Sharing the Cost of Success: A Game for Evaluating and Learning Collaborative Multi-Agent Instruction Giving and Following Policies

Philipp Sadler, **Sherzod Hakimov** and **David Schlangen**
CoLabPotsdam / Computational Linguistics
Department of Linguistics, University of Potsdam, Germany
firstname.lastname@uni-potsdam.de

Abstract

In collaborative goal-oriented settings, the participants are not only interested in achieving a successful outcome, but do also implicitly negotiate the effort they put into the interaction (by adapting to each other). In this work, we propose a challenging interactive reference game that requires two players to coordinate on vision and language observations. The learning signal in this game is a score (given after playing) that takes into account the achieved goal and the players' assumed efforts during the interaction. We show that a standard Proximal Policy Optimization (PPO) setup achieves a high success rate when bootstrapped with heuristic partner behaviors that implement insights from the analysis of human-human interactions. And we find that a pairing of neural partners indeed reduces the measured joint effort when playing together repeatedly. However, we observe that in comparison to a reasonable heuristic pairing there is still room for improvement—which invites further research in the direction of cost-sharing in collaborative interactions.

Keywords: vision-and-language, reinforcement learning, multi-agent

1. Introduction

Recent advances in natural language processing have led to language model-based systems that, at least at first sight, seem to do a good job at creating natural dialogue behaviour. However, the conversations with these models are often still very verbose (lengthy responses) and brittle (necessity to wait for response completion). In contrast, Clark and Wilkes-Gibbs (1986) observed that humans in a collaborative situation used the language as a coordination device (joint action; Clark (1996)) and that an adaption process takes place which is driven by effort reduction. In their experiments, a director instructed a listener to put cards with figures on them in a specific order without seeing the listener's cards. The observation was that the average number of speaking turns taken by the director per figure drastically reduced over the trials while the success outcomes stayed high. Thus, the later trials did not just lead to the desired outcome but were also more *efficient*.

Now, imagine a situation as sketched in Figure 1. An instruction giver guides a follower towards a piece that must be taken on a virtual board, but there are various other pieces which might distract the follower. A strategy for the guide could be to use short phrases and perform remote control (Guide A). The main effort stays with the guide, which has to provide accurate navigation instructions while the follower executes them without incurring any own planning effort costs. Another extreme would be a strategy (Guide B) where the guide initiates the interaction with a very detailed instruction and then stays silent. This puts most of the cognitive load on the follower's side which might now hesitate or actually take the wrong piece after all. The best strategy presumably lies in the middle (Guide M) where the guide – after having seen different boards previously and having interacted with the follower multiple times – initiates the interaction with a longer phrase but provides useful feedback

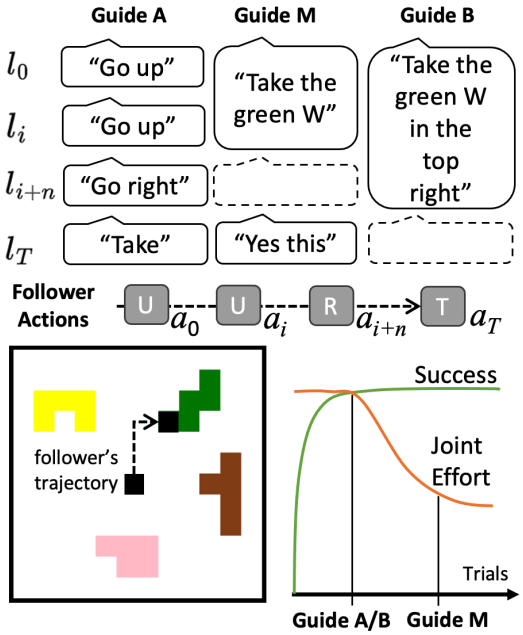

Figure 1: A guide and a follower observe the board with the pieces and the follower's gripper (the black dot). An optimal trajectory of actions for the follower would be: up (U), up, right (R), and take (T). The best strategy for the guide lies assumably in the middle (M) of the extremes (A/B) where the guide refers to a piece initially with l_0 and stays silent at until confirming the follower's choice with l_T. This strategy shares the cost for success between both.

when necessary. While all these strategies can be successful, the latter is the one that shares the cost of success the best between the partners. Such capabilities would be essential for future assisting agents to take a helpful part in society someday.

However, we notice that current research on language and vision coordination problems seems to neglect (a) the notion of required effort for the production and delivery of instructions and (b) the incremental aspects of the interaction. In vision and language navigation (Chevalier-Boisvert et al., 2019; Nguyen and III, 2019; Fried et al., 2018) a follower receives at each time step a (possibly lengthy) instruction that contains all relevant information. In interactive sub-goal generation, a planning model comes up with a goal formulation in "no-time" (a single step) (Chane-Sane et al., 2021; Sun et al., 2023; Lee and Kim, 2023). And in multi-agent environments, agents typically coordinate without using natural language at all (Bard et al., 2020; Samvelyan et al., 2019; Pan et al., 2022).

Can neural agents agree on a cooperative strategy that shares the cost of success more equally? In this work, we put the incremental aspect of the language and vision coordination problem to the fore again and weight an agent's actions by its assumed effort and time costs. Thus, agents have to trade off the production of costly but informative actions with the overall outcome of the game. To study neural agent capabilities under this constraint

- we propose a challenging reference game where two players have to coordinate on the selection of a piece among various distractors while the actual target piece is only known to one of them (the guide) and only the other can perform the selection (the follower),
- establish a strong baseline performance with heuristic partners that implement insights from the analysis of human-human interactions
- and show that neural partners in a multi-agent setting indeed strive towards an presumably more human-like strategy when effort matters.

2. A Game for Evaluating and Learning Collaborative Multi-Agent Policies

We propose a **Co**llaborative **G**ame of **R**eferential and **I**nteractive language with **P**entomino pieces (CoGRIP) to evaluate and learn neural policies for the aspect of cost sharing a multi-agent setting. In CoGRIP two players are forced to work together because of their asymmetry in knowledge and skill. A guide uses language utterances to instruct a follower to select a puzzle piece (Pentomino; Golomb (1996)). The guide can provide utterances but cannot move the gripper. The follower can move the gripper but is not allowed to provide an utterance. Zarrieß et al. (2016) found that such a setting leads to diverse language production on the guide's side. For example, there are references with delayed positional descriptions like "then you take the green W ... top right", detailed references like "the green object that looks like a T top left in the corner", directional reinforcements like "more to the left" and confirmations like "exactly". We virtualize this setting as shown in Figure 2 for better control and to apply neural learning algorithms in a multi-agent setting. And we frame this as a game where both players receive a score after playing that represents their success and the effort spent for completion. Next, we explain the details of the game, its scoring and the prepared instances.

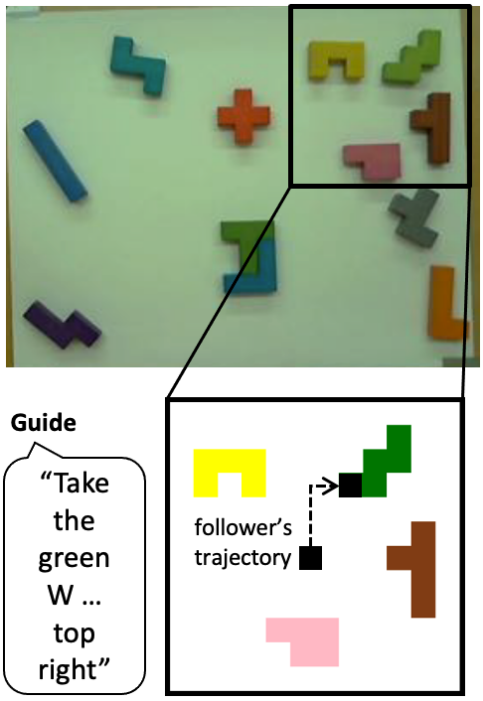

Figure 2: An example from Zarrieß et al. (2016) who found that a reference game leads to diverse language production on the guide's side. To study the aspects of cost sharing in such a collaborative interaction with neural agents, we propose CoGRIP along with a generator for virtual boards that eases the application of data-driven learning methods.

Actions. Formally, the guide's action space $\mathcal{A}_G$ spans all possible utterances of length L that are possible given the vocabulary V (in English) and includes an action for `silence`. Likewise the follower's action space $\mathcal{A}_F$ contains an action for hesitation (`wait`) and actions for movements (`left, right, up, down`) as well as an action to `take`. A board is internally represented as a grid of $M \times M$ tiles and the gripper can only move one tile at a time step. The gripper can move over pieces, but is not allowed to leave the boundaries of the board.

Effort. The efforts of the players represent the costs for success. We approximate the efforts based on the empirical observations from Zarrieß et al. (2016). The transcripts of the experiments suggest to group the guide's utterances into five categories to which we attach an effort estimate to each of the categories as follows:

$$E_{\mathsf{G}} = \sum_{t=1}^{T} \begin{cases} 0, & \text{if } a_t \in \{\texttt{silence}\} \\ 1.0, & \text{if } a_t \in \{\texttt{confirm}, \texttt{decline}\} \\ 2.0, & \text{if } a_t \in \{\texttt{directive}\} \\ 3.0, & \text{if } a_t \in \{\texttt{reference}\} \end{cases} \tag{1}$$

so that E_{G} represents the guide's effort in an episode with T steps. These action-based costs follow the assumed cognitive load for producing the according utterances: Here, staying `silent` (only watching) is the zero baseline. A `confirm` ("yes this way") carries the similar meaning (of detecting an action towards the goal) with the additional function of re-assuring the follower to act correctly by the cost of producing a short phrase. A `decline` phrase ("not this piece") signals the contradiction between the guides predicted action and the follower ones with the prospect of a follower's correction or if not, it buys time to produce a more demanding instruction. Such an instruction could be a `directive` ("go left/right/up/down", "take") which requires the comparison between the gripper's position and the target piece. Or even comparing all pieces with each other to produce a `reference` ("take the green W") which is reflected with the highest effort cost. For the follower we approximate the effort E_{F} with

$$E_{\mathsf{F}} = \sum_{t=1}^{T} \begin{cases} 0, & \text{if } a_t \in \{\texttt{wait}\} \\ 2.0, & \text{if } a_t \in \{\texttt{movement}\} \\ 3.0, & \text{if } a_t \in \{\texttt{take}\} \end{cases} \tag{2}$$

based on the assumed energy costs for performing the action (physically), i.e., lifting an object is harder than moving on a plane.

Score. We measure the quality of an episode of the game with a scoring function that follows the reward formulation of Chevalier-Boisvert et al. (2019)

$$\mathsf{S}(x) = 1 - 0.9 * (x / T_{\mathsf{max}}) \tag{3}$$

where T_{max} is a hyper-parameter that determines the maximal number of possible time steps in an episode of the game. Now, the quality score of an episode is given by the required time steps T to reach a terminal state $S_{\mathsf{Time}} = S(T)$, the joint effort score S_{Effort} and the overall outcome of the game S_{Outcome} which is $+1$ when the correct piece or a penalty of -1 if the wrong or no piece has been taken at all, so that:

$$\mathsf{S}_{\mathsf{Game}} = (\mathsf{S}_{\mathsf{Time}} + \mathsf{S}_{\mathsf{Effort}})/2 + \mathsf{S}_{\mathsf{Outcome}} \tag{4}$$

where $\mathsf{S}_{\mathsf{Effort}} = (\mathsf{S}(E_{\mathsf{G}}) + \mathsf{S}(E_{\mathsf{F}}))/2$. Given this formulation, the players have to to be active (not just wait until T_{max} is reached), achieve the goal (receive $S_{\mathsf{Outcome}} = +1$) and reduce their individual efforts (stay mostly `silent` or `wait` when the utterance is not understood) to reach the highest score. Thus the score ranges from about -2 (high effort, long and failure) to $+2$ (low effort, quick and successful).

Game Instance. An instance of the reference game is defined by the size of the board, a target piece and numerous distractor pieces. The appearance and position of the pieces is derived from symbolic piece representations: a tuple of shape (7), color (6), and position area (9). We use 315 $(7 \cdot 6 \cdot 9$ minus a holdout) of these symbolic pieces to create game instances and split them into distinct sets, so that the target pieces for the testing tasks differ from the ones seen during training (they might share color and shape but are, for example, positioned elsewhere). We provide 1750 training, 210 validation, and 245 testing instances of the task for three board sizes $(12, 21, 27)$. On these boards, a piece occupies five adjacent tiles and is not allowed to overlap with another one.

Evaluation. We quantify the performance on the task for a particular pair of follower and guide by letting them play all test game instances (where the follower always starts in the center of a map). We compute the achieved scores for these N testing episodes and average them to constitute the mean task score (mTS) for a pair of guide and follower. Furthermore, we are interested in the mean success rate (mSR) as the number of episodes where the correct piece was selected

$$\text{mSR} = \frac{1}{N} \sum_{i=1}^{N} s_i \text{ where } s_i = \begin{cases} 1, & \text{for correct piece} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

as well as the mean episode length (mEPL) as the number of time steps needed to take a piece (with the upper bound T_{max}) and the mean joint effort spent by the pair at each time step (mJE)

$$\text{mJE} = \frac{1}{N} \sum_{i=1}^{N} \frac{(E_{\mathsf{G}_i} + E_{\mathsf{F}_i})/2}{T_i} \tag{6}$$

which ranges from 0 to 3 (from the guide is always silent and the follower always waits to the guide utters a reference and follower performs take at each time step in an episode).

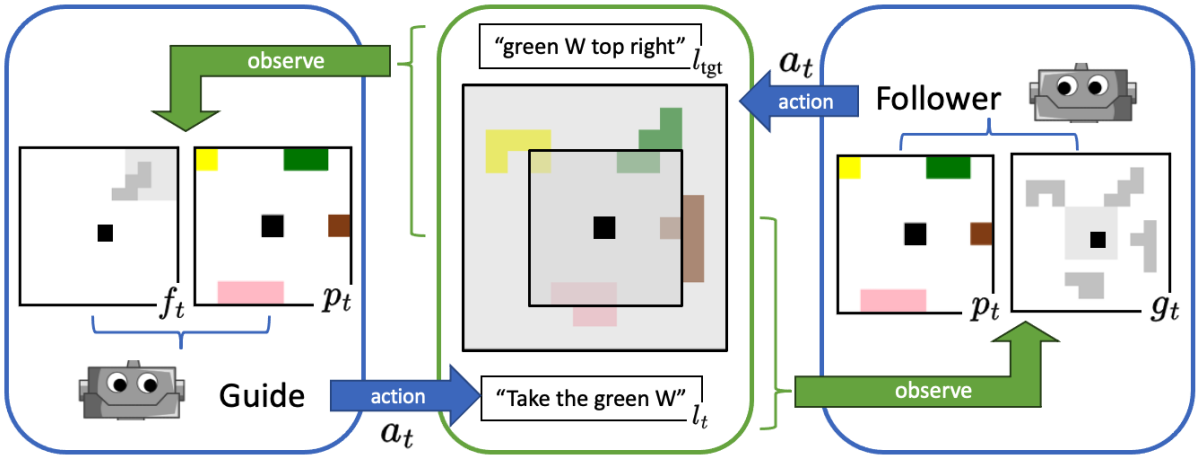

Figure 3: The general information and decision-making flow during an episode of the reference game. The guide observes a constant textual target piece descriptor l_{tgt}, the partial view p_t and a peripheral overview g_t of the scene. Given this, the guide chooses to produce a language action a_t which could mean "silence", a word, a phrase or a sentence that gets translated into an utterance l_t. The follower receives the utterance l_t, the partial view p_t and a peripheral overview f_t. Given this, the follower performs an action a_t that results into waiting, a movement (which changes the visual state) or an attempt to take a piece. The game ends when any piece is taken or the maximal number of time-steps T_{max} is reached.

3. Learning Neural Policies for Sharing the Cost of Success

Along with the newly proposed game for cost sharing, we determine baseline performances for neural policies and heuristic ones (which bootstrap them). The neural policies are supposed to learn the means of success in the game solely by playing with the partner. Here, the heuristic policies are supposed to help them to learn successful behavior in the reference game. We hypothesize that once the neural policies have learned how to achieve a successful outcome in the game (over a period of many cooperative interactions), a joint effort reduction takes place to achieve an even better score.

3.1. Problem Formulation

For our study, we methodologically frame this game as a reinforcement learning problem (Sutton and Barto, 1998) with sparse rewards. Thus, we treat the guide and follower from here on as *agents* that act in an *environment* (the game), which exposes observations to them. At each time-step t, given an observation $o_t \in \mathcal{O}$, the guide has to choose an action $a_t \in \mathcal{A}_{\text{G}}$ such that the overall resulting sequence of actions $(a_0, ..., a_t, ..., a_T)$ maximizes the sparse reward $\mathcal{R}(o_T) = S_{\text{Game}}$. Similarly, the follower has to choose an action $a_t \in \mathcal{A}_{\text{F}}$ at each time step to maximize the shared sparse reward. The follower and guide agents act at the same time-step but in consecutive order as depicted in Figure 3. An episode ends when a piece is selected by the follower or t reaches T_{max} so that an episode does not last forever and the trajectories do not become infinitely long.

3.2. Observations

Our intuition is that the overall task can be decomposed into two sub-tasks: First, the agents should agree on the area where the target piece is supposed to be located, e.g., the "top right". Then, after reaching this area with the follower's gripper, the agents have to coordinate to select the correct piece in that area (as shown in Table 1, as there is more than one candidate in the majority of cases).

Given these assumptions, we provide the learning agents with two visual perceptions of the scene: a lower-resolution peripheral overview (f_t or g_t) to agree on the target area by using positional utterances. And a colored higher resolution focus area (a partial view p_t, which is also commonly used in other vision-based reinforcement learning problems; (Hu et al., 2023; Chevalier-Boisvert et al., 2023)) to coordinate about the target piece with use of its shape and color attributes. The players

Size	N_{Pieces}	T_{max}	# DTA=0	# DTA≥1
12	4	30	430 / 58	1320 / 187
21	4–8	60	396 / 58	1354 / 187
27	4–16	80	360 / 48	1390 / 197

Table 1: The possible number of pieces (N_{Pieces}) for game instances with boards of the respective sizes and the maximal number of time steps (T_{max}). Game instances with board size 12 have always 4 pieces. For the other we choose uniform random from the range of piece amounts. In the majority of training/testing instances, there is at least one distractor (# DTA≥1) in the same positional area as the target piece e.g. both are in the "top right".

share the partial view p_t of the scene, which is centered around the gripper location. This can be interpreted as a behavior where the guide focuses on the follower's "hand". The overview observations f_t and g_t look slightly different for each agent to account for the knowledge asymmetry. The guide's overview g_t contains a mask of the target piece, the gripper position, and the ground-truth target area in respective channels. At the same time, the follower's overview f_t channels contain a mask for all pieces, the gripper position, and the current area, respectively. Furthermore, the guide receives at each time step a constant textual description l_{tgt} of the target piece (e.g., "blue T top right") while the follower receives the current utterance l_t produced by the guide (which could be silence).

3.3. Model Architecture

For both guide and follower, we use the same policy architecture as depicted in Figure 4. While the architecture is the same, the agents receive slightly different observations based on their role in the game (as described above). The observations o_t are first encoded into a 128-dimensional feature vector $\tilde{x}_t \in \mathbb{R}$. Then, the feature vector $\tilde{x}_t$ is fed through an LSTM (Hochreiter and Schmidhuber, 1997) to produce the memory-conditioned feature vector $\tilde{o}_t$. The LSTM passes a modifiable state vector h_t forward in time (which works as a memory). With this mechanism the follower could memorize already observed utterances (which allows the guide to stay silent), and the guide can anticipate a direction in which the follower is moving (and thus avoid repetitive utterance productions).

3.4. Learning Algorithm

We use *Proximal Policy Optimization* (PPO) (Schulman et al., 2017) to learn a parameterized actor-critic policy $\pi(\tilde{o}_t; \theta) \sim a_t$ where the actor predicts a distribution over the action space and the critic estimates the value of the states. The algorithm basically maximizes the surrogate objective

$$L(\theta) = \hat{\mathbb{E}}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}\left[r_t(\theta)\hat{A}_t\right] \quad (7)$$

but clips the ratio $r_t(\theta)$ if necessary to stabilize learning. This means when the critic favors the new state $\hat{A}_t > 0$ then the policy update is proportional to the ratio $r_t(\theta) \in [0, 1+\epsilon]$ which prevents a too large divergence. And for negative advantages $\hat{A}_t < 0$ the probability distribution over action is updated in the opposite direction proportional to the ratio $r_t(\theta) \in [1-\epsilon, \infty]$ which effectively reverts the increase in taking the less favorable action.

We use the recurrent PPO implementation of *StableBaselines3-Contrib* v2.1.0 (Raffin et al., 2021), because we have the state vector h_t that is passed forward in time as a memory mechanism. The implementation performs back-propagation through time until the first step in an episode.

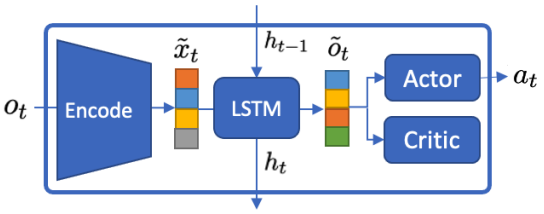

Figure 4: The neural agent's recurrent model architecture includes a memory mechanism (LSTM). At each time-step the observation o_t is encoded and then the resulting embedding $\tilde{x}_t$ is combined with a state representation h_{t-1} of previous time-steps.

3.5. Neural and Heuristic Policies

Learning cooperative neural agents in this environment from scratch requires a lot from them: the agents must learn (a) that the goal is to take a specific piece and none of the others, (b) the quality score is higher for strategies with less effort, and (c) the visual grounding of utterances themselves (reinforcement signals, references or directives). If training both agents at the same time from scratch, they may solve the task by learning a policy that amounts to a language that is inaccessible to humans (emergent languages; (Mul et al., 2019; Lowe et al., 2019)) because the vocabulary items can be freely associated with actions (meanings) that are different from what humans understand e.g. "left" may become (the action) `right`. Thus, we pair the learning neural agents with fixed heuristic ones that represent a proxy for competent speaker behavior.

3.5.1. A Neural Follower (NIF)

The role of the neural follower (NIF) is to take the piece described by the guide. For this the follower can perform a move, wait, or take action. Formally described, the follower receives at each time-step a vision and language observation $o_t = \{l_t, p_t, f_t\}$ and has to choose an action $a_t \in \{\texttt{wait}, \texttt{left}, \texttt{right}, \texttt{up}, \texttt{down}, \texttt{take}\}$, so that the sequence of actions $(a_0, ..., a_t, ..., a_T)$ maximizes the sparse reward $\mathcal{R}(o_T) = S_{Game}$.

3.5.2. A Heuristic Guide (HIG)

We pair the neural follower with a heuristic guide behavior (a fixed policy) that has been shown to lead to collaborative success with humans (Götze et al., 2022). The heuristic guide always has access to the ground-truth symbolic representations of the pieces on the board and the current gripper

position. Initially, the guide provides a referring expression l_0 that contains the properties necessary to distinguish the target, e.g., "Take the piece at the top right". Then the guide provides an utterance l_i at a time-step $t_{>0}$ only when the follower is over a piece or a pre-defined distance/time threshold $R \in \mathbb{N}$ has been exceeded (by comparison of the gripper's last and current position). This can be formally described with the following rules:

- `wait_thresh` → `reference` or `directive(dir)`
- `dist_thresh` → `dist_closer` or `dist_farther`
- `dist_closer` → `confirm`
- `dist_farther` → `decline` or `directive(dir)`
- `over_target` → `confirm` or `directive(take)`
- `over_other` → `decline` or `directive(dir)`

If none of these rules apply, the guide stays `silent` → `silence`. Note that the heuristic guide switches between the alternatives on the production side to provide more informative utterances than simply repeating. The production rules follow the effort categorization of Section 2. The utterance realization is based on the following templates:

- `silence` → `<empty string>`
- `confirm` → `Yes this [way|<piece>]`
- `decline` → `Not this [way|<piece>]`
- `directive(take)` → `Take <piece>`
- `directive(dir)` → `Go <dir>`
- `reference` → `Take the <IA(PO)>`

where `<piece>` resolves to a piece's color and shape when the current gripper position is located over a piece (or otherwise simply `piece`). The direction `<dir>` resolves to the necessary direction of movement. The reference production follows the Incremental Algorithm (`IA`; a cognitive algorithm by Dale and Reiter (1995)) that receives a preference over target piece properties (`PO`).

Here, the heuristic guide is supposed to mimic the intrinsic preference of humans (van Deemter, 2016). The most preferred property is usually the *type* of an object (Rosch and Lloyd, 1978), but in our visual domain all objects are "pieces" which makes this attribute uninformative. Although the *shape* could be a proxy for the type, the players would first need to agree on the idea that the pieces represent characters ("W", "T" etc.) and to use it successfully (Goudbeek and Krahmer, 2012). Instead the *color* is likely to be preferred by humans (Pechmann, 1989). Thus, when the follower's gripper is within the target piece area – meaning that the target piece is most likely visible – then the heuristic guide prefers color and shape to discriminate the target from its distractors. And otherwise, the guide prefers to mention the target piece's *position* to lead the follower into the target's position.

3.5.3. A Neural Guide (NIG)

The neural guide has to produce utterances that help the follower to select the target piece. More formally, the guide receives at each time step an observation $o_t = \{l_{tgt}, p_t, g_t\}$ and has to choose an action $a_t \in \{$`silence, confirm, decline, left, right, up, down, take, pcs, psc, cps, csp, spc, scp`$\}$ such that the overall resulting sequence of actions $(a_0, ..., a_t, ..., a_T)$ maximizes the sparse reward $\mathcal{R}(o_T) = S_{Game}$. The chosen actions are realized as utterances with the same mechanism that is used for the heuristic guide to reduce the burden on action space exploration. Note that the actions can be grouped into the five effort categories from Section 2 where `directive`'s are `left, right, up, down, take` and `reference`'s are the preferences orders `pcs, psc, cps, csp, spc, scp` (`c`=color, `s`=shape, `p`=position).

3.5.4. A Heuristic Follower (HIF)

The heuristic follower to be paired up with the neural guide should be similarly constrained as the neural follower (working with a partial view) so that both neural agents can play together after training with the heuristic partners. Thus, we took inspiration from Sun et al. (2023) and implemented a limited horizon planner that keeps track of and repeatedly revises a plan with up to 6 actions (the number of actions that is necessary to reach the diagonal corner of the partial view). The heuristic follower always has access to the ground-truth symbolic representation in the partial view and the current gripper position.

The actions in the plan are associated with a probability $p(a_i) = \max(\phi^i, \mathrm{L})$ of being executed where $\phi \in [0, 1]$ is a discount factor and $\mathrm{L} \in [0, 1]$ a lower threshold. This introduces a notion of *confidence*: either the planned action is executed, or a wait action occurs (hesitation). Furthermore, this conceptualizes that a follower becomes increasingly unsure about the continuation of the plan without receiving feedback. If an utterance is received, then its category is determined, and accordingly, one of five sub-programs is run to alter or revise the plan:

- `on_silence`: the follower executes, with respect to the confidence, the next action in the plan (if available). If the plan is empty, then the follower performs the `on_reference` sub-program.
- `on_confirm`: the follower sets the confidence for all actions in the current plan to 1. Then, the next action or `wait` is performed.
- `on_decline`: the follower erases the current plan and performs `wait`.

	12x12				21x21				27x27			
Pairing	**mSR**	**mEPL**	**mTS**	**mJE**	**mSR**	**mEPL**	**mTS**	**mJE**	**mSR**	**mEPL**	**mTS**	**mJE**
HIF-HIG	**1.00**	7.16	1.75	**1.36**	**0.99**	13.40	**1.74**	**1.33**	**0.98**	**17.64**	**1.73**	**1.33**
R=1	1.00	6.66	1.76	1.46	1.00	13.02	1.76	1.46	1.00	17.66	1.76	1.46
R=4	1.00	7.66	1.74	1.26	0.97	13.78	1.72	1.19	0.95	17.62	1.69	1.20
NIF-HIG	0.50	9.30	0.79	1.46	0.26	26.23	0.10	1.46	0.17	41.24	-0.18	1.48
R=1	0.57	10.42	0.82	1.60	0.29	29.12	0.06	1.62	0.21	44.62	-0.20	1.64
R=4	0.43	8.18	0.75	1.31	0.22	23.33	0.13	1.29	0.13	37.85	-0.16	1.32
HIF-NIG	**1.00**	**5.19**	**1.79**	1.51	0.96	**12.22**	1.66	1.77	0.90	20.29	1.46	1.84
NIF-PNIG*	0.99	6.15	1.73	1.77	0.95	16.23	1.54	1.90	0.93	23.30	1.47	1.94
PNIF-PNIG	0.96	7.20	1.63	1.71	0.87	19.35	1.33	1.79	0.69	37.27	0.77	1.90
NIF-NIG	0.95	8.04	1.57	1.63	0.73	27.61	0.87	1.77	0.55	47.7	0.34	1.80

Table 2: The performance of the neural and heuristic pairings on the test instances. We measure the mean success rates (mSR↑), the mean episode length (mEPL↓), the mean task scores (mTS↑) and the mean joint efforts (mRJE↓). The best values for a board size are in **bold**. The best neural-neural performance is underlined. PNIG* was kept frozen during training. PNIF-PNIG evaluated with last checkpoint.

- `on_directive`: the follower parses the utterances for directions or take. If the directive suggests taking, then the current plan is erased, and the `take` action is executed, assuming that this is the last action to be performed. Otherwise, the plan is overwritten with actions towards that direction, and the next action is executed.

- `on_reference`: the follower updates its internal target descriptor (color, shape, position) based on the current utterance (which might be empty when coming from `on_silence`). Afterward, the partial view is scanned for candidate coordinates based on the target descriptor, e.g., green coordinates given a reference to "Take the green piece at the top right". If the descriptor contains a position that is not yet reached, then moving towards that position is prioritized. Otherwise, if the positional area is unknown or already reached, then the shortest path to a candidate coordinate is established as the new plan. If the follower is already in the target area but has no information about shape and color, then a randomly chosen piece in the view is approached. In other cases, the follower `wait`s.

3.6. Experimental Setup

We pair the neural agents with the heuristic ones (HIF($\phi = 0.99$) and HIG($R = \{1, 4\}$) to bootstrap learning, and for comparison, we also run an experiment where they learn from scratch. We train each pairing on the 12×12 boards from the training split with four environments in parallel (batch size) and 10 million time-steps in total (for a multi-agent learning this means that each agent trains for 5 million steps). Thus, each board in the training split is seen at least 190 times. Every 100k steps during training, we evaluate the policies against the validation set. We keep the policies that achieve the highest mean episode reward based on the validation runs for later evaluation on the testing boards. We do this procedure for three different random seeds (49184, 92999, 98506) and average the results where applicable.

3.7. Results & Discussion

HIF-HIG is a very strong baseline pairing. The results in Table 2 show that the HIF-HIG pair achieves a 100% success rate along with the least joint effort (1.36) on the 12×12 test instances and generalizes to bigger map sizes as well. This very strong performance is supposedly achievable *after* a pairing went through an optimization process as observed by Clark and Wilkes-Gibbs (1986) which results in a collaborative strategy where utterances are mutually understood, properly grounded and produced in such a way that the individual effort is reduced without preventing a successful outcome.[1] The downside of the heuristic policies is that they cannot easily adapt to others or improve further.

HIF-NIG pairing exhibits "Guide A" strategy. Thus we pair a learning agent (neural guide) with a heuristic follower that can properly ground the utterances, so that the guide can easier learn to use the intents in a successful way. And indeed the HIF-NIG pair achieves a 100% success rate on 12×12 test boards with even less time steps as the heuristic pair (5.19) resulting in the highest task score (1.79). We also notice that the HIF-NIG pair generalizes to other map sizes. We find that the main reason for this superb performance is that the learnt strategy puts the most effort single-sided onto the guide: it provides a movement directive at almost every time step (see Figure 5). As hypothesised in the introduction, although this strategy is highly successfully, it does not result in the least joint effort. And indeed the mean effort is still higher (1.51) than the one of the heuristic pair (1.36).

[1] The NIF-HIG pairing does not fulfill this criteria as the low success rate (50%) indicates that the neural follower has not properly learned the goal condition of the game.

NIF-NIG pairings strive towards "Guide M". We hypothesized initially that the best strategy for the guide would be to initially produce a reference and then intervene *only when necessary* (the "M"iddle way of the extremes of producing an utterance at each time step or only initially). Such a strategy is presumably reached by an *adaption process* between the two collaborators. Since our heuristic policies cannot adapt to their counterparts we train also a pairing of learning agents (neural guides and followers). This NIF-NIG pairing achieves a remarkable success rate (95%) on the 12×12 boards based on a strategy that involves the whole repertoire of utterances (see Figure 5). Notably the guide learns to stay silent in almost 10% of the steps which reduces its effort.

Still, the neural policies might converge to a communication protocol that is inaccessible to humans. Thus, we pair a neural follower (NIF) with the pre-trained neural guide and keep the guide's parameters frozen (PNIG*). This guide learnt to produce the utterances in a way that humans (the heuristic follower) would understand. The results show that the neural follower successfully adapts to the "Guide A" strategy: The utterance production is about the same for HIF-NIG and NIF-PNIG* (see Figure 5). Consequently, the pair achieves a similar high success rate (99%) and the shortest episodes (6.15) among the neural-neural pairings.

Now the communication strategy of the neural-neural pairing (PNIF-PNIG) is more accessible to humans, but the joint effort (1.77) is above the one from the NIF-NIG (1.63). We continue training the pre-trained agents (using their best checkpoints as starting points) in a multi-agent fashion another 10M time-steps, so that the neural agents can further *adapt to each other*. We see that the neural pairings strive towards a strategy that further reduces the joint effort while maintaining the high success rates as shown in Figure 6. The best seed achieves a mean joint effort of 1.53, which is just above the heuristics, and the resulting overall mean efforts (1.71) are lower than before (1.77). The neural agents that went through the *adaption process* conduct a strategy that involves more references and also more silence (see Figure 5). This indicates that the neural agents strive towards a "Guide M" strategy that shares the cost of success better.

Pairing	mSR ↑	mEPL ↓	mTS ↑	mJE ↓
HIF-NIG^{W}	**1.00**	**5.71**	**1.80**	**1.13**
NIF-PNIG^{W}*	**1.00**	5.93	1.79	1.25
PNIF-PNIG^{W}	0.94	7.84	1.61	1.24
NIF-NIG^{W}	0.84	10.48	1.34	1.23

Table 3: The results for the word-level pairings of guide and heuristic follower on the test boards of size 12. The assumed effort per word is here 1.0 (and thus not directly comparable with Table 2).

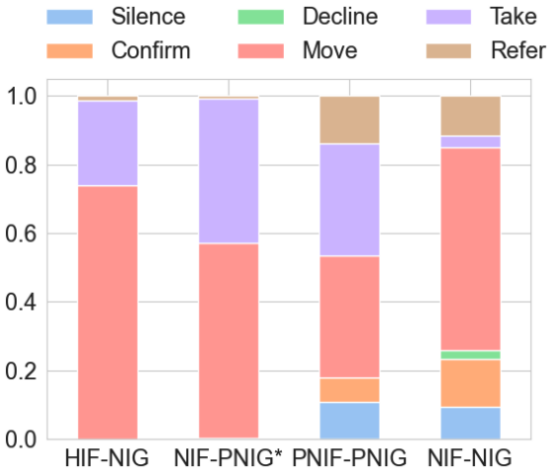

Figure 5: The relative usage of utterance categories per time-step for the guide in various pairings.

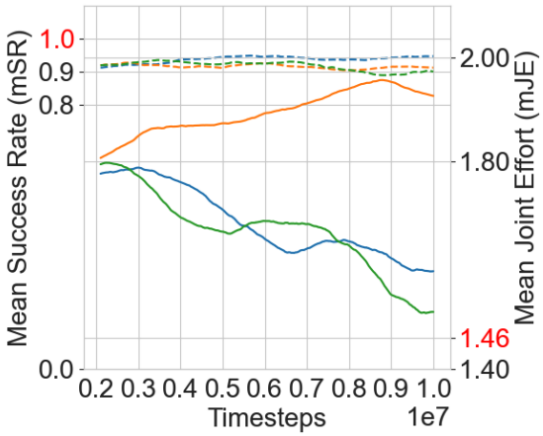

Figure 6: The (smoothed) training curves for the PNIF-PNIG pairing show the mean success rate (dashed lines) stays high and that the mean joint effort (mJE) reduces further (for 2 of 3 seeds). The HIF-HIG performance is indicated by the red ticks.

Interactive language learner mimics "Guide A". The previously described guides produce references by choosing a preference order. This more abstract prediction level allows the guide to focus on learning a useful coordination strategy. Nonetheless, the follower has to understand the actual realization of the reference to perform its actions. Thus we additionally investigate, if a neural guide can learn a useful language production from the interaction alone. We convert "intent"-actions to words and let a neural guide (NIG^{W}) choose actual property values (colors, shapes and positions) which leads to 24 "word"-actions in total. We adjust the heuristic follower to categorize the words correctly and assume an effort of 1 for a guide's action. A produced word is fed back to the guide in addition to the other observations. The results in Table 3 show that the NIG^{W} achieves high success rates. And a qualitative analysis (e.g. Figure 7) reveals that these results are based on a "Guide A" strategy.

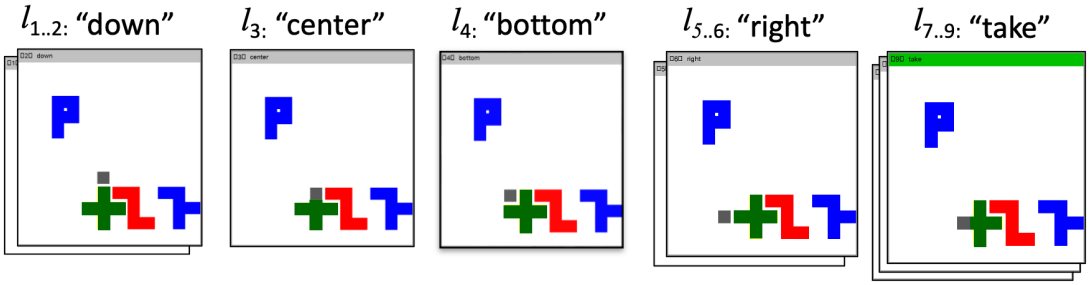

Figure 7: An example episode of the PNIF-PNIG^{W} pair on the validation boards after training. The guide produces a word at each time step to almost "remote control" the follower resulting in a high success rate.

4. Related Work

As an initial study our work connects ideas from the linguistic interaction field which asks how language shapes an interaction of interlocutors (and vice versa) (Gandolfi et al., 2022) and the vision and language field where the actions are visually-grounded (Zhang and Kordjamshidi, 2022; Dainese et al., 2023). We realize this connection via the paradigm of reinforcement learning (RL) that introduces a notion of time and allows for incremental processing (which has been recently studied for interactive dictation by Li et al. (2023)). We notice that recent work towards adaptive NLG (Ohashi and Higashinaka, 2022) and language feedback (Yan et al., 2023) are not visually grounded or approaches that involve vision have no (adaptive) interactive feedback (Zhang and Kordjamshidi, 2022; Dainese et al., 2023) that has to be learnt from the interaction alone without pre-trained on a pre-collected dataset. Our work is an attempt to connect the research ideas of these fields.[2]

Vision and language navigation. The proposed reference game shares similarities with vision and language navigation as the follower has to select (navigate to) a specific piece given an utterance. Nevertheless, in navigation tasks, there is usually a lengthy and detailed initial instruction which is accessible at every time step, and the metrics of interest are success rate alone (Chevalier-Boisvert et al., 2019), or additionally episode length (Nguyen and III, 2019; Fried et al., 2018). We are especially interested in the behavior under the constraint of an assumed joint effort and focus on the incremental aspects of language and vision coordination. In our setting, the agents are required to perceptually ground the language to produce a movement (see also Chevalier-Boisvert et al. (2023) as an example of a popular abstract navigation domain) or to produce a language act (here a verbalized intent to reduce space complexity) given the vision inputs at each time-step.

[2]Source code is publicly available at: https://github.com/clp-research/cost-sharing-reference-game

Cooperative multi-agent RL environments. Multi-player games present a useful environment to study multi-agent behavior with reinforcement learning, as there are usually well-defined constraints and rewards. Though to our knowledge, the communication between agents is usually not done via language utterances (Bard et al., 2020; Samvelyan et al., 2019; Pan et al., 2022; Mohanty et al., 2020; Kurach et al., 2020). The most similar environment we found is from Mordatch and Abbeel (2018), which studied cooperative communication where a listener has to navigate to one of three landmarks. The target is only known by a speaker that can not move. The agents learned from a dense reward signal, which is the distance to the ground-truth landmark. In our game, we only provide a sparse reward and are interested in the behavior after learning to be successful.

5. Conclusion & Further Work

In this work, we proposed a new game to study cooperative multi-agent behavior for cost-sharing, and we presented neural and heuristic policies for learning in this environment. We showed that an off-the-shelf learning algorithm (PPO) with a simple reward mechanism (sparse) learns policies that are successful in the game and that continue reducing an assumed joint effort. Nevertheless, the resulting agents lack variety in their coordination strategies (converge to remote control) and still require more effort than a sensible heuristic pairing. Thus our reference game provides a useful foundation and suggests further research in this interesting topic, so that future neural agents learn more diverse (human-like) language-based coordination behaviors and share the cost of success even better with their interaction counterparts.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback. This work was funded by the *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation) – 423217434 ("RECO-

LAGE") grant.

Limitations

Limits on visual naturalness. We chose this relatively abstract setting so as to be able to investigate in detail the contribution of each modelling decision. Moving to a more realistic and visually more complex environment is a necessary, but logically later, step. Nevertheless, we think our approach can also be applied to photo-realistic environments (Ramakrishnan et al., 2021; Kolve et al., 2017).

Limits on the visual variety. The variety of pieces is limited to 7 different shapes and 6 different colors. Furthermore, the pieces show no texture but are drawn with a solid color fill. Nevertheless, the visualisations are fast to compute and despite of their simplicity we observed that such a setting produces interesting and complex interactions between a follower and a guide. We leave experimentation on visually even more complex scenes or scenes with ambiguity for future work.

6. Bibliographical References

Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. 2020. The hanabi challenge: A new frontier for AI research. *Artif. Intell.*, 280:103216.

Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. 2021. Goal-conditioned reinforcement learning with imagined subgoals. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 1430–1440. PMLR.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. 2019. Babyai: A platform to study the sample efficiency of grounded language learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. 2023. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831.

Herbert H. Clark. 1996. *Using Language*. 'Using' Linguistic Books. Cambridge University Press, Cambridge.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39. Place: Netherlands Publisher: Elsevier Science.

Nicola Dainese, Pekka Marttinen, and Alexander Ilin. 2023. Reader: Model-based language-instructed reinforcement learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 16583–16599. Association for Computational Linguistics.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cogn. Sci.*, 19(2):233–263.

Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3318–3329.

Greta Gandolfi, Martin J. Pickering, and Simon Garrod. 2022. Mechanisms of alignment: shared control, social cognition and metacognition. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 378(1870):20210362. Publisher: Royal Society.

Solomon W. Golomb. 1996. *Polyominoes: Puzzles, Patterns, Problems, and Packings*. Princeton University Press.

Martijn Goudbeek and Emiel Krahmer. 2012. Alignment in interactive reference production: Content planning, modifier ordering, and referential overspecification. *Topics in Cognitive Science*, 4(2):269–289. Place: United Kingdom Publisher: Wiley-Blackwell Publishing Ltd.

Jana Götze, Karla Friedrichs, and David Schlangen. 2022. Interactive and Cooperative Delivery of Referring Expressions: A Comparison of Three Algorithms. In *Proceedings of the 26th Workshop on the Semantics and Pragmatics of Dialogue* -

Full Papers, Virtually and at Dublin, Ireland. SEMDIAL.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Chengpeng Hu, Ziqi Wang, Tianye Shu, Hao Tong, Julian Togelius, Xin Yao, and Jialin Liu. 2023. Reinforcement learning with dual-observation for general video game playing. *IEEE Trans. Games*, 15(2):202–216.

Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. 2017. AI2-THOR: an interactive 3d environment for visual AI. *CoRR*, abs/1712.05474.

Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. 2020. Google research football: A novel reinforcement learning environment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 4501–4510. AAAI Press.

Gyeong Taek Lee and Kang Jin Kim. 2023. A controllable agent by subgoals in path planning using goal-conditioned reinforcement learning. *IEEE Access*, 11:33812–33825.

Belinda Z. Li, Jason Eisner, Adam Pauls, and Sam Thomson. 2023. Toward interactive dictation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15319–15338. Association for Computational Linguistics.

Ryan Lowe, Jakob N. Foerster, Y-Lan Boureau, Joelle Pineau, and Yann N. Dauphin. 2019. On the pitfalls of measuring emergent communication. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pages 693–701. International Foundation for Autonomous Agents and Multiagent Systems.

Sharada Prasanna Mohanty, Erik Nygren, Florian Laurent, Manuel Schneider, Christian Scheller, Nilabha Bhattacharya, Jeremy D. Watson, Adrian Egli, Christian Eichenberger, Christian Baumberger, Gereon Vienken, Irene Sturm, Guillaume Sartoretti, and Giacomo Spigler. 2020. Flatland-rl : Multi-agent reinforcement learning on trains. *CoRR*, abs/2012.05893.

Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1495–1502. AAAI Press.

Mathijs Mul, Diane Bouchacourt, and Elia Bruni. 2019. Mastering emergent language: learning to guide in simulated navigation. *CoRR*, abs/1908.05135.

Khanh Nguyen and Hal Daumé III. 2019. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 684–695. Association for Computational Linguistics.

Atsumoto Ohashi and Ryuichiro Higashinaka. 2022. Adaptive natural language generation for task-oriented dialogue via reinforcement learning. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 242–252. International Committee on Computational Linguistics.

Xuehai Pan, Mickel Liu, Fangwei Zhong, Yaodong Yang, Song-Chun Zhu, and Yizhou Wang. 2022. MATE: benchmarking multi-agent reinforcement learning in distributed target coverage control. In *NeurIPS*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Thomas Pechmann. 1989. Incremental speech production and referential overspecification. 27(1):89–110. Publisher: De Gruyter Mouton Section: Linguistics.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. 2021. Stable-baselines3: Reliable reinforcement learning implementations. *J. Mach. Learn. Res.*, 22:268:1–268:8.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. 2021. Habitat-matterport 3d dataset (HM3D): 1000 large-scale 3d environments for embodied AI. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Eleanor Rosch and Barbara B. Lloyd, editors. 1978. *Cognition and categorization*. Cognition and categorization. Lawrence Erlbaum, Oxford, England. Pages: viii, 328.

Mikayel Samvelyan, Tabish Rashid, Christian Schröder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob N. Foerster, and Shimon Whiteson. 2019. The starcraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pages 2186–2188. International Foundation for Autonomous Agents and Multiagent Systems.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. 2023. Adaplanner: Adaptive planning from feedback with language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning - An Introduction*. Adaptive computation and machine learning. MIT Press.

Kees van Deemter. 2016. *Computational Models of Referring: A Study in Cognitive Science*. The MIT Press.

Hao Yan, Saurabh Srivastava, Yintao Tai, Sida I. Wang, Wen-tau Yih, and Ziyu Yao. 2023. Learning to simulate natural language feedback for interactive semantic parsing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 3149–3170. Association for Computational Linguistics.

Sina Zarrieß, Julian Hough, Casey Kennington, Ramesh R. Manuvinakurike, David DeVault, Raquel Fernández, and David Schlangen. 2016. Pentoref: A corpus of spoken references in task-oriented dialogues. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Yue Zhang and Parisa Kordjamshidi. 2022. Lovis: Learning orientation and visual signals for vision and language navigation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 5745–5754. International Committee on Computational Linguistics.

A. Appendix

Robot image in Figure 1 adjusted from `https://commons.wikimedia.org/wiki/File:Cartoon_Robot.svg`. That file was made available under the Creative Commons CC0 1.0 Universal Public Domain Dedication.

A.1. Observation Details

The environment exposes at each time-step all relevant observations so that any combination of the policies can be used within the environment. This means that the neural policy learners use the partial visual observation and the tokenized language utterance along with the positional mask as follows ($|V| = 54$ is the vocabulary size, $L = 16$ is the maximum sentence length and M is the map size.)

Neural follower observations:

- `RGB_PARTIAL` $= \{p_t \in \mathbb{N}_0^{7\times7\times3} | p_t \leq 255\}$
- `POS_FULL_CURRENT` $= g_t \in \{0,1\}^{M\times M\times4}$
- `LANGUAGE` $= \{l_t \in \mathbb{N}^L | 0 \leq l_t \leq |V|\}$

where `RGB_PARTIAL` is the partial RGB view around the current gripper position, `POS_FULL_CURRENT` contains masks for the board, the current gripper position, the pieces on the board and the current positional area and `LANGUAGE` contains the last produced utterance.

Neural guide observations:

- RGB_PARTIAL $= \{p_t \in \mathbb{N}_0^{7\times7\times3} | p_t \leq 255\}$
- POS_FULL_TARGET $= f_t \in \{0,1\}^{M\times M\times 4}$
- TARGET_DESC $= \{l_{tgt} \in \mathbb{N}_0^L | l_{tgt} \leq |V|\}$

where RGB_PARTIAL is the partial RGB view around the current gripper position, POS_FULL_CURRENT contains masks for the board, the current gripper position, the target piece on the board and the target's positional area and TARGET_DESC contains the tokenized properties of the target. And the heuristic policies use the symbolic equivalents of the observations as follows:

Heuristic follower observations:

- SYM_PARTIAL $= \{P_t \in \mathbb{N}_0^{7\times7\times3} | P_t \leq 255\}$
- SYM_AREA $= A_t \in \{1, ..., 9\}$
- SYM_POS $= \{G_t \in \mathbb{N}_0^2 | G_t \leq M\}$
- LANGUAGE $= \{l_t \in \mathbb{N}_0^L | l_t \leq |V|\}$

where SYM_PARTIAL is the partial symbolic view around the current gripper position with the symbolic colors, shapes and object id channels, SYM_AREA is the symbolic representation of the positional area the gripper is currently in, SYM_POS are the gripper's current x, y-coordinates and LANGUAGE contains the last produced utterance.

The symbolic representations for the shapes are: P (2), X (3), T (4), Z (5), W (6), U (7), F(8). The colors are encoded as: red (2), green (3), blue (4), yellow (5), brown (6), purple (7). The 0-symbol is reserved for out-of-world tiles which can occur in the partial view and peripheral view masks. The 1-symbol is reserved for an empty tile. The positional areas are enumerate as: top left (1), top center (2), top right (3), right center (4), bottom right (5), bottom center (6), bottom left (7), left center (8), center (9).

Heuristic guide observations:

- SYM_POS $= \{G_t \in \mathbb{N}_0^2 | G_t \leq M\}$

where SYM_POS are the gripper's current x, y-coordinates. In addition, the heuristic guide receives the information about the target's attributes and position at the start of each episode. The distances between two coordinates (p_1, p_2) are calculated as the euclidean distance.

A.2. Neural Policy Details

The agents encode two streams of visual inputs: one is the partial visual observation of the scene in colors (RGB) and the other is an overview of the scenes that encodes the position of the gripper on the board and the targeted (or current) positional area. Each of the vision embeddings is input to a FiLM layer that conditions the vision inputs on the target piece descriptor (for the Guide) or the utterance (for the Follower). These language-conditioned vision embeddings are then concatenated and input to the policy network. All model implementations are done in PyTorch v1.13.0 (Paszke et al., 2019).

Partial View Encoding. For the encoding of the partial view we use a CNN with 4 blocks of convolutions, batch norm and relu activations. The first block applies 32 kernels of size 5 with stride 1 and padding. This layer is supposed to learn edges and colors. The second layer applies 64 kernels of size 5 with stride 5 and no padding to shrink the input to the original spatial dimensions of 7×7. Then layer 3 and 4 apply 128 kernels of size 3 and padding each resulting in 128 7×7 feature maps that embed the high level visual information of the partial view.

Overview Encoding. For the encoding of the overview we also use a CNN with 4 blocks of convolutions, batch norm and relu activations. The first block applies 32 kernels of size 1 with stride 1 and no padding. This block is supposed to learn whether the gripper is located in the target area. The other blocks apply 64, 128, 128 kernels of size 3 with padding respectively resulting in 128 $W \times H$ feature maps that embed the high-level positional information of the overview.

Language-conditioning. The language observations of the agents (the target descriptor for the Guide and the utterances for the Follower) are embedded to 32-dimensional word vectors and then encoded with a GRU which has 128 hidden state dimensions. The last state of the GRU is given as the language encoding to the FiLM layers: one layer that conditions the partial view and one layer that conditions the overview encoding.

Recurrent Policy Network. The two language-conditioned visual embeddings are added and passed to an LSTM with 128 hidden dimensions. The LSTM embeds the observations over time and can keep track of previous actions. The LSTM's last hidden state is then given to the actor-critic policy network. The actor and the critic are 2-layer feedforward networks where each layer has 64 parameters.

feature_dims	128
normalize_images	True
shared_lstm	True
enable_critic_lstm	False
n_lstm_layers	1
lstm_hidden_size	128
net_arch	[[64,64], [64,64]]

Table 4: Policy arguments for the the neural agents.

Hyperparameters. We use the RecurrentPPO implementation from StableBaselines-Contrib v2.1.0 with the default learning hyper-parameters and the policy parameters as given in Table 4.

Experiments. We trained the pairings in parallel on 8 GeForce GTX 1080 Ti (11GB) where each of them consumed about 4GB of GPU memory. The training of an individual pairing (and seed) for the 5 million steps took about 1 day.

For the multi-agent training we switched the agent to be updated after each policy update so that the training took 10 million steps in total.

A.3. Incremental Algorithm (IA)

Both the neural and heuristic guide employ the Incremental Algorithm for referring expression generation via the selection of a preference order (except the NIG^{W} which produces the actual words directly).

Algorithm. The Algorithm 1, in the formulation of (Dale and Reiter, 1995), is supposed to find the properties that uniquely identify an object among others given a preference over properties. To accomplish this the algorithm is given the property values $\mathcal{P}$ of distractors in M and of a referent r. Then the algorithm excludes distractors in several iterations until either M is empty or every property of r has been tested. During the exclusion process the algorithm computes the set of distractors that do *not* share a given property with the referent and stores the property in $\mathcal{D}$. These properties in $\mathcal{D}$ are the ones that distinguish the referent from the others and thus will be returned.

Preference order. The algorithm has a meta-parameter $\mathcal{O}$, indicating the *preference order*, which determines the order in which the properties of the referent are tested against the distractors. In our domain, for example, when *color* is the most preferred property, the algorithm might return BLUE, if this property already excludes all distractors. When *shape* is the preferred property and all distractors do *not* share the shape T with the referent, T would be returned. Hence even when the referent and distractor pieces are the same, different preference orders might lead to different expressions.

Algorithm 1 The IA on symbolic properties as based on the formulation by van Deemter (2016)

Require: A set of distractors M, a set of property values $\mathcal{P}$ of a referent r and a linear preference order $\mathcal{O}$ over the property values $\mathcal{P}$

$\mathcal{D} \leftarrow \emptyset$
for P in $\mathcal{O}(\mathcal{P})$ **do**
  $\mathcal{E} \leftarrow \{m \in M : \neg P(m)\}$
  if $\mathcal{E} \neq \emptyset$ **then**
    Add P to $\mathcal{D}$
    Remove $\mathcal{E}$ from M
  return $\mathcal{D}$

Templates. There are 3 expression templates that are used when only a single property value of the target piece is returned by the Incremental Algorithm (IA):

- *take the [color] piece*
- *take the [shape]*
- *take the piece at [position]*

Then there are 3 expression templates that are selected when two properties are returned:

- *take the [color] [shape]*
- *take the [color] piece at [position]*
- *take the [shape] at [position]*

And finally there is one expression templates that lists all property values to identify a target piece:

- *take the [color] [shape] at [position]*

A.4. Task Generation

Symbolic piece splits. For the task generation we first split the set of the 378 possible symbolic pieces (a combination of color, shape and position) into different subsets, so that training, validation and testing splits do not overlap. This results into $250/30/35$ symbolic pieces for training/validation/testing respectively (and a holdout of 63 symbols that we did not use).

Utterance type-oriented sampling. Then we iterate through the symbolic pieces in the split and treat each of them as the target piece once. For the target piece we sample a set of distractor pieces in such a way that the IA's reference production would lead to one of the templates (from above) once. This means that per target piece 7 different distractor sets are sampled which leads to $1750/210/245$ tasks for training/validation/testing respectively. For each task the pieces are put on an initially empty board starting with the target. And then the other pieces are tried to be placed. If a piece cannot be placed on a board without collision, then we choose another coordinate and try this up to 100 times for each placement, until all pieces are placed.