A Cordial Sync: Going Beyond Marginal Policies for Multi-Agent Embodied Tasks

Unnat Jain[1⋆], Luca Weihs[2⋆], Eric Kolve[2], Ali Farhadi[3], Svetlana Lazebnik[1], Aniruddha Kembhavi[2,3], and Alexander Schwing[1]

[1] University of Illinois at Urbana-Champaign
[2] Allen Institute for AI [3] University of Washington

Abstract. Autonomous agents must learn to collaborate. It is not scalable to develop a new centralized agent every time a task's difficulty outpaces a single agent's abilities. While multi-agent collaboration research has flourished in gridworld-like environments, relatively little work has considered visually rich domains. Addressing this, we introduce the novel task FurnMove in which agents work together to move a piece of furniture through a living room to a goal. Unlike existing tasks, FurnMove requires agents to coordinate at every timestep. We identify two challenges when training agents to complete FurnMove: existing decentralized action sampling procedures do not permit expressive joint action policies and, in tasks requiring close coordination, the number of failed actions dominates successful actions. To confront these challenges we introduce SYNC-policies (synchronize your actions coherently) and CORDIAL (coordination loss). Using SYNC-policies and CORDIAL, our agents achieve a 58% completion rate on FurnMove, an impressive absolute gain of 25 percentage points over competitive decentralized baselines. Our dataset, code, and pretrained models are available at https://unnat.github.io/cordial-sync.

Keywords: Embodied agents, multi-agent reinforcement learning, collaboration, emergent communication, AI2-THOR

1 Introduction

Progress towards enabling artificial embodied agents to learn collaborative strategies is still in its infancy. Prior work mostly studies collaborative agents in gridworld like environments. Visual, multi-agent, collaborative tasks have not been studied until very recently [23,41]. While existing tasks are well designed to study some aspects of collaboration, they often don't require agents to closely collaborate *throughout* the task. Instead such tasks either require initial coordination (distributing tasks) followed by almost independent execution, or collaboration at a task's end (*e.g.*, verifying completion). Few tasks require frequent coordination, and we are aware of none within a visual setting.

⋆ denotes equal contribution by UJ and LW

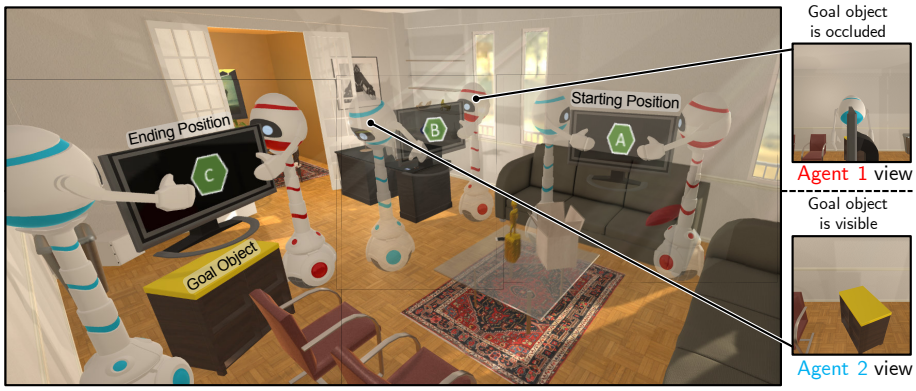


Fig. 1: Two agents communicate and synchronize their actions to move a heavy object through an indoor environment towards a goal. (a) Agents begin holding the object in a randomly chosen location. (b) Given only egocentric views, successful navigation requires agents to communicate their intent to reposition themselves, and the object, while contending with collisions, mutual occlusion, and partial information. (c) Agents successfully moved the object above the goal

To study our algorithmic ability to address tasks which require close and frequent collaboration, we introduce the furniture moving (FurnMove) task (see Fig. 1), set in the AI2-THOR environment. Agents hold a lifted piece of furniture in a living room scene and, given only egocentric visual observations, must collaborate to move it to a visually distinct goal location. As a piece of furniture cannot be moved without both agents agreeing on the direction, agents must explicitly *coordinate at every timestep*. Beyond coordinating actions, FurnMove requires agents to visually anticipate possible collisions, handle occlusion due to obstacles and other agents, and estimate free space. Akin to the challenges faced by a group of roommates relocating a widescreen television, this task necessitates extensive and ongoing coordination amongst all agents at every time step.

In prior work, collaboration between multiple agents has been enabled primarily by (i) sharing observations or (ii) learning low-bandwidth communication. (i) is often implemented using a *centralized* agent, *i.e.*, a single agent with access to all observations from all agents [9,70,89]. While effective it is also unrealistic: the real world poses restrictions on communication bandwidth, latency, and modality. We are interested in the more realistic *decentralized* setting enabled via option (ii). This is often implemented by one or more rounds of message passing between agents before they choose their actions [27,57,41]. Training decentralized agents when faced with FurnMove's requirement of coordination at each timestep leads to two technical challenges. Challenge 1: as each agent independently samples an action from its policy at every timestep, the joint probability tensor of all agents' actions at any given time is rank-one. This severely limits which multi-agent policies are representable. Challenge 2: the number of possible mis-steps or failed actions increases dramatically when requiring that agents closely coordinate with each other, complicating training.

Addressing challenge 1, we introduce SYNC (**S**ynchronize **Y**our actio**N**s **C**oherently) policies which permit expressive (*i.e.*, beyond rank-one) joint policies for decentralized agents while using interpretable communication. To ameliorate challenge 2 we introduce the **Co**o**rdi**n**a**tion **L**oss (CORDIAL) that replaces the standard entropy loss in actor-critic algorithms and guides agents away from actions that are mutually incompatible. A 2-agent system using SYNC and CORDIAL obtains a 58% success rate on test scenes in FurnMove, an impressive absolute gain of 25 percentage points over the baseline from [41] (76% relative gain). In a 3-agent setting, this difference is even more extreme.

In summary, our contributions are: (i) FurnMove, a new multi-agent embodied task that demands ongoing coordination, (ii) SYNC, a collaborative mechanism that permits expressive joint action policies for decentralized agents, (iii) CORDIAL, a training loss for multi-agent setups which, when combined with SYNC, leads to large gains, and (iv) open-source improvements to the AI2-THOR environment including a 16× faster gridworld equivalent for prototyping.

2 Related work

Single-agent embodied systems: Single-agent embodied systems have been considered extensively in the literature. For instance, literature on visual navigation, *i.e.*, locating an object of interest given only visual input, spans geometric and learning based methods. Geometric approaches have been proposed separately for mapping and planning phases of navigation. Methods entailing structure-from-motion and SLAM [87,76,25,13,71,77] were used to build maps. Planning algorithms on existing maps [14,45,51] and combined mapping & planning [26,49,48,30,6] are other related research directions.

While these works propose geometric approaches, the task of navigation can be cast as a reinforcement learning (RL) problem, mapping pixels to policies in an end-to-end manner. RL approaches [67,1,20,33,43,88,61,82] have been proposed to address navigation in synthetic layouts like mazes, arcade games, and other visual environments [96,8,46,53,42,80]. Navigation within photo-realistic environments [11,75,15,47,98,5,35,97,58] led to the development of *embodied* AI agents. The early work [103] addressed object navigation (find an object given an image) in AI2-THOR. Soon after, [35] showed how imitation learning permits agents to learn to build a map from which they navigate. Methods also investigate the utility of topological and latent memory maps [35,74,37,95], graph-based learning [95,99], meta-learning [94], unimodal baselines [86], 3D point clouds [93], and effective exploration [91,74,16,72] to improve embodied navigational agents. Extensions of embodied navigation include instruction following [38,4,78,91,3], city navigation [18,63,62,90], question answering [21,22,34,93,24], and active visual recognition [101,100]. Recently, with visual and acoustic rendering, agents have been trained for audio-visual embodied navigation [19,31].

In contrast to the above single-agent embodied tasks and approaches, we focus on collaboration between multiple embodied agents. Extending the above single-agent architectural novelties (or a combination of them) to multi-agent systems such as ours is an interesting direction for future work.

Non-visual MARL: Multi-agent reinforcement learning (MARL) is challenging due to non-stationarity when learning. Several methods have been proposed to address such issues [84,85,83,29]. For instance, permutation invariant critics have been developed recently [56]. In addition, for MARL, cooperation and competition between agents has been studied [50,69,59,12,68,36,57,28,56]. Similarly, communication and language in the multi-agent setting has been investigated [32,44,10,60,52,27,79,66,7] in maze-based setups, tabular tasks, and Markov games. These algorithms mostly operate on low-dimensional observations (*e.g.*, position, velocity, *etc.*) and top-down occupancy grids. For a survey of centralized and decentralized MARL methods, kindly refer to [102]. Our work differs from the aforementioned MARL works in that we consider complex visual environments. Our contribution of SYNC-Policies is largely orthogonal to RL loss function or method. For a fair comparison to [41], we used the same RL algorithm (A3C) but it is straightforward to integrate SYNC into other MARL methods [73,28,57] (for details, see Sec. A.3 of the supplement).
Visual MARL: Jain *et al.* [41] introduced a collaborative task for two embodied visual agents, which we refer to as FurnLift. In this task, two agents are randomly initialized in an AI2-THOR living room scene, must visually navigate to a TV, and, in a singe coordinated PickUp action, work to lift that TV up. FurnLift doesn't demand that agents coordinate their actions at each timestep. Instead, such coordination only occurs at the last timestep of an episode. Moreover, as success of an action executed by an agent is independent (with the exception of the PickUp action), a high performance joint policy need not be complex, *i.e.*, it may be near low-rank. More details on this analysis and the complexity of our proposed FurnMove task are provided in Sec. 3. Similarly, Chen *et al.* [17] proposes a visual hide-and-seek task, where agents can move independently. Das *et al.* [23] enable agents to learn who to communicate with, on predominantly 2D tasks. In visual environments they study the task where multiple agents jointly navigate to the same object. Jaderberg *et al.* [40] recently studied the game of Quake III and Weihs *et al.* [92] develop agents to play an adversarial hiding game in AI2-THOR. Collaborative perception for semantic segmentation and recognition classification have also been investigated [54,55].

To the best of our knowledge, all prior work in decentralized MARL uses a single marginal probability distribution per agent, *i.e.*, a rank-1 joint distribution. Moreover, FurnMove is the first multi-agent collaborative task in a visually rich domain requiring close coordination between agents at every timestep.

3 The furniture moving task (FurnMove)

We describe our new multi-agent task FurnMove, grounded in the real-world experience of moving furniture. We begin by introducing notation.
RL background and notation. Consider $N \geq 1$ collaborative agents $A^1, \ldots, A^N$. At every timestep $t \in \mathbb{N} = \{0, 1, \ldots\}$ the agents, and environment, are in some state $s_t \in \mathcal{S}$ and each agent A^i obtains an observation o_t^i recording some partial information about s_t. For instance, o_t^i might be the egocentric visual view of an agent A^i embedded in some simulated environment. From observation o_t^i

and history h^i_{t-1}, which records prior observations and decisions made by the agent, each agent A^i forms a policy $\pi^i_t : \mathcal{A} \to [0, 1]$ where $\pi^i_t(a)$ is the probability that agent A^i chooses to take action $a \in \mathcal{A}$ from a finite set of options $\mathcal{A}$ at time t. After the agents execute their respective actions $(a^1_t, \ldots, a^N_t)$, which we call a *multi-action*, they enter a new state s_{t+1} and receive individual rewards $r^1_t, \ldots, r^N_t \in \mathbb{R}$. For more on RL see [81,64,65].

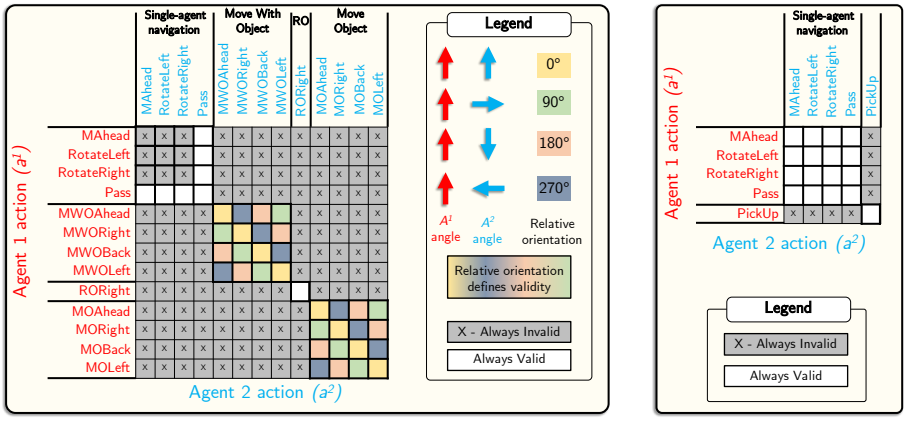
Task definition. FurnMove is set in the near-photorealistic and physics-enabled simulated environment AI2-THOR [47]. In FurnMove, N agents collaborate to move a lifted object through an indoor environment with the goal of placing this object above a visually distinct target as illustrated in Fig. 1. Akin to humans moving large items, agents must navigate around other furniture and frequently walk in-between obstacles on the floor.

In FurnMove, each agent at every timestep receives an egocentric observation (a 3×84×84 RGB image) from AI2-THOR. In addition, agents are allowed to communicate with other agents at each timestep via a low-bandwidth communication channel. Based on their local observation and communication, each agent executes an action from the set $\mathcal{A}$. The space of actions $\mathcal{A} = \mathcal{A}^{\text{NAV}} \cup \mathcal{A}^{\text{MWO}} \cup \mathcal{A}^{\text{MO}} \cup \mathcal{A}^{\text{RO}}$ available to an agent is comprised of the four single-agent navigational actions $\mathcal{A}^{\text{NAV}} = \{$MoveAhead, RotateLeft, RotateRight, Pass$\}$ used to move the agent independently, four actions $\mathcal{A}^{\text{MWO}} = \{$MoveWith ObjectX $\mid X \in \{$Ahead, Right, Left, Back$\}\}$ used to move the lifted object and the agents simultaneously in the same direction, four actions $\mathcal{A}^{\text{MO}} = \{$MoveObjectX$\mid X \in \{$Ahead, Right, Left, Back$\}\}$ used to move the lifted object while the agents stay in place, and a single action used to rotate the lifted object clockwise $\mathcal{A}^{\text{RO}} = \{$RotateObjectRight$\}$. We assume that all movement actions for agents and the lifted object result in a displacement of 0.25 meters (similar to [41,58]) and all rotation actions result in a rotation of 90 degrees (counter-)clockwise when viewing the agents from above.

Close and on-going collaboration is required in FurnMove due to restrictions on the set of actions which can be successfully completed jointly by all the agents. These restrictions reflect physical constraints: for instance, if two people attempt to move in opposite directions while carrying a heavy object they will either fail to move or drop the object. For two agents, we summarize these restrictions using the *coordination matrix* shown in Fig. 2a. For comparison, we include a similar matrix in Fig. 2b corresponding to the FurnLift task from [41]. We defer a more detailed discussion of these restrictions to Sec. A.1 of the supplement. Generalizing the coordination matrix shown in Fig. 2a, at every timestep t we let S_t be the $\{0, 1\}$-valued $|\mathcal{A}|^N$-dimensional tensor where $(S_t)_{i_1,\ldots,i_N} = 1$ if and only if the agents are configured such that multi-action $(a^{i_1}, \ldots, a^{i_N})$ satisfies the restrictions detailed in Sec. A.1. If $(S_t)_{i_1,\ldots,i_N} = 1$ we say the actions $(a^{i_1}, \ldots, a^{i_N})$ are *coordinated*.

3.1 Technical challenges

As we show in our experiments in Sec. 6, standard communication-based models similar to the ones proposed in [41] perform rather poorly when trained to



(a) FurnMove (b) FurnLift

Fig. 2: **Coordination matrix for tasks.** The matrix S_t records the validity of multi-action (a^1, a^2) for different relative orientations of agents A^1 & A^2. (a) S_t for all 4 relative orientations of two agents, for FurnMove. Only $16/169 = 9.5\%$ of multi-actions are coordinated at any given relative orientation, (b) FurnLift where single agent actions are always valid and coordination is needed only for PickUp action, i.e. at least $16/25 = 64\%$ actions are always valid.

complete the FurnMove task. In the following we identify two key challenges that contribute to this poor performance.

Challenge 1: rank-one joint policies. In classical multi-agent work [12,69,57], each agent A^i samples its action $a_t^i \sim \pi_t^i$ independently of all other agents. Due to this independent sampling, at time t, the probability of the agents taking multi-action $(a^1, \ldots, a^N)$ equals $\prod_{i=1}^N \pi_t^i(a^i)$. This means that the joint probability tensor of all actions at time t can be written as the rank-one tensor $\Pi_t = \pi_t^1 \otimes \cdots \otimes \pi_t^N$. This rank-one constraint limits the joint policy that can be executed by the agents, which has real impact. Sec. A.2 considers two agents playing rock-paper-scissors with an adversary: the rank-one constraint reduces the expected reward achieved by an optimal policy from 0 to -0.657 (minimal reward being -1). Intuitively, a high-rank joint policy is not well approximated by a rank-one probability tensor obtained via independent sampling.

Challenge 2: exponential failed actions. The number of possible multi-actions $|\mathcal{A}|^N$ increases exponentially as the number of agents N grows. While this is not problematic if agents act relatively independently, it's a significant obstacle when the agents are *tightly coupled*, *i.e.*, when the success of agent A^i's action a^i is highly dependent on the actions of the other agents. Just consider a randomly initialized policy (the starting point of almost all RL problems): agents stumble upon positive rewards with an extremely low probability which leads to slow learning. We focus on small N, nonetheless, the proportion of coordinated action tuples is small (9.5% when $N = 2$ and 2.1% when $N = 3$).

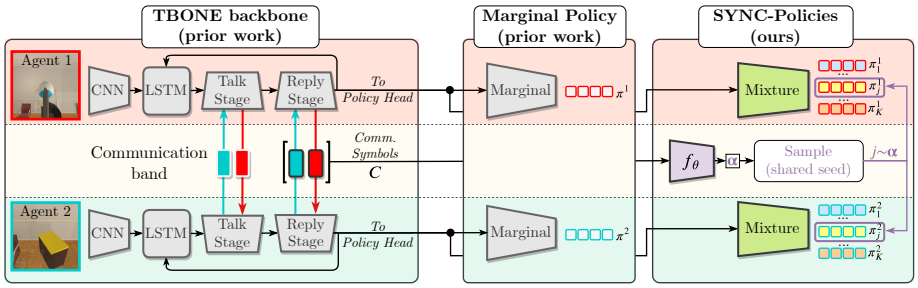


Fig. 3: Model overview for 2 agents in the decentralized setting. *Left*: all decentralized methods in this paper have the same TBONE [41] backbone. *Right*: marginal vs SYNC-policies. With marginal policies, the standard in prior work, each agent constructs its own policy and independently samples from it. With SYNC-policies, agents communicate to construct a distribution α over multiple "strategies" which they then sample from using a shared random seed

4 A cordial sync

To address the above challenges we develop: (a) a novel action sampling procedure named **S**ynchronize **Y**our actio**N**s **C**oherently (SYNC) and (b) an intuitive & effective multi-agent training loss named the **Co**ordi**n**ation **L**oss (CORDIAL).
Addressing challenge 1: SYNC. For concreteness we let $N = 2$, so the joint probability tensor Π_t is matrix of size $|\mathcal{A}|\times|\mathcal{A}|$, and provide an overview in Fig. 3. Recall our goal: using little communication, multiple agents should sample their actions from a high-rank joint policy. This is difficult as (i) little communication means that, except in degenerate cases, no agent can form the full joint policy and (ii) even if all agents had access to the joint policy it is not obvious how to ensure that the decentralized agents will sample a valid coordinated action.

To achieve our goal recall that, for any rank $m \leq |\mathcal{A}|$ matrix $L \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$, there are vectors $v_1, w_1, \ldots, v_m, w_m \in \mathbb{R}^{|\mathcal{A}|}$ such that $L = \sum_{j=1}^{m} v_j \otimes w_j$. Here, $\otimes$ denotes the outer product. Also, the *non-negative rank* of a matrix $L \in \mathbb{R}_{\geq 0}^{|\mathcal{A}|\times|\mathcal{A}|}$ equals the smallest integer s such that L can be written as the sum of s non-negative rank-one matrices. A non-negative matrix $L \in \mathbb{R}_{\geq 0}^{|\mathcal{A}|\times|\mathcal{A}|}$ has non-negative rank bounded above by $|\mathcal{A}|$. Since Π_t is a $|\mathcal{A}|\times|\mathcal{A}|$ joint probability matrix, *i.e.*, Π_t is non-negative and its entries sum to one, it has non-negative rank $m \leq |\mathcal{A}|$, *i.e.*, there exist non-negative vectors $\alpha \in \mathbb{R}_{\geq 0}^{m}$ and $p_1, q_1, \ldots, p_m, q_m \in \mathbb{R}_{\geq 0}^{|\mathcal{A}|}$ whose entries sum to one such that $\Pi_t = \sum_{j=1}^{m} \alpha_j \cdot p_j \otimes q_j$. We call a sum of the form $\sum_{j=1}^{m} \alpha_j \cdot p_j \otimes q_j$ a *mixture-of-marginals*. With this decomposition at hand, randomly sampling action pairs (a^1, a^2) from $\sum_{j=1}^{m} \alpha_j \cdot p_j \otimes q_j$ can be interpreted as a two-step process: first sample an index $j \sim \text{Multinomial}(\alpha)$ and then sample $a^1 \sim \text{Multinomial}(p_j)$ and $a^2 \sim \text{Multinomial}(q_j)$.

This stage-wise procedure suggests a strategy for sampling actions in a multi-agent setting, which we refer to as SYNC-*policies*. Generalizing to an N agent setup, suppose that agents $(A^i)_{i=1}^{N}$ have access to a shared random stream of

numbers. This can be accomplished if all agents share a random seed or if all agents initially communicate their individual random seeds and sum them to obtain a shared seed. Furthermore, suppose that all agents locally store a shared function $f_\theta : \mathbb{R}^K \to \Delta_{m-1}$ where θ are learnable parameters, K is the dimensionality of all communication between the agents in a timestep, and Δ_{m-1} is the standard $(m-1)$-probability simplex. Finally, at time t suppose that each agent A^i produces not a single policy π_t^i but instead a collection of policies $\pi_{t,1}^i, \ldots, \pi_{t,m}^i$. Let $C_t \in \mathbb{R}^K$ be all communication sent between agents at time t. Each agent A^i then samples its action as follows: (i) compute the shared probabilities $\alpha_t = f_\theta(C_t)$, (ii) sample an index $j \sim \text{Multinomial}(\alpha_t)$ using the shared random number stream, (iii) sample, independently, an action a^i from the policy $\pi_{t,j}^i$. Since both f_θ and the random number stream are shared, the quantities in (i) and (ii) are equal across all agents despite being computed individually. This sampling procedure is equivalent to sampling from the tensor $\sum_{j=1}^m \alpha_j \cdot \pi_{t,j}^1 \otimes \ldots \otimes \pi_{t,j}^N$ which, as discussed above, may have rank up to m. Intuitively, SYNC enables decentralized agents to have a more expressive joint policy by allowing them to agree upon a strategy by sampling from α_t.

Addressing challenge 2: CORDIAL**.** We encourage agents to rapidly learn to choose coordinated actions via a new loss. In particular, letting Π_t be the joint policy of our agents, we propose the *coordination loss* (CORDIAL)

$$\text{CL}_\beta(S_t, \Pi_t) = -\beta \cdot \langle S_t, \log(\Pi_t) \rangle \;/\; \langle S_t, S_t \rangle, \tag{1}$$

where log is applied element-wise, $\langle *, * \rangle$ is the usual Frobenius inner product, and S_t is defined in Sec. 3. CORDIAL encourages agents to have a near uniform policy over the actions which are coordinated. We use this loss to replace the standard entropy encouraging loss in policy gradient algorithms (*e.g.*, the A3C algorithm [65]). Similarly to the parameter for the entropy loss in A3C, β is chosen to be a small positive constant so as to not overly discourage learning.

The coordination loss is less meaningful when $\Pi_t = \pi^1 \otimes \cdots \otimes \pi^N$, *i.e.*, when Π_t is rank-one. For instance, suppose that S_t has ones along the diagonal, and zeros elsewhere, so that we wish to encourage the agents to all take the same action. In this case it is straightforward to show that $\text{CL}_\beta(S_t, \Pi_t) = -\beta \sum_{i=1}^N \sum_{j=1}^M (1/M) \log \pi_t^i(a^j)$ so that $\text{CL}_\beta(S_t, \Pi_t)$ simply encourages each agent to have a uniform distribution over its actions and thus actually encourages the agents to place a large amount of probability mass on uncoordinated actions. Indeed, Tab. 4 shows that using CORDIAL without SYNC leads to poor results.

5 Models

We study four distinct policy types: *central*, *marginal*, *marginal w/o comm*, and *SYNC*. *Central* samples actions from a joint policy generated by a central agent with access to observations from all agents. While often unrealistic in practice due to communication bottlenecks, *central* serves as an informative baseline. *Marginal* follows prior work, *e.g.*, [41]: each agent independently samples its actions from its individual policy after communication. *Marginal w/o comm*

is identical to *marginal* but does not permit agents to communicate explicitly (agents may still see each other). Finally, *SYNC* is our newly proposed policy described in Sec. 4. For a fair comparison, all decentralized agents (*i.e.*, *SYNC*, *marginal*, and *marginal w/o comm*), use the same TBONE backbone architecture from [41], see Fig. 3. We have ensured that parameters are fairly balanced so that our proposed *SYNC* has close to (and never more) parameters than the *marginal* and *marginal w/o comm* nets. We train *central* and *SYNC* with CORDIAL, and the *marginal* and *marginal w/o comm* without it. This choice is mathematically explained in Sec. 4 and empirically validated in Sec. 6.3.

Architecture: For clarity, we describe the policy and value net for the 2 agent setup, extending to any number of agents is straightforward. Decentralized agents use the TBONE backbone from [41]. Our primary architectural novelty extends TBONE to SYNC-policies. An overview of the TBONE backbone and differences between sampling with *marginal* and *SYNC* policies is shown in Fig. 3.

As a brief summary of TBONE, agent i observes at time t inputs o_t^i, *i.e.*, a 3×84×84 RGB image returned from AI2-THOR which represents the i-th agent's egocentric view. Each o_t^i is encoded by a 4-layer CNN and combined with an agent-specific learned embedding (encoding the agent's ID) along with the history embedding h_{t-1}^i. The resulting vector is fed into an LSTM [39] unit to produce a 512-dimensional embedding $\tilde{h}_t^i$ corresponding to the i^{th} agent. The agents then undergo two rounds of communication resulting in two final hidden states h_t^1, h_t^2 and messages $c_{t,j}^i \in \mathbb{R}^{16}$, $1 \leq i, j \leq 2$ with message $c_{t,j}^i$ being produced by agent i in round j and then sent to the other agent in that round. In [41], the value of the agents' state as well as logits corresponding to the policy of the agents are formed by applying linear functions to h_t^1, h_t^2.

We now show how SYNC can be integrated into TBONE to allow our agents to represent high-rank joint distributions over multi-actions (see Fig. 3). First each agent computes the logits corresponding to α_t. This is done using a 2-layer MLP applied to the messages sent between the agents, at the second stage. In particular, $\alpha_t = \mathbf{W}_3\ \text{ReLU}(\mathbf{W}_2\ \text{ReLU}(\mathbf{W}_1\ [c_{t,2}^1; c_{t,2}^2] + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3$ where $\mathbf{W}_1 \in \mathbb{R}^{64\times 32}, \mathbf{W}_2 \in \mathbb{R}^{64\times 64}$, $\mathbf{W}_3 \in \mathbb{R}^{m\times 64}$, $\mathbf{b}_1 \in \mathbb{R}^{32}, \mathbf{b}_2 \in \mathbb{R}^{64}$, and $\mathbf{b}_3 \in \mathbb{R}^m$ are a learnable collection of weight matrices and biases. After computing α_t we compute a collection of policies $\pi_{t,1}^i, \ldots, \pi_{t,m}^i$ for $i \in \{1, 2\}$. Each of these policies is computed following the TBONE architecture but using $m-1$ additional, and learnable, linear layers per agent.

6 Experiments

6.1 Experimental setup

Simulator. We evaluate our models in the AI2-THOR environment [47] with several novel upgrades including support for initializing lifted furniture and a top-down gridworld version of AI2-THOR for faster prototyping (16× faster than [41]). For details about framework upgrades, see Sec. A.3 of the supplement.

Tasks. We compare against baselines on FurnMove, Gridworld-FurnMove, and FurnLift [41]. FurnMove is the novel task introduced in this work

(Sec. 3): agents observe egocentric visual views (90° field-of-view). In Gridworld-FurnMove the agents are provided a top-down egocentric 3D tensor as observations. The third dimension of the tensor contains semantic information such as, if the location is navigable by an agent or navigable by the lifted object, or whether the location is occupied by another agent, the lifted object, or the goal object. Hence, Gridworld-FurnMove agents do not need visual understanding, but face other challenges of the FurnMove task – coordinating actions and planning trajectories. We consider only the harder variant of FurnLift, where communication was shown to be most important ('constrained' with no implicit communication in [41]). In FurnLift, agents observe egocentric visual views.

Data. As in [41], we train and evaluate on a split of the 30 living room scenes. As FurnMove is already quite challenging, we only consider a single piece of lifted furniture (a television) and a single goal object (a TV-stand). Twenty rooms are used for training, 5 for validation, and 5 for testing. The test scenes have very different lighting conditions, furniture, and layouts. For evaluation our test set includes 1000 episodes equally distributed over the five scenes.

Training. For training we augment the A3C algorithm [65] with CORDIAL. For our studies in the visual domain, we use 45 workers and 8 GPUs. Models take around two days to train. For more details, see Sec. A.3 of the supplement.

6.2 Metrics

For completeness, we consider a variety of metrics. We adapt SPL, *i.e.*, Success weighted by (normalized inverse) Path Length [2], so that it doesn't require shortest paths but still provides similar semantic information[4]: We define a Manhattan Distance based SPL as $\textit{MD-SPL} = N_{\text{ep}}^{-1} \sum_{i=1}^{N_{\text{ep}}} S_i \frac{m_i / d_{\text{grid}}}{\max(p_i, m_i / d_{\text{grid}})}$, where i denotes an index over episodes, N_{ep} equals the number of test episodes, and S_i is a binary indicator for success of episode i. Also p_i is the number of actions taken per agent, m_i is the Manhattan distance from the lifted object's start location to the goal, and d_{grid} is the distance between adjacent grid points, for us 0.25m. We also report other metrics capturing complementary information. These include mean number of actions in an episode per agent (*Ep len*), success rate (*Success*), and distance to target at the end of the episode (*Final dist*).

We also introduce two metrics unique to coordinating actions: *TVD*, the mean total variation distance between Π_t and its best rank-one approximation, and *Invalid prob*, the average probability mass allotted to uncoordinated actions, *i.e.*, the dot product between $1 - S_t$ and Π_t. By definition, *TVD* is zero for the *marginal* model, and higher values indicate divergence from independent marginal sampling. Without measuring *TVD* we would have no way of knowing if our SYNC model was actually using the extra expressivity we've afforded it. Lower *Invalid prob* values imply an improved ability to avoid uncoordination actions as detailed in Sec. 3 and Fig. 2.

[4] For FurnMove, each location of the lifted furniture corresponds to 404, 480 states, making shortest path computation intractable (more details in Sec. A.4).

Table 1: Quantitative results on three tasks. ↑ (or ↓) indicates that higher (or lower) value of the metric is desirable while ↕ denotes that no value is, a priori, better than another. †denotes that a centralized agent serves only as an upper bound to decentralized methods and cannot be fairly compared with. Among decentralized agents, our SYNC model has the best metric values across all reported metrics (**bolded** values). Values are highlighted in green if their 95% confidence interval has no overlap with the confidence intervals of other values

Methods	MD-SPL ↑	Success ↑	Ep len ↓	Final dist ↓	Invalid prob. ↓	TVD ↕
FurnMove (ours)						
Marginal w/o comm [41]	0.032	0.164	224.1	2.143	0.815	0
Marginal [41]	0.064	0.328	194.6	1.828	0.647	0
SYNC	**0.114**	**0.587**	**153.5**	**1.153**	**0.31**	0.474
Central†	0.161	0.648	139.8	0.903	0.075	0.543
Gridworld-FurnMove (ours)						
Marginal w/o comm [41]	0.111	0.484	172.6	1.525	0.73	0
Marginal [41]	0.218	0.694	120.1	0.960	0.399	0
SYNC	**0.228**	**0.762**	**110.4**	**0.711**	**0.275**	0.429
Central†	0.323	0.818	87.7	0.611	0.039	0.347
Gridworld-FurnMove-3Agents (ours)						
Marginal [41]	0	0	250.0	3.564	0.823	0
SYNC	**0.152**	**0.578**	**149.1**	**1.05**	**0.181**	0.514
Central†	0.066	0.352	195.4	1.522	0.138	0.521

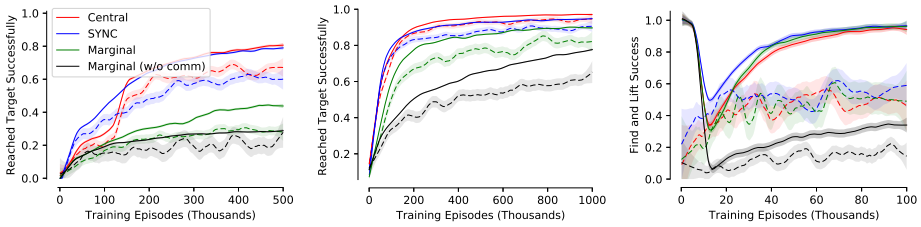
Table 2: Quantitative results on the FurnLift task. For legend, see Tab. 1

Methods	MD-SPL ↑	Success ↑	Ep len ↓	Final dist ↓	Invalid prob. ↓	TVD ↕	Failed pickups ↓	Missed pickups ↓
FurnLift [41] ('constrained' setting with no implicit communication)								
Marginal w/o comm [41]	0.029	0.15	229.5	2.455	0.11	0	25.219	6.501
Marginal [41]	**0.145**	**0.449**	**174.1**	2.259	0.042	0	8.933	1.426
SYNC	0.139	0.423	176.9	**2.228**	**0**	0.027	**4.873**	**1.048**
Central†	0.145	0.453	172.3	2.331	0	0.059	5.145	0.639

6.3 Quantitative evaluation

We conduct four studies: (a) performance of different methods and relative difficulty of the three tasks, (b) effect of number of components on SYNC performance, (c) effect of CORDIAL (ablation), and (d) effect of number of agents.

Comparing methods and tasks. We compare models detailed in Sec. 5 on tasks of varying difficulty, report metrics in Tab. 1, and show the progress of metrics over training episodes in Fig. 4. In our FurnMove experiments, *SYNC* performs better than the best performing method of [41] (*i.e.*, *marginal*) on all metrics. Success rate increases by 25.9% and 6.8% absolute percentage points on FurnMove and Gridworld-FurnMove respectively. Importantly, SYNC is significantly better at allowing agents to coordinate their actions: for FurnMove, the joint policy of SYNC assigns, on average, 0.31 probability mass to invalid actions pairs while the *marginal* and *marginal w/o comm* models assign 0.647 and 0.815 probability mass to invalid action pairs. Additionally, SYNC goes beyond rank-one *marginal* methods by capturing a more expressive joint policy using the mixture of marginals. This is evidenced by the high TVD of 0.474 *vs.* 0 for *marginal*. In Gridworld-FurnMove, oracle-perception of a 2D gridworld helps raise performance of all methods, though the trends are similar. Tab. 2 shows similar trends for FurnLift but, perhaps surprisingly, the *Success* of



(a) FurnMove (b) Gridworld-FurnMove (c) FurnLift

Fig. 4: **Success rate during training.** Train (solid lines) and validation (dashed lines) performance of our agents for FurnMove, Gridworld-FurnMove, and FurnLift (with 95% confidence intervals). For additional plots, see Sec. A.4

Table 3: Effect of number of mixture components m on *SYNC*'s performance (in FurnMove). Generally, larger m means better performance and larger *TVD*.

K in SYNC	MD-SPL ↑	Success ↑	Ep len ↓	Final dist ↓	Invalid prob. ↓	TVD ↕
FurnMove						
1 component	0.064	0.328	194.6	1.828	0.647	0
2 components	0.084	0.502	175.5	1.227	**0.308**	0.206
4 components	0.114	0.569	154.1	**1.078**	0.339	0.421
13 components	**0.114**	**0.587**	**153.5**	1.153	0.31	0.474

SYNC is somewhat lower than the *marginal* model (2.6% lower, within statistical error). As is emphasized in [41] however, *Success* alone is a poor measure of model performance: equally important are the *failed pickups* and *missed pickups* metrics (for details, see Sec. A.4 of the supplement). For these metrics, *SYNC* outperforms the *marginal* model. That *SYNC* does not completely outperform *marginal* in FurnLift is intuitive, as FurnLift does not require continuous close coordination the benefits of *SYNC* are less pronounced.

While the difficulty of a task is hard to quantify, we will consider the relative test-set metrics of agents on various tasks as an informative proxy. Replacing the complex egocentric vision in FurnMove with the semantic 2D gridworld in Gridworld-FurnMove, we see that all agents show large gains in *Success* and *MD-SPL*, suggesting that Gridworld-FurnMove is a dramatic simplification of FurnMove. Comparing FurnMove to FurnLift is particularly interesting: the *MD-SPL* and *Success* metrics for the *central* agent do not provide a clear picture of relative task difficulty. However, the much higher *TVD* for the *central* agent for FurnMove and the superior *MD-SPL* and *Success* of the *Marginal* agents for FurnLift suggest that FurnMove requires more coordination and more expressive joint policies than FurnLift.

Effect of number of mixture components in SYNC**.** Recall (Sec. 4) that the number of mixture components m in SYNC is a hyperparameter controlling the maximal rank of the joint policy. *SYNC* with $m = 1$ is equivalent to *marginal*. In Tab. 3 we see *TVD* increase from 0.206 to 0.474 when increasing m from 2 to 13. This suggests that SYNC learns to use the additional expressivity. Moreover, we see that this increased expressivity results in better performance. A

Table 4: Ablation study of CORDIAL on *marginal* [41], *SYNC*, and *central* methods. *Marginal* performs better without CORDIAL whereas *SYNC* and *central* show improvement with CORDIAL. For legend, see Tab. 1

Method	CORDIAL	MD-SPL ↑	Success ↑	Ep len ↓	Final dist ↓	Invalid prob. ↓	TVD ↕
			FurnMove				
Marginal	✗	0.064	0.328	194.6	1.828	0.647	0
Marginal	✓	0.015	0.099	236.9	2.134	0.492	0
SYNC	✗	0.091	0.488	170.3	1.458	0.47	0.36
SYNC	✓	**0.114**	**0.587**	**153.5**	**1.153**	**0.31**	0.474
Central†	✗	0.14	0.609	146.9	1.018	0.155	0.6245
Central†	✓	0.161	0.648	139.8	0.903	0.075	0.543

success rate jump of 17.4% from $m = 1$ to $m = 2$ demonstrates that substantial benefits are obtained by even small increases in expressitivity. Moreover with more components, *i.e.*, $m = 4$ & $m = 13$ we obtain more improvements. There are, however, diminishing returns with the $m = 4$ model performing nearly as well as the $m = 13$ model. This suggests a trade-off between the benefits of expressivity and the increasing complexities in optimization.

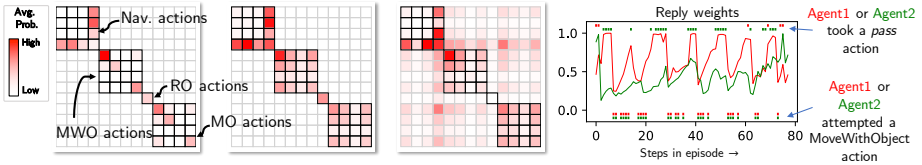
Effect of CORDIAL**.** In Tab. 4 we quantify the effect of CORDIAL. When adding CORDIAL to *SYNC* we obtain a 9.9% improvement in success rate. This is accompanied by a drop in *Invalid prob.* from 0.47 to 0.31, which signifies better coordination of actions. Similar improvements are seen for the *central* model. In 'Challenge 2' (Sec. 4) we mathematically laid out why *marginal* models gain little from CORDIAL. We substantiate this empirically with a 22.9% drop in success rate when training the *marginal* model with CORDIAL.

Effect of more agents. The final three rows of Tab. 1 show the test-set performance of *SYNC*, *marginal*, and *central* models trained to accomplish a 3-agent variant of our Gridworld-FurnMove task. In this task the *marginal* model fails to train at all, achieving a 0% success rate. *SYNC*, on the other hand, successfully completes the task 57.8% of the time. *SYNC*'s success rate drops by nearly 20 percentage points when moving from the 2- to the 3-agent variant of the task: clearly increasing the number of agents substantially increases the task's difficult. Surprisingly, the *central* model performs worse than *SYNC* in this setting. A discussion of this phenomenon is deferred to Supp. Sec. A.4.

6.4 Qualitative evaluation

We present three qualitative results on FurnMove: joint policy summaries, analysis of learnt communication, and visualizations of agent trajectories.

Joint policy summaries. In Fig. 5 we show summaries of the joint policy captured by the *central*, *SYNC*, and *marginal* models. These matrices average over action steps in the test-set episodes for FurnMove. Other tasks show similar trends, see Sec. A.5 of the supplement. In Fig. 5a, the sub-matrices corresponding to $\mathcal{A}^{\text{MWO}}$ and $\mathcal{A}^{\text{MO}}$ are diagonal-dominant, indicating that agents are looking in the same direction (0° relative orientation in Fig. 2). Also note the high probability associated to (Pass, RotateX) and (RotateX, Pass),



a. Central *(upper bound)* b. SYNC *(ours)* c. Marginal *(prior work)* d. Communication analysis of reply stage

Fig. 5: **Qualitative results.** (a,b,c) joint policy summary (Π_t averaged over steps in test episodes in FurnMove) and (d) communication analysis.

within the $\mathcal{A}^{\text{NAV}}$ block. Together, this means that the *central* method learns to coordinate single-agent navigational actions to rotate one of the agents (while the other holds the TV by executing Pass) until both face the same direction. They then execute the same action from $\mathcal{A}^{\text{MO}}$ ($\mathcal{A}^{\text{MWO}}$) to move the lifted object. Comparing Fig. 5b *vs*. Fig. 5c, shows the effect of CORDIAL. Recall that the *marginal* model doesn't support CORDIAL and thus suffers by assigning probability to invalid action pairs (color outside the block-diagonal submatrices). The banded nature of Fig. 5c suggesting agents frequently fail to coordinate.

Communication analysis. A qualitative discussion of communication follows. Agent are colored red and green. We defer a quantitative treatment to Sec. A.5 of the supplement. As we apply SYNC on the TBONE backbone introduced by Jain *et al*. [41], we use similar tools to understand the communication emerging with SYNC policy heads. In line with [41], we plot the weight assigned by each agent to the first communication symbol in the reply stage. Fig. 5d strongly suggests that the reply stage is directly used by the agents to coordinate the modality of actions they intend to take. In particular, the large weight being assigned to the first reply symbol is consistently associated with the other agent taking a Pass action. Similarly, we see that small reply weights coincide with agents taking a MoveWithObject action. The talk weights' interpretation is intertwined with the reply weights, and is deferred to Supp. Sec. A.5.

Agent trajectories. Our supplementary video includes examples of policy rollouts. These clips include both agents' egocentric views and a top-down trajectory visualization. This enables direct comparisons of *marginal* and SYNC on the same test episode. We also allow for hearing patterns in agents' communication: we convert scalar weights (associated with reply symbols) to audio.

7 Conclusion

We introduce FurnMove, a collaborative, visual, multi-agent task requiring close coordination between agents and develop novel methods that allow for moving beyond existing marginal action sampling procedures, these methods lead to large gains across a diverse suite of metrics.

Acknowledgements: This material is based upon work supported in part by the National Science Foundation under Grants No. 1563727, 1718221, 1637479, 165205, 1703166, Samsung, 3M, Sloan Fellowship, NVIDIA Artificial Intelligence Lab, Allen Institute for AI, Amazon, AWS Research Awards, and Siebel Scholars Award. We thank M. Wortsman and K.-H. Zeng for their insightful comments.

References

1. Abel, D., Agarwal, A., Diaz, F., Krishnamurthy, A., Schapire, R.E.: Exploratory gradient boosting for reinforcement learning in complex domains. arXiv preprint arXiv:1603.04119 (2016)
2. Anderson, P., Chang, A., Chaplot, D.S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., et al.: On evaluation of embodied navigation agents. arXiv preprint arXiv:1807.06757 (2018)
3. Anderson, P., Shrivastava, A., Parikh, D., Batra, D., Lee, S.: Chasing ghosts: Instruction following as bayesian state tracking. In: NeurIPS (2019)
4. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: CVPR (2018)
5. Armeni, I., Sax, S., Zamir, A.R., Savarese, S.: Joint 2d-3d-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105 (2017)
6. Aydemir, A., Pronobis, A., Göbelbecker, M., Jensfelt, P.: Active visual object search in unknown environments using uncertain semantics. In: IEEE Trans. on Robotics (2013)
7. Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., Mordatch, I.: Emergent tool use from multi-agent autocurricula. arXiv preprint arXiv:1909.07528 (2019)
8. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: An evaluation platform for general agents. J. of Artificial Intelligence Research (2013)
9. Boutilier, C.: Sequential optimality and coordination in multiagent systems. In: IJCAI (1999)
10. Bratman, J., Shvartsman, M., Lewis, R.L., Singh, S.: A new approach to exploring language emergence as boundedly optimal control in the face of environmental and cognitive constraints. In: Proc. Int.'l Conv. on Cognitive Modeling (2010)
11. Brodeur, S., Perez, E., Anand, A., Golemo, F., Celotti, L., Strub, F., Rouat, J., Larochelle, H., Courville, A.: Home: A household multimodal environment. arXiv preprint arXiv:1711.11017 (2017)
12. Busoniu, L., Babuska, R., Schutter, B.D.: A comprehensive survey of multiagent reinforcement learning. In: IEEE Trans. on Systems, Man and Cybernetics (2008)
13. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Trans. on Robotics (2016)
14. Canny, J.: The complexity of robot motion planning. MIT Press (1988)
15. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. In: 3DV (2017)
16. Chaplot, D.S., Gupta, S., Gupta, A., Salakhutdinov, R.: Learning to explore using active neural mapping. In: ICLR (2020)
17. Chen, B., Song, S., Lipson, H., Vondrick, C.: Visual hide and seek. arXiv preprint arXiv:1910.07882 (2019)
18. Chen, H., Suhr, A., Misra, D., Snavely, N., Artzi, Y.: Touchdown: Natural language navigation and spatial reasoning in visual street environments. In: CVPR (2019)
19. Chen*, C., Jain*, U., Schissler, C., Gari, S.V.A., Al-Halah, Z., Ithapu, V.K., Robinson, P., Grauman, K.: Audio-visual embodied navigation. In: ECCV (2020), * equal contribution

20. Daftry, S., Bagnell, J.A., Hebert, M.: Learning transferable policies for monocular reactive mav control. In: Proc. ISER (2016)
21. Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., Batra, D.: Embodied Question Answering. In: CVPR (2018)
22. Das, A., Gkioxari, G., Lee, S., Parikh, D., Batra, D.: Neural Modular Control for Embodied Question Answering. In: ECCV (2018)
23. Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., Pineau, J.: Tarmac: Targeted multi-agent communication. In: ICML (2019)
24. Das*, A., Carnevale*, F., Merzic, H., Rimell, L., Schneider, R., Abramson, J., Hung, A., Ahuja, A., Clark, S., Wayne, G., et al.: Probing emergent semantics in predictive agents via question answering. In: ICML (2020), * equal contribution
25. Dellaert, F., Seitz, S., Thorpe, C., Thrun, S.: Structure from Motion without Correspondence. In: CVPR (2000)
26. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. Computer (1989)
27. Foerster, J.N., Assael, Y.M., de Freitas, N., Whiteson, S.: Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In: NeurIPS (2016)
28. Foerster, J.N., Farquhar, G., Afouras, T., NArdelli, N., Whiteson, S.: Counterfactual Multi-Agent Policy Gradients. In: AAAI (2018)
29. Foerster, J.N., Nardelli, N., Farquhar, G., Torr, P.H.S., Kohli, P., Whiteson, S.: Stabilising experience replay for deep multi-agent reinforcement learning. In: ICML (2017)
30. Fraundorfer, F., Heng, L., Honegger, D., Lee, G.H., Meier, L., Tanskanen, P., Pollefeys, M.: Vision-based autonomous mapping and exploration using a quadrotor mav. In: IROS (2012)
31. Gao, R., Chen, C., Al-Halah, Z., Schissler, C., Grauman, K.: Visualechoes: Spatial image representation learning through echolocation. In: ECCV (2020)
32. Giles, C.L., Jim, K.C.: Learning communication for multi-agent systems. In: Proc. Innovative Concepts for Agent-Based Systems (2002)
33. Giusti, A., Guzzi, J., Cireşan, D.C., He, F.L., Rodríguez, J.P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Di Caro, G., et al.: A machine learning approach to visual perception of forest trails for mobile robots. IEEE Robotics and Automation Letters (2015)
34. Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., Farhadi, A.: IQA: Visual Question Answering in Interactive Environments. In: CVPR (2018)
35. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: CVPR (2018)
36. Gupta, J.K., Egorov, M., Kochenderfer, M.: Cooperative Multi-Agent Control Using Deep Reinforcement Learning. In: AAMAS (2017)
37. Henriques, J.F., Vedaldi, A.: Mapnet: An allocentric spatial memory for mapping environments. In: CVPR (2018)
38. Hill, F., Hermann, K.M., Blunsom, P., Clark, S.: Understanding grounded language learning agents. arXiv preprint arXiv:1710.09867 (2017)
39. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation (1997)
40. Jaderberg, M., Czarnecki, W.M., Dunning, I., Marris, L., Lever, G., Castaneda, A.G., Beattie, C., Rabinowitz, N.C., Morcos, A.S., Ruderman, A., et al.: Human-level performance in 3d multiplayer games with population-based reinforcement learning. Science (2019)

41. Jain*, U., Weihs*, L., Kolve, E., Rastegari, M., Lazebnik, S., Farhadi, A., Schwing, A.G., Kembhavi, A.: Two body problem: Collaborative visual task completion. In: CVPR (2019), * equal contribution
42. Johnson, M., Hofmann, K., Hutton, T., Bignell, D.: The malmo platform for artificial intelligence experimentation. In: IJCAI (2016)
43. Kahn, G., Zhang, T., Levine, S., Abbeel, P.: Plato: Policy learning using adaptive trajectory optimization. In: ICRA (2017)
44. Kasai, T., Tenmoto, H., Kamiya, A.: Learning of communication codes in multi-agent reinforcement learning problem. In: Proc. IEEE Soft Computing in Industrial Applications (2008)
45. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE transactions on Robotics and Automation (1996)
46. Kempka, M., Wydmuch, M., Runc, G., Toczek, J., Jakowski, W.: Vizdoom: A doom-based ai research platform for visual reinforce- ment learning. In: Proc. IEEE Conf. on Computational Intelligence and Games (2016)
47. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon, D., Zhu, Y., Gupta, A., Farhadi, A.: AI2-THOR: an interactive 3d environment for visual AI. arXiv preprint arXiv:1712.05474 (2019)
48. Konolige, K., Bowman, J., Chen, J., Mihelich, P., Calonder, M., Lepetit, V., Fua, P.: View-based maps. Intl. J. of Robotics Research (2010)
49. Kuipers, B., Byun, Y.T.: A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. Robotics and autonomous systems (1991)
50. Lauer, M., Riedmiller, M.: An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In: ICML (2000)
51. Lavalle, S.M., Kuffner, J.J.: Rapidly-exploring random trees: Progress and prospects. Algorithmic and Computational Robotics: New Directions (2000)
52. Lazaridou, A., Peysakhovich, A., Baroni, M.: Multi-agent cooperation and the emergence of (natural) language. In: arXiv preprint arXiv:1612.07182 (2016)
53. Lerer, A., Gross, S., Fergus, R.: Learning physical intuition of block towers by example. In: ICML (2016)
54. Liu, Y.C., Tian, J., Glaser, N., Kira, Z.: When2com: Multi-agent perception via communication graph grouping. In: CVPR (2020)
55. Liu, Y.C., Tian, J., Ma, C.Y., Glaser, N., Kuo, C.W., Kira, Z.: Who2com: Collaborative perception via learnable handshake communication. In: ICRA (2020)
56. Liu*, I.J., Yeh*, R., Schwing, A.G.: PIC: Permutation Invariant Critic for Multi-Agent Deep Reinforcement Learning. In: CoRL (2019), * equal contribution
57. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In: NeurIPS (2017)
58. Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., Batra, D.: Habitat: A Platform for Embodied AI Research. In: ICCV (2019)
59. Matignon, L., Laurent, G.J., Fort-Piat, N.L.: Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In: IROS (2007)
60. Melo, F.S., Spaan, M., Witwicki, S.J.: QueryPOMDP: POMDP-based communication in multiagent systems. In: Eurpoean Workshop on Multi-Agent Systems (2011)

61. Mirowski, P., Pascanu, R., Viola, F., Soyer, H., Ballard, A., Banino, A., Denil, M., Goroshin, R., Sifre, L., Kavukcuoglu, K., et al.: Learning to navigate in complex environments. In: ICLR (2017)
62. Mirowski, P., Banki-Horvath, A., Anderson, K., Teplyashin, D., Hermann, K.M., Malinowski, M., Grimes, M.K., Simonyan, K., Kavukcuoglu, K., Zisserman, A., et al.: The streetlearn environment and dataset. arXiv preprint arXiv:1903.01292 (2019)
63. Mirowski, P., Grimes, M., Malinowski, M., Hermann, K.M., Anderson, K., Teplyashin, D., Simonyan, K., Zisserman, A., Hadsell, R., et al.: Learning to navigate in cities without a map. In: NeurIPS (2018)
64. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature (2015)
65. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: ICML (2016)
66. Mordatch, I., Abbeel, P.: Emergence of Grounded Compositional Language in Multi-Agent Populations. In: AAAI (2018)
67. Oh, J., Chockalingam, V., Singh, S., Lee, H.: Control of memory, active perception, and action in minecraft. In: ICML (2016)
68. Omidshafiei, S., Pazis, J., Amato, C., How, J.P., Vian, J.: Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In: ICML (2017)
69. Panait, L., Luke, S.: Cooperative multi-agent learning: The state of the art. Autonomous Agents and Multi-Agent Systems. In: AAMAS (2005)
70. Peng, P., Wen, Y., Yang, Y., Yuan, Q., Tang, Z., Long, H., Wang, J.: Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. arXiv preprint arXiv:1703.10069 (2017)
71. R. C. Smith, R.C., Cheeseman, P.: On the representation and estimation of spatial uncertainty. Intl. J. Robotics Research (1986)
72. Ramakrishnan, S.K., Jayaraman, D., Grauman, K.: An exploration of embodied visual exploration. arXiv preprint arXiv:2001.02192 (2020)
73. Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., Whiteson, S.: Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In: ICML (2018)
74. Savinov, N., Dosovitskiy, A., Koltun, V.: Semi-parametric topological memory for navigation. In: ICLR (2018)
75. Savva, M., Chang, A.X., Dosovitskiy, A., Funkhouser, T., Koltun, V.: Minos: Multimodal indoor simulator for navigation in complex environments. arXiv preprint arXiv:1712.03931 (2017)
76. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
77. Smith, R.C., Self, M., Cheeseman, P.: Estimating uncertain spatial relationships in robotics. In: UAI (1986)
78. Suhr, A., Yan, C., Schluger, J., Yu, S., Khader, H., Mouallem, M., Zhang, I., Artzi, Y.: Executing instructions in situated collaborative interactions. In: EMNLP (2019)
79. Sukhbaatar, S., Szlam, A., Fergus, R.: Learning multiagent communication with backpropagation. In: NeurIPS (2016)

80. Sukhbaatar, S., Szlam, A., Synnaeve, G., Chintala, S., Fergus, R.: Mazebase: A sandbox for learning from games. arXiv preprint arXiv:1511.07401 (2015)
81. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (1998)
82. Tamar, A., Wu, Y., Thomas, G., Levine, S., Abbeel, P.: Value iteration networks. In: NeurIPS (2016)
83. Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., Aru, J., Vicente, R.: Multiagent cooperation and competition with deep reinforcement learning. In: PloS (2017)
84. Tan, M.: Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In: ICML (1993)
85. Tesauro, G.: Extending q-learning to general adaptive multi-agent systems. In: NeurIPS (2004)
86. Thomason, J., Gordon, D., Bisk, Y.: Shifting the baseline: Single modality performance on visual navigation & qa. In: NAACL (2019)
87. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. IJCV (1992)
88. Toussaint, M.: Learning a world model and planning with a self-organizing, dynamic neural system. In: NeurIPS (2003)
89. Usunier, N., Synnaeve, G., Lin, Z., Chintala, S.: Episodic exploration for deep deterministic policies: An application to starcraft micromanagement tasks. In: ICLR (2016)
90. de Vries, H., Shuster, K., Batra, D., Parikh, D., Weston, J., Kiela, D.: Talk the walk: Navigating new york city through grounded dialogue. arXiv preprint arXiv:1807.03367 (2018)
91. Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.F., Wang, W.Y., Zhang, L.: Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In: CVPR (2019)
92. Weihs, L., Kembhavi, A., Han, W., Herrasti, A., Kolve, E., Schwenk, D., Mottaghi, R., Farhadi, A.: Artificial agents learn flexible visual representations by playing a hiding game. arXiv preprint arXiv:1912.08195 (2019)
93. Wijmans, E., Datta, S., Maksymets, O., Das, A., Gkioxari, G., Lee, S., Essa, I., Parikh, D., Batra, D.: Embodied Question Answering in Photorealistic Environments with Point Cloud Perception. In: CVPR (2019)
94. Wortsman, M., Ehsani, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In: CVPR (2019)
95. Wu, Y., Wu, Y., Tamar, A., Russell, S., Gkioxari, G., Tian, Y.: Bayesian relational memory for semantic visual navigation. ICCV (2019)
96. Wymann, B., Espié, E., Guionneau, C., Dimitrakakis, C., Coulom, R., Sumner, A.: Torcs, the open racing car simulator (2013), http://www.torcs.org
97. Xia, F., Shen, W.B., Li, C., Kasimbeg, P., Tchapmi, M., Toshev, A., Martín-Martín, R., Savarese, S.: Interactive gibson: A benchmark for interactive navigation in cluttered environments. arXiv preprint arXiv:1910.14442 (2019)
98. Xia, F., Zamir, A.R., He, Z., Sax, A., Malik, J., Savarese, S.: Gibson env: Real-world perception for embodied agents. In: CVPR (2018)
99. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Visual curiosity: Learning to ask questions to learn visual recognition. In: CoRL (2018)
100. Yang, J., Ren, Z., Xu, M., Chen, X., Crandall, D., Parikh, D., Batra, D.: Embodied amodal recognition: Learning to move to perceive objects. In: ICCV (2019)

101. Yang, W., Wang, X., Farhadi, A., Gupta, A., Mottaghi, R.: Visual semantic navigation using scene priors. In: ICLR (2018)
102. Zhang, K., Yang, Z., Başar, T.: Multi-agent reinforcement learning: A selective overview of theories and algorithms. arXiv preprint arXiv:1911.10635 (2019)
103. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A.: Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. In: ICRA (2017)