Interpretation of Emergent Communication in Heterogeneous Collaborative Embodied Agents

Shivansh Patel1* Saim Wani2* Unnat Jain3* Alexander Schwing3
Svetlana Lazebnik3 Manolis Savva1 Angel X. Chang1
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. Video summary: https://youtu.be/kLv2rxO9t0g

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.

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]. 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 [62] 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 **FIND** \}. **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.
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 programatically 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 $b_O$.

$A_N$ 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 $b_N$.

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: $r_t = \mathbb{1}_{t\text{[reached subgoal]}} r_{\text{goal}} + r_{\text{closer}} + r_{\text{time penalty}}$ where $\mathbb{1}_{t\text{[reached subgoal]}}$ is a binary indicator of finding a goal at time step $t$, $r_{\text{goal}}$ is the reward for finding a goal, $r_{\text{closer}}$ 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 $r_{\text{goal}}$ 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 $300 \times 300$ 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):
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.

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

---

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%.
What does \( A_O \) tell \( A_N \) in \( m_{O \rightarrow N}^1 \)? This is the second message that \( A_O \) sends 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 \( m_{O \rightarrow N}^1 \) 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 1st element and higher values for the 2nd element of \( m_{O \rightarrow N}^1 \). We observe that the emergent communication exhibits an angular pattern. To quantify this observation, we again fit linear probes. Given \( m_{O \rightarrow N}^2 \), we predict the value 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: \([0^\circ, 45^\circ), [45^\circ, 90^\circ), [90^\circ, 135^\circ), [135^\circ, 180^\circ)\). Given \( m_{O \rightarrow N}^2 \), 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 \( m_{O \rightarrow N}^2 \) 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. Communication analysis

Here, we interpret the emergent communication between the agents. We use the notation \( m_{\text{sender} \rightarrow \text{receiver}}^{\text{round}} \) Hence \( m_{O \rightarrow N}^1 \) denotes the message sent by \( A_O \) to \( A_N \) for round one. At each step, four messages are sent between the agents: \( m_{O \rightarrow N}^1, m_{O \rightarrow N}^2, m_{O \rightarrow N}^3, \) and \( m_{O \rightarrow N}^4 \). We interpret \( m_{O \rightarrow N}^1 \) and \( m_{O \rightarrow N}^2 \) in the main paper, and discuss the interpretation of \( m_{O \rightarrow N}^3 \) in the supplement. We do not interpret \( m_{O \rightarrow N}^4 \) as it is used to refine belief \( b_{O \rightarrow O} \) which is not used anywhere. For \( U\text{-Comm} \), we interpret messages of length 2 and for \( S\text{-Comm} \), 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 \( m_{N \rightarrow O}^1 \)? 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 \( m_{N \rightarrow O}^1 \) 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 \( m_{N \rightarrow O}^1 \) and the goal object. We fit linear probes [1] on \( m_{N \rightarrow O}^1 \) 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 \( m_{N \rightarrow O}^1 \) as input. The probe achieves an accuracy of 69.7\% on the val split, supporting our hypothesis that \( m_{N \rightarrow O}^1 \) communicates the goal object color.

## 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 \( m_{N \rightarrow O}^1 \) and \( m_{O \rightarrow N}^2 \) 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: \( \Delta_1 \) (\( p_1 < 0.2 \)), \( \Delta_2 \) (\( 0.2 \leq p_1 \leq 0.8 \)), or \( \Delta_3 \) (\( p_1 > 0.8 \)). Here, we only inter-
pret for vocabulary size 2 and defer the interpretation for vocabulary size 3 to the supplement.

**What does \( A_N \) tell \( A_O \) in \( m_{2, O ightarrow N}^1 \)?** We again hypothesize that \( A_N \) uses \( m_{2, O ightarrow N}^1 \) 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 \( \Delta_1 \) when the goal object is a red, white or black, and sends \( \Delta_2 \) 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 \( m_{2, O ightarrow N}^1 \) 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 \( m_{2, O ightarrow N}^2 \)?** Similar to \( U-Comm \), \( A_O \) utilizes \( m_{2, O ightarrow N}^2 \) 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 \( \Delta_1, \Delta_2 \) or \( \Delta_3 \) 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 \( \Delta_1, \Delta_2 \) and \( \Delta_3 \) (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 \( m_{2, O ightarrow N}^3 \)?** 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 \( \Delta_3 \). This is intuitive as \( \Delta_3 \) 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 \( \Delta_3 \) as
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 through 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 Košecka, 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, Aleksandar 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 Jacobs, 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] Joshua 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. 1

[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 Cod袂vila, 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,
[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
A. Supplementary material

This supplemental document provides the following additional contents to support the main paper:

- A.1 Overview of notation used in the paper
- A.2 Additional quantitative evaluation
- A.3 Interpretation of $m_{O \rightarrow N}^1$ for 1-ON
- A.4 Interpretation of $S$-Comm using vocabulary size 3
- A.5 Interpretation of U-Comm and S-Comm for 2-ON
- A.6 Additional analyses to check for information content of messages
- A.7 Additional analyses to check for information content of messages
- B Implementation details

A.1. Notation overview

Table 3 provides a summary of the definitions of important notations used in the paper and this supplementary document.

A.2. Additional quantitative evaluation

We report only the PPL and PROGRESS metrics in the main paper. Table 4 summarizes the complete set of evaluation metrics we use: PROGRESS, PPL, SUCCESS and SPL. The trends in SUCCESS and SPL are similar to those in PROGRESS and PPL. We also perform generalization experiments by evaluating models trained on 3-ON on the more difficult 4-ON and 5-ON tasks. Table 4 summarizes those results as well. We observe that S-Comm outperforms U-Comm in these generalization experiments as well.

Table 5 shows the effect of increasing message length on the task performance. 2-dimensional message results are repeated from Table 1. S-Comm still outperforms U-Comm on equal message lengths. Overall, there are small improvements with increasing message dimension. We hypothesize that $A_O$ can encode the goal location in 2-dimensional messages, thus higher-dimensional messages provide small improvements.

A.3. Interpretation of $m_{O \rightarrow N}^1$ for 1-ON

In the CoMON task, $A_N$ has knowledge of the goal to visit while $A_O$ has knowledge of the semantic map (with goal positions) as well as the position and orientation of $A_O$. In the main paper, we showed that $m_{O \rightarrow N}^1$ is used to communicate the location of the goal. Here we consider what information is in the initial message sent from $A_O$ to $A_N$ ($m_{O \rightarrow N}^1$) for 1-ON. Note that in the case of 1-ON, there is only one goal, so $A_O$ can already send information about that goal without waiting for $m_{N \rightarrow O}^1$. Using similar methods as we employed in the main paper for $m_{O \rightarrow N}^1$, we find that $m_{O \rightarrow N}^1$ is also used to communicate the location of the goal relative to $A_N$ for U-Comm and S-Comm.

A.3.1 Interpretation of $m_{O \rightarrow N}^1$ in U-Comm

Figure 8 shows the distribution of $m_{O \rightarrow N}^1$ w.r.t. the relative coordinates of the goal object from $A_N$, using a similar visualization as for $m_{O \rightarrow N}^1$ (Figure 5 in the main paper). We note there is a correlation between $m_{O \rightarrow N}^1$ and the location of the current goal object: with the first element indicating whether the goal is to the left of the agent, and the second element whether the goal is to the right. To quantify the relation, we again fit linear probes. As the target of the linear probe, we bin the angles into 8 bins each of $45^\circ$ (see dashed lines in Figure 8). Our probe attains classification accuracy of 51% (compared to chance accuracy of 12.5%) supporting our hypothesis that $m_{O \rightarrow N}^1$ includes information about the location of the current goal object relative to $A_N$.

A.3.2 Interpretation of $m_{O \rightarrow N}^1$ in S-Comm

As in the main paper (Section 6.2), we perform a similar analysis for $m_{O \rightarrow N}^1$ as for $m_{O \rightarrow N}^2$, where we use thresholds to group the messages into $\Delta_1$, $\Delta_2$, and $\Delta_3$. Figure 9 plots the distribution of each symbol w.r.t. the relative location of the current goal relative to $A_N$ (similar to Figure 6 in the main paper). We observe that $m_{O \rightarrow N}^1$ is again used to convey the goal object location, but the correlation between the communicated message and the goal object location is weaker than that of $m_{O \rightarrow N}^2$. This is evident from the higher overlap of the regions corresponding to each symbol (compared to Figure 6 in the main paper). This observation is confirmed by the lower classification accuracy of 83% (vs 89% for $m_{O \rightarrow N}^2$) after training a random forest classifier to predict the communicated symbol from the $(x, y)$ coordinate of the current goal object.

A.4. Interpretation of $S$-Comm for vocabulary size 3

We provide details of the analysis of S-Comm with vocabulary size 3. Similar to our analysis for vocabulary size 2 (see section 6.2 in main paper), we bin the messages probabilities based on the observed probabilities. Due to the larger vocabulary size, we bin the messages into six classes (vs 3 classes for vocabulary size of 2): $\Delta_1$, $\Delta_2$, $\Delta_3$, $\Delta_4$, $\Delta_5$, or $\Delta_6$. See B.0.4 for more details about the binning process. When we examine the messages, we see a consistent pattern as we observed for vocabulary size 2.

What does $A_N$ tell $A_O$ in $m_{N \rightarrow O}^1$? Here also, we observe that $A_N$ uses $m_{N \rightarrow O}^1$ to convey the goal object to $A_O$. $A_N$ send $\Delta_1$ when the goal object is a red, green, pink, or cyan cylinder. It sends $\Delta_2$ for blue and yellow cylinders, and it sends $\Delta_3$ otherwise. We find that $\Delta_4$, $\Delta_5$, $\Delta_6$ are not used for $m_{N \rightarrow O}^1$, and are only used in $m_{O \rightarrow N}^1$ and $m_{O \rightarrow N}^2$.

What does $A_O$ tell $A_N$ in $m_{O \rightarrow N}^2$? We perform the same interpretation analysis as we did for vocabulary size 2 in the main paper. We again observe that $A_O$ utilizes $m_{O \rightarrow N}^2$ to
Table 5. Effect of message length on performance. S-Comm outperforms U-Comm on similar message length and there is a slight improvement with increasing message dimension.

Table 3. Summary of notation. Subscript $t$ denotes the corresponding notation at time step $t$.

Table 4. Additional quantitative metrics on 1-ON, 2-ON and 3-ON tasks and generalization to 4-ON and 5-ON. All agents are trained on 3-ON and evaluated on the task indicated in each column.

6.2), we observe that more symbols are allocated to the front of the agent than at its back.

What does $A_O$ tell $A_N$ in $m_{O \rightarrow N}^1$? For this message, our observations are again consistent with those of vocabulary size 2 (see Figure 11). $A_O$ sends different symbols for different goal locations, but there is more overlap between the regions allocated to the symbols as compared to that in $m_{O \rightarrow N}^2$.

A.5. Interpretation for 2-ON

Most of our analysis thus far has focused on 1-ON. Here we analyze what is communicated in U-Comm and S-Comm for 2-ON using the same methodology.

A.5.1 Interpretation of U-Comm for 2-ON

What does $A_N$ tell $A_O$ in $m_{N \rightarrow O}^1$? We observe that the distribution of $m_{N \rightarrow O}^1$ is similar to that in Figure 4 of the main paper. This is expected as $A_N$ would send similar
message irrespective of the number of goals in an episode.

What does $A_O$ tell $A_N$ in $m^2_{O \rightarrow N}$? In Figure 12, we show how the distribution of $m^2_{O \rightarrow N}$ 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. Our observations are consistent with what we observed for 1-ON. $m^2_{O \rightarrow N}$ is used to convey the goal location to $A_N$.

### A.5.2 Interpretation of S–Comm for 2-ON

In this setting we use a vocabulary size of 2 and group the messages into three symbols as in the main paper. Because the number of symbols is less than the number of possible goals, we observe that the agents use a partitioning scheme when sending messages. This phenomenon has been observed in Kottur et al. and is consistent with game theory results.

What does $A_N$ tell $A_O$ in $m^1_{N \rightarrow O}$? Here also, $A_O$ sends similar $m^1_{N \rightarrow O}$ as in 1-ON. That is, $A_O$ sends $\Delta_1$ when the goal object is a red, white or black cylinder, and sends $\Delta_2$ otherwise. $A_N$ partitions the goal objects into two sets: $P_1$ with 3 categories and $P_2$ with 5 categories.

What does $A_O$ tell $A_N$ in $m^2_{O \rightarrow N}$? As $m^1_{N \rightarrow O}$ only sends $\Delta_1$ or $\Delta_2$, $A_O$ cannot infer the precise current goal object.
If both of these objects lie in $P_2$, $A_N$ would send $\Delta_2$ to $A_O$ throughout the episode. Therefore, $A_O$ would not know which of the 2 objects $A_N$ is looking for at the moment. Instead, if one of the target objects lies in $P_1$ and the other in $P_2$, $A_O$ can infer the current target object $A_N$ is looking for. We plot the message $m_{O \rightarrow N}^2$ for the two cases separately. In Figure 13, first row represents the case when the current goal can be distinguished from $m_{N \rightarrow O}^1$. Note that the current goal is said to be distinguishable from $m_{N \rightarrow O}^1$ if the two goals for the episode lie in separate partitions $P_1$ and $P_2$. $m_{O \rightarrow N}^2$ correlates more strongly with the location of the current goal in the former case, where $A_O$ could infer the current goal before sending $m_{O \rightarrow N}^2$. This is reflected in the symbols being more well separated in the first row of Figure 13 than in the second row. This can be observed by the overlaps between symbols. Distinguishable goals have less overlap between symbol regions as compared to indistinguishable goals. To quantify the separation of symbols in the two plots, we also train a random forest classifier to predict the communication symbol given the x,y coordinates of the symbol in the plots as input. The prediction accuracy for distinguishable goals is 84% and for indistinguishable goals it is 76%.

**A.6. Episode map visualizations**

In Figure 14, we provide a visualization of egocentric observations and map state for S-Comm at several points on the trajectory to show correlations between the communication symbols for $m_{N \rightarrow O}$ (shown on the trajectory) and what the agent observes at each position. Similarly in Figure 15, we show the correlations for $m_{O \rightarrow N}$.

**A.7. Are messages conveying other information?**

We investigated other information that the messages might be conveying, but did not find a strong signal. We checked if $m_{O \rightarrow N}^2$ or $m_{O \rightarrow N}^2$ conveys the optimal action and if $m_{N \rightarrow O}^1$ conveys whether the current goal is in $A_N$’s view. We also checked whether messages from $A_N$ to $A_O$ contain direction, and messages from $A_O$ to $A_N$ contain color, and did not find any correlations.

**B. Implementation details**

**B.0.1 Architecture details**

Here we report the architectural details. $A_O$ has an oracle map $M$ of spatial dimension $300 \times 300$. This contains occupancy and goal object information. $M$ is converted to egocentric map $E$ of spatial dimension $45 \times 45$. Each of the occupancy and goal object information is converted to 16 dimensional embeddings for each grid location so the map is of dimension $45 \times 45 \times 32$. This is passed through a map encoder comprising of a two layered CNN and a linear layer to obtain 256–dimensional belief $\hat{b}_O$. $b_O$ is a 256–dimensional vector as well.

RGGD observations of $A_O$ are passed through an image encoder. It consists of three CNN layers and a linear layer to obtain an image embedding $v_i$ of shape 512. The current goal embedding $v_g$ and previous action embedding $v_a$ are both 16–dimensional vectors. The belief $b_N$ and $b_O$ are of shape 512. The state representation vector $s$ is of shape 528.

**B.0.2 Details about random baselines**

Here, we present the implementation details for Rand U-Comm and Rand S-Comm. In Rand U-Comm, we replace the message by a random vector sampled from a multivariate gaussian distribution with mean and variance equal to the mean and variance of the corresponding message sent in the validation set. For Rand S-Comm, we replace the message by random probabilities sampled from a random multinomial probability vectors and these probabilities sum up to 1.

**B.0.3 S-Comm classifier implementation**

To establish the existence of various correlations between the communication symbols exchanged between the agents in S-Comm, we train random forest classifiers that predict the communication symbol given a quantity $Q$ as input. We report the classification accuracy for $m_{N \rightarrow O}^1$ and $m_{O \rightarrow N}^2$ in Section 6.2 and for $m_{O \rightarrow N}^1$ above. For all of these, the data for training/evaluating the classifier is obtained by evaluating the model on the validation set of 1,000 episodes and accumulating the relevant metrics at each environment step across the 1,000 episodes. At each environment step, we log the following: \{ $m_{N \rightarrow O}^1$, $m_{O \rightarrow N}^2$, $m_{O \rightarrow N}^2$, current object goal category, relative location of current goal in $A_N$’s egocentric reference frame. \} We first balance the dataset such that each symbol $\Delta_i$ has equal number of training examples. The collected data, where each data point corresponds to an environment step, is divided into train and val sets in 3/1 ratio. The classifier is then trained to predict the communication symbol $\Delta_i$ from quantity $q$ using the train set. We report the classification accuracy on the val set.

**B.0.4 Binning of probabilities in S-Comm**

Here, we describe the implementation of binning to create the discrete symbols used in our interpretation of S-Comm.

**Vocabulary size 2.** Let the probability vector output by the final softmax layer of communication module be $p = [p_1, p_2]$ and let the binned vector be $d$. If $p_1 < 0.2$, $d = [0, 1]$; if $p_1 > 0.8$, $d = [1, 0]$; and if $0.2 \leq p_1 \leq 0.8$, $d = [0.5, 0.5]$. As such, each agent sends one of the three categorical vectors during each round of communication.
Vocabulary size 3. Here, the model outputs a probability vector \( p \) of length 3: \([p_1, p_2, p_3]\). The procedure for obtaining the binned vector \( d \) is described below:

\[
d = \begin{cases} 
[1, 0, 0] & \text{if } p_1 > 0.75 \\
[0, 1, 0] & \text{if } p_2 > 0.75 \\
[0, 0, 1] & \text{if } p_3 > 0.75 \\
[0, 0.5, 0.5] & \text{if } \max(p_1, p_2, p_3) < 0.75 \\
& \text{and } p_1 < p_2, p_3 \\
[0.5, 0.5] & \text{if } \max(p_1, p_2, p_3) < 0.75 \\
& \text{and } p_2 < p_1, p_3 \\
[0.5, 0.5, 0] & \text{if } \max(p_1, p_2, p_3) < 0.75 \\
& \text{and } p_3 < p_2, p_1
\end{cases}
\]

Under this formulation, each agent can be considered to send only a discrete symbol to the other agent during communication.
Figure 10. Egocentric visualization of $S$-Comm communication symbol $m^O_{A \rightarrow N}$ for vocabulary size 3. 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 three. 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 six plots on the left 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.

Figure 11. Egocentric visualization of $S$-Comm communication symbol $m^O_{O \rightarrow N}$ for vocabulary size 3. 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.
Figure 12. Egocentric visualization of S-Comm communication symbol $m^2_{O \rightarrow N}$ for 2-ON. 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 1st dimension of the message, while the plot on the right corresponds to the 2nd dimension. The value of each dimension is indicated by the color hue.

Figure 13. Egocentric visualization of S-Comm communication symbol $m^3_{O \rightarrow N}$. First and second row show the case when the goals are distinguishable and indistinguishable by $A_O$ respectively. 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. Notice that first row symbols have lesser overlap than the second row symbols.
Figure 14. **Example navigation episode with communication message** $m_{N \rightarrow O}$ **on the agent trajectory for S-Comm**. The message $m_{N \rightarrow O}$ is depicted by the color of the arrow symbol at various points on $A_N$’s trajectory on the top-down map. The sequence of maps from top-to-bottom visualizes the trajectory at different points in time. Egocentric observations and map representations at specific agent positions are given to the right of each map. Note the changed communication symbol (from blue agent symbol to green) after the first green goal is found and the agent proceeds to the next black goal.
Figure 15. Example navigation episode with communication message $m_{O \rightarrow N}$ on the agent trajectory for S-Comm. The message $m_{O \rightarrow N}$ is depicted by the color of the arrow symbol at various points on $A_N$’s trajectory on the top-down map. The sequence of maps from top-to-bottom visualizes the trajectory at different points in time. Egocentric observations and map representations at specific agent positions are given to the right of each map. Note how the communication symbol changes as the relative location of the goal object with respect to the agent changes: when the goal is not ahead of the agent, it is blue; when the goal is ahead of the agent but far away, it is green; and when the goal is in front of the agent, it is orange.