Interpretation of Emergent Communication in Heterogeneous Collaborative Embodied Agents

Shivansh Patel[1]* Saim Wani[2]* Unnat Jain[3]* Alexander Schwing[3]
Svetlana Lazebnik[3] Manolis Savva[1] Angel X. Chang[1]
[1]Simon Fraser University [2]IIT Kanpur [3]UIUC
https://shivanshpatel35.github.io/comon

Abstract

Communication between embodied AI agents has received increasing attention in recent years. Despite its use, it is still unclear whether the learned communication is interpretable and grounded in perception. To study the grounding of emergent forms of communication, we first introduce the collaborative multi-object navigation task 'CoMON.' In this task, an 'oracle agent' has detailed environment information in the form of a map. It communicates with a 'navigator agent' that perceives the environment visually and is tasked to find a sequence of goals. To succeed at the task, effective communication is essential. CoMON hence serves as a basis to study different communication mechanisms between heterogeneous agents, that is, agents with different capabilities and roles. We study two common communication mechanisms and analyze their communication patterns through an egocentric and spatial lens. We show that the emergent communication can be grounded to the agent observations and the spatial structure of the 3D environment.

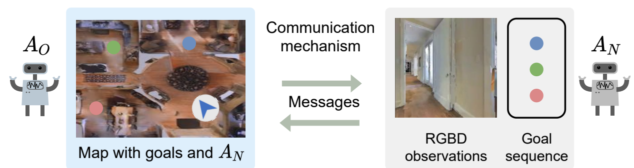

Figure 1. We propose a collaborative multi-object navigation task (CoMON) where an oracle agent A_O communicates with a navigator agent A_N. The oracle A_O possesses a global map and the navigator A_N needs to perceive and navigate a 3D environment to find a sequence of goal objects while avoiding collisions. Through this task, we study the impact of structured and unstructured communication mechanisms on navigation performance, and the emergence of messages grounded in egocentric perception.

1. Introduction

Research in embodied AI agents that learn to perceive, act, and communicate within 3D environments has become popular in recent years [3, 6, 9]. Collaboration between multiple agents has also received an increasing amount of attention. Consequently, there has been renewed interest in studying communication mechanisms that increase the effectiveness of collaborative agents [42].

A key goal of communication is to transmit information. Therefore, to analyze communication it is common to study collaborative tasks where agents have heterogeneous abilities or asymmetric access to information [10, 40]. A heterogeneous agent setup also corresponds to real-world scenarios such as guiding a delivery driver to our home over a phone call. However, prior work on emergent communication has adopted simplified settings like reference games [5, 43] or agents communicating within 2D environments [10]. Work involving communication in 3D environments has focused on whether communication can lead to improved performance through cooperation in solving the task [32–34], rather than detailed interpretation of the emergent communication patterns. Despite this rich literature studying emergent communication, there has been no systematic analysis and interpretation of emergent communication in realistic 3D environments.

In this paper, we focus on interpreting emergent communication through an *egocentric* and *spatially grounded* analysis. To do this, we define a collaborative multi-object navigation task (CoMON), which extends the recently proposed multi-object navigation (MultiON) task [64]. The CoMON task requires a pair of agents—an oracle with privileged knowledge of the environment in the form of a map, and a navigator who can perceive and navigate the environment—to communicate with each other in order for the navigator to find and reach a sequence of goal objects (see Figure 1). The primary role of this task is to study the emergent communication between heterogeneous agents in visually realistic 3D environments.

*denotes equal contribution by SP, SW, and UJ

We conduct a rigorous comparison and interpretation of two families of communication mechanisms: *unstructured* and *structured* communication. The first is commonly adopted in non-visual RL settings [25, 46] and corresponds to the 'continuous communication' of DIAL by Foerster et al. [25]. The latter introduces an inductive bias by imposing a discrete message structure and has been adopted by the Embodied AI community [33, 34].

We find that: 1) structured communication can achieve higher navigation performance than unstructured communication; 2) agents using structured communication come close to matching the success rate of 'oracle' communication but are less efficient; 3) interpretable messages akin to 'I am looking for the red goal' emerge in both communication mechanisms; and 4) both communication mechanisms lead to the emergence of egocentrically-grounded messages such as 'goal is close in front of you,' and 'goal is behind you.'

2. Related work

Our work is related to cooperation and coordination between multiple agents [14, 24, 27, 31, 41, 44, 45, 47, 51, 54, 55, 67]. We discuss relevant work in emergent communication, collaborative embodied AI, and embodied navigation tasks.

Emergent communication. Work on understanding the emergence of communication through simulations has a long history. Batali [5] studied this by encoding simple phrases into a series of characters that need to be decoded by another agent. Steels [60] studied a similar experiment with robots that had to generate a shared lexicon to perform well in a guessing game. Foerster et al. [26] showed that RL agents can learn successful communication protocols. Foerster et al. [25] then showed that agents can learn communication protocols in the form of messages that are sent to each other. When the agents are allowed to communicate, interesting communication patterns emerge [11, 13, 29, 30, 37, 43, 49, 52, 53, 61]. More recently, Lazaridou et al. [43] show emergence of natural language in referential games. Das et al. [20] propose a cooperative image guessing game between two static heterogeneous agents where the agents communicate through natural language. Mordatch and Abbeel [53] investigate the emergence of grounded compositional language in multi-agent populations. For a survey of emergent communication methods we refer the reader to Lazaridou and Baroni [42].

Our work is similar in spirit to Kottur et al. [40], in that we study and analyze emergent communication patterns and what information they communicate. Unlike that work and other work in emergent communication, we are less interested in whether compositional language emerges when using discrete symbols, but rather on whether there is consistent interpretation of messages between the two agents, and whether they correspond to available visual information. Kajić et al. [36] study how agents develop interpretable communication mechanisms in grid-world navigation environments and visualize agent policies conditioned on messages. We have a similar focus but we study continuous communication in realistic 3D environments.

Collaborative embodied AI tasks. While single agent embodied tasks have been studied in depth, there is less work on collaborative embodied agents. Das et al. [21] develop a targeted multi-agent communication architecture where agents select which of the other agents to communicate with. Jain et al. [33] introduce a furniture-lifting task where two agents must navigate to a furniture item. These agents must coordinate to satisfy spatial constraints for lifting the heavy furniture. Followup work studies a furniture-moving task where the agents relocate lifted furniture items [34, 35]. However, the agents are homogeneous and no map representation is studied in these prior works. Iqbal and Sha [32] study coordinated exploration by introducing handcrafted intrinsic rewards to incentivize agents to explore 'novel' states. Here, agents do not explicitly communicate with each other. Our work is focused on studying a spectrum of communication mechanisms for heterogeneous agents in visually realistic indoor 3D environments.

Navigation tasks in Embodied AI. Agents capable of navigating in complex, visual, 3D environments [2, 4, 12, 15, 19, 22, 38, 39, 65, 68, 69] have been studied extensively. Anderson et al. [3] divide embodied navigation tasks into point goal navigation (PointNav), object goal navigation (ObjectNav) and room goal navigation (RoomNav). Pertinent to this work, ObjectNav agents are given goal cues such as an object category label or an image of the goal object [7, 16–18, 70, 71, 73]. Long-horizon navigation tasks are most relevant to our work [8, 23, 63, 66, 72]. Map-based navigation methods have been benchmarked on multi-object navigation (multiON) *i.e.* navigating to an ordered sequence of goal objects [64]. Since we study communication involving map-based memory, we extend multiON to a collaborative setting.

3. Task

Here, we describe the collaborative multiON (CoMON) task, the agent observation and action spaces, and discuss alternatives to sharing information between the agents.

Background task (multiON). In an episode of multiON [64], an agent must navigate to an ordered sequence of goal objects G placed within an environment. The agent indicates discovery of a goal object by executing a FOUND action within a threshold distance from the goal. The objects in G are sampled from a set of k unique categories. An episode is a failure if the agent calls FOUND while not in the vicinity of the current goal, or if the allocated time budget is exceeded. We use m-ON to denote an episode with m sequential goals.

CoMON task. In Collaborative MultiON (CoMON), an episode involves two heterogeneous agents A_O and A_N. A_O is a disembodied oracle, which cannot navigate in the environment. However, A_O has access to oracle information (detailed later) of the environment's state. A_N is an embodied navigator, which navigates and interacts with the environment. A_N carries out a multiON [64] task. To optimize the team's (shared) rewards, both agents must collaborate. For this, A_O and A_N perform the task collaboratively by communicating via a limited-bandwidth channel.

Agent observations. A_O has access to a fixed top-down view of the scene along with A_N's position and orientation. The scene is discretized and represented as an oracle map M, a 3D tensor. The first two dimensions correspond to the horizontal and vertical axes of the top-down view, and the third contains semantic information in each cell $M[i, j]$:

- *Occupancy*: whether location $[i, j]$ is free space (*i.e.*, navigable), occupied, or out of the scene bounds.
- *Goal objects*: categorical variable denoting which goal object is located at $[i, j]$ or a 'no object' indicator.

The observations of A_N are consistent with multiON [64], allowing architectures trained on the single-agent task to be used in our collaborative setting. At time-step t, the observations of A_N include:

- *RGBD*: egocentric vision and depth frame o_t.
- *Object*: categorical variable denoting the current goal object as one-hot vector g_t.
- *Previous action*: agent action at previous time step as one-hot vector a_{t-1}.

Agent action space. At each time step, both A_O and A_N send messages to each other. A_N additionally takes an environment action following the communication round. The action space consists of four actions: { FORWARD, TURN LEFT, TURN RIGHT, and FOUND }. FORWARD takes the agent forward by 0.25m and turns are 30° each.

Task design alternatives. We note that there are other choices for how to distribute information between A_O and A_N. For example, the goal sequence information could be given to A_O. This would correspond to the practical scenario of a dispatch operator communicating with a taxi driver. However, this would lead to most information being concentrated with A_O and obviate the need for frequent two-way communication between A_O and A_N. Yet another setting would hide A_N's position and orientation on the map from A_O. Our preliminary investigations included experiments in this setting, with no information about A_N's position on the map being given to A_O. We empirically observed that this was a hard learning problem, with the agents failing to acquire meaningful task performance or communication strategies. We hypothesize that this may be partly due to a strong coupling with the independently challenging localization problem (*i.e.*, determining A_N's position and orientation in the map through egocentric observations from A_N's perspective). Since there is a rich literature for localization based on egocentric visual data (*e.g.*, see Fuentes-Pacheco et al. [28] for a survey), we factor out this aspect allowing a deeper focus on interpretation of emergent communication.

4. Agent models

We provide an overview of our agent models by describing the communication mechanisms, the agent network architectures, the reward structure and implementation details.

4.1. Communication mechanisms

We study two types of communication mechanisms: unstructured [25, 46] and structured [33, 34]. Their key difference is that the unstructured mechanism implements free-form communication via a real-valued vector, whereas the structured communication mechanism has an inductive bias through the imposed message structure. Figure 2 illustrates these two types of communication. Each round of communication involves the two agents synchronously sending a message to each other. The receiving agent uses the message to refine its internal representation (*i.e.*, belief). The same architecture is used for both agents and for each communication round.

Unstructured communication (`U-Comm`). The agent communicates a real-valued vector message. For sending the message, the belief is passed through a linear layer to produce the sent message. On the receiving side, the received message is concatenated with the belief and passed through two fully connected layers and skip connected through the belief to obtain the refined belief.

Structured communication (`S-Comm`). The agent has a vocabulary of K words $w_1, \ldots, w_K$, implemented as learnable embeddings. Note that the embeddings for the two rounds, and the two agents differ and are separately learned. The sent message is a set of probabilities $p_1, \ldots, p_K$ (where $\sum_{l=1}^{K} p_l = 1$) corresponding to the K words. These probabilities are obtained by passing the belief through a linear layer followed by a softmax layer. On the receiving side, the agent decodes these incoming message probabilities by linearly combining its word embeddings using the probabilities as weights, *i.e.*, it computes $\sum_{l=1}^{K} p_l w_l$. Similar to the previous mechanism, this decoded message is concatenated with the belief and passed through two fully connected layers and skip connected to obtain the refined belief. In early experiments, we tried using discrete tokens instead of a weighted sum. To make the model differentiable, we used the Gumbel-Softmax trick but found the agents could not be trained successfully. We hypothesize this is due to the high-dimensional input space and the numerical instability

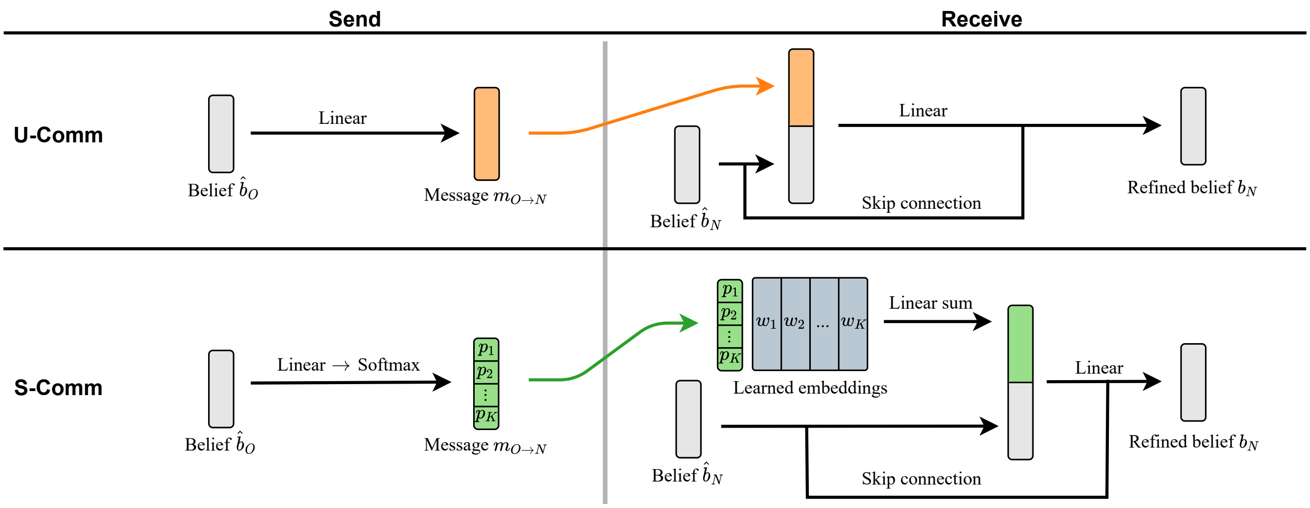

Figure 2. **Architecture of the send and receive branches for the unstructured (U-Comm) and structured (S-Comm) communication mechanisms.** On the sending branch, the agent creates a message by passing through a linear layer for U-Comm and by passing through a linear layer and a softmax layer for S-Comm. On the receiving branch for U-Comm, the message is concatenated with the belief and passed through a linear layer and skip connected to obtain the refined belief. For S-Comm, the message is first *decoded* by linearly combining the word embeddings w_k while using the probabilities p_k as weights ($\sum_{k=1}^{K} p_k w_k$). The embeddings are learned for each agent and round.

of Gumbel-Softmax [56].

4.2. Agent network architecture

Figure 3 illustrates the network architecture. We adapt the TBONE architecture which has been shown to be successful for multi-agent embodied tasks [33, 34]. For readability we drop the subscript t denoting the time step. A_O encodes the map by storing two 16-dimensional learnable embeddings for the occupancy and goal object category information at each grid location. Since A_O has access to A_N's position and orientation, it programmatically crops and rotates the map M around A_N's position and orientation to build an egocentric map E. This implicitly encodes A_N's position and orientation into E which is then passed through a CNN and a linear layer to obtain A_O's initial belief $\hat{b}_O$.

A_N passes its RGBD observations o through a CNN and a linear layer to obtain an observation embedding v_o. It also passes the object category g and previous action a_{t-1} through separate embedding layers to obtain a real-valued goal embedding v_g and action embedding v_a respectively. v_o and v_g are concatenated to obtain A_N's initial belief $\hat{b}_N$.

Both A_O and A_N go through two rounds of communication (as detailed in Section 4.1) to obtain their final beliefs b_O and b_N respectively. A_N concatenates its final belief b_N with the previous action embedding v_a and passes it through a GRU to obtain a state vector s. Following Jain et al. [33, 34], we use an actor-critic architecture where the state vector s is passed through: i) an actor head to estimate the distribution over the action space; and ii) a critic head that outputs a value estimating the utility of the state. b_O is left unused and hence it is discarded.

4.3. Reward structure

We model our multi-agent setup using the centralized training and decentralized execution paradigm [24, 47, 50, 59, 62]. In this paradigm, a central critic estimates the value function $V(s)$ of all the agents. Execution during testing is decentralized and each agent takes independent actions. The agents are trained using the navigator (A_N) reward: $r_t = \mathbb{1}_t^{\text{[reached subgoal]}} r^{\text{goal}} + r_t^{\text{closer}} + r^{\text{time penalty}}$ where $\mathbb{1}_t^{\text{[reached subgoal]}}$ is a binary indicator of finding a goal at time step t, r^{goal} is the reward for finding a goal, r_t^{closer} is the decrease in geodesic distance to the goal between the previous and the current time step, and $r^{\text{time penalty}}$ is the penalty per time step.

4.4. Implementation details

Following Wani et al. [64], we set r^{goal} to 3 and $r^{\text{time penalty}}$ to -0.01. We train with PPO [58], using 16 parallel threads with 4 mini-batches and 2 epochs per PPO update. Agents are trained until 50M steps accumulate across worker threads. The map M is of dimension 300×300 and each cell corresponds to a $0.8m \times 0.8m$ patch on the ground. See the supplement for more details.

5. Experiments

Here, we describe the experimental setup we adopt to study both communication mechanisms.

5.1. Agent models

All agent models share the base architecture explained in Sec. 4. For ablations each model is adjusted as follows (see supplement for details):

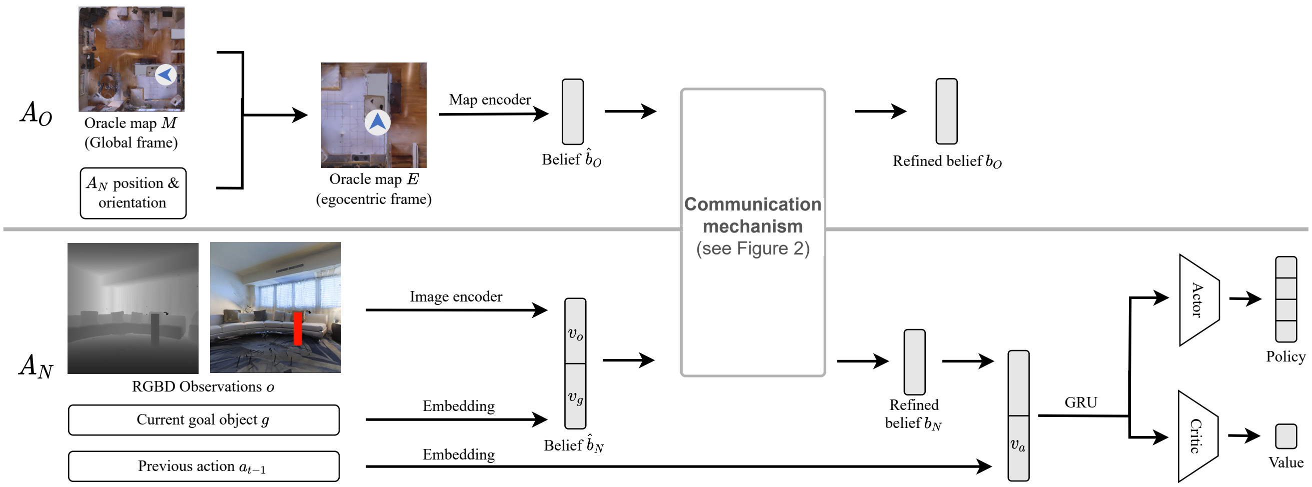

Figure 3. **Overall agent model architecture.** A_O and A_N process their respective inputs to get initial beliefs $\hat{b}_O$ and $\hat{b}_N$ which encode the agent's *belief* about the current observation. These are refined by a communication channel into final beliefs b_O and b_N. The belief b_N is concatenated with the previous action, and passed through a GRU to actor and critic heads to obtain policy and value function estimates.

`NoCom` [64] is the model without agent A_O. This represents the case where navigator A_N can't receive help from an oracle. It hence represents the 'no communication' scenario.

`Rand U-Comm` represents a model using unstructured communication while the messages sent between the agents are Gaussian random vectors. This provides a lower bound for unstructured communication.

`Rand S-Comm` represents a model using structured communication while the messages sent between the agents are random multinomial probability vectors. This provides the lower bound for structured communication.

`U-Comm` represents a model using unstructured communication as explained in Sec. 4.1.

`S-Comm` represents a model using structured communication as explained in Sec. 4.1.

`OracleMap` [64] combines both A_O and A_N into a single agent. Effectively, this agent has access to the map and it has to navigate in the environment without a need for communication. Hence, it sets an upper bound for performance.

	PROGRESS (%)			PPL (%)		
	1-ON	2-ON	3-ON	1-ON	2-ON	3-ON
`NoCom`	56	39	26	35	26	16
`Rand U-Comm`	59	40	28	36	28	18
`Rand S-Comm`	50	31	24	33	24	16
`U-Comm`	**87**	77	63	60	51	39
`S-Comm`	85	**80**	**70**	**67**	**59**	**50**
`OracleMap`	89	80	70	74	64	52

Table 1. **Task performance metrics for different communication mechanisms evaluated on the** 1-ON, 2-ON **and** 3-ON **tasks.** `Rand S-Comm` and `S-Comm` have a vocabulary size of two. For a fair comparison, both `Rand U-Comm` and `U-Comm` have the same message length of two elements. The random baselines perform poorly, and are close to the `NoCom` (*i.e.* 'no communication') baseline. Both the `U-Comm` and `S-Comm` communication mechanisms perform much better and approach `OracleMap`, with `S-Comm` being mostly more successful (higher PROGRESS) and more efficient (higher PPL), especially as the task becomes more challenging. Variance in PROGRESS for all models in 3-ON is less than 2%.

5.2. Datasets

We use the multiON dataset [64] based on the AI Habitat simulator [57]. This dataset contains episodes with agent starting position, orientation, and goal locations. There are eight goal objects with identical cylindrical shape but different colors. The episodes are generated from Matterport3D [15] scenes. We follow the standard scene-based Matterport3D train/val/test split with episodes established by Wani et al. [64]. Each scene contains 50,000 episodes for the train split and 12,500 episodes for the val and test splits. We train models for 3-ON (3 sequential goals) and evaluate on 1-ON, 2-ON, 3-ON, 4-ON and 5-ON.

5.3. Quantitative evaluation

We adopt the four metrics used in Wani et al. [64]. **SUCCESS**: episode success rate; **PROGRESS**: fraction of goal objects found in an episode; **SPL**: success weighted by path length; **PPL**: progress weighted by path length.

We summarize our experimental findings in Table 1. We report PROGRESS and PPL for 1,000 val episodes. As expected, `OracleMap` has the highest performance among all the agent models, with significant gains over `NoCom`. `Rand U-Comm` and `Rand S-Comm` perform close to `NoCom` which shows that the learnt messages indeed contain useful

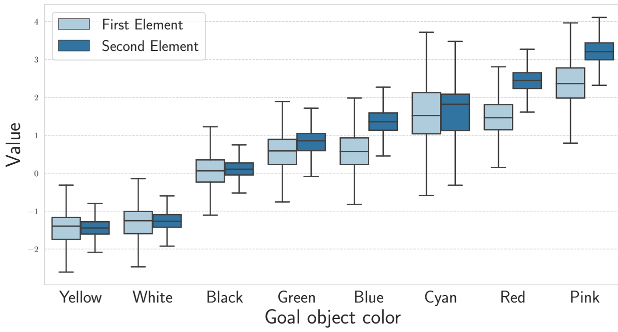

Figure 4. **Value of first and second element of $m^1_{N\to O}$ message plotted against goal object color in U-Comm.** Goal object colors are on the x-axis and the distribution of $m^1_{N\to O}$ values is on the y-axis. The box plots show 0^{th}, 25^{th}, 50^{th}, 75^{th}, and 100^{th} quartiles after removing outliers. Note that A_N sends different messages for differently colored objects. The ordering of the colors by average message value appears to respect color hue similarity (*e.g.*, red and pink are close together and far from yellow and white).

information. We observe that `S-Comm` performs better than `U-Comm`. The difference is more pronounced as the task difficulty increases. PPL decreases by 10.44% for 1-ON, 13.5% for 2-ON, and 22% for 3-ON. This shows that the imposed communication structure helps learn more efficient communication strategies. `NoCom` and `OracleMap` are the same as in Wani et al. [64] but we train for 50M steps instead of 40M steps. To test generalization, we also evaluate on 4-ON (`S-Comm` PROGRESS is 63% *vs.* `U-Comm` 41%) and 5-ON (`S-Comm` PROGRESS is 52% *vs.* `U-Comm` 26%). This indicates that `S-Comm` agents are better able to generalize to harder tasks (see supplement for more details).

6. Communication analysis

Here, we interpret the emergent communication between the agents. We use the notation $m^{\text{round}}_{\text{sender}\to\text{receiver}}$. Hence $m^1_{O\to N}$ denotes the message sent by A_O to A_N for round one. At each step, four messages are sent between the agents: $m^1_{O\to N}$, $m^1_{N\to O}$, $m^2_{O\to N}$, and $m^2_{N\to O}$. We interpret $m^1_{N\to O}$ and $m^2_{O\to N}$ in the main paper, and discuss the interpretation of $m^1_{O\to N}$ in the supplement. We do not interpret $m^2_{N\to O}$ as it is used to refine belief $\tilde{b}_O$ to b_O which is not used anywhere. For `U-Comm`, we interpret messages of length 2 and for `S-Comm`, we interpret vocabulary of size 2 and 3 (see supplement for vocabulary size 3).

6.1. U-Comm interpretation

What does A_N tell A_O in $m^1_{N\to O}$? This is the first message that A_N sends to A_O. We hypothesize that it is used to communicate the goal object color. This is intuitive as A_O needs to know the goal to which A_N must navigate. This is similar to a human asking "where is the green object goal located?" Figure 4 shows the distribution of the two elements of $m^1_{N\to O}$ w.r.t. goal object category (x-axis). The data for the plot is collected from each step across 1,000 validation episodes. It appears that A_N sends different messages for different objects. To test this hypothesis, we quantify the correlation between $m^1_{N\to O}$ and the goal object. We fit linear probes [1] on $m^1_{N\to O}$ to classify goal objects. Linear probes use linear classifiers to map input data to output and are trained using a cross-entropy loss. We use the same data for this analysis as for Figure 4. We split the data into train and val with a ratio of 3:1 and train the probe to predict the goal object category with $m^1_{N\to O}$ as input. The probe achieves an accuracy of 69.7% on the val split, supporting our hypothesis that $m^1_{N\to O}$ communicates the goal object color.

What does A_O tell A_N in $m^2_{O\to N}$? This is the second message that A_O send to A_N. We hypothesize that A_O uses it to communicate the relative position of the goal w.r.t. A_N. This is akin to a human saying "the goal you asked for is in front of you." Figure 5 shows the distribution of the two elements of $m^2_{O\to N}$ against the current object goal in the spatial reference frame defined by the position and orientation of A_N (egocentric frame) at the environment step when the message was sent. In the figure, the agent is facing up and the field-of-view is marked by red lines. When the goal is in front of A_N, A_O sends smaller values for the 1^{st} element and higher values for the 2^{nd} element of $m^2_{O\to N}$. We observe that the emergent communication exhibits an angular pattern. To quantify this observation, we again fit linear probes. Given $m^2_{O\to N}$, we predict the angle of the goal object w.r.t. A_N's heading direction (+y axis). Since the plot is mostly symmetric about the y-axis, we take the absolute value of the angle from the heading direction and bin the angles into 4 bins: $[0°, 45°)$, $[45°, 90°)$, $[90°, 135°)$, $[135°, 180°)$. Given $m^2_{O\to N}$, our probe has to predict the bin to which the goal location would belong. We observe a classification accuracy of 58% (compared to chance accuracy of 25%), providing support for our hypothesis that $m^2_{O\to N}$ conveys the egocentric relative position of the goal.

Since both A_O and A_N send messages that are statistically dependent on their respective observations, we can conclude that they exhibit positive signaling [48] (sending messages related to their observations or actions).

6.2. S-Comm interpretation

In this communication mechanism, the messages exchanged between the agents consist of probabilities p_1 and p_2 for words w_1 and w_2 respectively. In Figure 7 we plot the distribution of p_1 for messages $m^1_{N\to O}$ and $m^2_{O\to N}$ on all val set episodes (note that $p_2 = 1 - p_1$, which can hence be inferred from the distribution of p_1). We observe that most probabilities for vocabulary of size 2 are close to 0 or 1. Based on this observation, for vocabulary size 2, we bin the probabilities into three classes: Δ_1 ($p_1 < 0.2$), Δ_2 ($0.2 \leq p_1 \leq 0.8$), or Δ_3 ($p_1 > 0.8$). Here, we only inter-

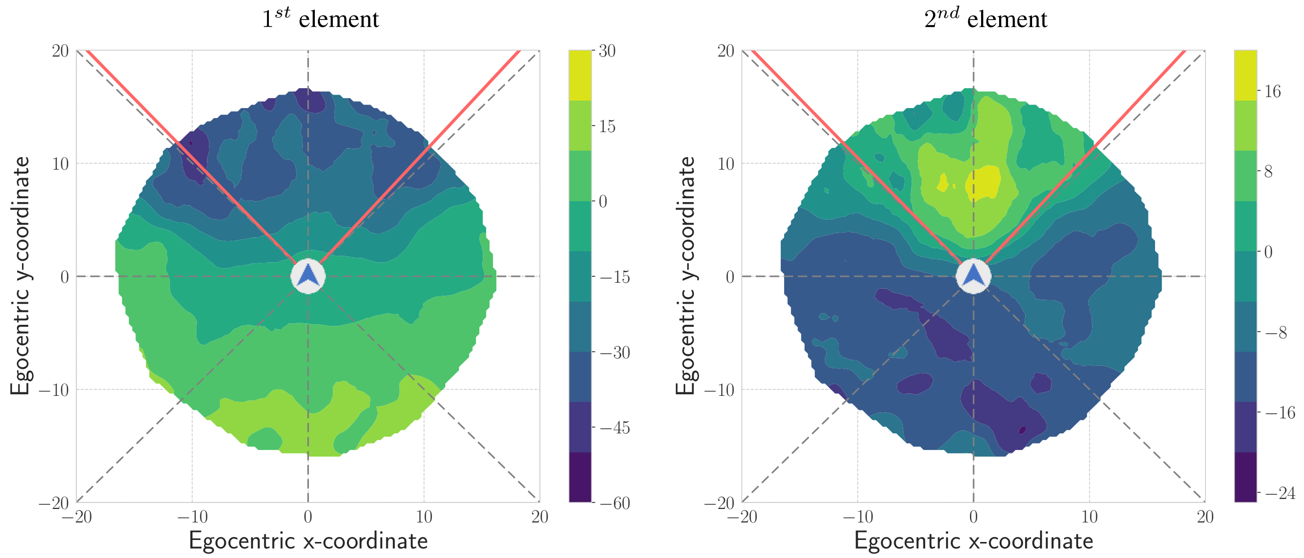

Figure 5. **Egocentric visualization of `U-Comm` communication symbol** $m^2_{O\to N}$. The two plots visualize the value of the first and second element of the message plotted w.r.t. the relative coordinates of the goal object from A_N. The navigator agent A_N is facing the +y axis and its field-of-view is marked with red lines. The plot on the left corresponds to the 1^{st} dimension of the message, while the plot on the right corresponds to the 2^{nd} dimension. The value of each dimension is indicated by the color hue. We observe that higher values of the 1^{st} dimension correspond to 'farther behind', while higher values of the 2^{nd} dimension are clustered 'close and in front' of the agent.

pret for vocabulary size 2 and defer the interpretation for vocabulary size 3 to the supplement.

What does A_N tell A_O in $m^1_{N\to O}$? We again hypothesize that A_N uses $m^1_{N\to O}$ to communicate which goal it has to navigate to. Since there are eight goal categories, A_N needs to communicate which one is the current goal. We observe that A_N sends Δ_1 when the goal object is a red, white or black, and sends Δ_2 otherwise. To quantify the correlation between the communication symbol and the current goal, we train a random forest classifier that predicts the communication symbol given the object category. Here, we use random forest classifiers rather than linear probes to better handle the non-linear decision boundaries that emerge in the $m^2_{O\to N}$ interpretation as seen in Figure 6. Note that to interpret `U-Comm`, we predict properties like goal category or goal direction using the messages. In contrast, to interpret `S-Comm`, we predict communication symbols using properties. In both cases, we predict a discrete variable like object category or goal direction in `U-Comm` and communication symbol in `S-Comm`. The classifier used here is trained using data from all the validation episodes. The data is split into train and test sets and our classifier attains almost 100% accuracy on the test set (see supplement).

What does A_O tell A_N in $m^2_{O\to N}$? Similar to `U-Comm`, A_O utilizes $m^2_{O\to N}$ to communicate the goal location. Figure 6 shows the symbols sent by A_O against the relative location of the current object goal in the egocentric frame of A_N when the message was sent (similar to Figure 5). Points are accumulated across 1000 validation episodes of 1-ON. We observe that A_O communicates Δ_1, Δ_2 or Δ_3 depending on the position of the current target object with respect to A_N. To verify this observation, we train a random forest classifier to predict the communication symbol from the (x, y) coordinate of the current target goal in A_N's reference frame. We observe an accuracy of about 89% with high precision and recall for all three classes Δ_1, Δ_2 and Δ_3 (details in supplement). With a larger vocabulary of size 3 A_O can send even more fine-grained information about the location of the current goal (see supplement). In both cases, we observe that the majority of symbols are associated with areas within the field of view of A_N (delineated in red). Thus, A_O uses a higher proportion of the communication bandwidth to communicate to A_N the location of the current goal if it is in A_N's field of view. Possibly, it is more advantageous for A_N to have precise information about the goal location when it is in front. If the goal is in the field of view, A_O sends a different symbol depending on the distance of the current goal from A_N. Here also, messages sent by A_O and A_N are dependent on their observations. Hence, both of them exhibit positive signaling.

Are A_N's actions influenced by $m^2_{O\to N}$? Agents exhibit positive listening [48] if the received messages influence the agent policy. Table 2 reports the percentage of each action taken for communication symbols in `S-Comm` (with vocabulary size 2). We observe that A_N never calls FOUND when it receives Δ_3. This is intuitive as Δ_3 is communicated when the goal is far ahead of A_N. We also observe that A_N is more likely to move forward when it receives Δ_3 as

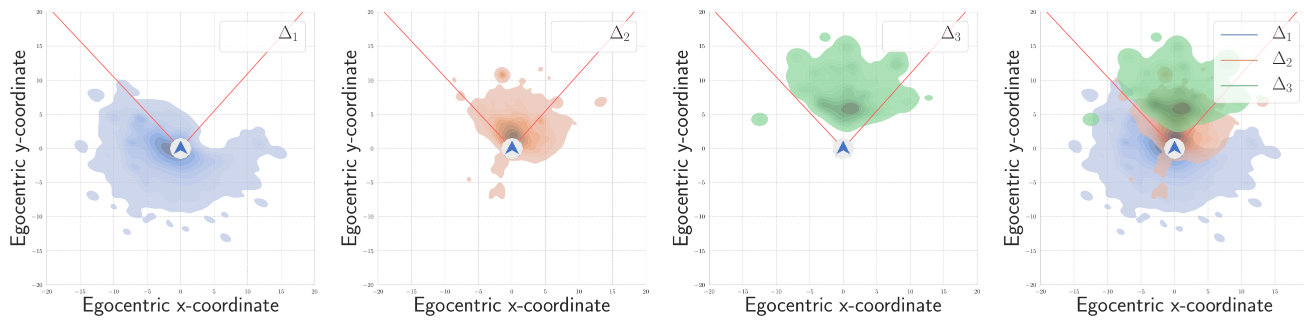

Figure 6. **Egocentric visualization of `S-Comm` communication symbol $m^2_{O \to N}$.** The plots show the relative coordinates of the current goal object from A_N's perspective when A_O communicates the symbol through `S-Comm` with vocabulary size two. The navigator agent (A_N) is facing the +y axis and its field-of-view is marked with red lines. Data points are accumulated across all validation episodes, and we plot contour lines of the bivariate density distribution. Each data point is a message with (x, y) coordinates determined from the coordinates of the current goal object in A_N's egocentric reference frame when the message was sent. The first three plots are for each communication symbol, and the right-most combines all symbols. Note how each symbol represents distinct regions that are egocentrically organized around the agent: Δ_1 captures 'behind and not visible', Δ_2 corresponds mostly to 'close, in front', and Δ_3 is 'farther in front'.

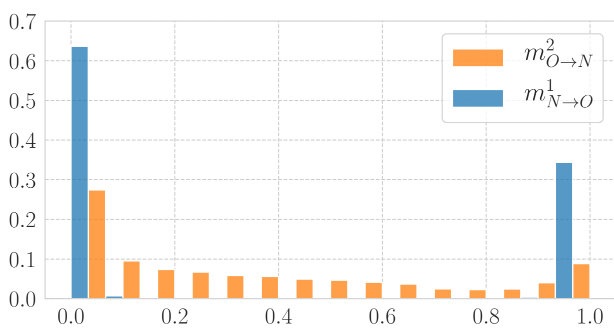

Figure 7. **Distribution of probability weight p_1 associated with w_1 in messages $m^1_{N \to O}$ and $m^2_{O \to N}$ for `S-Comm`.** The vocabulary consists of two words, w_1 and w_2. Since $p_1 + p_2 = 1$, we only plot p_1 here. For $m^1_{N \to O}$, probabilities are concentrated at $p_1 = 0$ and $p_1 = 1$. For $m^1_{N \to O}$, distribution is comparatively uniform with higher probabilities at $p_1 = 0$ and $p_1 = 1$.

	FOUND	FORWARD	TURN LEFT	TURN RIGHT
Δ_1	0.8	43.9	24.7	30.6
Δ_2	0.3	52.2	28.7	18.8
Δ_3	0.0	63.4	18.9	17.7

Table 2. **Distribution over actions taken by A_N upon receiving each `S-Comm` communication symbol (vocabulary size 2).** Values in each row report percentage out of all actions taken when that symbol is received. Note that Δ_3 leads to a high percentage of FORWARD actions and no FOUND actions. This is intuitive in light of the spatial distribution of goal positions relative to A_N when Δ_3 is communicated, as visualized in Figure 6.

compared to Δ_1 or Δ_2. This is also intuitive as A_N is more likely to move forward when the goal is far ahead.

What happens when the goal is in A_N's view? The distribution of the exchanged messages remains unchanged, but how A_N acts based on the received messages is different. We performed two experiments at evaluation time to study this case. **1)** A_O sends random messages when the goal is visible to A_N. We find this does not change the overall performance of A_N. **2)** We insert an incorrect goal in the scene while keeping A_O's map unchanged. PROGRESS and PPL metrics drop to 29% and 7% respectively. We conclude that when the goal is visible, A_N ignores messages from A_O and relies on its perception to navigate.

7. Conclusion

We proposed the collaborative multi-object navigation task (CoMON) for studying the grounding of learned communication between heterogeneous agents. Using this task, we investigated two families of communication mechanisms (structured and unstructured communication) between heterogeneous agents. We analyzed the emergent communication patterns though an egocentric and spatially grounded lens. We found the emergence of interpretable perception-specific messages such as 'I am looking for X' and egocentric instructions such as 'look behind' and 'goal is close in front.' We believe the CoMON task, along with the interpretation framework for communication between agents that we presented will help to enable systematic study of grounded communication for embodied AI navigation agents.

Acknowledgements We thank the anonymous reviewers for their suggestions and feedback. This work was funded in part by a Canada CIFAR AI Chair, a Canada Research Chair and NSERC Discovery Grant, and enabled in part by support provided by WestGrid and Compute Canada. This work is supported in part by NSF under Grant #1718221, 2008387, 2045586.

References

[1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016. 6

[2] Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Košecká, and Alexander C Berg. A dataset for developing and benchmarking active vision. In *ICRA*, 2017. 2

[3] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 1, 2

[4] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2D-3D-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 2

[5] John Batali. Computational simulations of the emergence of grammar. *Approach to the Evolution of Language*, 1998. 1, 2

[6] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, Manolis Savva, and Hao Su. Rearrangement: A challenge for embodied AI. *arXiv preprint arXiv:2011.01975*, 2020. 1

[7] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020. 2

[8] Edward Beeching, Christian Wolf, Jilles Dibangoye, and Olivier Simonin. EgoMap: Projective mapping and structured egocentric memory for deep RL. In *ECML-PKDD*, 2020. 2

[9] Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience grounds language. In *EMNLP*, 2020. 1

[10] Ben Bogin, Mor Geva, and Jonathan Berant. Emergence of communication in an interactive world with consistent speakers. In *Emergent Communication Workshop at NeurIPS*, 2018. 1

[11] Jeshua Bratman, Michael Shvartsman, Richard L Lewis, and Satinder Singh. A new approach to exploring language emergence as boundedly optimal control in the face of environmental and cognitive constraints. In *ICCM*, 2010. 2

[12] Simon Brodeur, Ethan Perez, Ankesh Anand, Florian Golemo, Luca Celotti, Florian Strub, Jean Rouat, Hugo Larochelle, and Aaron Courville. Home: A household multimodal environment. *arXiv preprint arXiv:1711.11017*, 2017. 2

[13] Kalesha Bullard, Franziska Meier, Douwe Kiela, Joelle Pineau, and Jakob Foerster. Exploring zero-shot emergent communication in embodied multi-agent populations. *Deep RL Workshop at NeurIPS*, 2020. 2

[14] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *SMC*, 2008. 2

[15] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *3DV*, 2017. 2, 5

[16] Matthew Chang, Arjun Gupta, and Saurabh Gupta. Semantic visual navigation by watching youtube videos. In *NeurIPS*, 2020. 2

[17] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *NeurIPS*, 2020.

[18] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological SLAM for visual navigation. In *CVPR*, 2020. 2

[19] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *ECCV*, 2020. 2

[20] Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *CVPR*, 2017. 2

[21] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Michael Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *ICML*, 2019. 2

[22] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *CoRL*, 2017. 2

[23] Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. In *CVPR*, 2019. 2

[24] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *AAAI*, 2018. 2, 4

[25] Jakob N Foerster, Yannis M Assael, Nando De Freitas, and Shimon Whiteson. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In *NeurIPS*, 2016. 2, 3

[26] Jakob N Foerster, Yannis M Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate to solve riddles with deep distributed recurrent Q-networks. In *Deep Learning Workshop at IJCAI*, 2016. 2

[27] Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *AAMAS*, 2017. 2

[28] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 2015. 3

[29] C Lee Giles and Kam-Chuen Jim. Learning communication for multi-agent systems. In *Workshop on Radical Agent Concepts*, 2002. 2

[30] Abhinav Gupta, Cinjon Resnick, Jakob Foerster, Andrew Dai, and Kyunghyun Cho. Compositionality and capacity in emergent languages. In *Workshop on Representation Learning for NLP*, 2020. 2

[31] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *AAMAS*, 2017. 2

[32] Shariq Iqbal and Fei Sha. Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning. *arXiv preprint arXiv:1905.12127*, 2019. 1, 2

[33] Unnat Jain, Luca Weihs, Eric Kolve, Mohammad Rastegari,

Svetlana Lazebnik, Ali Farhadi, Alexander G. Schwing, and Aniruddha Kembhavi. Two body problem: Collaborative visual task completion. In *CVPR*, 2019. 2, 3, 4

[34] Unnat Jain, Luca Weihs, Eric Kolve, Ali Farhadi, Svetlana Lazebnik, Aniruddha Kembhavi, and Alexander G. Schwing. A cordial sync: Going beyond marginal policies for multi-agent embodied tasks. In *ECCV*, 2020. 1, 2, 3, 4

[35] Unnat Jain, Iou-Jen Liu, Svetlana Lazebnik, Aniruddha Kembhavi, Luca Weihs, and Alexander Schwing. GridToPix: Training embodied agents with minimal supervision. *arXiv preprint arXiv:2105.00931*, 2021. 2

[36] Ivana Kajić, Eser Aygün, and Doina Precup. Learning to cooperate: Emergent communication in multi-agent navigation. In *CogSci*, 2020. 2

[37] Tatsuya Kasai, Hiroshi Tenmoto, and Akimoto Kamiya. Learning of communication codes in multi-agent reinforcement learning problem. In *ICSCIA*, 2008. 2

[38] Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. VizDoom: A doom-based AI research platform for visual reinforcement learning. In *CIG*, 2016. 2

[39] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: an interactive 3D environment for visual AI. *arXiv preprint arXiv:1712.05474*, 2019. 2

[40] Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In *EMNLP*, 2017. 1, 2

[41] Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *ICML*, 2000. 2

[42] Angeliki Lazaridou and Marco Baroni. Emergent multi-agent communication in the deep learning era. *arXiv preprint arXiv:2006.02419*, 2020. 1, 2

[43] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *ICLR*, 2017. 1, 2

[44] Iou-Jen Liu, Raymond A Yeh, and Alexander G Schwing. Pic: permutation invariant critic for multi-agent deep reinforcement learning. In *CoRL*, 2020. 2

[45] Iou-Jen Liu, Unnat Jain, Raymond A Yeh, and Alexander Schwing. Cooperative exploration for multi-agent deep reinforcement learning. In *ICML*, 2021. 2

[46] Yen-Cheng Liu, Junjiao Tian, Chih-Yao Ma, Nathan Glaser, Chia-Wen Kuo, and Zsolt Kira. Who2com: Collaborative perception via learnable handshake communication. In *ICRA*, 2020. 2, 3

[47] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NeurIPS*, 2017. 2, 4

[48] Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, and Yann Dauphin. On the pitfalls of measuring emergent communication. In *AAMAS*, 2019. 6, 7

[49] Ryan Lowe, Abhinav Gupta, Jakob Foerster, Douwe Kiela, and Joelle Pineau. On the interaction between supervision and self-play in emergent communication. In *ICLR*, 2020. 2

[50] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. In *NeurIPS*, 2019. 4

[51] Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *IROS*, 2007. 2

[52] Francisco S Melo, Matthijs TJ Spaan, and Stefan J Witwicki. Querypomdp: Pomdp-based communication in multiagent systems. In *EUMAS*, 2011. 2

[53] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *AAAI*, 2018. 2

[54] Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P How, and John Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *ICML*, 2017. 2

[55] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *AAMAS*, 2005. 2

[56] Andres Potapczynski, Gabriel Loaiza-Ganem, and John P Cunningham. Invertible gaussian reparameterization: Revisiting the gumbel-softmax. *NeurIPS*, 2020. 4

[57] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *ICCV*, 2019. 5

[58] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 4

[59] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. *ICML*, 2019. 4

[60] Luc Steels. Evolving grounded communication for robots. *Trends in Cognitive Sciences*, 2003. 2

[61] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *NeurIPS*, 2016. 2

[62] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017. 4

[63] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *arXiv preprint arXiv:2106.14405*, 2021. 2

[64] Saim Wani, Shivansh Patel, Unnat Jain, Angel X. Chang, and Manolis Savva. Multi-ON: Benchmarking Semantic Map Memory using Multi-Object Navigation. In *NeurIPS*, 2020. 1, 2, 3, 4, 5, 6

[65] Luca Weihs, Jordi Salvador, Klemen Kotar, Unnat Jain, Kuo-Hao Zeng, Roozbeh Mottaghi, and Aniruddha Kembhavi. AllenAct: A framework for embodied AI research. *arXiv preprint arXiv:2008.12760*, 2020. 2

[66] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *CVPR*, 2021. 2

[67] Luca Weihs, Aniruddha Kembhavi, Kiana Ehsani, Sarah M Pratt, Winson Han, Alvaro Herrasti, Eric Kolve, Dustin Schwenk, Roozbeh Mottaghi, and Ali Farhadi. Learning generalizable visual representations via interactive gameplay. In *ICLR*, 2021. 2

[68] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *CVPR*, 2018. 2

[69] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. Interactive gibson: A benchmark for interactive navigation in cluttered environments. In *ICRA*, 2020. 2

[70] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Visual curiosity: Learning to ask questions to learn visual recognition. In *CoRL*, 2018. 2

[71] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectnav. *arXiv preprint arXiv:2104.04112*, 2021. 2

[72] Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L. Berg, and Dhruv Batra. Multi-target embodied question answering. In *CVPR*, 2019. 2

[73] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 2017. 2