Two Body Problem: Collaborative Visual Task Completion

Unnat Jain[1]*† Luca Weihs[2]* Eric Kolve[2] Mohammad Rastegari[2,4]
Svetlana Lazebnik[1] Ali Farhadi[2,3,4] Alexander Schwing[1] Aniruddha Kembhavi[2]

[1]UIUC [2]PRIOR @ Allen Institute for AI [3]University of Washington [4]Xnor.ai

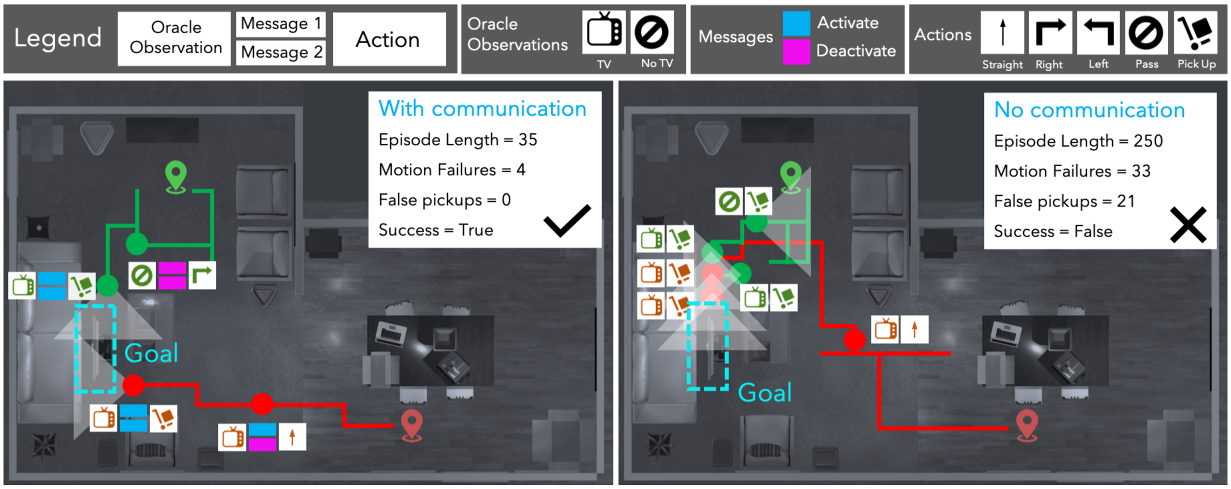


Figure 1: Two agents learn to successfully navigate through a previously unseen environment to find, and jointly lift, a heavy TV. Without learned communication, agents attempt many failed actions and pickups. With learned communication, agents send a message when they observe or when they intend to interact with the TV. The agents also learn to grab the opposite ends of the TV and coordinate to do so.

Abstract

Collaboration is a necessary skill to perform tasks that are beyond one agent's capabilities. Addressed extensively in both conventional and modern AI, multi-agent collaboration has often been studied in the context of simple grid worlds. We argue that there are inherently visual aspects to collaboration which should be studied in visually rich environments. A key element in collaboration is communication that can be either explicit, through messages, or implicit, through perception of the other agents and the visual world. Learning to collaborate in a visual environment entails learning (1) to perform the task, (2) when and what to communicate, and (3) how to act based on these communications and the perception of the visual world. In this paper we study the problem of learning to collaborate directly from pixels in AI2-THOR and demonstrate the benefits of explicit and implicit modes of communication to perform visual tasks. Refer to our project page for more details: https://prior.allenai.org/projects/two-body-problem

1. Introduction

Developing collaborative skills is known to be more cognitively demanding than learning to perform tasks independently. In AI, multi-agent collaboration has been studied in more conventional [32, 43, 9, 58] and modern settings [53, 28, 79, 35, 56, 61]. These studies have mainly been performed on grid-worlds and have factored out the role of perception in collaboration.

In this paper we argue that there are aspects of collaboration that are inherently visual. Studying collaboration in simplistic environments does not permit to observe the interplay between perception and communication, which is necessary for effective collaboration. Imagine moving a piece of furniture with a friend. Part of the collaboration is rooted in explicit communication through exchanging messages, and some part of it is done through implicit communication through interpreting perceivable cues about the other agents behavior. If you see your friend going around the furniture to grab it, you would naturally stay on the opposite side to avoid toppling it over. Additionally, communication and collaboration should be considered jointly with the task itself. The way you communicate, either explicitly or implicitly, in a soccer game is very different from when you move furniture. This suggests that factoring out per-

*indicates equal contributions.
†work partially done as an intern at Allen Institute for AI

ception and studying collaboration in isolation (grid-world) might not result in an ideal outcome.

In short, learning to perform tasks collaboratively in a visual environment entails joint learning of (1) how to perform tasks in that environment, (2) when and what to communicate, and (3) how to act based on implicit and explicit communication. In this work, we develop one of the first frameworks that enables the study of explicitly and implicitly communicating agents collaborating together in a photo-realistic environment.

To this end we consider the problem of finding and lifting bulky items, ones which cannot be lifted by a single agent. While conceptually simple, attaining proficiency in this task requires multiple stages of communication. The agents must search for the object of interest in the environment (possibly communicating their findings to each other), position themselves appropriately (for instance, opposing each other), and then lift the object simultaneously. If the agents position themselves incorrectly, lifting the object will cause it to topple over. Similarly, if the agents pick up the object at different time steps, they will not succeed.

To study this task, we use the AI2-THOR virtual environment [48], a photo-realistic, physics-enabled environment of indoor scenes used in past work to study single agent behavior. We extend AI2-THOR to enable multiple agents to communicate and interact.

We explore collaboration along several modes: (1) The benefits of communication for spatially constrained tasks (*e.g.*, requiring agents to stand across one another while lifting an object) *vs*. unconstrained tasks. (2) The ability of agents to implicitly and explicitly communicate to solve these tasks. (3) The effect of the expressivity of the communication channel on the success of these tasks. (4) The efficacy of these developed communication protocols on known environments and their generalizability to new ones. (5) The challenges of egocentric visual environments *vs*. grid-world settings.

We propose a Two Body Network, or TBONE, for modeling the policies of agents in our environments. TBONE operates on a visual egocentric observation of the 3D world, a history of past observations and actions of the agent, as well as messages received from other agents in the scene. At each time step, agents go through two rounds of communication, akin to sending a message each and then replying to messages that are received in the first round. TBONE is trained with a warm start using a variant of DAgger [70], followed by a minimization of a sum of an A3C loss and a cross entropy loss between the agents actions and the actions of an expert policy.

We perform a detailed experimental analysis of the impact of communication using metrics including accuracy, number of failed pickup actions, and episode lengths. Following our above research questions, our findings show that: (1) Communication clearly benefits both constrained and unconstrained tasks but is more advantageous for constrained tasks. (2) Both explicit and implicit communication are exploited by our agents and both are beneficial, individually and jointly. (3) For our tasks, large vocabulary sizes are beneficial. (4) Our agents generalize well to unseen environments. (5) Abstracting our environments towards a grid-world setting improves accuracy, confirming our notion that photo-realistic visual environments are more challenging than grid-world like settings. This is consistent with findings by past works for single agent scenarios.

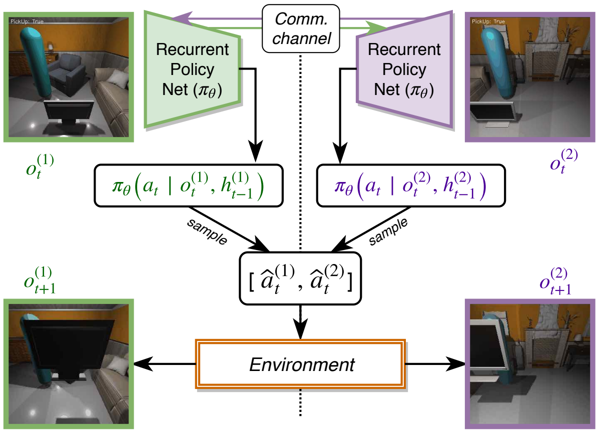

PickUp: True
Comm. channel
Recurrent Policy Net (π_θ)
Recurrent Policy Net (π_θ)
PickUp: True
$o_t^{(1)}$
$\pi_\theta\left(a_t \mid o_t^{(1)}, h_{t-1}^{(1)}\right)$
$\pi_\theta\left(a_t \mid o_t^{(2)}, h_{t-1}^{(2)}\right)$
$o_t^{(2)}$
sample
sample
$o_{t+1}^{(1)}$
$[\,\hat{a}_t^{(1)}, \hat{a}_t^{(2)}]$
$o_{t+1}^{(2)}$
Environment

Figure 2: A schematic depicting the inputs to the policy network. An agent's policy operates on a partial observation of the scene's state and a history of previous observations, actions, and messages received.

Finally we interpret the explicit mode of communication between agents by fitting logistic regression models to the messages to predict the values such as oracle distance to target, next action, *etc*., and find strong evidence matching our intuitions about the usage of messages between agents.

2. Related Work

We now review related work in the directions of visual navigation, navigation and language, visual multi-agent reinforcement learning (RL), and virtual learning environments employed in past works to evaluate algorithms.

Visual Navigation: A large body of work focuses on visual navigation, *i.e.*, locating a target using only visual input. Prominent early map-based navigation methods [47, 6, 7, 64] use a global map to make decisions. More recent approaches [76, 87, 23, 85, 46, 71] reconstruct the map on the fly. Simultaneous localization and mapping [84, 74, 24, 12, 67, 77] consider mapping in isolation. Upon having obtained a map of the environment, planning methods [13, 44, 52] yield a sequence of actions to achieve the goal. Combinations of joint mapping and planning have also been discussed [27, 50, 49, 31, 3]. Map-less methods [38, 54, 69, 72, 66, 92, 36] often formulate the task as obstacle avoidance given an input image or reconstruct a map implicitly. Conceptually, for visual navigation, we

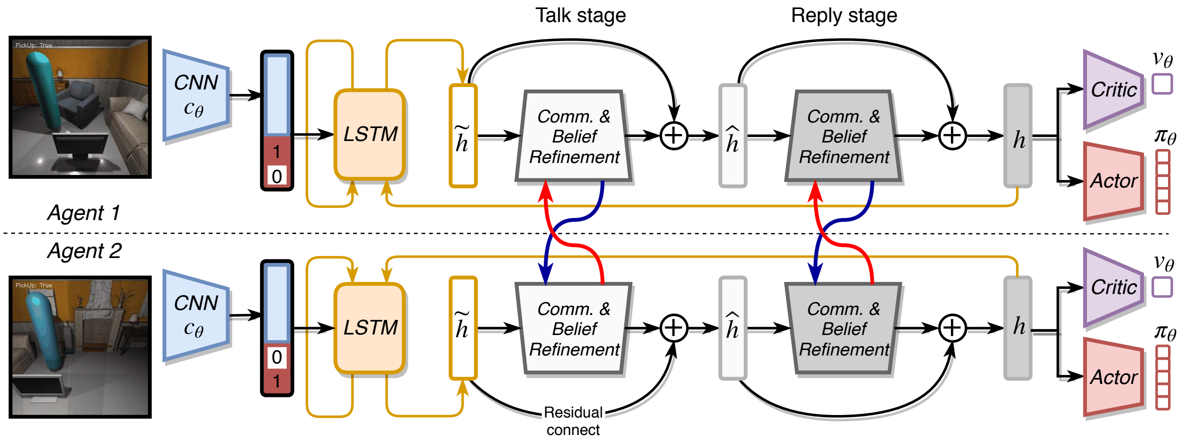


Figure 3: Overview of our TBONE architecture for collaboration.

must learn a mapping from visual observations to actions which influence the environment. Consequently the task is well suited for an RL formulation, a perspective which has become popular recently [62, 1, 16, 17, 33, 42, 86, 59, 5, 8, 90, 25, 36, 91, 37]. Some of these approaches compute actions from observations directly while others attempt to explicitly/implicitly reconstruct a map.

Following recent techniques, our proposed approach also uses RL for visual navigation. While our proposed approach could be augmented with explicit or implicit maps, our focus is upon multi-agent communication. In the spirit of factorizing out orthogonal extensions from the model, we defer such extensions to future work.

Navigation and Language: Another line of work has focused on communication between humans and virtual agents. These methods more accurately reflect real-world scenarios since humans are more likely to interact with an agent using language rather than abstract specifications. Recently Das *et al.* [19, 21] and Gordon *et al.* [34] proposed to combine question answering with robotic navigation. Chaplot *et al.* [15], Anderson *et al.* [2] and Hill *et al.* [39] propose to guide a virtual agent via language commands.

While language directed navigation is an important task, we consider an orthogonal direction where multiple agents need to collaboratively solve a specified task. Since visual multi-agent RL is itself challenging, we refrain from introducing natural language complexities. Instead, in this paper, we are interested in developing a systematic understanding of the utility and character of communication strategies developed by multiple agents through RL.

Visual Multi-Agent Reinforcement Learning: Multi-agent systems result in non-stationary environments posing significant challenges. Multiple approaches have been proposed over the years to address such concerns [82, 83, 81, 30]. Similarly, a variety of settings from multiple cooperative agents to multiple competitive ones have been investigated [51, 65, 57, 11, 63, 35, 56, 29, 61].

Among the plethora of work on multi-agent RL, we want to particularly highlight work by Giles and Jim [32], Kasai *et al.* [43], Bratman *et al.* [9], Melo *et al.* [58], Lazaridou *et al.* [53], Foerster *et al.* [28], Sukhbaatar *et al.* [79] and Mordatch and Abbeel [61], all of which investigate the discovery of communication and language in the multi-agent setting using maze-based tasks, tabular setups, or Markov games. For instance, Lazaridou *et al.* [53] perform experiments using a referential game of image guessing, Foerster *et al.* [28] focus on switch-riddle games, Sukhbaatar *et al.* [79] discuss multi-turn games on the MazeBase environment [80], and Mordatch and Abbeel [61] evaluate on a rectangular environment with multiple target locations and tasks. Most recently, Das *et al.* [20] demonstrate, especially in grid-world settings, the efficacy of targeted communication where agents must learn to whom they should send messages.

Our work differs from the above body of work in that we consider communication for visual tasks, *i.e.*, our agents operate in rich visual environments rather than a grid-like maze, a tabular setup or a Markov game. We are particularly interested in investigating how communication and perception support each other.

Reinforcement Learning Environments: As just discussed, our approach is evaluated on a rich visual environment. Suitable environment simulators are AI2-THOR [48], House3D [88], HoME [10], MINOS [73] for Matterport3D [14] and SUNCG [78]. Common to these environments is the goal of modeling real world living environments with substantial visual diversity. This is in contrast to other RL environments such as the arcade environment [4], Vizdoom [45], block towers [55], Malmo [41], TORCS [89], or MazeBase [80]. Of these environments, we chose AI2-THOR as it was easy to extend, provides high fidelity images, and has interactive physics enabled scenes, opening up interesting multi-agent research directions beyond this current work.

3. Collaborative Task Completion

We are interested in understanding how two agents can learn, from pixels, to communicate so as to effectively and collaboratively solve a given task. To this end, we develop a

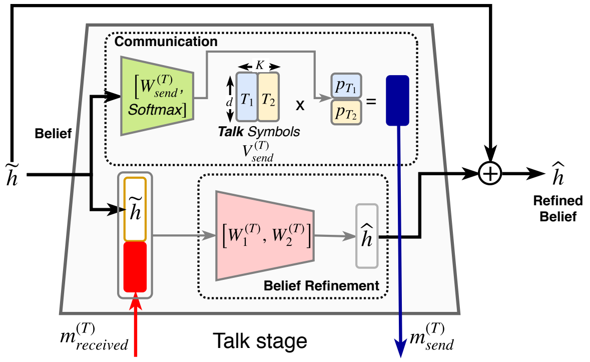


Figure 4: Communication and belief refinement module for the *talk* stage (marked with the superscript of (T)) of explicit communication. Here our vocab. is of size $K = 2$.

task for two agents which consists of two components, each tailored to a desirable skill for indoor agents. The components are: (1) visual navigation, which the agents may solve independently, but which may also benefit from some collaboration; and (2) jointly synchronized interaction with the environment, which typically requires collaboration to succeed. The choice of these components stems from the fact that navigating to a desired position in an environment or to locate a desired object is a quintessential skill for an indoor agent, and synchronized interaction is fundamental to understanding any collaborative multi-agent setting.

We first discuss the collaborative task more formally, then detail the components of our network, TBONE, used to complete the task.

3.1. Task: Find and Lift Furniture

We task two agents to lift a heavy target object in an environment, a task that cannot be completed by a single agent owing to the weight of the object. The two agents as well as the target object are placed at random locations in a randomly chosen AI2-THOR living room scene. Both agents must locate the target, approach it, position themselves appropriately, and then simultaneously lift it.

To successfully complete the task, both agents perform actions over time according to the same learned policy (Fig. 2). Since our agents are homogeneous, we share the policy parameters for both agents. Previous works [35, 61] have found this to train agents more efficiently. For an agent, the policy operates on (1) an ego-centric observation of the environment as well as a previous history of (a) observations, (b) actions taken by the agent, and (c) messages sent by the other agent. At each time step, the two agents process their current observations and then perform two rounds of explicit communication. Each round of communication involves each of the agents sending a single message to the other. The agents also have the ability to watch the other agent (when in view) and possibly even recognize their actions over time, thereby using implicit communication as a means of gathering information.

More formally, an agent perceives the scene at time t in the form of an image o_t and chooses its action $a_t \in \mathcal{A}$ by computing a policy, *i.e.*, a probability distribution $\pi_\theta(a_t|o_t, h_{t-1})$, over all actions $a_t \in \mathcal{A}$. In our case, the images o_t are first-person views obtained from AI2-THOR. Following classical recurrent models, our policy leverages information computed in the previous time-step via the representation h_{t-1}. The set of available actions $\mathcal{A}$ consists of the five options MOVEAHEAD, ROTATELEFT, ROTATERIGHT, PASS, and PICKUP. The actions MOVEAHEAD, ROTATELEFT, and ROTATERIGHT allow the agent to navigate. To simplify the complexities of continuous time movement we let a single MOVEAHEAD action correspond to a step of size 0.25 meters, a single ROTATERIGHT action correspond to a 90 degree rotation clockwise, and a single ROTATELEFT action correspond to a 90 degree rotation anti-clockwise. The PASS action indicates that the agent should stand-still and PICKUP is the agent's attempt to pick up the target object. Critically, the PICKUP action has the desired effect only if three preconditions are met, namely both agents must (1) be within 1.5 meters of the object and be looking directly at it, (2) be a minimum distance away from one another, and (3) carry out the PICKUP action simultaneously. Note that asking agents to be at a minimum distance from one another amounts to adding specific constraints on their relative spatial layouts with regards to the object and hence requires the agents to reason about such relationships. This is akin to requiring the agents to stand across each other when they pick up the object. The motivation to model spatial constraints with a minimum distance constraint is to allow us to easily manipulate the complexity of the task. For instance, setting this minimum distance to 0 loosens the constraints and only requires agents to meet two of the above preconditions.

In our experiments, we train agents to navigate within and interact with 30 indoor environments. Specifically, an episode is considered successful if both agents navigate to a known object and, jointly, lift it within a fixed number of time steps. As our focus is the study of collaboration and not primarily object recognition, we keep the sought object, a television, constant. Importantly, environments as well as the agents' start locations and the target object location are randomly assigned at the start of each episode. Consequently, the agents must learn to (1) search for the target object in different environments, (2) navigate towards it, (3) stay within the object's vicinity until the second agent arrives, (4) coordinate that both agents are apart from each other by at least the specified distance, and (5) finally and jointly perform the pickup action.

Intuitively, we expect the agents to perform better on this task if they can communicate with each other. We conjecture that explicit communication will allow them to both signal when they have found the object and, after naviga-

Data	Accuracy	Reward	Missed pickups	Unsuccess. pickups
Visual	59.0 ±4.0	-2.7 ±0.3	0.3 ±0.09	2.9 ±0.8
Visual+depth	65.7 ±3.9	-2.0 ±0.3	0.4 ±0.1	3.2 ±0.9
Grid-world	**78.2 ±3.4**	**-0.6 ±0.2**	**0.1 ±0.05**	**0.7 ±0.1**

Table 1: Effect of adding oracle depth as well as moving to a grid-world setting on unseen scenes, *Constrained* task.

tion, help coordinate when to attempt a PICKUP, whereas implicit communication will help to reason about their relative locations with regards to each other and the object. To measure the impact of explicit and implicit means of communication in the given task, we train models with and without message passing as well as by making agents (in)visible to one another. Explicit communication would seem to be especially important in the case where implicit communication isn't possible. Without any communication, there seems to be no better strategy than for both agents to independently navigate to the object and then repeatedly try PICKUP actions in the hope that they will be, at some point, in sync. The expectation that such a policy may be forthcoming gives rise to one of our metrics, namely the count of failed pickup events among both agents in an episode. We discuss metrics and results in Section 4.

3.2. Network Architecture

In the following we describe the learned policy (actor) $\pi_\theta(a_t|o_t, h_{t-1})$ and value (critic) $v_\theta(o_t, h_{t-1})$ functions for each agent in greater detail. See Fig. 3 for a high level visualization of our network structure. Let θ represent a catch-all parameter encompassing all the learnable weights in TBONE. At the t-th timestep in an episode we obtain as an agent's observation, from AI2-THOR, a $3 \times 84 \times 84$ RGB image o_t which is then processed by a four layer CNN c_θ into the 1024-dimensional vector $c_\theta(o_t)$. Onto $c_\theta(o_t)$ we append an 8-dimensional learnable embedding e which, unlike all other weights in the model, is not shared between the two agents. This agent embedding e gives the agents the capacity to develop distinct complementary strategies. The concatenation of $c_\theta(o_t)$ and e is fed, along with historical embeddings from time $t-1$, into a long-short-term-memory (LSTM) [40] cell resulting in a 512-dimensional output vector $\widetilde{h}_t$ capturing the beliefs of the agent given its prior history and most recent observation. Intuitively, we now would like the two agents to refine their beliefs via communication before deciding on a course of action. We consider this process in several stages (Fig. 4).

Communication: We model communication by allowing the agents to send one another a d-dimensional vector derived by performing soft-attention over a vocabulary of a fixed size K. More formally, let $\mathbf{W}_{\text{send}} \in \mathbb{R}^{K\times 512}$, $\boldsymbol{b}_{\text{send}} \in \mathbb{R}^{512}$, and $\boldsymbol{V}_{\text{send}} \in \mathbb{R}^{d\times K}$ be (learnable) weight matrices with the columns of $\boldsymbol{V}_{\text{send}}$ representing our vocabulary. Then, given the representation $\widetilde{h}_t$ described above, the agent computes soft-attention over the vocabulary producing the message $m_{\text{send}} = \boldsymbol{V}_{\text{send}}\,\text{softmax}(\mathbf{W}_{\text{send}}\,\widetilde{h}_t + \boldsymbol{b}_{\text{send}}) \in \mathbb{R}^d$, which is relayed to the other agent.

Belief Refinement: Given the agents' current beliefs $\widetilde{h}_t$ and the message m_{received} from the other agent, we model the process of refining one's beliefs given new information using a two layer fully connected neural network with a residual connection. In particular, $\widetilde{h}_t$ and m_{received} are concatenated, and new beliefs $\hat{h}_t$ are formed by computing $\hat{h}_t = \widetilde{h}_t + \text{ReLU}(\mathbf{W}_2\,\text{ReLU}(\mathbf{W}_1[\widetilde{h}_t\,;\,m_{\text{received}}] + \boldsymbol{b}_1) + \boldsymbol{b}_2)$, where $\mathbf{W}_1 \in \mathbb{R}^{512\times(512+d)}$, $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^{512}$, and $\mathbf{W}_2 \in \mathbb{R}^{512\times 512}$ are learnable weight matrices. We set the value of d to 8.

Reply and Additional Refinement: The above step is followed by one more round of communication and belief refinement by which the representation $\hat{h}_t$ is transformed into h_t. These additional stages have new sets of learnable parameters including a new vocabulary matrix. Note that, unlike in the standard LSTM framework where $\widetilde{h}_{t-1}$ would be fed into the cell at time t, we instead give the LSTM cell the refined vector h_{t-1}.

Linear Actor and Critic: Finally the policy and value functions are computed as $\pi_\theta(a_t|o_t, h_{t-1}) = \text{softmax}(\mathbf{W}_{\text{actor}}\,h_t + \boldsymbol{b}_{\text{actor}})$, and $v_\theta(o_t, h_{t-1}) = \mathbf{W}_{\text{critic}}\,h_t + \boldsymbol{b}_{\text{critic}}$ where $\mathbf{W}_{\text{actor}} \in \mathbb{R}^{5\times 512}$, $\boldsymbol{b}_{\text{actor}} \in \mathbb{R}^5$, $\mathbf{W}_{\text{critic}} \in \mathbb{R}^{1\times 512}$, and $\boldsymbol{b}_{\text{critic}} \in \mathbb{R}^1$ are learned.

3.3. Learning

Similar to others [19, 36, 18, 22], we found training of our agents from scratch to be infeasible when using a pure reinforcement learning (RL) approach, *e.g.*, with asynchronous actor critic (A3C) [60], even in simplified settings, without extensive reward shaping. Indeed, often the agents must make upwards of 60 actions to navigate to the object and will only successfully complete the episode and receive a reward if they jointly pick up the object. This setting of extremely sparse rewards is a well known failure mode of standard RL techniques. Following the above prior work, we use a "warm-start" by training with a variant of DAgger [70]. We train our models online using imitation learning for 10,000 episodes with actions for episode i sampled from the mixture $(1-\alpha_i)\pi_{\theta_{i-1}} + \alpha_i\pi^*$ where θ_{i-1} are the parameters learned by the model up to episode i, π^* is an expert policy (described below), and α_i decays linearly from 0.9 to 0 as i increases. This initial warm-start allows the agents to learn a policy for which rewards are far less sparse, allowing traditional RL approaches to be applicable. Note that our expert supervision only applies to the actions, there is no supervision for how agents should communicate. Instead the agents must learn to communicate in such a way that would increase the probability of expert actions.

After the warm-start period, trajectories are sampled purely from the agent's current policy and we train our agents by minimizing the sum of an A3C loss, and a cross entropy loss between the agents' actions and the actions of

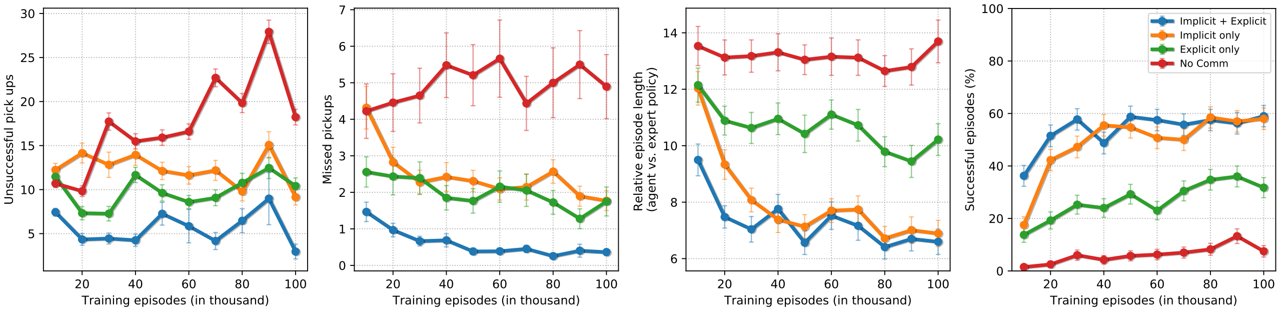


Figure 5: Unseen scenes metrics (*Constrained* task): (a) Failed pickups (b) Missed pickups (c) Relative ep. len (d) Accuracy.

an expert policy. The A3C and cross entropy losses here are complementary, each helping correct for a deficiency in the other. Namely, the gradients from an A3C loss tend to be noisy and can, at times, derail or slow training; the gradients from the cross entropy loss are noise free and thereby stabilize training. A pure cross entropy loss however fails to sufficiently penalize certain undesirable actions. For instance, diverging from the expert policy by taking a MOVEAHEAD action when directly in front of a wall should be more strongly penalized than when the area in front of the agent is free as the former case may result in damage to the agent; both these cases are penalized equally by a cross entropy loss. The A3C loss, on the other hand, accounts for such differences easily so long as they are reflected by the rewards the agent receives.

We now describe the expert policy. If both agents can see the TV, are within 1.5 meters of it, and are at least a given minimum distance apart from one another then the expert action is to PICKUP for both agents. Otherwise given a fixed scene and TV position we obtain, from AI2-THOR, the set $T = \{t_1, \ldots, t_m\}$ of all positions (on a grid with square size 0.25 meters) and rotations within 1.5 meters of the TV from which the TV is visible. Letting ℓ_{ik} be the length of the shortest path from the current position of agent $i \in \{0, 1\}$ to t_k we then assign each $(t_j, t_k) \in T \times T$ the score $s_{jk} = \ell_{0j} + \ell_{1k}$. We then compute the lowest scoring tuple $(s, t) \in T \times T$ for which s and t are at least a given minimum distance apart and assign agent 0 the expert action corresponding to the first navigational step along the shortest path from agent 0 to s (and similarly for agent 1 whose expert goal is t).

Note that our training strategy and communication scheme can be extended to more than two agents. We defer such an analysis to future work, a careful analysis of the two-agent setting being an appropriate first step.

Implementation Details. Each model was trained for 100,000 episodes. Each episode is initialized in a random train (seen) scene of AI2-THOR. Rewards provided to the agents are: 1 to both agents for a successful pickup action, constant -0.01 step penalty to discourage long trajectories, -0.02 for any failed action (*e.g.*, running into a wall) and -0.1 for a failed pickup action. Episodes run for a maximum of 500 steps (250 steps for each agent) after which the episode is considered failed.

4. Experiments

In this section, we present our evaluation of the effect of communication towards collaborative visual task completion. We first briefly describe the multi-agent extensions made to AI2-THOR, the environments used for our analysis, the two tasks used as a test bed and metrics considered. This is followed by a detailed empirical analysis of the tasks. We then provide a statistical analysis of the explicit communication messages used by the agents to solve the tasks, which sheds light on their content. Finally we present qualitative results.

Framework and Data. We extend the AI2-THOR environment to support multiple agents that can each be independently controlled. In particular, we extend the existing initialization action to accept an `agentCount` parameter allowing an arbitrarily large number of agents to be specified. When additional agents are spawned, each is visually depicted as a capsule of a distinct color. This allows agents to observe each other's presence and impact on the environment, a form of implicit communication. We also provide a parameter to render agents invisible to one another, which allows us to study the benefits of implicit communication. Newly spawned agents have the full capabilities of a single agent, being able to interact with the environment by, for example, picking up and opening objects. These changes are publicly available with AI2-THOR v1.0. We consider the 30 AI2-THOR living room scenes for our analysis, since they are the largest in terms of floor area and also contain a large amount of furniture. We train on 20 and test on the 20 seen scenes as well as the remaining 10 unseen ones.

Tasks. We consider two tasks, both requiring the two agents to simultaneously pick up the TV in the environment: (1) *Unconstrained*: No constraints are imposed here with regards to the locations of the agents with respect to each other. (2) *Constrained*: The agents must be at least 8 steps from each other (akin to requiring them to stand across each other when they pick up the object). Intuitively, we expect the *Constrained* setting to be more difficult than the *Unconstrained*, since it requires the agents to spatially reason

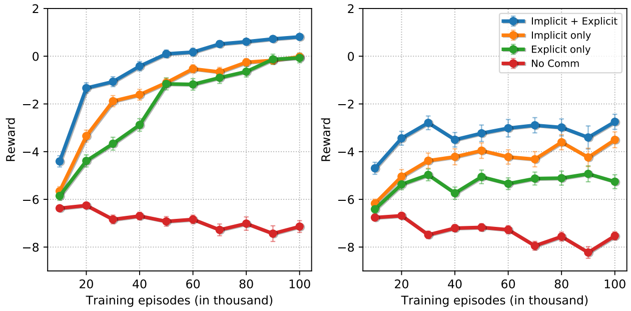


Figure 6: Reward *vs*. training episodes on the *Constrained* task. (left) Seen scenes (right) Unseen scenes.

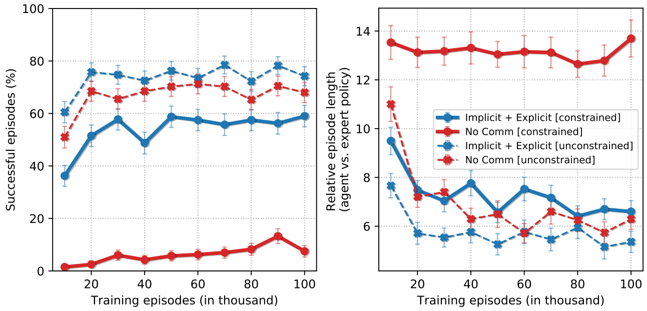


Figure 7: *Constrained vs. unconstrained* task (on unseen scenes): (left) Accuracy, (right) Relative episode length.

about themselves and objects in the scene. For each of the above tasks, we train 4 variants of TBONE, resulting from switching explicit and implicit communication on and off. Switching off implicit communication amounts to rendering the *other* agent invisible.

Metrics. We consider the following metrics: (1) *Reward*, (2) *Accuracy*: % successful episodes, (3) Number of *Failed pickups*, (4) Number of *Missed pickups*: where both agents could have picked up the object but did not, (5) *Relative episode length*: relative to an oracle. These metrics are aggregated over 400 random initializations (Unseen scenes: 10 scenes × 40 inits, Seen scenes: 20 scenes × 20 inits). Note that accuracy alone isn't revealing enough. Naïve agents that wander around and randomly pick up objects will eventually succeed. Also, agents that correctly locate the TV and then keep attempting a pickup in the hope of synchronizing with the other agent will also succeed. Both these cases will however do poorly on the other metrics.

Quantitative analysis. All plots and metrics referenced in this section contain 90% confidence intervals.

Fig. 5 compares the four metrics: Accuracy, Failed pickups, Missed pickups, and Relative episode length for unseen scenes and the *Constrained* task. With regards to accuracy, explicit+implicit communication fares only moderately better than implicit communication, but the need for explicit communication is dramatic in the absence of an implicit one. But when one considers all metrics, the benefits of having both explicit and implicit communication are clearly visible. The number of failed and missed pickups is lower, while episode lengths are a little better than just using implicit communication. The differences between just explicit *vs*. just implicit also shrink when looking at all metrics together. However, across the board, it is clear that communicating is advantageous over not communicating.

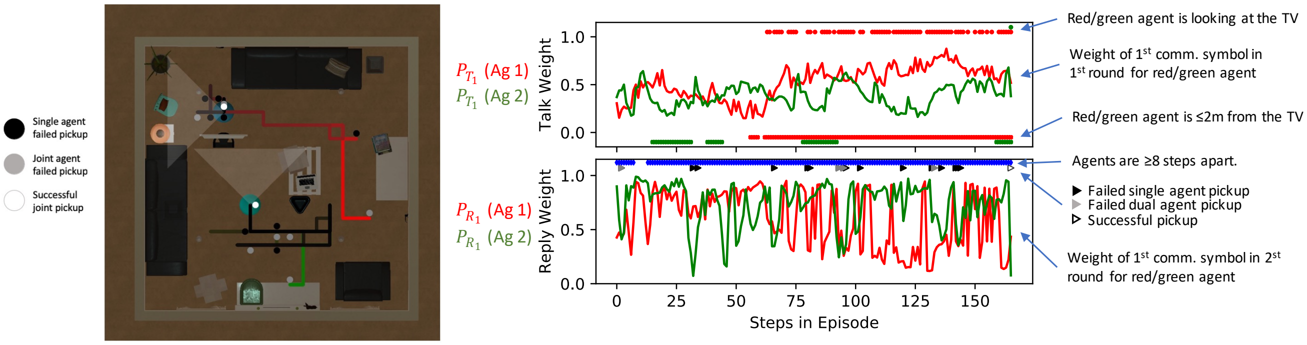
Fig. 6 shows the rewards obtained by the 4 variants of our model on seen and unseen environments for the *Constrained* task. While rewards on seen scenes are unsurprisingly higher, the models with communication do generalize well to unseen environments. Adding the two means of communication is more beneficial than either and far better than not having any means of communication. Interestingly just implicit communication fares better than just explicit, on accuracy.

Fig. 7 presents the accuracy and relative episode lengths metrics for the unseen scenes and *Unconstrained* task in contrast to the *Constrained* task. In these plots, for brevity we only consider the extreme cases of having full communication *vs*. no communication. As expected, the *Unconstrained* setting is easier for the agents with higher accuracy and lower episode lengths. Communication is also advantageous in the *Unconstrained* setting, but its benefits are lesser compared to the *Constrained* setting.

Table 1 shows a large jump in accuracy when we provide a perfect depth map as an additional input on the *Constrained* task, indicating that improved perception is beneficial to task completion. We also obtained significant jumps in accuracy (from 31.8 ± 3.8 to 37.2 ± 4.0) when we increase the size of our vocabulary from 2 to 8. This analysis was performed in the explicit-only communication and *Constrained* environment setup. However, note that even with a vocabulary of 2, agents may be using the full continuous spectrum to encode more nuanced events.

Grid-world abstraction. In order to assess impact of learning to communicate from pixels rather than, as in most prior work, from grid-world environments, we perform a direct translation of our task into a grid-world and compare its performance to our best model. We transform the 1.25m × 2.75m area in front of our agent into a 5×11 grid where each square is assigned a 16 dimensional embedding based on whether it is free space, occupied by another agent, occupied by the target object, otherwise unreachable, or unknown (in the case the grid square leaves the environment). The agents then move in AI2-THOR but perceive this partially observable grid-world. Agents in this setting acquire a large bump in accuracy on the *Constrained* task (Table 1), confirming our claim that photo-realistic visual environments are more challenging than grid-world like settings.

Interpreting Communication. While we have seen, in Section 4, that communication can substantially benefit our task, we now investigate *what* these agents have learned to communicate. We focus on the communication strategies learned by agents with a vocabulary of two in the



(a) *Constrained* setting agent trajectories

(b) Communication between agents

Figure 8: Single episode trajectory with associated agent communication.

	$\beta^{\leq}$	$\beta_t^{\leq}$	$\beta_r^{\leq}$	β^{see}	β_t^{see}	β_r^{see}
Est.	0.35	1.23	-0.35	0.88	0.59	-1.1
SE	0.013	0.019	0.013	0.013	0.015	0.013

	β^{pick}	$\beta_{t,0}^{\text{pick}}$	$\beta_{r,0}^{\text{pick}}$	$\beta_{t,1}^{\text{pick}}$	$\beta_{r,1}^{\text{pick}}$	$\beta_{\vee,r}^{\text{pick}}$
Est	1.06	-0.01	-0.04	0	-0.03	-1.09
SE	0.012	0.007	0.006	0.007	0.006	0.021

Table 2: Estimates, and corresponding robust bootstrap standard errors, of the parameters from Section 4.

Constrained setting. Fig. 8 displays one episode trajectory of the two agents with the corresponding communication. From Fig. 8(b) we generate hypotheses regarding communication strategies. Suppressing the dependence on episode and step, for $i \in \{0, 1\}$ let t_i be the weight assigned by agent i to the 1st element of the vocabulary in the 1st round of communication, and similarly let r_i be as t_i but for the 2nd round of communication. When the agent with the red trajectory (henceforth called agent 0 or A_0) begins to see the TV the weight t_0 increases and remains high until the end of the episode. This suggests that the 1st round of communication may be used to signify closeness to or visibility of the TV. On the other hand, the pickup actions taken by the two agents are associated with the agents making r_0 and r_1 simultaneously small.

To add evidence to these hypotheses we fit logistic regression models to predict, from (functions of) t_i and r_i, two oracle values (*e.g.*, whether the TV is visible) and whether or not the agents will attempt a pickup action. As the agents are largely symmetric we take the perspective of A_0 and define the models $\sigma^{-1}\ P\big(A_0 \text{ is} \leq \text{2m from the TV}\big) = \beta^{\leq} + \beta_t^{\leq} t_0 + \beta_r^{\leq} r_0$, $\sigma^{-1}\ P\big(A_0 \text{ sees TV and is} \leq 1.5\text{m from it}\big) = \beta^{\text{see}} + \beta_t^{\text{see}} t_0 + \beta_r^{\text{see}} r_0$, and $\sigma^{-1}\ P\big(A_0 \text{ attempts a pickup action}\big) = \beta^{\text{pick}} + \sum_{i\in\{0,1\}}(\beta_{t,i}^{\text{pick}} t_i + \beta_{r,i}^{\text{pick}} r_i) + \beta_{\vee,r}^{\text{pick}} \max(r_0, r_1)$ where σ^{-1} is the logit function. Details of how these models are fit can be found in the appendix.

From Table 2, which displays the estimates of the above parameters along with their standard errors, we find strong evidence for the above intuitions. Note, for all of the estimates discussed above, the standard errors are very small, suggesting highly statistically significant results. The large positive coefficients associated with $\beta_t^{\leq}$ and β_t^{see} suggest that, conditional on r_0 being held constant, an increase in the weight t_0 is associated with a higher probability of A_0 being near, and seeing, the TV. Note also that the estimated value of β_r^{see} is fairly large in magnitude and negative. This is very much in line with our prior hypothesis that r_0 is made small when agent 0 wishes to signal a readiness to pickup the object. Finally, essentially all estimates of coefficients in the final model are close to 0 except for $\beta_{\vee,r}^{\text{pick}}$ which is large and negative. Hence, conditional on other values being fixed, $\max(r_0, r_1)$ being small is associated with a higher probability of a subsequent pickup action. Of course $r_0, r_1 \leq \max(r_0, r_1)$ again lending evidence to the hypothesis that the agents coordinate pickup actions by setting r_0, r_1 to small values.

5. Conclusion

We study the problem of learning to collaborate in visual environments and demonstrate the benefits of learned explicit and implicit communication to aid task completion. We compare performance of collaborative tasks in photo-realistic visual environments to an analogous grid-world environment, to establish that the former are more challenging. We also provide a statistical interpretation of the communication strategy learned by the agents.

Future research directions include extensions to more than two agents, more intricate real-world tasks and scaling to more environments. It would be exciting to enable natural language communication between the agents which also naturally extends to involving human-in-the-loop.

Acknowledgements: This material is based upon work supported in part by the National Science Foundation under Grants No. 1563727, 1718221, 1637479, 165205, 1703166, Samsung, 3M, Sloan Fellowship, NVIDIA Artificial Intelligence Lab, Allen Institute for AI, Amazon, AWS Research Awards and Thomas & Stacey Siebel Foundation.

References

[1] D. Abel, A. Agarwal, F. Diaz, A. Krishnamurthy, and R. E. Schapire. Exploratory gradient boosting for reinforcement learning in complex domains. *arXiv preprint arXiv:1603.04119*, 2016. 3

[2] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proc. CVPR*, 2018. 3

[3] A. Aydemir, A. Pronobis, M. Gbelbecker, and P. Jensfelt. Active visual object search in unknown environments using uncertain semantics. In *IEEE Trans. on Robotics*, 2013. 2

[4] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *J. of Artificial Intelligence Research*, 2013. 3

[5] S. Bhatti, A. Desmaison, O. Miksik, N. Nardelli, N. Siddharth, and P. H. S. Torr. Playing doom with slam-augmented deep reinforcement learning. *arXiv preprint arXiv:1612.00380*, 2016. 3

[6] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Trans. on Systems, Man and Cybernetics*, 1989. 2

[7] J. Borenstein and Y. Koren. The vector field histogram – fast obstacle avoidance for mobile robots. *IEEE Trans. on Robotics and Automation*, 1991. 2

[8] S. Brahmbhatt and J. Hays. Deepnav: Learning to navigate large cities. In *Proc. CVPR*, 2017. 3

[9] J. Bratman, M. Shvartsman, R. L. Lewis, and S. Singh. A new approach to exploring language emergence as boundedly optimal control in the face of environmental and cognitive constraints. In *Proc. Int.'l Conv. on Cognitive Modeling*, 2010. 1, 3

[10] S. Brodeur, E. Perez, A. Anand, F. Golemo, L. Celotti, F. Strub, J. Rouat, H. Larochelle, and A. Courville. HoME: a Household Multimodal Environment. In *https://arxiv.org/abs/1711.11017*, 2017. 3

[11] L. Busoniu, R. Babuska, and B. D. Schutter. A comprehensive survey of multiagent reinforcement learning. In *IEEE Trans. on Systems, Man and Cybernetics*, 2008. 3

[12] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. on Robotics*, 2016. 2

[13] J. Canny. *The complexity of robot motion planning*. MIT Press, 1988. 2

[14] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *International Conference on 3D Vision (3DV)*, 2017. 3

[15] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. R. Salakhutdinov. Gated-attention architectures for task-oriented language grounding. In *CoRR, abs/1706.07230*, 2017. 3

[16] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, 2015. 3

[17] S. Daftry, J. A. Bagnell, and M. Hebert. Learning transferable policies for monocular reactive mav control. In *Proc. ISER*, 2016. 3

[18] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? In *EMNLP*, 2016. 5

[19] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied Question Answering. In *Proc. CVPR*, 2018. 3, 5

[20] A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau. Tarmac: Targeted multi-agent communication. *arXiv preprint arXiv:1810.11187*, 2018. 3

[21] A. Das, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Neural Modular Control for Embodied Question Answering. In *Proc. ECCV*, 2018. 3

[22] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proc. ICCV*, 2017. 5

[23] A. J. Davison. Real time simultaneous localisation and mapping with a single camera. In *ICCV*, 2003. 2

[24] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. Structure from Motion without Correspondence. In *Proc. CVPR*, 2000. 2

[25] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel. Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016. 3

[26] B. Efron. Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, 7(1):1–26, 01 1979. 14

[27] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 1989. 2

[28] J. N. Foerster, Y. M. Assael, N. de Freitas, and S. Whiteson. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In *Proc. NIPS*, 2016. 1, 3

[29] J. N. Foerster, G. Farquhar, T. Afouras, N. NArdelli, and S. Whiteson. Coutnerfactual Multi-Agent Policy Gradients. In *Proc. AAAI*, 2018. 3

[30] J. N. Foerster, N. Nardelli, G. Farquhar, P. H. S. Torr, P. Kohli, and S. Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. In *CoRR, abs/1702.08887*, 2017. 3

[31] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys. Vision-based autonomous mapping and exploration using a quadrotor mav. In *Proc. IROS*, 2012. 2

[32] C. L. Giles and K. C. Jim. Learning communication for multi-agent systems. In *Proc. Innovative Concepts for Agent-Based Systems*, 2002. 1, 3

[33] A. Giusti, J. Guzzi, s. D. C. Cire F. L. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. CAro, et al. A machine learning approach to visual perception of forest trails for mobile robots. 2016. 3

[34] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi. IQA: Visual Question Answering in Interactive Environments. In *Proc. CVPR*, 2018. 3

[35] J. K. Gupta, M. Egorov, and M. Kochenderfer. Cooperative Multi-Agent Control Using Deep Reinforcement Learning. In *Proc. AAMAS*, 2017. 1, 3, 4

[36] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive Mapping and Planning for Visual Navigation. In *Proc. CVPR*, 2017. 2, 3, 5

[37] S. Gupta, D. Fouhey, S. Levine, and J. Malik. Unifying map and landmark based representations for visual navigation. *arXiv preprint arXiv:1712.08125*, 2017. 3

[38] H. Haddad, M. Khatib, S. Lacroix, and R. Chatila. Reactive navigation in outdoor environments using potential fields. In *Proc. ICRA*, 1998. 2

[39] F. Hill, K. M. Hermann, P. Blunsom, and S. Clark. Understanding grounded language learning agents. In *CoRR, abs/1710.09867*, 2017. 3

[40] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997. 5

[41] M. Johnson, K. Hofmann, T. Hutton, and D. Bignell. The malmo platform for artificial intelligence experimentation. In *Intl. Joint Conference on AI*, 2016. 3

[42] G. Kahn, T. Zhang, S. Levine, and P. Abbeel. Plato: Policy learning using adaptive trajectory optimization. In *Proc. ICRA*, 2017. 3

[43] T. Kasai, H. Tenmoto, and A. Kamiya. Learning of communication codes in multi-agent reinforcement learning problem. In *Proc. IEEE Soft Computing in Industrial Applications*, 2008. 1, 3

[44] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *RA*, 1996. 2

[45] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jakowski. Vizdoom: A doom-based ai research platform for visual reinforce- ment learning. In *Proc. IEEE Conf. on Computational Intelligence and Games*, 2016. 3

[46] K. Kidono, J. Miura, and Y. Shirai. Autonomous visual navigation of a mobile robot using a human guided experience. *Robotics and Autonomous Systems*, 2002. 2

[47] D. Kim and R. Nevatia. Symbolic navigation with a generic map. *Autonomous Robots*, 1999. 2

[48] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. In *https://arxiv.org/abs/1712.05474*, 2017. 2, 3

[49] K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. *Intl. J. of Robotics Research*, 2010. 2

[50] B. Kuipers and Y. T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and autonomous systems*, 1991. 2

[51] M. Lauer and M. Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proc. ICML*, 2000. 3

[52] S. M. Lavalle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. *Algorithmic and Computational Robotics: New Directions*, 2000. 2

[53] A. Lazaridou, A. Peysakhovich, and M. Baroni. Multi-agent cooperation and the emergence of (natural) language. In *arXiv preprint arXiv:1612.07182*, 2016. 1, 3

[54] S. Lenser and M. Veloso. Visual sonar: Fast obstacle avoidance using monocular vision. In *Proc. IROS*, 2003. 2

[55] A. Lerer, S. Gross, and R. Fergus. Learning physical intuition of block towers by example. In *Proc. ICML*, 2016. 3

[56] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Proc. NIPS*, 2017. 1, 3

[57] L. Matignon, G. J. Laurent, and N. L. Fort-Piat. Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *Proc. IROS*, 2007. 3

[58] F. S. Melo, M. Spaan, and S. J. Witwicki. QueryPOMDP: POMDP-based communication in multiagent systems. In *Proc. Multi-Agent Systems*, 2011. 1, 3

[59] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, et al. Learning to navigate in complex environments. In *Proc. ICLR*, 2017. 3

[60] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *https://arxiv.org/abs/1602.01783*, 2016. 5

[61] I. Mordatch and P. Abbeel. Emergence of Grounded Compositional Language in Multi-Agent Populations. In *Proc. AAAI*, 2018. 1, 3, 4

[62] J. Oh, V. Chockalingam, S. Singh, and H. Lee. Control of memory, active perception, and action in minecraft. In *Proc. ICML*, 2016. 3

[63] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *CoRR, abs/1703.06182*, 2017. 3

[64] G. U. G. Oriolo and M. Vendittelli. On-line map building and navigation for autonomous mobile robots. In *Proc. ICRA*, 1995. 2

[65] L. Panait and S. Luke. Cooperative multi-agent learning: The state of the art. Autonomous Agents and Multi-Agent Systems. In *Proc. AAMAS*, 2005. 3

[66] S. Phillips, A. Jaegle, and K. Daniilidis. Fast, robust, continuous monocular egomotion computation. In *Proc. ICRA*, 2016. 2

[67] R. C. R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *Intl. J. Robotics Research*, 1986. 2

[68] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in neural information processing systems*, pages 693–701, 2011. 12

[69] A. Remazeilles, F. Chaumette, and P. Gros. Robot motion control from a visual memory. In *Proc. ICRA*, 2004. 2

[70] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011. 2, 5

[71] E. Royer, J. Bom, M. Dhome, B. Thuillot, M. Lhuillier, and F. Marmoiton. Outdoor autonomous navigation using monocular vision. In *Proc. IROS*, 2005. 2

[72] P. Saeedi, P. D. Lawrence, and D. G. Lowe. Vision-based 3-d trajectory tracking for unknown environments. *IEEE Trans. on Robotics*, 2006. 2

[73] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun. MINOS: Multimodal indoor simulator for navigation in complex environments. 2017. 3

[74] J. L. Schnberger and J. M. Frahm. Structure-from-motion revisited. In *Proc. CVPR*, 2016. 2

[75] S. Seabold and J. Perktold. Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010. 14

[76] R. Sim and J. J. Little. Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters. In *Proc. IROS*, 2006. 2

[77] R. C. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Proc. UAI*, 1986. 2

[78] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *Proc. CVPR*, 2017. 3

[79] S. Sukhbaatar, A. Szlam, and R. Fergus. Learning multiagent communication with backpropagation. In *Proc. NIPS*, 2016. 1, 3

[80] S. Sukhbaatar, A. Szlam, G. Synnaeve, S. Chintala, and R. Fergus. Mazebase: A sandbox for learning from games. 2015. 3

[81] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente. Multiagent cooperation and competition with deep reinforcement learning. In *PloS*, 2017. 3

[82] M. Tan. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In *Proc. ICML*, 1993. 3

[83] G. Tesauro. Extending q-learning to general adaptive multiagent systems. In *Proc. NIPS*, 2004. 3

[84] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 1992. 2

[85] M. Tomono. 3-d object map building using dense object models with sift-based recognition features. In *Proc. IROS*, 2006. 2

[86] M. Toussaint. Learning a world model and planning with a self-organizing, dynamic neural system. In *Proc. NIPS*, 2003. 3

[87] D. Wooden. A guide to vision-based map building. *IEEE Robotics and Automation Magazine*, 2006. 2

[88] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian. Building Generalizable Agents with a Realistic and Rich 3D Environment. In *https://arxiv.org/abs/1801.02209*, 2018. 3

[89] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner. Torcs, the open racing car simulator, 2013. 3

[90] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard. Deep reinforcement learning with successor features for navigation across similar environments. *arXiv preprint arXiv:1612.05533*, 2016. 3

[91] Y. Zhu, D. Gordon, E. Kolve, D. Fox, L. Fei-Fei, A. Gupta, R. Mottaghi, and A. Farhadi. Visual Semantic Planning using Deep Successor Representations. In *https://arxiv.org/abs/1705.08080*, 2017. 3

[92] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. In *Proc. ICRA*, 2017. 2

A. Appendix

This appendix presents the following content:

1. Visualizations of the grid-world abstraction of our task,
2. Our learning algorithm,
3. Interplay between *talk* and *reply* stages of the communication and belief refinement module,
4. Implementation details of model
5. A detailed explanation of metrics used in our paper,
6. Quantitative evaluation of our models but now evaluated on seen scenes,
7. Statistical analysis of agent communication strategies but now demonstrated on unseen scenes,
8. Qualitative results of agents with different communication abilities deployed on unseen scenes. This includes clip summaries with agent communication signals for video https://youtu.be/9sQhD_Gin5M.

A.1. AI2-THOR to Grid-world

In order to assess the impact of learning to communicate directly from pixels rather than, as in most prior work, from grid-world environments, we perform a direct translation of our task into a grid-world and compare its performance to our best model. For this purpose we transform AI2-THOR into a grid-world environment. Figure 10 visualizes, for a single AI2-THOR scene, this transformation. To make our comparison fair, as our pixel-based agents only obtain partial information about their environment at any given timestep, we impose the same restriction on our grid-world agents by only providing them with an egocentric 5×11 view of their environment (see Figure 11).

A.2. Learning algorithm

Algorithm 1 succinctly summarizes our learning procedure as otherwise described in Section 3.3 of the main paper.

A.3. Talk and Reply stages

Explicit communication happens via two stages - talk and reply. As illustrated in Fig. 9, each stage has it's own weights (V_{send}, W_{send}, W_1, W_2). These are clearly marked using superscripts of $^{(T)}$ and $^{(R)}$ for the talk and reply stage, respectively.

A.4. Implementation Details.

We use the same hyperparameters and embedding dimensionality in all of our experiments. In our A3C loss we discount rewards with a factor of $\gamma = 0.99$ and weight the entropy maximization term with a factor of $\beta = 0.01$. We use the Adam optimizer with a learning rate of 10^{-4}, momentum values of 0.9 and 0.999 (for the first and second moments respectively), and share optimizer statistics across processes. Gradient steps are made in the hogwild approach, that is without explicit synchronization or locks between processes [68].

Algorithm 1 Learning Algorithm

```
Randomly initialize shared model weights $\theta_{\text{shared}}$
Set global episode counter $c \leftarrow 0$
**while** $c <$ maxEpisodes **in parallel do**
    $\theta \leftarrow \theta_{\text{shared}}$
    $c \leftarrow c + 1$
    Randomly choose environment
    Randomize agents' positions and TV location
    Set $\alpha \leftarrow 0.9 \cdot \max(1 - c/10000, 0)$
    Set $\pi \leftarrow (1 - \alpha) \cdot \pi_\theta + \alpha \cdot \pi^*$
    Roll out trajectory of length $\leq$ 500 from both agents using $\pi$.
    $L_{\text{a3c}} \leftarrow$ A3C loss for trajectory
    $L_{\text{cross}} \leftarrow$ cross entropy loss of trajectory w.r.t. $\pi^*$
    **if** no expert actions sampled in trajectory **then**
        $g \leftarrow \nabla_\theta (L_{\text{a3c}} + L_{\text{cross}})$
    **else**
        $g \leftarrow \nabla_\theta L_{\text{cross}}$
    Perform one gradient update of $\theta_{\text{shared}}$ using ADAM with gradients $g$ and statistics shared across processes
**end**
```

Each model was trained for 100,000 episodes. Each episode is initialized in a random train (seen) scene of AI2-THOR. Rewards provided to the agents are: 1 to both agents for a successful pickup action, constant -0.01 step penalty to discourage long trajectories, -0.02 for any failed action (*e.g.*, running into a wall) and -0.1 for a failed pickup action. Episodes run for a maximum of 500 total steps (250 steps for each agent) after which the episode is considered failed. The minimum aggregate achievable reward in an episode, obtained by successive attempting failed pickup actions by both agents is -65 while the maximum reward is 1.98 achieved by both agents immediately picking up the object as their first action and only receiving a single step penalty.

A.5. Metrics

We now present a more detailed explanation of the metrics we use to evaluate our models.

(1) Per agent reward structure:

- +1 for performing a successful joint pickup,
- -0.1 for a failed pickup action,
- -0.02 for any other failed action (trying to move into walls, furniture, etc.), and

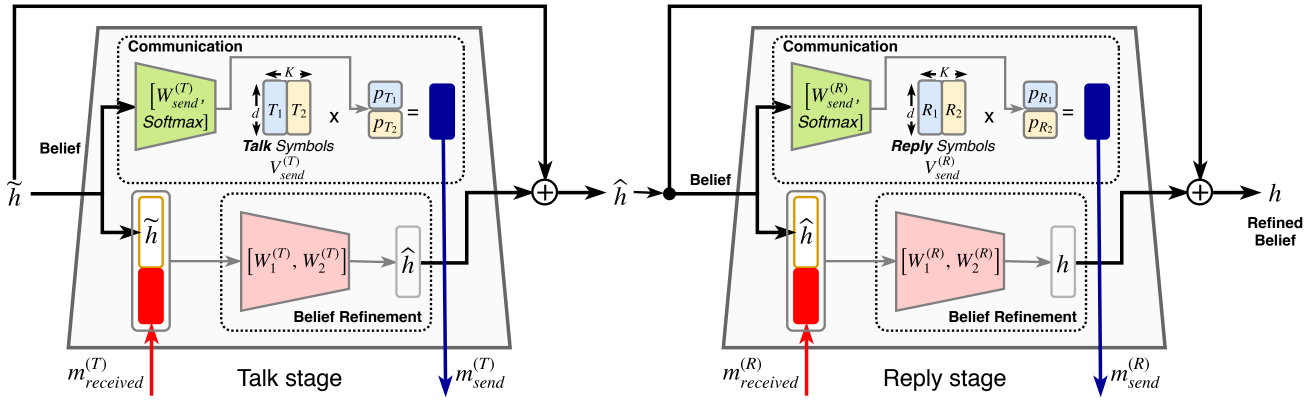


Figure 9: Two stages of communication and belief refinement module - *talk* and *reply*. The refined belief from the talk stage is further refined by another round of communication between agents at the reply stage. In this illustration the size of vocabulary is 2 *i.e.* $K = 2$.

- -0.01 for each step to encourage short trajectories.

(2) Accuracy: the percentage of episodes which led to the successful pickup action by both agents.

(3) Number of unsuccessful pickups: total number of pickup actions attempted by both agents which didn't lead to the target being picked up. The three preconditions necessary for a successful joint pickup action are as follows.

  (i) Both agents perform the pickup action simultaneously,

  (ii) Both agents are closer than 1.5m to the target and the target is visible, and

  (iii) Both agents are a minimum distance apart from each other (0 for the *Unconstrained* and 8 steps = 2 meters in the manhattan distance for the *Constrained* setting).

(4) Number of missed pickups: total number of episode steps where both agents could have picked up the object but did not. This is the number of opportunities where 3ii and 3iii were met, but the agents didn't perform simultaneous pickup actions.

(5) Relative episode length: the quantity

$$\frac{\text{Episode length following agent policy } (\pi)}{\text{Episode length following oracle policy } (\pi^*)}$$

As it has access to information not available to the agents, our expert policy is also referred to as the oracle policy. As mentioned in the paper, the oracle plans a shortest path from each agent location to the target. This is achieved by leveraging the full map of the scene (*i.e.*, free space, occupied areas, location of other agent, and the target location).

A.6. Quantitative evaluation

In this section we provide quantitative evaluation results of variants of TBONE. We provide results on seen (train) and unseen (test) scenes. Many of the unseen scenes results are already included in the main paper, but we reproduce the full suite of graphs here, for ease of comparison.

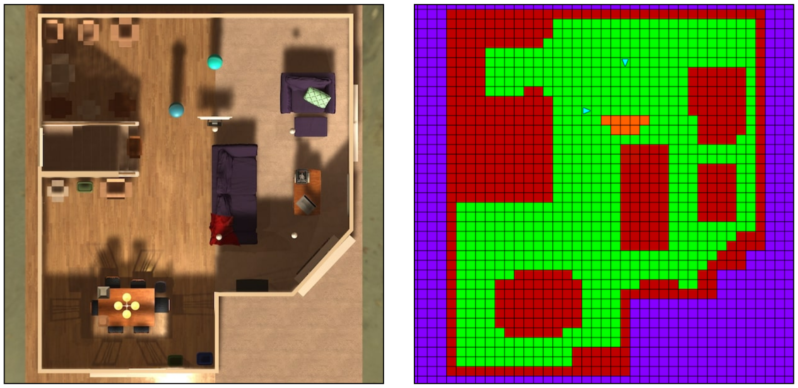
For the *Constrained* task, Fig. 12 and Fig. 13 show the above metrics on seen and unseen scenes, respectively. For the *Unconstrained* task, Fig. 14 and Fig. 15 show the above metrics on seen and unseen scenes, respectively.

On the *Constrained* task in seen scenes (Fig. 12), having both modes of communication clearly produces better rewards. And having either or both modes of communication easily outperforms agents with no means of communication. While the accuracy metric is similar to having only implicit means of communication, the number of unsuccessful pickups, missed pickups, and relative episode lengths metrics show the benefit of having both modes of communication over any one of them. A similar trend is seen in unseen scenes for the same task (Fig. 13).

On the *Unconstrained* task, the benefits of communication are, as expected, less dramatic (Fig. 14 and Fig. 15). Since the task is simpler and potentially can be solved without communication, agents with no means of communication are able to obtain high accuracies. But in the absence of communication, agents end up having a large number of unsuccessful pickups. This is expected. With no means of communication, agents simply go close to the TV and start attempting pickups. Only with communication can they lower this metric by coordinating with each other.

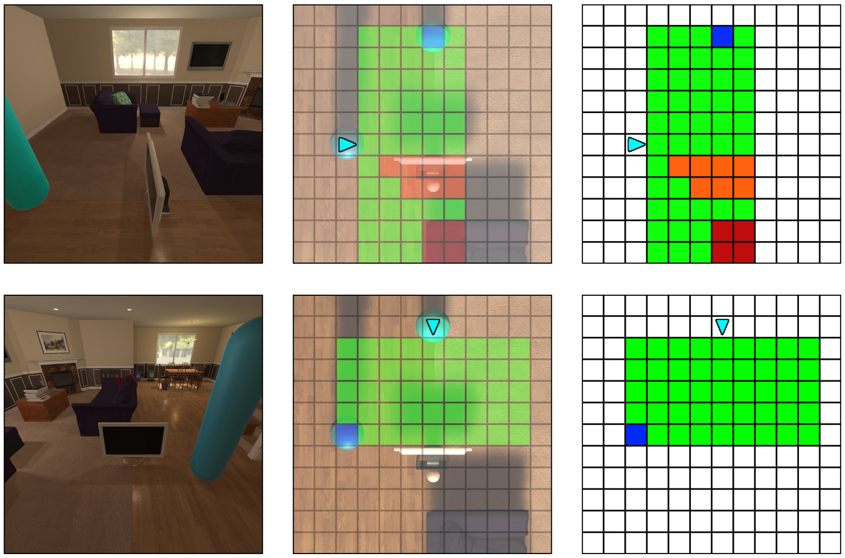
A.7. Interpreting Communication

To fit the logistic models described in Section 4 of the main paper we randomly initialize 2,687 episodes on the 20 training scenes from which we obtain a corresponding



(a) Top view of AI2-THOR scene (b) Corresponding grid-world

Figure 10: An AI2-THOR scene from a top-down view along the corresponding grid-world. Note that each agent (teal triangles) only observes a small portion of the grid-world at any given time-step, see Figure 11 for details. Here each color corresponds to a different category: freespace (green), impassable terrain (red), target object (orange), and unknown (purple).



(a) First-person AI2-THOR agent view (b) Agent partially observed grid-world overlayed on map view (c) Grid-world corresponding to agent view

Figure 11: First person viewpoints of agents in AI2-THOR and the corresponding grid-world observations. Note that white squares are unobserved and blue squares correspond to another agent, see Figure 10 for a description of the other colors.

number of agent trajectories. Treating each step in these trajectories as a single observation, this results in a dataset containing 143,401 samples. We fit these logistic models using the `statsmodels` package [75] in Python. As observations within a single episode are highly correlated, we use the bootstrap [26] to obtain robust standard errors for our estimates.

As the analysis above is done on the seen scenes, it begs the question of whether the same trends occur when agents communicate in unseen environments. To address this, we sample 1,333 agent episodes on the 10 test scenes resulting in a dataset of 201,738 samples. We fit identical logistic regression models to this dataset as in the main paper and report the resulting estimates and standard errors in Table 3. While several estimates differ, in a statistically significant way, from those on the seen scenes, all trends remain the same suggesting that agents communicate in largely the same way in unseen environments as they do in previously seen environments.

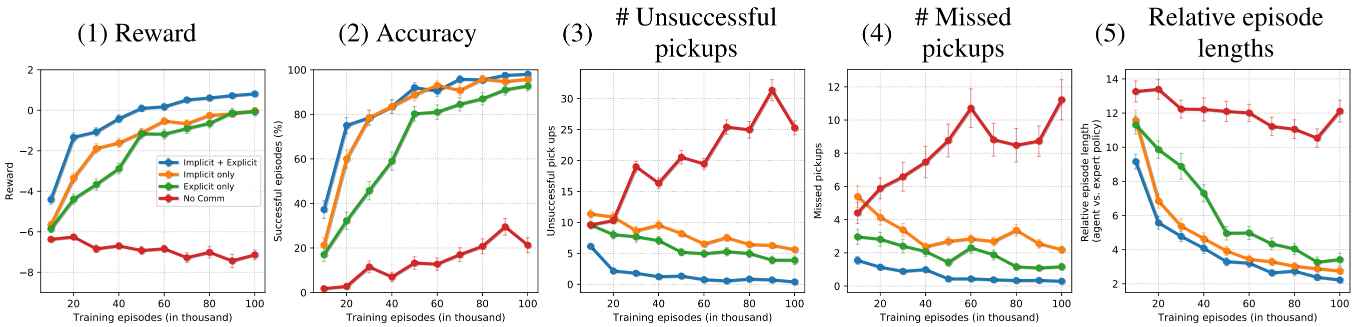


Figure 12: *Constrained* task, seen scenes.

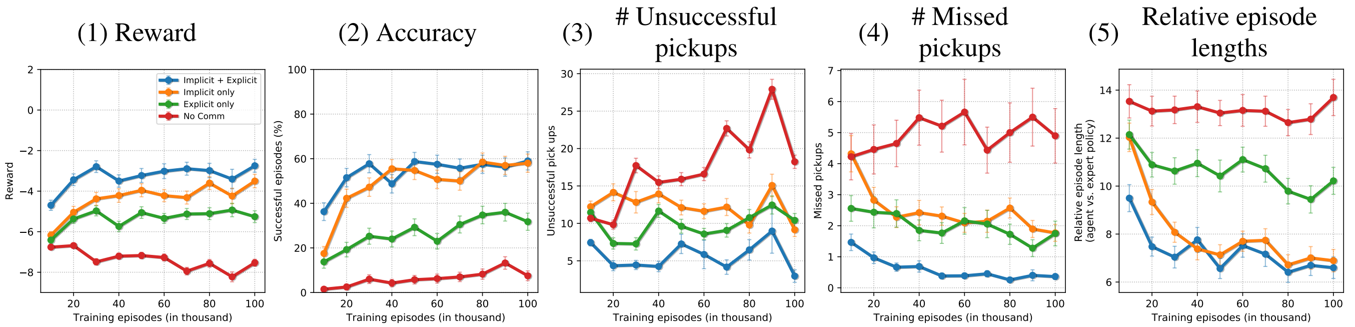


Figure 13: *Constrained* task, unseen scenes.

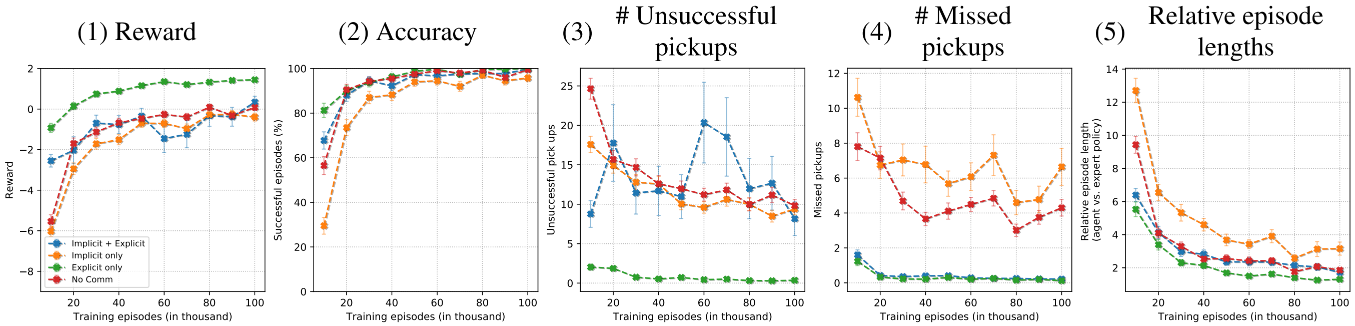


Figure 14: *Unconstrained* task, seen scenes.

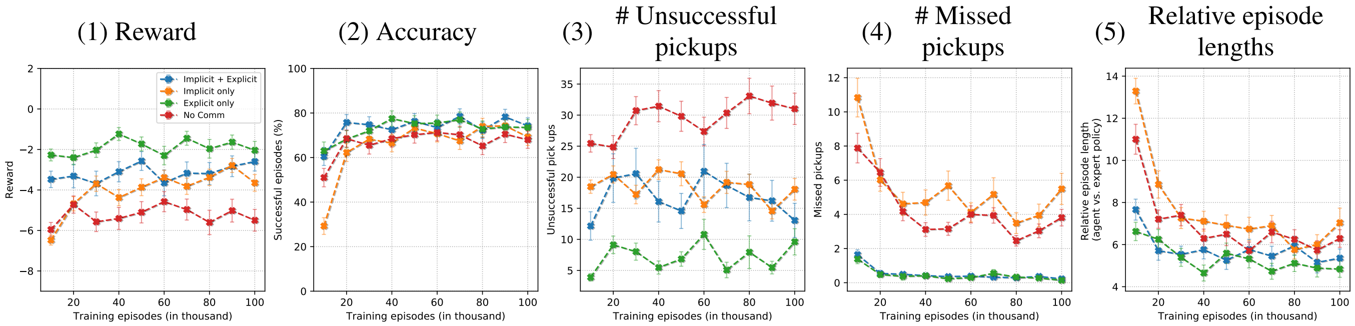


Figure 15: *Unconstrained* task, unseen scenes.

	$\beta^{\leq}$	$\beta_t^{\leq}$	$\beta_r^{\leq}$	β^{see}	β_t^{see}	β_r^{see}
Est.	0.07	1.29	-0.14	0.65	0.57	-0.88
SE	0.033	0.027	0.031	0.041	0.027	0.042

	β^{pick}	$\beta_{t,0}^{\text{pick}}$	$\beta_{r,0}^{\text{pick}}$	$\beta_{t,1}^{\text{pick}}$	$\beta_{r,1}^{\text{pick}}$	$\beta_{\vee,r}^{\text{pick}}$
Est	1.15	-0.0	-0.04	-0.01	-0.04	-1.17
SE	0.037	0.009	0.009	0.009	0.011	0.041

Table 3: Estimates, and corresponding robust bootstrap standard errors, of the parameters from the main paper's Section 4 when using trajectories sampled from the unseen scenes as described in Section A.7.

A.8. Qualitative results

A.8.1 Effect of communication

We present qualitative results of agents with three communication abilities: implicit + explicit vs. implicit only vs. no communication. We compare the effect by deploying this agents for a particular initialization of an episode i.e. the same scene, agents' start locations and target object location. We find both explicit and implicit communication help achieve the task faster as seen Fig. 16, Fig. 17 and Fig. 18 which have episode lengths of 86, 165 and 250 respectively. Another such initialization is compared in Fig. 19, Fig. 20 and Fig. 21 which have episode lengths of 17, 72 and 217 respectively.

A.8.2 Video

The associated video includes episode visualizations for the *Constrained* task on **Unseen scenes**, and can be found here: `https://youtu.be/9sQhD_Gin5M`. For these episodes we ran inference on the model with both explicit and implicit communication. The six clips in the video are summarized in Fig. 22, Fig. 23, Fig. 24, Fig. 25, Fig. 26 and Fig. 27. The first four culminated in successful pickup of the target object. The last two videos highlight typical error modes.

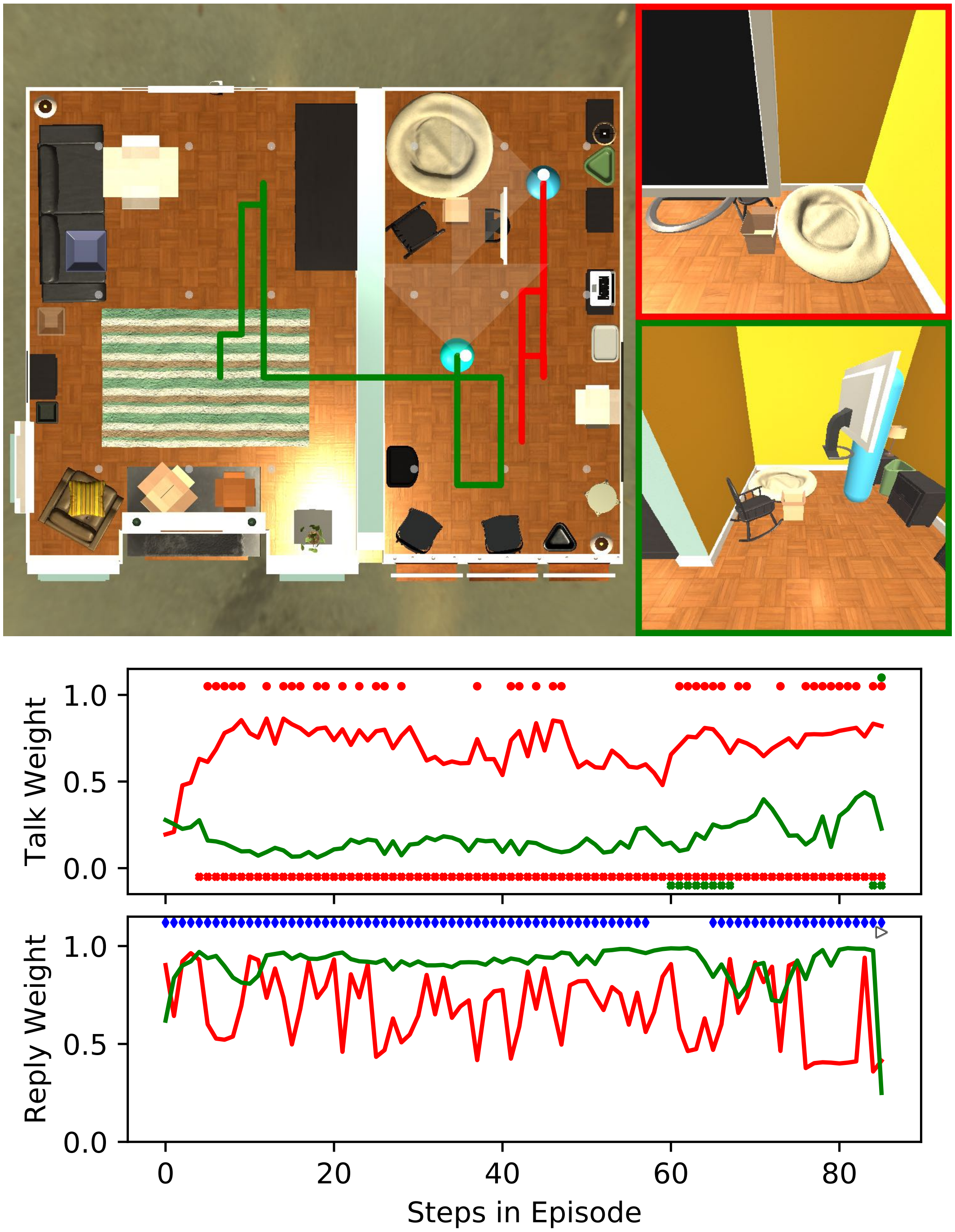


Figure 16: Initialization 1: With explicit and implicit communication, episode length is 86 per agent. Associated agent communication in plot below, see Figure 8 in the main paper for a legend.

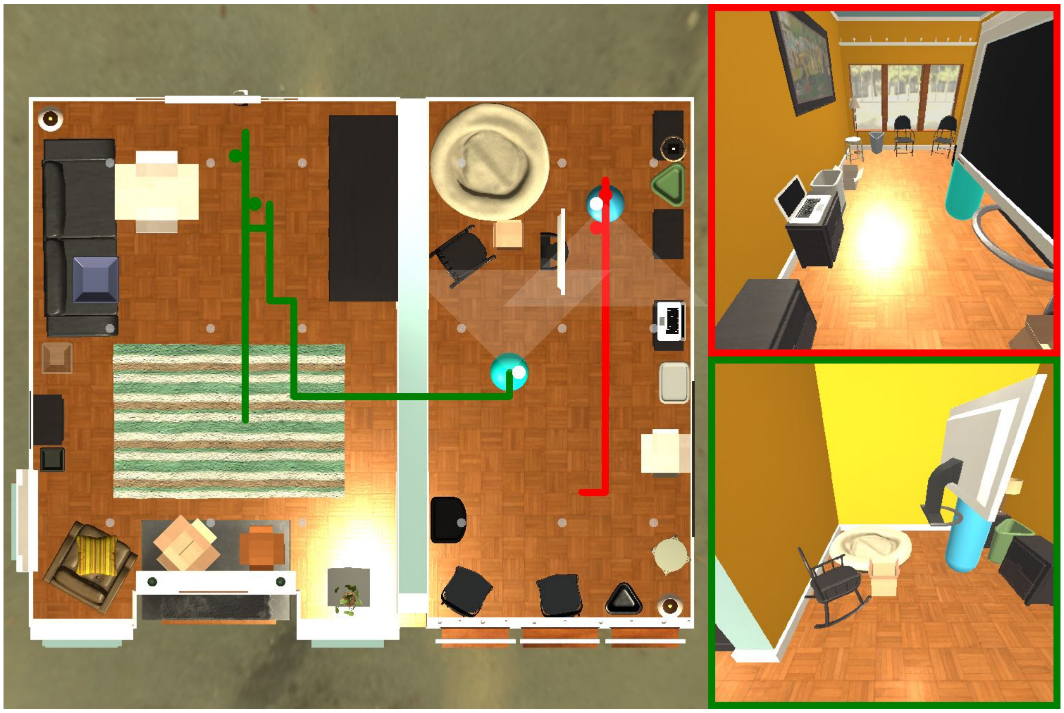


Figure 17: Initialization 1: With only implicit communication, episode length is 165 per agent.

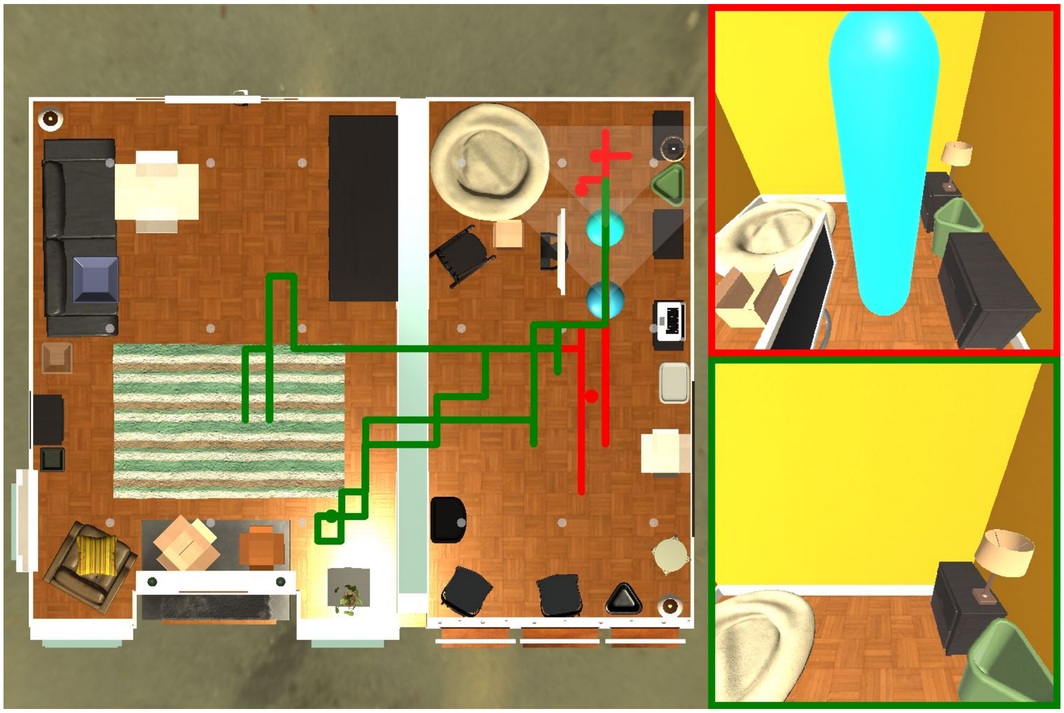


Figure 18: Initialization 1: With no communication, episode length is 250 per agent (unsuccessful).

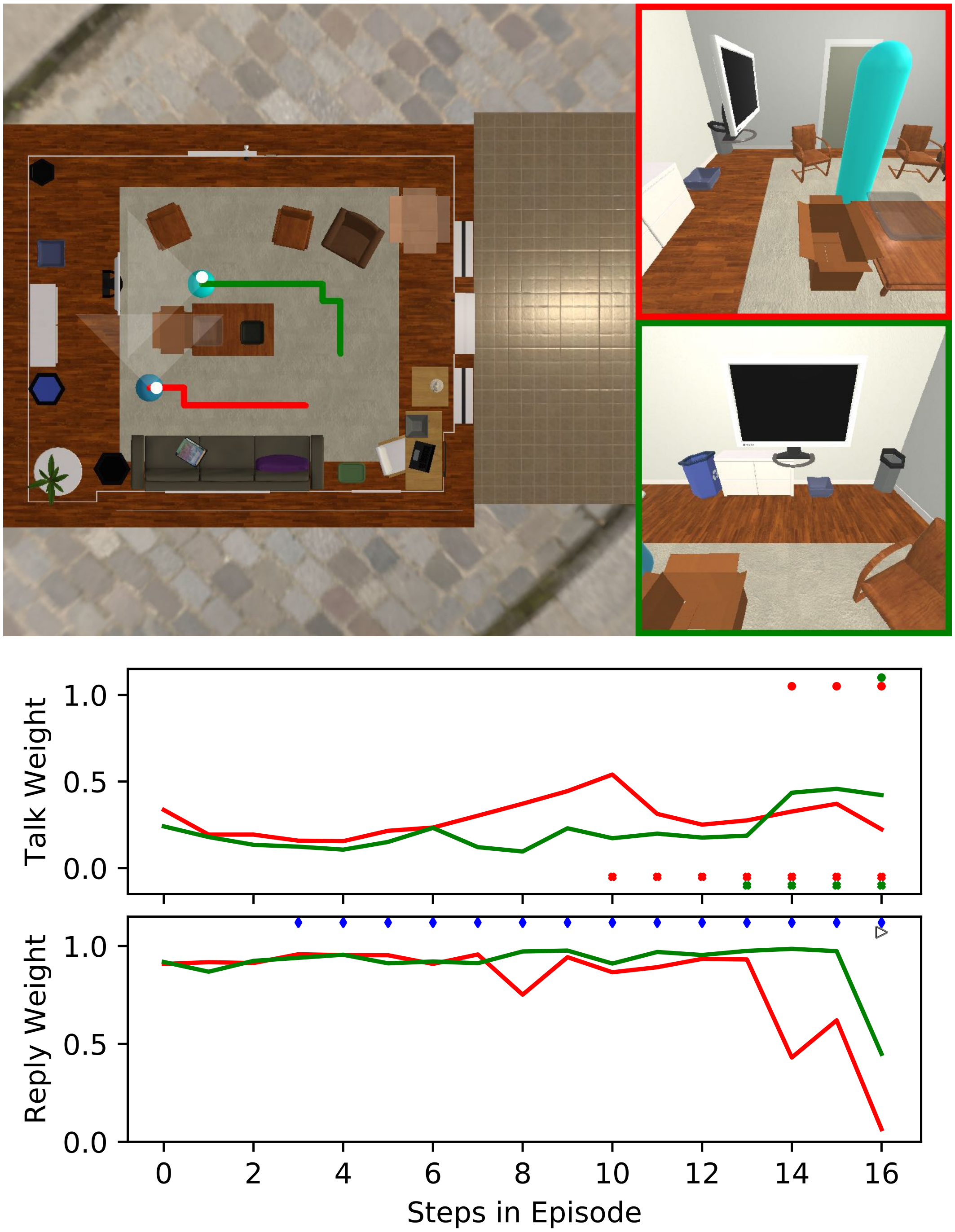


Figure 19: Initialization 2: With explicit and implicit communication, episode length is 17 per agent. Associated agent communication in plot below, see Figure 8 in the main paper for a legend.

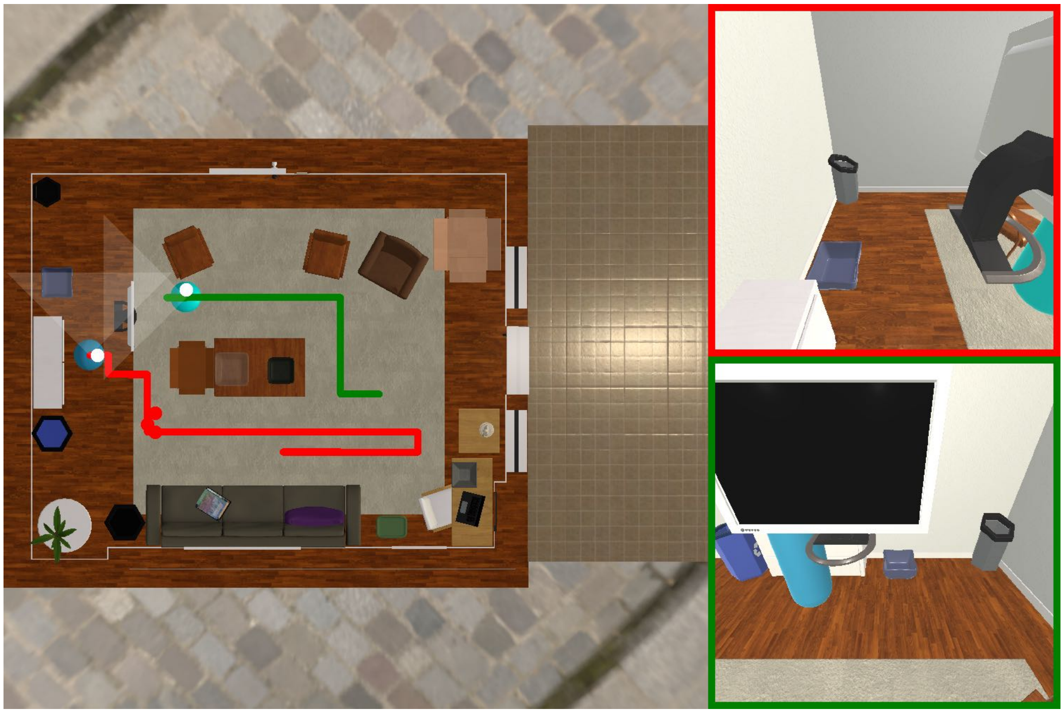


Figure 20: Initialization 2: With only implicit communication, episode length is 72 per agent.

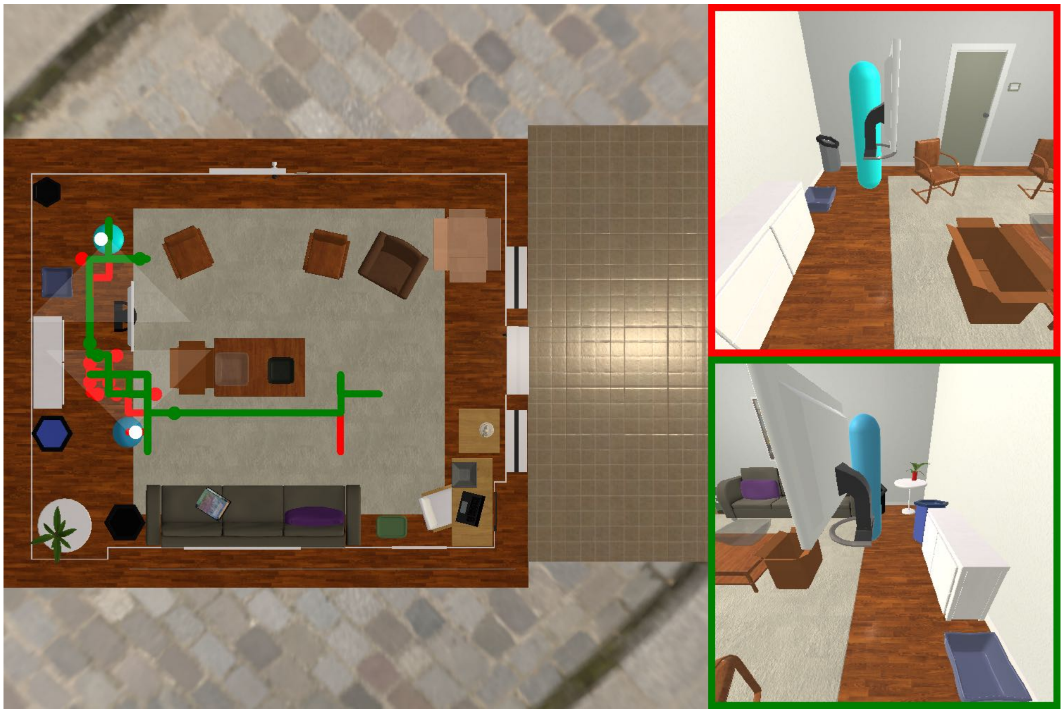


Figure 21: Initialization 2: With no communication, episode length is 217 per agent.

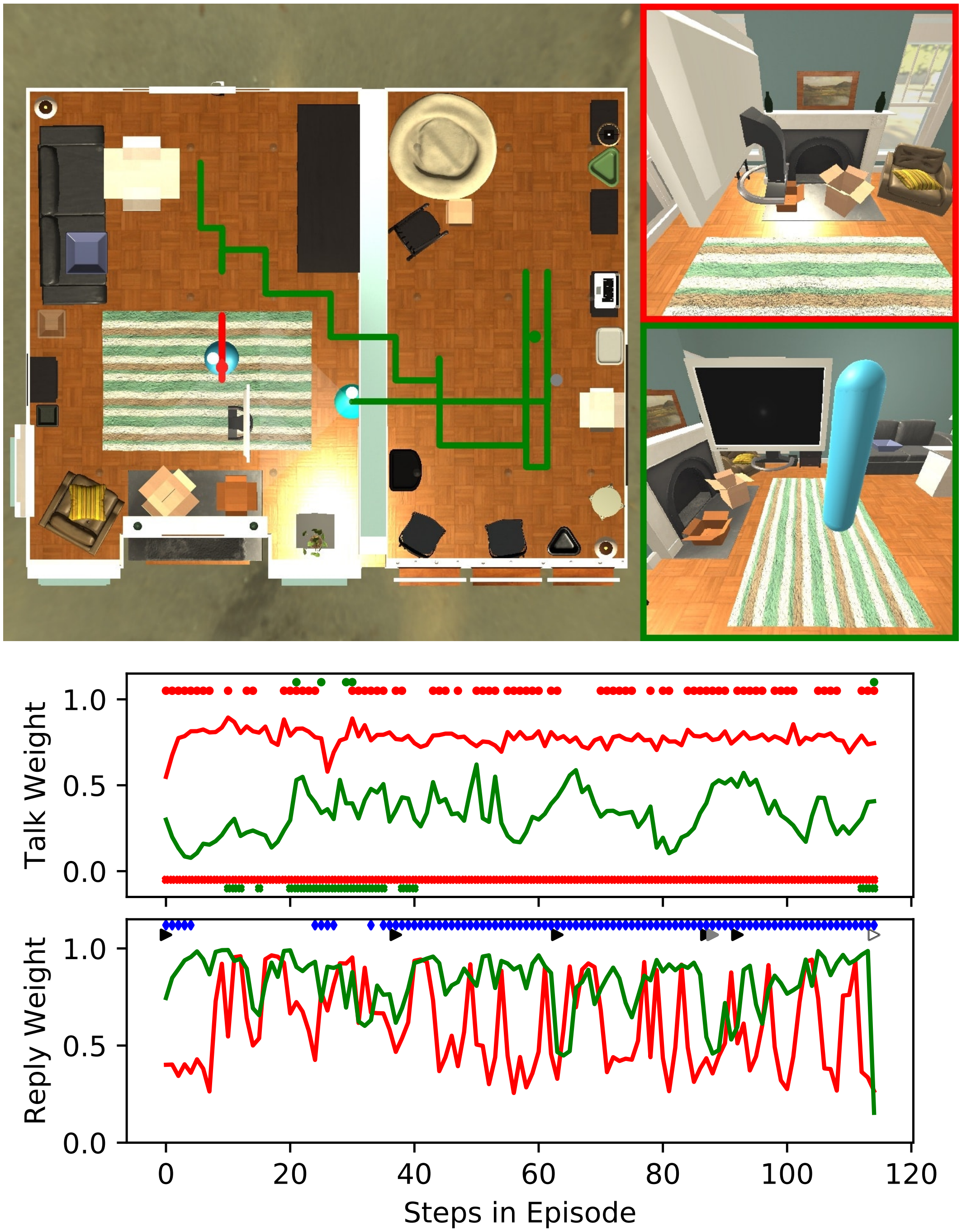


Figure 22: Clip 1 summary, see Figure 8 in the main paper for a legend.

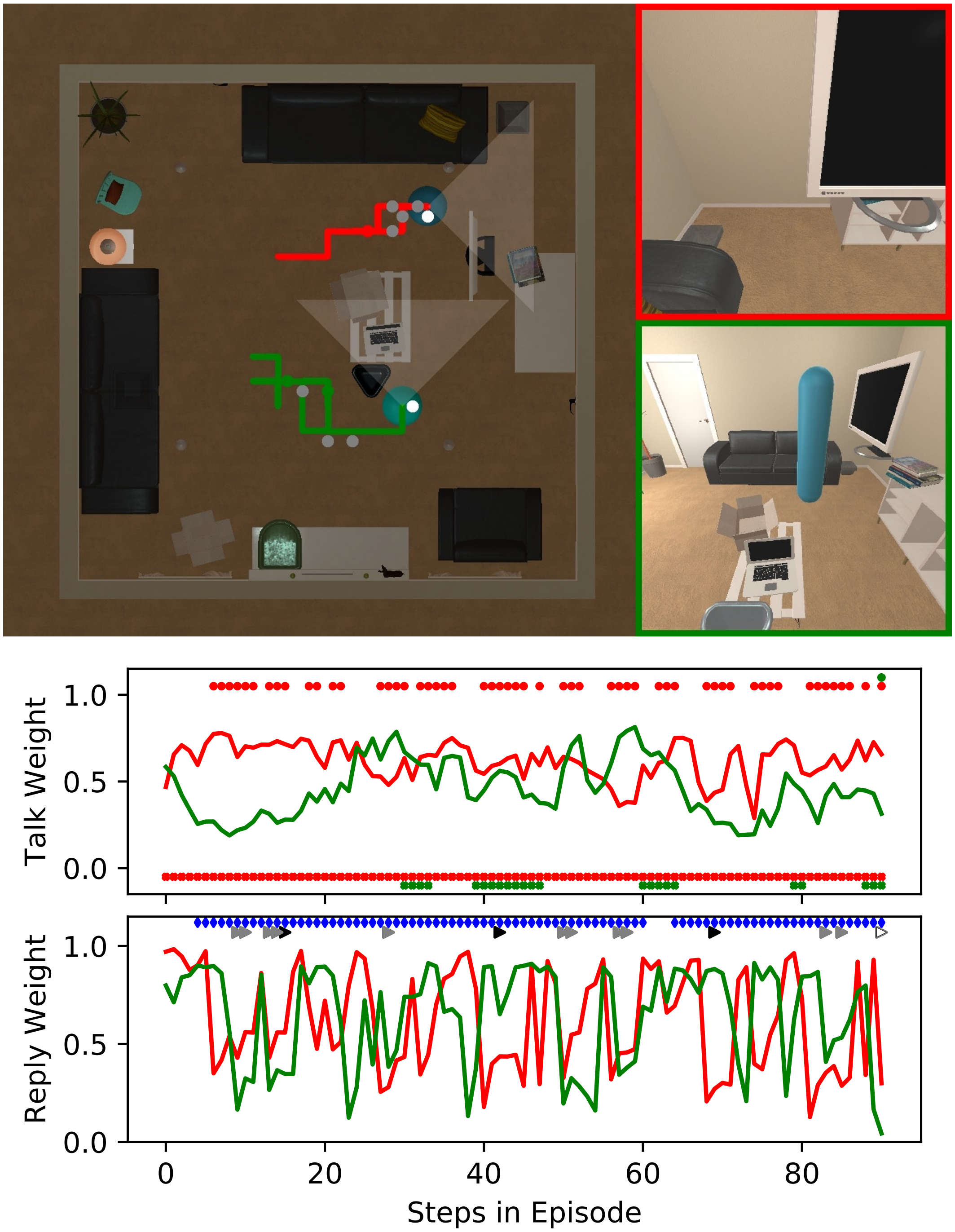


Figure 23: Clip 2 summary, see Figure 8 in the main paper for a legend.

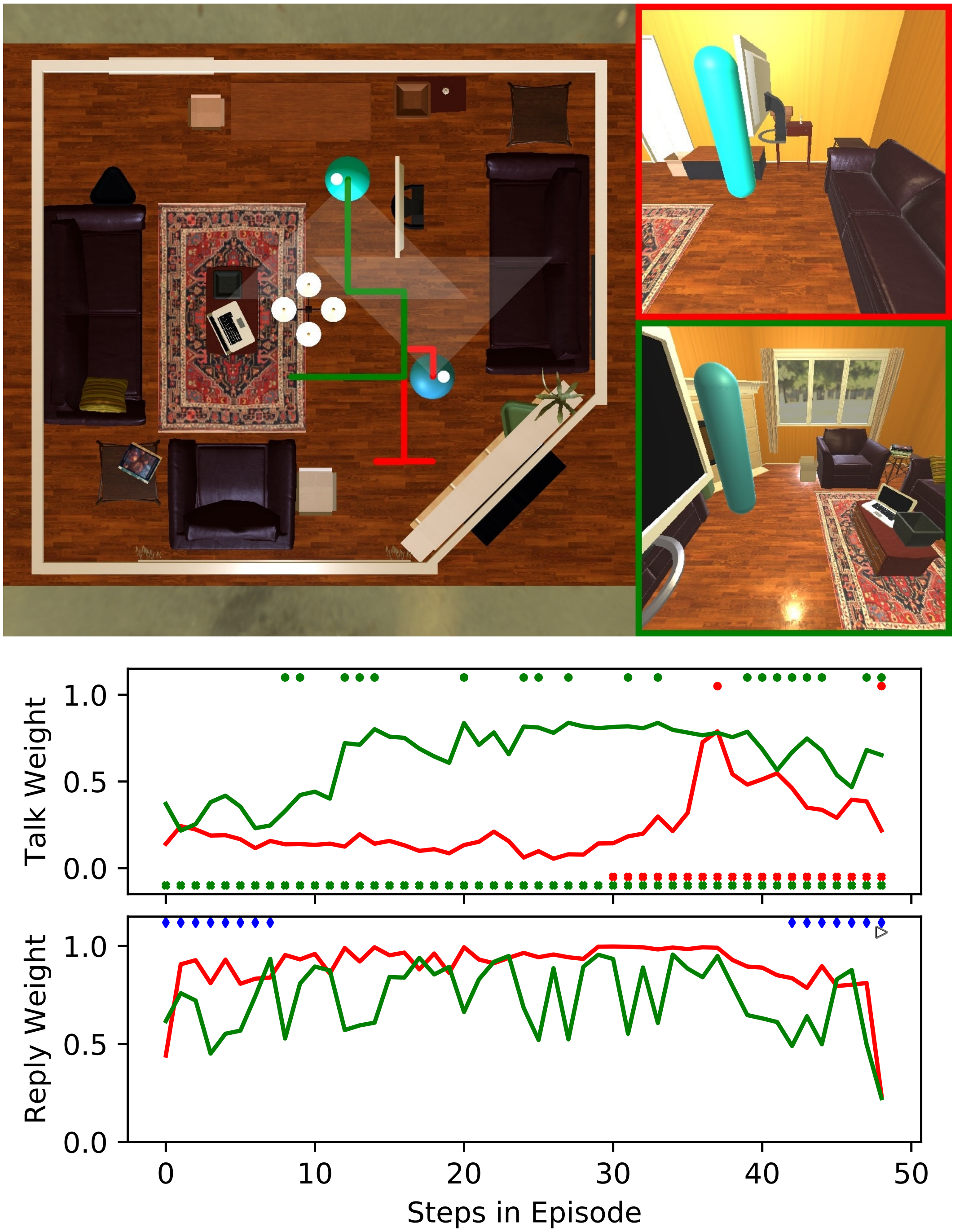


Figure 24: Clip 3 summary, see Figure 8 in the main paper for a legend.

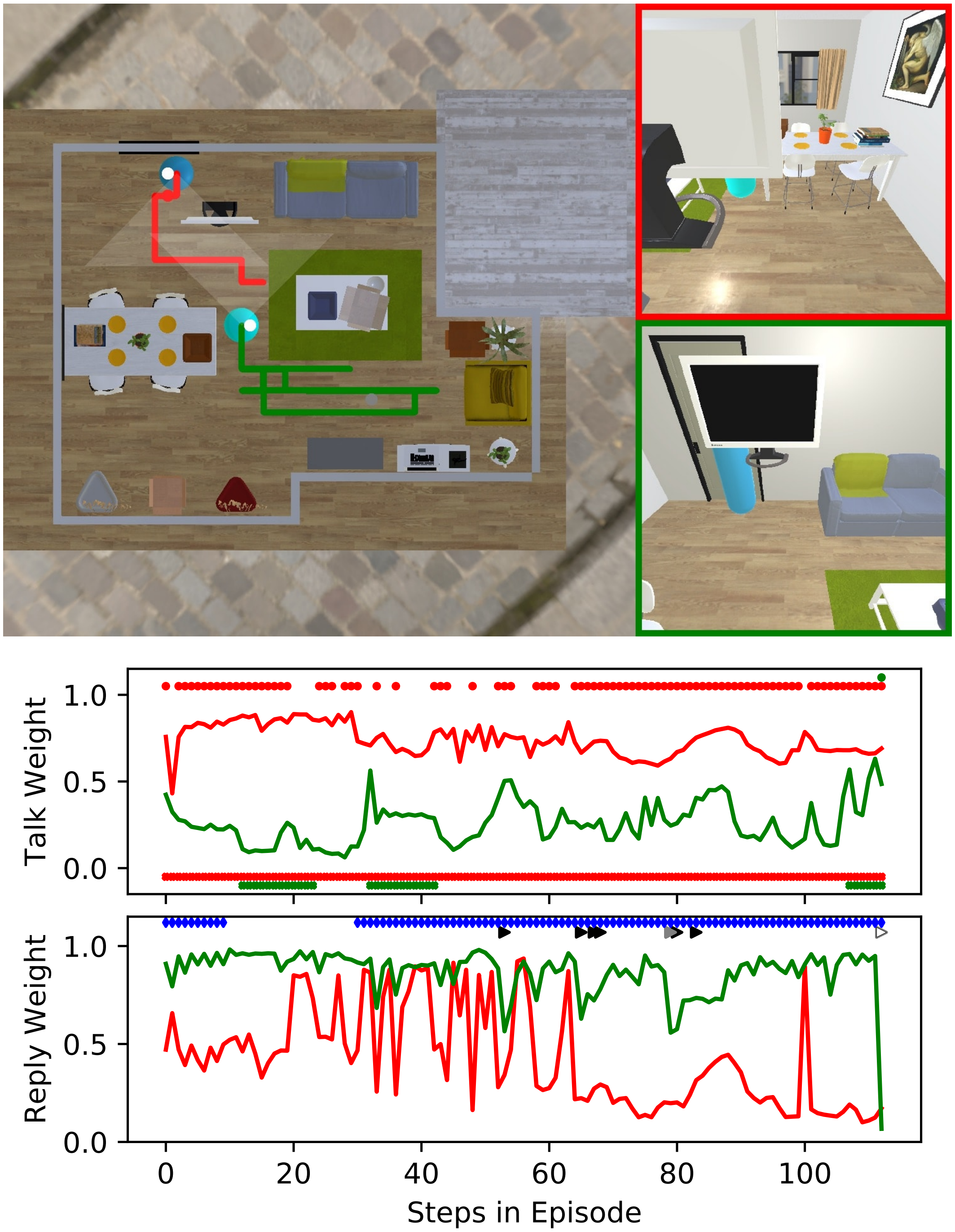


Figure 25: Clip 4 summary, see Figure 8 in the main paper for a legend.

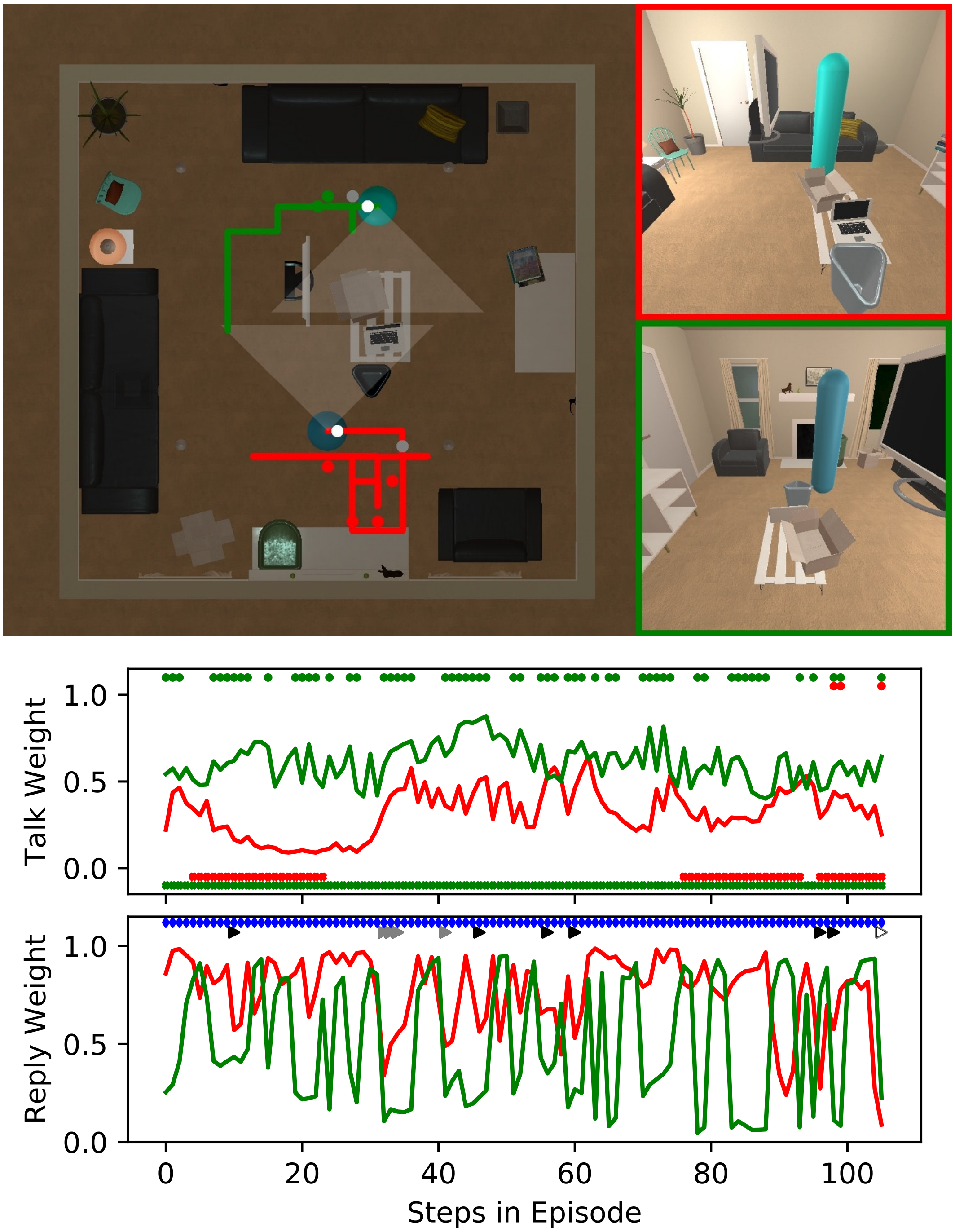


Figure 26: Clip 5 summary, see Figure 8 in the main paper for a legend.

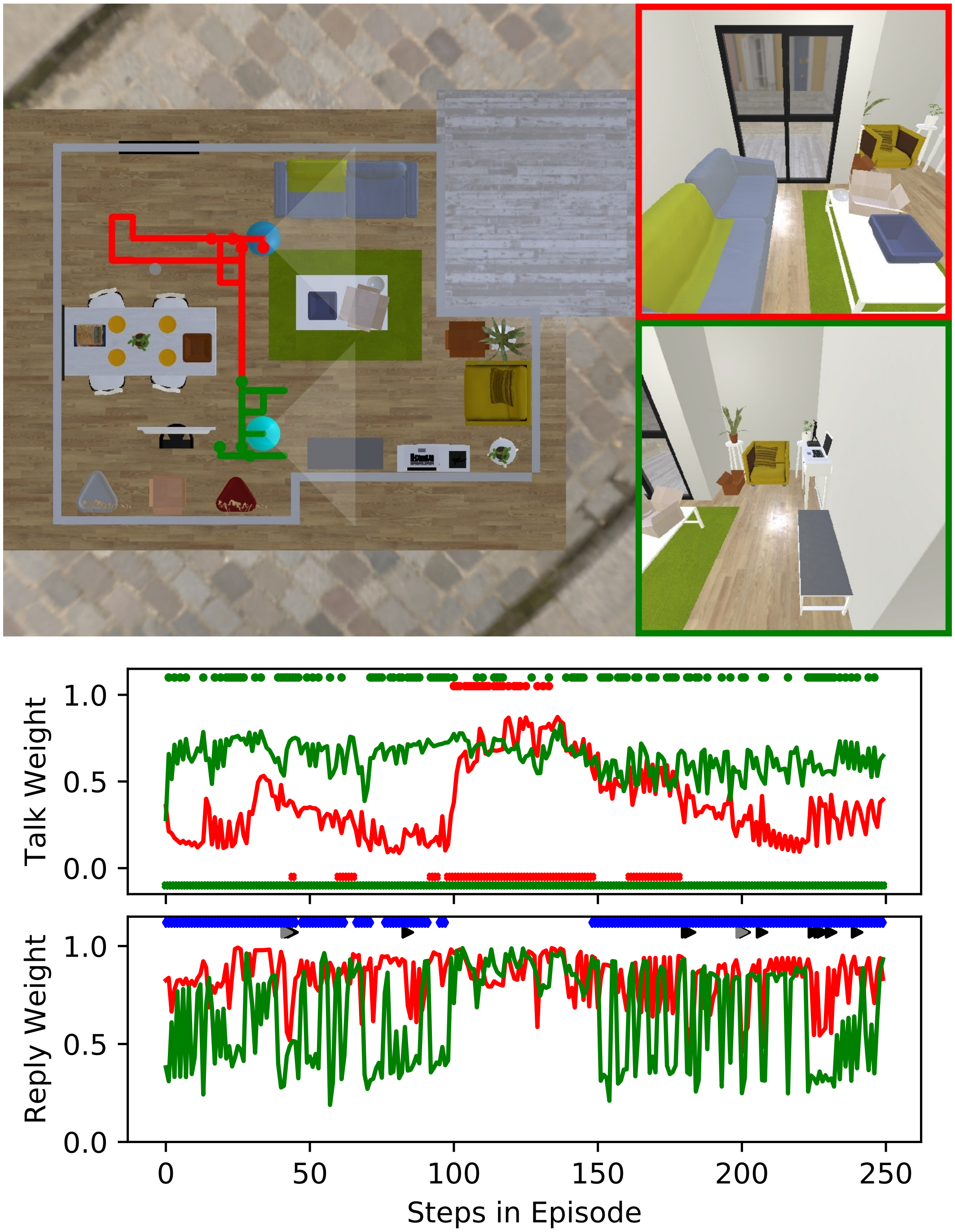


Figure 27: Clip 6 summary, see Figure 8 in the main paper for a legend.