evaluation by initializing the environment in the following ways:

Cooperative Navigation and Heterogenous Navigation: All agents are initialized at the origin (i.e. center of the world), and target landmarks are placed randomly on a circle perimeter with the maximum radius (i.e. world radius - the greatest landmark size) so that each agent is equidistant from each target landmark.

Physical Deception: Both agents and adversaries start at the origin. All the landmarks, including the goal, are randomly initialized on a circle perimeter.

Predator-prey: The collaborative agents are initialized at the center while the adversaries are placed randomly on a circle perimeter. All the landmarks are randomly initialized in the world.

Keep-away: All the cooperative agents are placed at the origin. Adversaries and landmarks, including the goal, are randomly initialized on a circle perimeter. In this task setup, we do not initialize the adversaries at the center because they are awarded for colliding with the cooperative agents.

A.3 Learning curves

Compared to the best baseline's highest performance at the 100th epoch, we find that it only takes the best ELIGN variant 31 (Coop nav), 74 (Hetero nav), 30 (Physical dec), 97 (Predator-prey) and 92 (Keep-away) epochs respectively in the five multi-agent particle tasks (8). This means that on average the best ELIGN variant requires 35 ± 32 fewer training steps to reach the same performance as curiosity or spare methods.

A.4 Assets and licenses

We use four assets in total, two of which are existing multi-agent reinforcement learning environments, and the other two are libraries for training reinforcement learning algorithms.

We conduct our evaluation on the multi-agent particle (Lowe et al., 2017) and Google Research football (Kurach et al., 2019) environments, which are under the MIT license and Apache-2.0 license respectively.

We adapt the tianshou (Weng et al., 2021) and rllib (part of the ray package) (Liang et al.) libraries to our experiments, and they are under the MIT license and Apache-2.0 license respectively.

Table 3: We report the mean test episode extrinsic rewards and standard errors of *decentralized* methods with different intrinsic rewards in fully observable environments.

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (3v0)**	**Hetero nav. (4v0)**	**Phy decep. (2v1)**	**Pred-prey (2v2)**	**Keep-away (2v2)**
Full observability	SPARSE[1]	154.00 ± 10.51	274.75 ± 19.74	82.97 ± 12.23	-10.48 ± 4.20	$\mathbf{4.95 \pm 2.96}$
	$\text{CURIO}_{\text{self}}$	154.71 ± 8.00	268.85 ± 15.61	$\mathbf{100.66 \pm 15.14}$	-8.74 ± 4.62	-2.00 ± 1.24
	$\text{ELIGN}_{\text{self}}$	$\mathbf{161.70 \pm 4.52}$	$\mathbf{280.16 \pm 17.12}$	87.50 ± 15.40	$\mathbf{-5.60 \pm 2.60}$	0.40 ± 1.92

[1] Lowe et al. (2017)

A.5 Societal impacts

While developing new intrinsic rewards to improve decentralized multi-agent training can help develop and deploy agents in a variety of applications, we foresee no immediate societal consequences of this work. However, our experiments thus far have not studied the possible degradation of behaviors when agents align to malicious teammates. We have also not tested how emergent properties promote better or worse human collaborators.

A.6 Additional tables

We include 20 tables of additional results that quantify the agents' performance under fully observable environments, with centralized training, and beyond extrinsic reward.

Table 3 reports the test episode rewards of decentralized methods under the fully observable multi-agent particle environment.

Table 4 reports the test episode rewards of centralized methods under partial observability.

Table 5 and 6 report two sets of metrics of *decentralized* methods trained with different intrinsic rewards in both partially and fully observable settings. Table 5 reports the average number of agent-target occupancies per step (or, we can understand it as: on average, the total number of goals occupied by the agents at any given timestep throughout an episode) and agent-adversary collisions in *Predator-prey*. Higher scores are better for the occupancy metric, and lower scores are better for collision. Table 6 reports the average minimum agent-to-target distance and agent-to-adversary distance. Agent-to-target distances measure the closest distance an agent achieves to the target location; lower scores are better on this metric. Agent-to-adversary distances measure the closest distance an adversary gets to a good agent; higher scores are better on this metric. Note that these distance-based metrics are not included in the reward functions, and should mainly be used to make comparisons in the case where primary metrics (i.e, reward and occupancy/collision count) have the same values.

Table 7 and 8 report the same metrics as 5 and 6 respectively, but in *scaled* environments with more agents.

Tables 9, 10 and 11 report the test episode reward and additional metrics of *decentralized* algorithms in the *symmetry-breaking* experiments conducted under "Investigating how alignment reward helps". Table 12, 13 and 14 report the same set of metrics but from experiments conducted in *scaled* and *symmetry-breaking* environments.

Table 15 reports the test mean episode rewards of *centralized* algorithms with different intrinsic rewards under full observability. Table 16 and 17 show the other two sets of metrics (i.e, occupancy/collision count and agent-target/agent-adversary distance) of *centralized* algorithms. Table 18, 19, and 20 contain the same metrics as 15, 16 and 17 respectively, but in *scaled* environments.

Finally, Tables 21 and 22 report the test episode reward values and secondary distance-based metrics for the zero-shot generalization experiments conducted under "Investigating how ELIGN reward helps". These experiments measure how well agents trained on different seeds generalized to new partners trained on other seeds.

A.7 Model architecture and hyperparameters

Table 23 presents the model architecture and hyperparameters used to train the algorithms in the multi-agent particle and Google Research football environments.

Table 4: We report the mean test episode extrinsic rewards and standard errors of *centralized* methods with different intrinsic rewards under partial observability.

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (3v0)**	**Hetero nav. (4v0)**	**Phy decep. (2v1)**	**Pred-prey (2v2)**	**Keep-away (2v2)**
Partial observability	SPARSE[1]	113.25 ± 8.10	178.62 ± 9.62	$\mathbf{117.45 \pm 10.63}$	-1.96 ± 1.45	$\mathbf{35.79 \pm 14.93}$
	CURIO[2] $_{\text{self}}$	128.77 ± 7.70	190.30 ± 7.73	111.08 ± 10.09	-1.63 ± 1.27	13.94 ± 12.56
	CURIO[3] $_{\text{team}}$	114.13 ± 11.84	189.80 ± 11.81	114.32 ± 5.46	-3.04 ± 1.09	6.01 ± 3.36
	ELIGN$_{\text{self}}$	$\mathbf{137.14 \pm 3.63}$	169.58 ± 14.99	93.27 ± 3.70	-0.41 ± 0.28	22.77 ± 9.91
	ELIGN$_{\text{team}}$	119.10 ± 10.89	$\mathbf{210.81 \pm 9.70}$	96.49 ± 6.46	-0.92 ± 0.72	24.94 ± 12.58
	ELIGN$_{\text{adv}}$	—	—	102.37 ± 6.98	$\mathbf{-0.13 \pm 0.03}$	8.70 ± 4.44

[1] Lowe et al. (2017),[2] Stadie et al. (2015),[3] Iqbal & Sha (2020)

Table 5: The average test occupancy/collision count per step and standard errors of *decentralized* methods with different intrinsic rewards under partial and full observability. Higher scores are better for the occupancy metric (↑), and lower scores are better for the collision metric (↓).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (3v0)** ↑	**Hetero nav. (4v0)** ↑	**Phy decep. (2v1)** ↑	**Pred-prey (2v2)** ↓	**Keep-away (2v2)** ↑
Partial observability	SPARSE	0.46 ± 0.05	0.57 ± 0.01	0.98 ± 0.07	0.02 ± 0.01	0.07 ± 0.02
	CURIO$_{\text{self}}$	0.43 ± 0.03	0.60 ± 0.01	0.99 ± 0.03	0.02 ± 0.01	0.14 ± 0.02
	CURIO$_{\text{team}}$	0.42 ± 0.05	0.59 ± 0.01	0.95 ± 0.01	0.02 ± 0.01	0.10 ± 0.03
	ELIGN$_{\text{self}}$	0.52 ± 0.03	0.61 ± 0.01	0.95 ± 0.02	0.03 ± 0.01	0.10 ± 0.02
	ELIGN$_{\text{team}}$	0.44 ± 0.04	0.58 ± 0.02	0.99 ± 0.07	0.03 ± 0.01	0.07 ± 0.02
	ELIGN$_{\text{adv}}$	—	—	1.00 ± 0.06	0.01 ± 0.01	0.15 ± 0.03
Full observability	SPARSE	0.46 ± 0.11	0.57 ± 0.01	0.88 ± 0.09	0.03 ± 0.01	0.06 ± 0.02
	CURIO$_{\text{self}}$	0.50 ± 0.07	0.59 ± 0.02	1.09 ± 0.13	0.03 ± 0.01	0.02 ± 0.00
	ELIGN$_{\text{self}}$	0.48 ± 0.11	0.58 ± 0.02	0.83 ± 0.10	0.02 ± 0.01	0.04 ± 0.01

Table 6: The average test agent-to-target (agt-target) and agent-to-adversary (agt-adv) distances and standard errors of *decentralized* methods with different intrinsic rewards under partial and full observability. Lower scores are better for agt-target (↓), and higher scores are better for agt-adv (↑).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (3v0)** ↓	**Hetero nav. (4v0)** ↓	**Phy decep. (2v1)** ↓	**Pred-prey (2v2)** ↑	**Keep-away (2v2)** ↓
Partial observability	SPARSE	0.30 ± 0.02	0.23 ± 0.00	0.26 ± 0.01	1.45 ± 0.11	1.41 ± 0.07
	CURIO$_{\text{self}}$	0.32 ± 0.02	0.25 ± 0.01	0.25 ± 0.00	1.36 ± 0.06	1.14 ± 0.09
	CURIO$_{\text{team}}$	0.31 ± 0.01	0.25 ± 0.01	0.26 ± 0.00	1.48 ± 0.13	1.31 ± 0.10
	ELIGN$_{\text{self}}$	0.33 ± 0.03	0.25 ± 0.01	0.26 ± 0.00	1.39 ± 0.12	1.26 ± 0.09
	ELIGN$_{\text{team}}$	0.33 ± 0.02	0.23 ± 0.01	0.25 ± 0.01	1.38 ± 0.13	1.38 ± 0.09
	ELIGN$_{\text{adv}}$	—	—	0.25 ± 0.01	1.54 ± 0.08	1.14 ± 0.09
Full observability	SPARSE	0.32 ± 0.09	0.23 ± 0.00	0.26 ± 0.01	1.23 ± 0.12	1.27 ± 0.09
	CURIO$_{\text{self}}$	0.28 ± 0.04	0.22 ± 0.01	0.23 ± 0.01	1.37 ± 0.15	1.53 ± 0.03
	ELIGN$_{\text{self}}$	0.30 ± 0.07	0.23 ± 0.01	0.27 ± 0.01	1.40 ± 0.13	1.41 ± 0.10

Table 7: The average test occupancy/collision count per step and standard errors of *decentralized* methods with different intrinsic rewards in *scaled* environments under partial and full observability. Higher scores are better for the occupancy metric (↑), and lower scores are better for the collision metric (↓).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (5v0)** ↑	**Hetero nav. (6v0)** ↑	**Phy decep. (4v2)** ↑	**Pred-prey (4v4)** ↓	**Keep-away (4v4)** ↑
Partial observability	SPARSE	0.50 ± 0.04	0.46 ± 0.08	1.20 ± 0.10	0.11 ± 0.02	0.08 ± 0.02
	CURIO$_{\text{self}}$	0.48 ± 0.03	0.63 ± 0.01	1.20 ± 0.08	0.07 ± 0.02	0.15 ± 0.06
	CURIO$_{\text{team}}$	0.53 ± 0.03	0.60 ± 0.02	1.20 ± 0.09	0.05 ± 0.02	0.06 ± 0.01
	ELIGN$_{\text{self}}$	0.49 ± 0.03	0.67 ± 0.00	1.30 ± 0.23	0.04 ± 0.02	0.14 ± 0.08
	ELIGN$_{\text{team}}$	0.56 ± 0.04	0.56 ± 0.00	1.21 ± 0.09	0.08 ± 0.02	0.10 ± 0.02
	ELIGN$_{\text{adv}}$	—	—	1.23 ± 0.10	0.08 ± 0.02	0.16 ± 0.07
Full observability	SPARSE	0.52 ± 0.11	0.46 ± 0.08	0.99 ± 0.09	0.21 ± 0.01	0.03 ± 0.00
	CURIO$_{\text{self}}$	0.39 ± 0.13	0.56 ± 0.01	0.86 ± 0.04	0.16 ± 0.03	0.04 ± 0.00
	ELIGN$_{\text{self}}$	0.55 ± 0.11	0.56 ± 0.00	1.04 ± 0.07	0.15 ± 0.03	0.06 ± 0.02

Table 8: The average test agent-to-target (agt-target) and agent-to-adversary (agt-adv) distances and standard errors of *decentralized* methods with different intrinsic rewards in *scaled* environments under partial and full observability. Lower scores are better for agt-target (↓), and higher scores are better for agt-adv (↑).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (5v0)** ↓	**Hetero nav. (6v0)** ↓	**Phy decep. (4v2)** ↓	**Pred-prey (4v4)** ↑	**Keep-away (4v4)** ↓
Partial observability	SPARSE	0.22 ± 0.01	0.27 ± 0.05	0.23 ± 0.02	2.03 ± 0.15	2.97 ± 0.17
	CURIO$_{\text{self}}$	0.30 ± 0.02	0.21 ± 0.01	0.24 ± 0.01	2.18 ± 0.13	2.70 ± 0.25
	CURIO$_{\text{team}}$	0.23 ± 0.02	0.22 ± 0.01	0.23 ± 0.02	2.29 ± 0.12	3.14 ± 0.08
	ELIGN$_{\text{self}}$	0.29 ± 0.03	0.19 ± 0.00	0.24 ± 0.02	2.39 ± 0.11	2.97 ± 0.30
	ELIGN$_{\text{team}}$	0.23 ± 0.04	0.21 ± 0.00	0.23 ± 0.01	2.16 ± 0.12	2.88 ± 0.19
	ELIGN$_{\text{adv}}$	—	—	0.22 ± 0.01	2.12 ± 0.16	2.66 ± 0.23
Full observability	SPARSE	0.23 ± 0.06	0.27 ± 0.05	0.21 ± 0.02	1.64 ± 0.02	3.28 ± 0.02
	CURIO$_{\text{self}}$	0.33 ± 0.09	0.21 ± 0.01	0.22 ± 0.01	1.81 ± 0.12	3.24 ± 0.08
	ELIGN$_{\text{self}}$	0.20 ± 0.04	0.21 ± 0.00	0.21 ± 0.01	1.82 ± 0.10	2.97 ± 0.17

Table 9: We report the mean test episode extrinsic rewards and standard errors of *decentralized* methods with different intrinsic rewards in *symmetry-breaking* settings under partial and full observability.

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (3v0)**	**Hetero nav. (4v0)**	**Phy decep. (2v1)**	**Pred-prey (2v2)**	**Keep-away (2v2)**
Partial observability	SPARSE	97.45 ± 10.49	184.18 ± 7.63	59.39 ± 21.10	-1.89 ± 1.69	3.85 ± 4.25
	CURIO$_{\text{self}}$	85.23 ± 10.88	184.07 ± 9.99	54.17 ± 27.40	-2.86 ± 1.19	19.57 ± 4.92
	CURIO$_{\text{team}}$	81.50 ± 15.78	141.78 ± 20.04	41.12 ± 13.37	-2.80 ± 1.91	10.21 ± 6.34
	ELIGN$_{\text{self}}$	110.29 ± 9.67	176.98 ± 6.38	98.90 ± 17.71	-4.00 ± 2.14	9.47 ± 3.99
	ELIGN$_{\text{team}}$	92.41 ± 10.70	187.42 ± 11.29	74.06 ± 21.58	-2.00 ± 1.39	3.32 ± 3.04
	ELIGN$_{\text{adv}}$	—	—	87.55 ± 15.35	-1.40 ± 1.25	13.77 ± 3.58
Full observability	SPARSE	150.42 ± 15.18	250.41 ± 14.23	69.06 ± 14.06	-7.62 ± 3.50	3.50 ± 4.00
	CURIO$_{\text{self}}$	149.48 ± 9.42	241.69 ± 19.58	52.69 ± 17.97	-10.40 ± 6.33	-1.10 ± 0.59
	ELIGN$_{\text{self}}$	152.08 ± 6.68	275.69 ± 7.49	75.79 ± 24.54	-4.44 ± 2.05	0.96 ± 3.14

Table 10: The average test occupancy/collision count per step and standard errors of *decentralized* methods with different intrinsic rewards in *symmetry-breaking* settings under partial and full observability. Higher scores are better for the occupancy metric (↑), and lower scores are better for the collision metric (↓).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (3v0)** ↑	**Hetero nav. (4v0)** ↑	**Phy decep. (2v1)** ↑	**Pred-prey (2v2)** ↓	**Keep-away (2v2)** ↑
Partial observability	SPARSE	0.26 ± 0.04	0.27 ± 0.02	0.67 ± 0.09	0.02 ± 0.01	0.04 ± 0.03
	CURIO$_{\text{self}}$	0.22 ± 0.01	0.28 ± 0.04	0.61 ± 0.06	0.02 ± 0.01	0.15 ± 0.04
	CURIO$_{\text{team}}$	0.26 ± 0.06	0.29 ± 0.02	0.65 ± 0.05	0.01 ± 0.01	0.08 ± 0.04
	ELIGN$_{\text{self}}$	0.29 ± 0.05	0.32 ± 0.03	0.62 ± 0.02	0.02 ± 0.01	0.08 ± 0.03
	ELIGN$_{\text{team}}$	0.27 ± 0.04	0.27 ± 0.02	0.72 ± 0.10	0.02 ± 0.01	0.05 ± 0.04
	ELIGN$_{\text{adv}}$	—	—	0.68 ± 0.07	0.00 ± 0.00	0.14 ± 0.04
Full observability	SPARSE	0.45 ± 0.12	0.54 ± 0.01	0.89 ± 0.11	0.03 ± 0.01	0.05 ± 0.02
	CURIO$_{\text{self}}$	0.48 ± 0.08	0.54 ± 0.01	1.13 ± 0.14	0.04 ± 0.02	0.00 ± 0.00
	ELIGN$_{\text{self}}$	0.46 ± 0.11	0.54 ± 0.01	0.86 ± 0.12	0.02 ± 0.01	0.02 ± 0.02

Table 11: The average test agent-to-target (agt-target) and agent-to-adversary (agt-adv) distances and standard errors of *decentralized* methods with different intrinsic rewards in *symmetry-breaking* settings under partial and full observability. Lower scores are better for agt-target (↓), and higher scores are better for agt-adv (↑).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (3v0)** ↓	**Hetero nav. (4v0)** ↓	**Phy decep. (2v1)** ↓	**Pred-prey (2v2)** ↑	**Keep-away (2v2)** ↓
Partial observability	SPARSE	0.53 ± 0.02	0.57 ± 0.02	0.35 ± 0.03	1.49 ± 0.14	1.57 ± 0.15
	CURIO$_{\text{self}}$	0.57 ± 0.04	0.55 ± 0.04	0.37 ± 0.02	1.29 ± 0.06	1.07 ± 0.17
	CURIO$_{\text{team}}$	0.53 ± 0.03	0.55 ± 0.03	0.35 ± 0.02	1.49 ± 0.13	1.37 ± 0.18
	ELIGN$_{\text{self}}$	0.68 ± 0.05	0.52 ± 0.03	0.34 ± 0.01	1.39 ± 0.13	1.26 ± 0.18
	ELIGN$_{\text{team}}$	0.55 ± 0.05	0.56 ± 0.02	0.31 ± 0.02	1.41 ± 0.10	1.53 ± 0.17
	ELIGN$_{\text{adv}}$	—	—	0.33 ± 0.04	1.61 ± 0.08	1.08 ± 0.15
Full observability	SPARSE	0.45 ± 0.12	0.29 ± 0.00	0.25 ± 0.01	1.28 ± 0.15	1.30 ± 0.15
	CURIO$_{\text{self}}$	0.33 ± 0.06	0.30 ± 0.01	0.22 ± 0.01	1.47 ± 0.17	1.71 ± 0.05
	ELIGN$_{\text{self}}$	0.46 ± 0.11	0.30 ± 0.00	0.25 ± 0.02	1.49 ± 0.16	1.53 ± 0.16

Table 12: We report the mean test episode extrinsic rewards and standard errors of *decentralized* methods with different intrinsic rewards in *scaled* and *symmetry-breaking* settings.

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (5v0)**	**Hetero nav. (6v0)**	**Phy decep. (4v2)**	**Pred-prey (4v4)**	**Keep-away (4v4)**
Partial observability	SPARSE	328.24 ± 24.17	405.08 ± 21.53	172.87 ± 32.43	-35.40 ± 8.63	1.37 ± 3.48
	CURIO$_{\text{self}}$	295.48 ± 20.54	436.17 ± 26.30	202.39 ± 26.06	-11.19 ± 3.65	9.24 ± 8.49
	CURIO$_{\text{team}}$	316.33 ± 14.44	422.71 ± 13.24	229.50 ± 28.29	-11.56 ± 6.37	-1.29 ± 1.58
	ELIGN$_{\text{self}}$	357.40 ± 19.52	412.39 ± 12.63	129.07 ± 51.08	-7.34 ± 5.12	11.97 ± 13.30
	ELIGN$_{\text{team}}$	354.14 ± 19.53	417.94 ± 22.29	184.21 ± 23.16	-19.37 ± 6.44	4.05 ± 5.78
	ELIGN$_{\text{adv}}$	—	—	148.69 ± 31.79	-23.42 ± 8.32	18.71 ± 14.78
Full observability	SPARSE	466.17 ± 28.16	471.19 ± 16.23	233.61 ± 25.44	-39.24 ± 6.63	-5.10 ± 0.26
	CURIO$_{\text{self}}$	509.91 ± 14.10	606.07 ± 7.55	256.13 ± 41.13	-38.66 ± 13.38	-6.58 ± 1.29
	ELIGN$_{\text{self}}$	520.25 ± 9.68	510.18 ± 25.71	222.31 ± 15.39	-30.56 ± 9.87	-4.27 ± 2.53

Table 13: The average test occupancy/collision count per step and standard errors of *decentralized* methods with different intrinsic rewards in *scaled* and *symmetry-breaking* settings. under partial and full observability. Higher scores are better for the occupancy metric (↑), and lower scores are better for the collision metric (↓).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (5v0)** ↑	**Hetero nav. (6v0)** ↑	**Phy decep. (4v2)** ↑	**Pred-prey (4v4)** ↓	**Keep-away (4v4)** ↑
Partial observability	SPARSE	0.37 ± 0.05	0.38 ± 0.03	0.75 ± 0.04	0.13 ± 0.03	0.04 ± 0.02
	CURIO$_{\text{self}}$	0.29 ± 0.03	0.43 ± 0.02	0.66 ± 0.06	0.05 ± 0.02	0.12 ± 0.08
	CURIO$_{\text{team}}$	0.32 ± 0.02	0.34 ± 0.01	0.78 ± 0.07	0.05 ± 0.02	0.02 ± 0.01
	ELIGN$_{\text{self}}$	0.29 ± 0.03	0.39 ± 0.02	0.96 ± 0.20	0.04 ± 0.03	0.11 ± 0.10
	ELIGN$_{\text{team}}$	0.37 ± 0.03	0.40 ± 0.03	0.72 ± 0.03	0.07 ± 0.03	0.06 ± 0.03
	ELIGN$_{\text{adv}}$	—	—	0.63 ± 0.09	0.07 ± 0.03	0.15 ± 0.10
Full observability	SPARSE	0.52 ± 0.11	0.43 ± 0.09	0.86 ± 0.07	0.19 ± 0.02	0.00 ± 0.00
	CURIO$_{\text{self}}$	0.39 ± 0.14	0.54 ± 0.00	0.81 ± 0.06	0.14 ± 0.04	0.00 ± 0.00
	ELIGN$_{\text{self}}$	0.55 ± 0.11	0.55 ± 0.00	0.94 ± 0.08	0.12 ± 0.03	0.02 ± 0.01

Table 14: The average test agent-to-target (agt-target) and agent-to-adversary (agt-adv) distances and standard errors of *decentralized* methods with different intrinsic rewards in *scaled* and *symmetry-breaking* settings under partial and full observability. Lower scores are better for agt-target (↓), and higher scores are better for agt-adv (↑).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (5v0)** ↓	**Hetero nav. (6v0)** ↓	**Phy decep. (4v2)** ↓	**Pred-prey (4v4)** ↑	**Keep-away (4v4)** ↓
Partial observability	SPARSE	0.36 ± 0.01	0.42 ± 0.02	0.35 ± 0.01	2.04 ± 0.18	3.10 ± 0.29
	$\text{CURIO}_{\text{self}}$	0.50 ± 0.04	0.37 ± 0.02	0.38 ± 0.03	2.35 ± 0.12	2.70 ± 0.35
	$\text{CURIO}_{\text{team}}$	0.43 ± 0.03	0.45 ± 0.01	0.35 ± 0.02	2.39 ± 0.15	3.37 ± 0.19
	$\text{ELIGN}_{\text{self}}$	0.52 ± 0.05	0.41 ± 0.02	0.37 ± 0.04	2.52 ± 0.15	3.22 ± 0.42
	$\text{ELIGN}_{\text{team}}$	0.42 ± 0.03	0.41 ± 0.02	0.37 ± 0.01	2.25 ± 0.15	3.00 ± 0.33
	$\text{ELIGN}_{\text{adv}}$	—	—	0.41 ± 0.05	2.27 ± 0.16	2.62 ± 0.30
Full observability	SPARSE	0.29 ± 0.07	0.37 ± 0.06	0.26 ± 0.01	1.81 ± 0.04	3.69 ± 0.05
	$\text{CURIO}_{\text{self}}$	0.42 ± 0.11	0.29 ± 0.00	0.26 ± 0.02	2.02 ± 0.16	3.63 ± 0.12
	$\text{ELIGN}_{\text{self}}$	0.26 ± 0.05	0.29 ± 0.00	0.27 ± 0.01	2.10 ± 0.12	3.24 ± 0.26

Table 15: We report the mean test episode extrinsic rewards and standard errors of *centralized* methods with different intrinsic rewards under full observability.

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (3v0)**	**Hetero nav. (4v0)**	**Phy decep. (2v1)**	**Pred-prey (2v2)**	**Keep-away (2v2)**
Full observability	SPARSE	106.02 ± 20.95	123.17 ± 18.77	130.90 ± 6.59	-1.90 ± 1.61	12.49 ± 9.83
	$\text{CURIO}_{\text{self}}$	86.52 ± 16.02	108.84 ± 6.89	107.84 ± 13.67	-1.69 ± 0.60	23.70 ± 12.95
	$\text{ELIGN}_{\text{self}}$	120.47 ± 12.26	134.30 ± 5.84	105.74 ± 9.72	-2.37 ± 1.39	22.92 ± 7.00

Table 16: The average test occupancy/collision count per step and standard errors of *centralized* methods with different intrinsic rewards under partial and full observability. Higher scores are better for the occupancy metric (↑), and lower scores are better for the collision metric (↓).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (3v0)** ↑	**Hetero nav. (4v0)** ↑	**Phy decep. (2v1)** ↑	**Pred-prey (2v2)** ↓	**Keep-away (2v2)** ↑
Partial observability	SPARSE	0.29 ± 0.10	0.50 ± 0.03	0.94 ± 0.06	0.00 ± 0.00	0.36 ± 0.12
	$\text{CURIO}_{\text{self}}$	0.28 ± 0.09	0.47 ± 0.03	0.94 ± 0.03	0.01 ± 0.00	0.17 ± 0.10
	$\text{CURIO}_{\text{team}}$	0.33 ± 0.10	0.47 ± 0.04	0.92 ± 0.01	0.01 ± 0.00	0.08 ± 0.03
	$\text{ELIGN}_{\text{self}}$	0.21 ± 0.10	0.50 ± 0.01	0.92 ± 0.02	0.00 ± 0.00	0.25 ± 0.08
	$\text{ELIGN}_{\text{team}}$	0.23 ± 0.09	0.55 ± 0.02	0.90 ± 0.07	0.01 ± 0.00	0.24 ± 0.11
	$\text{ELIGN}_{\text{adv}}$	—	—	0.94 ± 0.04	0.00 ± 0.00	0.10 ± 0.05
Full observability	SPARSE	0.34 ± 0.10	0.33 ± 0.07	0.88 ± 0.04	0.01 ± 0.00	0.26 ± 0.11
	$\text{CURIO}_{\text{self}}$	0.30 ± 0.07	0.32 ± 0.05	0.82 ± 0.02	0.01 ± 0.01	0.33 ± 0.16
	$\text{ELIGN}_{\text{self}}$	0.30 ± 0.11	0.40 ± 0.04	0.88 ± 0.05	0.01 ± 0.01	0.30 ± 0.07

Table 17: The average test agent-to-target (agt-target) and agent-to-adversary (agt-adv) distances and standard errors of *centralized* methods with different intrinsic rewards under partial and full observability. Lower scores are better for agt-target (↓), and higher scores are better for agt-adv (↑).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (3v0)** ↓	**Hetero nav. (4v0)** ↓	**Phy decep. (2v1)** ↓	**Pred-prey (2v2)** ↑	**Keep-away (2v2)** ↓
Partial observability	SPARSE	0.42 ± 0.05	0.29 ± 0.02	0.27 ± 0.01	1.54 ± 0.02	1.38 ± 0.13
	$\text{CURIO}_{\text{self}}$	0.42 ± 0.05	0.29 ± 0.01	0.27 ± 0.01	1.46 ± 0.05	1.40 ± 0.13
	$\text{CURIO}_{\text{team}}$	0.41 ± 0.06	0.29 ± 0.02	0.28 ± 0.01	1.49 ± 0.04	1.43 ± 0.14
	$\text{ELIGN}_{\text{self}}$	0.50 ± 0.07	0.29 ± 0.01	0.27 ± 0.01	1.60 ± 0.04	1.26 ± 0.12
	$\text{ELIGN}_{\text{team}}$	0.45 ± 0.05	0.27 ± 0.01	0.28 ± 0.01	1.52 ± 0.04	1.35 ± 0.14
	$\text{ELIGN}_{\text{adv}}$	—	—	0.28 ± 0.01	1.55 ± 0.03	1.45 ± 0.10
Full observability	SPARSE	0.38 ± 0.07	0.34 ± 0.04	0.25 ± 0.00	1.59 ± 0.06	1.43 ± 0.09
	$\text{CURIO}_{\text{self}}$	0.36 ± 0.05	0.32 ± 0.02	0.25 ± 0.01	1.53 ± 0.09	1.08 ± 0.15
	$\text{ELIGN}_{\text{self}}$	0.43 ± 0.08	0.30 ± 0.02	0.25 ± 0.00	1.51 ± 0.08	1.18 ± 0.15

Table 18: We report the mean test episode extrinsic rewards and standard errors of *centralized* methods with different intrinsic rewards in *scaled* environments.

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (5v0)**	**Hetero nav. (6v0)**	**Phy decep. (4v2)**	**Pred-prey (4v4)**	**Keep-away (4v4)**
Partial observability	SPARSE	100.63 ± 19.36	346.16 ± 18.95	-38.99 ± 16.18	-17.33 ± 4.29	-2.50 ± 2.64
	$\text{ELIGN}_{\text{self}}$	112.15 ± 19.69	375.21 ± 26.10	13.71 ± 29.53	-20.12 ± 1.42	-4.68 ± 1.21
	$\text{ELIGN}_{\text{team}}$	97.93 ± 25.23	372.41 ± 44.28	60.07 ± 13.26	-27.87 ± 0.99	1.72 ± 3.79
	$\text{ELIGN}_{\text{adv}}$	—	—	21.67 ± 48.17	-17.68 ± 5.59	-4.92 ± 1.81
Full observability	SPARSE	50.60 ± 13.10	153.76 ± 19.81	97.32 ± 17.95	-38.25 ± 5.06	-3.39 ± 2.77
	$\text{ELIGN}_{\text{self}}$	186.55 ± 53.15	127.97 ± 13.02	103.46 ± 28.91	-23.29 ± 5.00	-4.90 ± 0.67

Table 19: The average test occupancy/collision count per step and standard errors of *centralized* methods with different intrinsic rewards in *scaled* environments under partial and full observability. Higher scores are better for the occupancy metric (↑), and lower scores are better for the collision metric (↓).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (5v0)** ↑	**Hetero nav. (6v0)** ↑	**Phy decep. (4v2)** ↑	**Pred-prey (4v4)** ↓	**Keep-away (4v4)** ↑
Partial observability	SPARSE	0.11 ± 0.02	0.29 ± 0.06	0.56 ± 0.05	0.06 ± 0.02	0.07 ± 0.02
	ELIGN$_{\text{self}}$	0.23 ± 0.09	0.33 ± 0.04	0.56 ± 0.06	0.08 ± 0.00	0.05 ± 0.00
	ELIGN$_{\text{team}}$	0.27 ± 0.10	0.33 ± 0.05	0.50 ± 0.08	0.09 ± 0.00	0.09 ± 0.03
	ELIGN$_{\text{adv}}$	—	—	0.60 ± 0.09	0.05 ± 0.02	0.05 ± 0.01
Full observability	SPARSE	0.10 ± 0.04	0.16 ± 0.04	0.50 ± 0.03	0.12 ± 0.01	0.06 ± 0.01
	ELIGN$_{\text{self}}$	0.16 ± 0.09	0.11 ± 0.00	0.55 ± 0.02	0.10 ± 0.02	0.04 ± 0.01

Table 20: The average test agent-to-target (agt-target) and agent-to-adversary (agt-adv) distances and standard errors of *centralized* methods with different intrinsic rewards in *scaled* environments under partial and full observability. Lower scores are better for agt-target (↓), and higher scores are better for agt-adv (↑).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (5v0)** ↓	**Hetero nav. (6v0)** ↓	**Phy decep. (4v2)** ↓	**Pred-prey (4v4)** ↑	**Keep-away (4v4)** ↓
Partial observability	SPARSE	0.33 ± 0.01	0.29 ± 0.02	0.34 ± 0.02	2.27 ± 0.08	3.12 ± 0.18
	ELIGN$_{\text{self}}$	0.32 ± 0.04	0.28 ± 0.01	0.36 ± 0.02	2.32 ± 0.09	3.25 ± 0.05
	ELIGN$_{\text{team}}$	0.30 ± 0.04	0.28 ± 0.01	0.37 ± 0.03	2.29 ± 0.08	3.01 ± 0.20
	ELIGN$_{\text{adv}}$	—	—	0.33 ± 0.04	2.44 ± 0.08	3.23 ± 0.13
Full observability	SPARSE	0.37 ± 0.03	0.37 ± 0.02	0.29 ± 0.02	1.98 ± 0.07	3.13 ± 0.17
	ELIGN$_{\text{self}}$	0.36 ± 0.04	0.39 ± 0.00	0.27 ± 0.01	1.98 ± 0.08	3.25 ± 0.12

Table 21: We sample agents from different *decentralized* training runs and evaluate their zero-shot performance in *scaled* environments under partial observability. We report the mean test episode extrinsic rewards and standard errors of decentralized methods with different intrinsic rewards.

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (5v0)**	**Hetero nav. (6v0)**	**Phy decep. (4v2)**	**Pred-prey (4v4)**	**Keep-away (4v4)**
Partial observability	SPARSE	434.68 ± 6.42	561.16 ± 31.63	128.64 ± 17.31	-32.12 ± 3.63	-2.80 ± 2.91
	ELIGN$_{\text{self}}$	471.07 ± 5.00	676.01 ± 16.53	248.16 ± 6.62	-16.77 ± 2.25	-5.03 ± 1.06
	ELIGN$_{\text{team}}$	511.97 ± 6.95	699.56 ± 11.64	190.06 ± 29.10	-19.40 ± 2.86	-3.10 ± 3.09
	ELIGN$_{\text{adv}}$	—	—	228.53 ± 25.03	-31.03 ± 3.13	27.24 ± 4.48

Table 22: We sample agents from different *decentralized* training runs and evaluate their zero-shot performance in *scaled* environments under partial observability. We report the average test agent-to-target (agt-target) and agent-to-adversary (agt-adv) distances and standard errors of *decentralized* methods with different intrinsic rewards. Lower scores are better for agt-target (↓), and higher scores are better for agt-adv (↑).

		Cooperative		Competitive		
Task (Agt # vs. Adv #)		**Coop nav. (5v0)** ↓	**Hetero nav. (6v0)** ↓	**Phy decep. (4v2)** ↓	**Pred-prey (4v4)** ↑	**Keep-away (4v4)** ↓
Partial observability	SPARSE	0.22 ± 0.00	0.23 ± 0.00	0.23 ± 0.00	1.93 ± 0.00	3.16 ± 0.01
	ELIGN$_{\text{self}}$	0.19 ± 0.00	0.20 ± 0.00	0.17 ± 0.00	2.33 ± 0.00	3.31 ± 0.01
	ELIGN$_{\text{team}}$	0.43 ± 0.01	0.19 ± 0.00	0.24 ± 0.00	2.04 ± 0.01	3.15 ± 0.01
	ELIGN$_{\text{adv}}$	—	—	0.21 ± 0.00	2.11 ± 0.01	2.31 ± 0.01

Table 23: Model and training hyperparameters

Parameter	Multi-agent particle	Google Research football
SAC actor model architecture	FC layers [128,128]	FC layers [256,256]
SAC critic model architecture	FC layers [128,128]	FC layers [256,256]
World model architecture	FC layers [128,128]	FC layers [128,128]
Replay buffer size	1,000,000	1,000,000
Batch size	1,024	256
Actor learning rate	0.001	0.0003
Critic learning rate	0.001	0.0003
Discount factor gamma	0.95	0.99
SAC soft update coefficient	0.01	0.005
SAC policy entropy regularization coefficient	0.1	1.0 (initial)