Learning Communication Policies for Different Follower Behaviors in a Collaborative Reference Game

Philipp Sadler[1], **Sherzod Hakimov**[1] and **David Schlangen**[1,2]

[1]CoLabPotsdam / Computational Linguistics
Department of Linguistics, University of Potsdam, Germany
[2]German Research Center for Artificial Intelligence (DFKI), Berlin, Germany
firstname.lastname@uni-potsdam.de

Abstract

Albrecht and Stone (2018) state that modeling of changing behaviors remains an open problem "due to the essentially unconstrained nature of what other agents may do". In this work we evaluate the adaptability of neural artificial agents towards assumed partner behaviors in a collaborative reference game. In this game success is achieved when a knowledgeable Guide can verbally lead a Follower to the selection of a specific puzzle piece among several distractors. We frame this language grounding and coordination task as a reinforcement learning problem and measure to which extent a common reinforcement training algorithm (PPO) is able to produce neural agents (the Guides) that perform well with various heuristic Follower behaviors that vary along the dimensions of confidence and autonomy. We experiment with a learning signal that in addition to the goal condition also respects an assumed communicative effort. Our results indicate that this novel ingredient leads to communicative strategies that are less verbose (staying silent in some of the steps) and that with respect to that the Guide's strategies indeed adapt to the partner's level of confidence and autonomy.

1 Introduction

Sometimes we feel like we could continue another person's sentence. This happens in particular with people we know well or we often interact with. A common phrase coined to this phenomenon is that "people are on the same wavelength". And indeed Davidesco et al. (2023) found that brain activities somewhat synchronize between teachers and students during lessons. Even more surprising, synchronicity becomes a good predictor of the learning success of the students. A psycho-linguistic study by Clark and Wilkes-Gibbs (1986) observed the language use of collaborative partners during an ongoing goal-oriented interaction: They (implicitly) agree on newly introduced noun phrases and a common strategy to achieve the goal together. Interestingly, the number of used words drastically decrease during the collaboration. The participants strive towards reduced individual efforts while the number of successful outcomes stays high. We see that human interaction is characterized by synchronicity (adaption) and the reduction of the individual effort. In this work we study how (artificial) learning agents adapt to an assumed partner's behavior. For this, we propose a simple, but still challenging vision and language grounding task where two players have to coordinate on the selection of a puzzle piece (Pentomino; Golomb (1996)) while (i) the actual target piece is only known to one of them (the Guide), and (ii) only the other can perform the selection (the Follower). See Figure 1 for an example illustration of this goal-oriented collaborative game. Clark (1996) points out that in these situations language acts as a device for solving the coordination problem: If the participants agree on a mutually desired outcome (the goal; for example taking a specific piece) then their individual participatory actions take part in a joint action. The regularity in behavior, common ground, and the recurrence of the coordination prob-

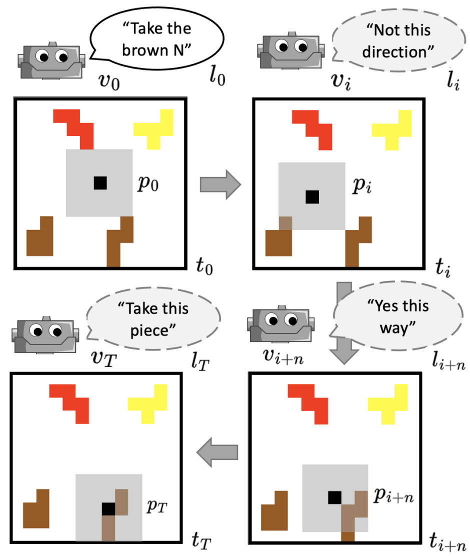

Figure 1: An exemplary interaction between a Guide and a Follower that controls the gripper (the black dot). The Guide observes the scene v_0 and refers to a piece initially with l_0. The Follower has only a partial view p_0 (the grey box) and might go wrong. The Guide can provide further information based on the Follower's actions until a piece is selected at time step T. The Guide should learn that less utterances are necessary with a more autonomous and confident Follower.

lem lets them settle on conventions – and ultimately adapt to each other. We think that such capabilities would be essential for future assisting agents that might take part in society someday (see Park et al. (2023) for a toy example). Albrecht and Stone (2018) found that modeling of changing behaviors (or different others) remains an open problem "due to the essentially unconstrained nature of what other agents may do". Are neural agents capable to adapt to their interactants and converging to strategies that become useful only during the dynamic interaction itself (when the partner's behavior becomes apparent)?

In this paper, we frame the language coordination and grounding task as a reinforcement learning problem (Sutton and Barto 2018) and evaluate, if and to which extent a common training algorithm Proximal Policy Optimization (PPO) (Schulman et al. 2017) is able to produce guiding neural agents that perform well with a variety of Follower behaviors in a collaborative setting where the Guide's utterances become language actions. In this scenario, an agent (or possibly multiple ones) take step-wise actions in an observable and dynamic environment to maximize a reward signal.

The main idea is that we assume an ongoing interaction in which the Follower's behavior changes. After some time the Follower should become more autonomous and more confident in choosing actions and executing its own plan (as pointed out by Clark and Wilkes-Gibbs (1986)). But instead of treating this as a multi-agent setting directly, we follow Yang et al. (2022) (with the notion of assigning different agents to different sub-tasks) and learn separate Guides for each of the (hand-crafted) Follower behaviors (sub-tasks). The resulting policies represent a Guide's communicative strategy at certain points in time of the assumed ongoing interaction. Our expectations on the learned communicative strategies of the Guide are that in the beginning (with a less autonomous, less confident Follower) more is to be said. And that with a more autonomous and confident Follower the Guide learns that it "does not need to say anything" to be successful (and consequently reducing the effort). Our contributions are the following[1]:

- We propose a challenging RL environment: a reference game in which a neural agent (the Guide) has to learn communication strategies that are **successful and reduce an assumed effort**, and
- contribute a plausible Follower policy (the training partner) that is variable on two dimensions: **confidence** and **autonomy**, and
- present strong baseline Guide policies for this difficult cooperative reference game that are indeed able to balance out episode success and their individual effort by **learning to stay silent**.

2 Related Work

Vision and language navigation. The use of natural language to guide an instruction following agent has been heavily studied for the vision and language navigation task (Gu et al. 2022; Nguyen et al. 2019; Nguyen and III 2019; Fried et al. 2018; Thomason et al. 2019). For example, Nguyen and III (2019) train an instruction giver (IG) on a pre-collected dataset of instructions. The Follower is then allowed to ask the IG for more information during task execution. Although the setting is very similar, we distinguish from these works as our Guide has to learn itself when to provide more information to the Follower. In our setting, the language back-channel for the Follower is cut, so that the Guide's timing and utterance choice becomes essential.

Natural language goals in RL. Using natural language to describe the goal state in an RL problem has become a common theme (Chevalier-Boisvert et al. 2019; Gao et al. 2022; Padmakumar et al. 2022; Pashevich, Schmid, and Sun 2021; Suhr and Artzi 2022). This research direction is interesting because it could allow humans to interact more easily with learned agents. There is work that shows that intermediate language inputs are a valuable signal in task-oriented visual environments (Co-Reyes et al. 2019; Mu et al. 2022). Indeed Huang, Lipovetzky, and Cohn (2023) found that natural language can "provide a gradient" towards the goal state. But they also point out the "brittleness" of these signals because the language input might align badly with sub-trajectories. A key challenge here is the variability of expressions in language that can be produced and understood in the defined action space. Even in relatively simple environments, there might arise an overwhelming amount of situations for an agent to handle (Chevalier-Boisvert et al. 2019). We weaken the action space exploration problem by using ideas from natural language understanding (Moon et al. 2020; E et al. 2019) and let the guide produce language actions in a well-defined reduced "intent space". These intents are then verbalized (using templates; which could be a conditioned pre-trained language model) and given to the follower.

Interactive sub-goal generation in RL. Sun et al. (2023) use a pre-trained large language model to generate possible plans (in the form of source code) for the completion of a task. They introduce a distinction between implicit and explicit closed-loop systems that are able to either refine single actions or an entire plan respectively. Indeed neural agents perform better when they self-predict sub-goals to be achieved (with an intrinsic reward) instead of reaching for the final goal immediately (Jurgenson and Tamar 2023; Chane-Sane, Schmid, and Laptev 2021; Pertsch et al. 2020; Jeon et al. 2022). For example, Lee and Kim (2023) study the task of finding the best route in a simple visual domain by training a sub-goal system that predicts intermediate coordinates. In contrast to them, our guiding agent has to produce a natural language utterance to describe a sub-goal (and we use referring expressions or directions). Gürtler, Büchler, and Martius (2021) also address the question of "when to provide sub-goals", which is necessary in our task. Nevertheless, in distinction to these works, we treat the sub-goal generation not just as additional information for the follower's success but are interested in the learned communicative strategies themselves. We treat the sub-goal providing guide as an individual participant in the environment similar to a multi-agent setting.

[1]Source code is publicly available under: https://github.com/clp-research/different-follower-behaviors

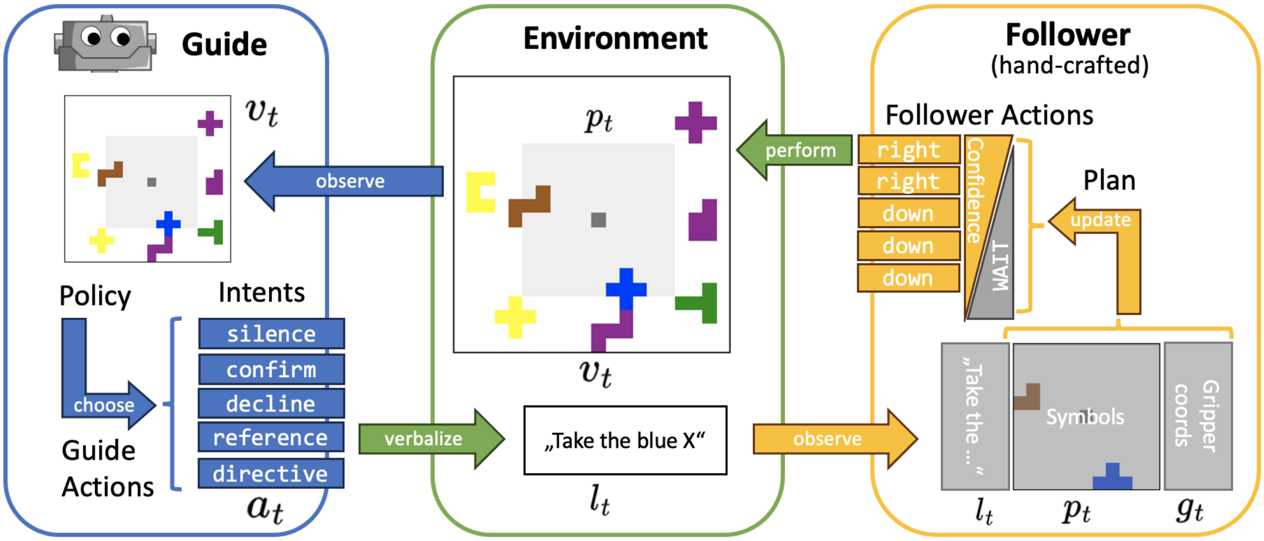

Figure 2: The general information and decision-making flow of the reference game. The Guide observes v_t which contains the full scene in pixel space and additionally the gripper position (4th-channel) and target piece (5th-channel). Given this, the Guide chooses an intent action a_t that gets verbalized into a natural language sentence l_t. Then, the Follower receives the utterance l_t, the gripper coordinate g_t and a symbolic representation of a partial view of the scene p_t. The hand-crafted policy updates the plan accordingly based on its given representation of the world. Finally, the Follower's next planned action (or `wait`) is performed with a certain chance defined by the attached confidence. The process repeats until a piece is taken or time runs out.

Skill learning in cooperative multi-agent RL. We treat both guide and follower as agents in a cooperative setting and follow work that uses hand-crafted policies (Wang et al. 2021; Ghosh et al. 2020; Xie et al. 2020) (here a follower that is able to mimic behavior that varies in autonomy and confidence). In this sense, our approach is similar to heterogeneous skill learning (Chang et al. 2022; Liu et al. 2022; Hu et al. 2023) where a single agent is trained to acquire a variety of skills (in our case communicative strategies). This is, in particular, helpful due to the differences in the action spaces of the guide (language acts) and the follower (movements). In addition, this method (of having a fixed hand-crafted follower policy) allows us to avoid the problem of emergent communication where agents agree on a language that becomes inaccessible by humans (Lowe et al. 2019; Mul, Bouchacourt, and Bruni 2019; Kolb et al. 2019).

3 The CoGRIP-GL Reference Game

We use the **Co**llaborative **G**ame of **R**eferential and **I**nteractive language with **P**entomino pieces (CoGRIP) (reference suppressed) and extend it for **G**uidance **L**earning (CoGRIP-GL). A Guide uses natural language to instruct a Follower to select a specific target piece using a gripper. In this setting, both players are constrained as follows: The Guide can provide utterances but cannot move the gripper. The Follower can move the gripper but is not allowed to provide an utterance. This asymmetry in knowledge and skill forces them to work together and coordinate. Zarrieß et al. (2016) found that such a reference game leads to diverse language use on the Guide's side. The most similar environment is from Mordatch and Abbeel (2018) who 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 speaker has to learn how to make use of a restricted vocabulary based on a dense reward signal (the listener's distance to the ground-truth landmark). In our game, we only provide a sparse reward and the communication signals become verbalized into language utterances.

3.1 Problem Formulation

The Guide has to provide utterances that are useful for the Follower to navigate and select the correct target piece. We frame this task as an RL problem with sparse rewards. At each time-step t, given an observation $o_t \in \mathcal{O}$ of the environment, the Guide has to choose an action a_t such that the overall resulting sequence of actions $(a_0, ..., a_t, ..., a_T)$ (which become verbalized into $(l_0, ..., l_t, ..., l_T)$) maximizes the sparse reward $\mathcal{R}(o_T) = r$ that is given on episode end when a piece is selected by the Follower or t reaches $T_{max} = 30$. This maximal number of steps is sufficient to navigate to the target piece with some room for error on our 21×21 tile maps as the Follower is always starting in the center of the map (the farthest tile would be 10 horizontal plus 10 vertical steps away) and allows quicker training.

3.2 Actions

We let the Guide predict "intent" actions and translate them into sentences instead of predicting words directly to reduce the agent's burden on action space exploration (later this verbalization process could be done by a language generation system). Here we focus on the Guide's choice among five intent categories: `silence`, `confirm`, `decline, directive, reference` (Figure 2). For the `directive`s, we allow more fine-grained control over the utterance production, so that the agent has to choose between `left, right, up, down` and `take`. And similarly, for the `reference`s this means that the agent has to choose among possible preference orders `PCS, PSC, SPC, CPS, SCP` and `CSP`, in which `P`, `C` and `S` stand for piece, color, and shape, respectively. These preference orders (PO) define the order in which properties are compared between the target piece and its distractors. This means, for example, that a `CSP`-based reference is likely to mention the target piece's color because the color is tried first to distinguish the target from its distractors (and it is very unlikely that all pieces share the same color). These six `reference` actions, five `directive` actions, `silence`, `confirm` and `decline` lead to a total of $|A| = 14$ actions. In comparison, the vocabulary contains 37 tokens and the maximal sentence length is 12 which results in 37^{12} possible utterances when predicting individual words instead of intents.

3.3 Verbalization

The chosen intent is then verbalized based on templates by application of the following rules:

- `silence` → `<empty string>`
- `confirm` → `Yes this [way|<piece>]`
- `decline` → `Not this [way|<piece>]`
- `directive(take)` → `Take <piece>`
- `directive(dir)` → `Go <dir>`
- `reference(PO)` → `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>` resolve to the according intent name. The fine-grained reference intent (`PO`) is given to the "Incremental Algorithm" (Dale and Reiter 1995), which produces the referring expression for reference verbalization (Appendix A.1).

Rewards Following Chevalier-Boisvert et al. (2019), we define a basic sparse reward for playing the game:

$$\mathcal{R}_{\text{Game}} = 1 - 0.9 * (T/T_{\max}) \tag{1}$$

In addition, we introduce a sparse reward for the Guide's individual effort in an episode:

$$\mathcal{R}_{\text{Guide}} = 1 - 0.9 * (E_{\text{Guide}}/T_{\max}) \tag{2}$$

where the Guide's effort E_{Guide} is the sum over the assumed efforts of taking the respective actions:

$$E_{\text{Guide}} = \sum_{t=1}^{T} \begin{cases} 0, & \text{if } a_t \in \{\texttt{silence}\} \\ 1.0, & \text{if } a_t \in \{\texttt{confirm}, \texttt{decline}\} \\ 1.1, & \text{if } a_t \in \{\texttt{directive}\} \\ 1.2, & \text{if } a_t \in \{\texttt{reference}\} \end{cases} \tag{3}$$

These action-based efforts follow the assumed cognitive load for producing them i.e. saying nothing is the cheapest and comparing pieces with each other to produce a reference is the highest.

	TPS	Tasks	Boards
Training	275	2500	700
Validation	25	175	175
Testing	60	420	420

Table 1: The number of tasks and boards in each data split. The target pieces for the tasks are chosen from non-overlapping sub-sets of target piece symbols (TPS). For evaluation splits, we mix-in training pieces as distractors. We construct boards with up to 7 distractors (and at least 1).

Finally, we give an additional reward ($\mathcal{R}_{\text{Outcome}}$) of $+1$ when the correct piece or a penalty of -1 if the wrong or no piece has been taken at all, so that:

$$\mathcal{R} = (\mathcal{R}_{\text{Game}} + \mathcal{R}_{\text{Guide}})/2 + \mathcal{R}_{\text{Outcome}} \tag{4}$$

Given this formulation, the Guide has to play the game by being active (not just stay silent), achieve the goal (get the bonus) and reduce its individual effort (stay mostly silent) to reach a high reward.

3.4 Observations

The environment exposes at each time-step t an observation o_t that contains the following:

- the Follower's gripper coordinates $g_t = (x, y)$
- the Guide's utterance l_t (might be empty)
- a full view of the scene v_t for the Guide
- a partial view p_t of the scene for the Follower

The visual observations are represented as 2-dimensional images (with RGB color channels), but the Follower only receives a 11×11-sized cut out, centered on the gripper position (see Figure 2). We add a 4th channel to the visual observations to indicate the gripper position by setting the values to zero at g_t and one otherwise. In addition, the Guide is informed about the target piece coordinates by setting the according values to zero for the target piece and ones otherwise on the 5th channel of its visual observation. A piece occupies five adjacent tiles and is not allowed to overlap with another one. For our purposes, the Follower receives a symbolic representation of the partial view (as a neural learner might receive) where color and shapes are directly represented as numbers (and not pixels; see Appendix A.1).

3.5 Tasks

The task is that a Guide provides utterances to a Follower that has to take an intended target piece among several other pieces (the distractors). Thus, a game instance of this task is defined by (i) the number and identity of pieces on the board, (ii) including which of these is the the target piece, (iii) and by the size of the board (see Figure 2 for an example).

The appearance and positioning of the pieces is derived from symbolic piece representations: a tuple of shape (9), color (6), and position (8). We experiment with 360 of these symbolic pieces which include all shapes, colors, and positions and split them into distinct sets. Therefore, the target symbols for the testing tasks are distinct from the ones seen during training (they might share color and shape though, but are for example positioned elsewhere). We ensure the reproducibility of our experiments by constructing 2500 training, 175 validation, and 420 testing tasks representing scenes with a map size of 21×21 tiles (see Table 1 and Appendix A.2 for the detailed generation process).

4 The Follower Behaviors

For the Follower, we take inspiration from Sun et al. (2023) who suggest a plan-based approach towards solving text-based tasks with language models: given a task's natural language instruction the model initially produces a plan, which is then executed and repeatedly refined or revised. We implement a policy that keeps track of a plan that contains up to 10 actions (the plan horizon; which is exactly the number of actions needed to reach the diagonal corner of the partial view). Our Follower's behavior of following the plan is adjustable along two dimensions: confidence and autonomy.

Confidence. The actions in the plan are associated with a decreasing probability of being executed (the "confidence triangle" in Figure 2) so that given a discount factor $\phi \in [0,1]$ and a lower threshold $\mathrm{L} \in [0,1]$ we calculate:

$$\text{Confidence}(a_i) = \max(\phi^i, \mathrm{L}) \tag{5}$$

Which 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 from the Guide.

Autonomy. The revision process for our Follower policy is conceptually divided into five sub-programs that run after the Guide's utterance is received, parsed and the assumed intent type is determined, as follows:

- `on_silence`: The Follower executes, based on confidence, the next action in the plan (if available). Otherwise, it waits.
- `on_confirm`: The Follower sets the confidence for all actions in the current plan to 1. Then the next action is chosen as described under `on_silence`.
- `on_decline`: The Follower erases the current plan. As the plan is then empty, a wait action will be returned.
- `on_directive`: The Follower parses the utterances for the concrete directives (a direction or a "take" prompt). For "take", the plan is replaced with take action under the assumption that this is the last action to be performed. Otherwise, the plan is filled with actions that align with the direction prompt. Then, the next action is chosen as described under `on_silence`.
- `on_reference`: The Follower updates its internal target descriptor (color, shape, position) based on the new reference. Given this updated descriptor, the Follower identifies candidate coordinates in the symbolic representation of the current field of view, for example, coordinates that are blue given a reference "Take the blue piece". If such a coordinate is identified and the Follower has not already approached it, then the shortest path to that candidate is established as a new plan. Otherwise, if the descriptor only contains a position, then a direction towards that position is approached. In the case where the Follower is already in that position, a randomly chosen piece in the field of view is approached. When none of this matches, then the current plan proceeds as described under `on_silence`.

Now, the autonomy defines which procedures the Follower undertakes, when intermediate feedback *is missing* (the Guide stays silent). The **cautious** Follower is performing solely the previously defined procedures: when the plan is exhausted, then it waits until a new directive or reference is given. If this Follower is over an assumed target piece, then it waits until the "take" directive is given by the Guide. In contrast, the **eager** Follower aims to actually take an assumed target piece when approaching it in the current field of view. Furthermore, the eager Follower autonomously looks for target candidates at each step (as described in the `on_reference` procedure) and potentially revises the plan (also when the Guide stays silent).

5 Learning Communication Policies for Different Follower Behaviors

Mnih et al. (2015) showed that vision-driven reinforcement learning policies can achieve human-level performance in pixel-based environments like Atari games. Similarly, the Guide as an agent in our environment has the challenging task to learn:

- when to produce an utterance (or stay silent),
- what to produce (confirm, decline, direct, refer), and
- how to produce it (which directive or preference order)

based solely on visual observation of the board state and the follower actions.

5.1 The Guide

The observation $o_t = (v_t)$ with $v_t \in \mathbb{R}^{21\times21\times5}$ is encoded into a 128-dimensional feature vector $\tilde{v}_t \in \mathbb{R}$ using a 4-layer convolutional neural network similar to that by Chevalier-Boisvert et al. (2019). Then, the feature vector $\tilde{v}_t$ is fed through an LSTM (Hochreiter and Schmidhuber 1997) which functions as a memory mechanism (updating a state vector h_t that is passed forward in time). Given the resulting memory-conditioned visual feature vector $\tilde{x}_t$, we learn a parameterized actor-critic-based policy $\pi(\tilde{x}_t;\theta) \sim a_t$ where

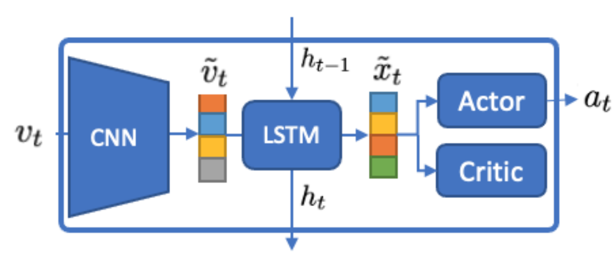

Figure 3: The Guide's recurrent vision network.

the actor predicts a distribution over the action space (intents) and the critic estimates the value of the current state (Figure 3). For the recurrent policy, we use the implementation of *StableBaselines3-Contrib* v1.8.0 (Raffin et al. 2021), which performs back-propagation through time until the first step in an episode.

5.2 Experiment Setup

In this work, we evaluate if and to which extent the PPO algorithm (Schulman et al. 2017) is able to produce guiding neural agents in a challenging reference game where the learning signal is a sparse reward that also involves the assumed accumulated effort over actions. In particular, we are interested in the question of whether the resulting learned policies (the Guides) are adapted towards the Follower behaviors in such ways that align with expectations based on the Follower's dimensions of confidence and autonomy. Thus, for the experiments, we initiate **cautious** and **eager** Follower's with increasing confidence discount factors so that $\phi \in [0.75, 0.85, 0.90, 0.95, 0.97, 0.99]$.

We use *StableBaselines3* v1.8.0 (Raffin et al. 2021) to train for each of these Follower behaviors a policy. We train each policy with 4 parallel running environments (batch size) and 1 million time steps in total. This means that each board in the training split is seen at least 13 times (and even more often when mean $T < 30$). Every 100k steps during training, we evaluate the policies against the validation set. And we keep the policies (the Guides) that achieve the highest mean episode reward based on these validation runs. We conduct the experiments with three different seeds.

5.3 Results and Discussion

Overall Results. The overall results in Table 2 show that learned policies are communicative strategies that can successfully guide the Follower (towards the target piece) in most of the cases (on average in 92% of the test episodes). This indicates that the Guide learned the goal of the game and hereby almost reaches the best episode length (on average only 1.93 steps longer than the shortest path). The overall average effort (9.72) covers only about 71.5% of the average episode length (13.58) which means that the policies altogether produce an utterance in about 2 out of 3 steps.

Has the Guide learned to stay silent? Indeed, Figure 4 shows that the policies converge to a mode where the `silence` intent is chosen in at least 23% of the steps: The policies are in general able to learn to say nothing. The most chosen intent is `reference`, which is reasonable as it is directly providing additional information (the target piece description) to the Follower and triggers a plan revision.

What preference orders are chosen for the `reference` production? The `reference` intents define the order in which properties are compared between the target piece and its distractors. This means, for example, that a CSP reference is likely to mention the target piece's color because the color attribute is first compared to distinguish the target from its distractors (and it is very likely that at least one distractor gets excluded, because otherwise all pieces would share the

Metrics:	mR ↑	mSR ↑	mEPL ↓	mEff. ↓
— Cautious —				
100% Silent	0.00	0.00	30.00	0.00
100% Ref.	-1.04	0.00	30.00	34.8
PPO-Guide	1.55	0.94	13.97	10.72
ϕ=75	1.52	0.93	15.02	11.07
ϕ=85	1.47	**0.96**	14.13	14.63
ϕ=90	1.59	0.95	13.87	10.33
ϕ=95	1.57	0.94	13.67	10.49
ϕ=97	1.57	0.93	13.27	10.00
ϕ=99	1.57	0.90	13.88	7.78
— Eager —				
100% Silent	0.45	0.23	16.78	0.00
100% Ref.	0.86	0.75	18.57	21.09
PPO-Guide	1.57	0.91	13.19	8.72
ϕ=75	1.54	0.92	13.54	10.04
ϕ=85	**1.60**	0.89	14.28	**6.15**
ϕ=90	1.49	0.92	13.24	11.67
ϕ=95	1.59	0.92	12.86	8.39
ϕ=97	1.58	0.90	12.64	7.28
ϕ=99	1.59	0.93	**12.58**	8.76
— Overall —				
100% Silent	0.23	0.11	23.39	0.00
100% Ref.	-0.09	0.37	24.29	27.94
PPO-Guide	1.56	0.92	13.58	9.72

Table 2: The mean rewards (mR), success rates (mSR in %), episodes lengths (mEPL) and efforts of the agents on the test tasks for the chosen autonomy and confidence combinations of the Follower (averaged over all seeds). A shortest path solver reaches 11.65 mEPL (3.13 std). Given this, the upper bound for the mean reward is 1.83. Best values in bold.

Chosen Intent:	S	C	D	O	R
— Cautious —					
PPO-Guide	0.27	0.04	/	0.09	0.60
ϕ=75	0.27	0.08	/	0.08	0.56
ϕ=85	0.06	0.08	/	0.09	0.78
ϕ=90	0.29	0.09	/	0.08	0.53
ϕ=95	0.28	/	/	0.09	0.63
ϕ=97	0.30	/	/	0.09	0.61
ϕ=99	0.43	/	/	0.09	0.48
— Eager —					
PPO-Guide	0.34	0.06	0.06	0.09	0.46
ϕ=75	0.25	0.26	0.03	0.08	0.38
ϕ=85	0.53	0.01	0.09	0.08	0.29
ϕ=90	0.16	0.05	0.11	0.08	0.59
ϕ=95	0.34	/	0.13	0.09	0.45
ϕ=97	0.42	/	/	0.11	0.47
ϕ=99	0.33	0.02	/	0.08	0.57
— Overall —					
PPO-Guide	0.31	0.05	0.03	0.09	0.53

Table 3: The intent's mean chance of being chosen at a step (for each policy evaluated on the test split) broken down by a Follower's confidence and autonomy. The intents are abbreviated as follows: `silence` (S), `confirm` (C), `decline` (D), `directive` (O) and `reference` (R). It appears reasonable that the cautious Follower's actions are never declined because the behavior is to always wait for the Guide's instructions (in contrast to the eager ones that explore occasionally on their own). Similarly, the higher confidence Follower's require less re-assurance (confirms) of their actions.

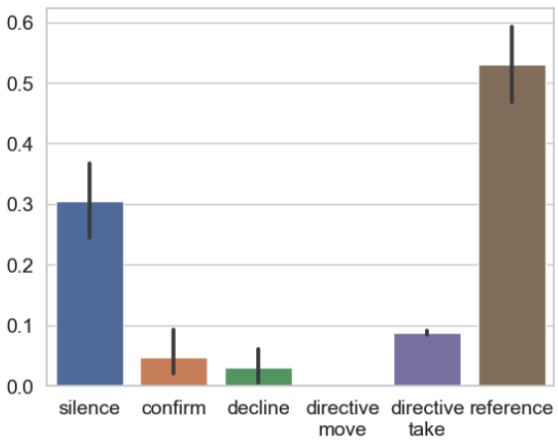

Figure 4: An intent's mean chance of being chosen at a step (for all learnt policies evaluated on the test split).

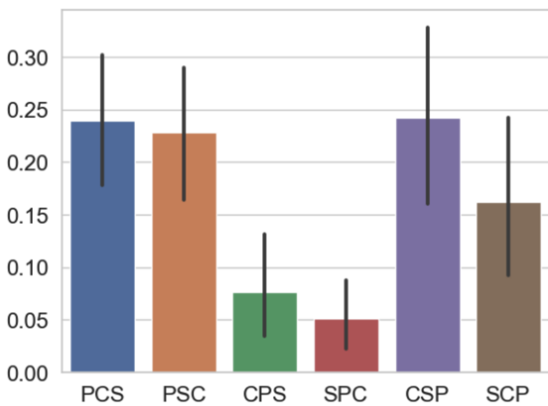

Figure 5: The distribution of the preference order choices for the `reference` action (from Figure 4). The preferences over position (P), shape (S) and color (C) are given to the IA for reference production.

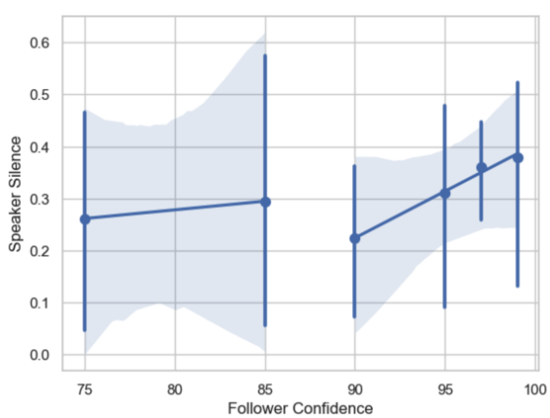

Figure 6: The linear regressions with a confidence interval of 99% for the mean silence rates measured during the test episodes for all learnt policies (3 seeds per follower). Fitted separately for the confidences $\{75, 85\}$ and $\{90, 95, 97, 99\}$.

same color). Thus, it is reasonable that there are communicative strategies learnt that choose CSP in the majority of cases. This means that the guide produces a reference that likely includes the shape and the color of the target piece. These properties are indeed useful for the follower to identify and approach the target in its field of view. An the other hand, preference orders that test positions first (PCS and PSC) are also chosen rather often. These strategies lead the Follower to the target piece without having it necessarily already in the field of view.

The effects of the Follower's autonomy mode. We experimented with two levels of autonomy of the Follower. The results in Table 2 show that the policies that learn from interactions with the **eager** Follower require on average 2.00 points less effort than the **cautious** one. This is reasonable as the eager Follower is autonomously updating the plan and looking for target candidates at each step. Along these lines, it is also reasonable that the decline intent is never selected for the cautious Follower (see Table 3) because it never tried to approach a target piece without the Guide referencing it.

The effects of the Follower's confidence. The differences in the intent selection strategy of the learned policies (Guides) shown in Table 3 indicate that Guides learnt from interaction with more confident Follower's ($\phi > 0.9$) produce less or no `confirm` actions. This seems reasonable as the decrease in the execution probability of these Followers is less steep and a `reference` action has a similar effect. Furthermore, we see a slight tendency of Guide's to stay quieter (on average) when trained with more confident Followers as shown in Figure 6. Though we cannot see such a tendency for Guide's trained with less confident Followers.

6 Conclusions

In this work, we examined an interesting intersection between psycho-linguistic studies and deep learning with reinforcement learning. We considered neural agents as possible interaction partners (for humans) in a challenging reference game where a Guide has to learn when, what, and how information (actionable intents) is to be provided to a Follower. As a proxy for different Follower behaviors, we implemented a hand-crafted policy that is controllable along two dimensions: autonomy in exploration and confidence in executing an action. We experimented with a learning signal that in addition to the goal condition also respects an assumed communicative effort. Our results indicate that this formulation of the learning signal leads to communicative strategies that are less verbose (stay silent more often) and that the resulting Guide behaviors are adapted (in terms of intent selection distributions) to the Follower's autonomy and confidence levels. We think this work presents a useful case study towards neural agents that have to learn adapted communication strategies in an interactive setting (possibly with humans). In future work, we want to investigate other reward formulations for our proven reference game and evaluate the learning of communication policies in an even more incremental setting where the utterance production process spans multiple time steps (one word at a time) and must be possibly interrupted and re-adjusted during the interaction.

Acknowledgements

We want to thank the anonymous reviewers for their comments. This work was funded by the *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation) – 423217434 (“RECOLAGE”) grant.

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 Environment Details

Board The internal representation of the visual state is a 2-dimensional grid that spans $W \times H$ tiles where W and H are defined by the map size. A tile is either empty or holds an identifier for a piece (the tile is then occupied). The pieces are defined by their colour, shape and coordinates and occupy five adjacent tiles (within a virtual box of 5×5 tiles). The pieces are not allowed to overlap with another piece’s tiles. For a higher visual variation, we also apply rotations to pieces, but we ignore the rotation for expression generation, though this could be an extension of the task. The colors are described in Table 4.

Name	HEX	RGB
red	#ff0000	(255, 0, 0)
green	#008000	(0, 128, 0)
blue	#0000ff	(0, 0, 255)
yellow	#ffff00	(255, 255, 0)
brown	#8b4513	(139, 69, 19)
purple	#800080	(128, 0, 128)

Table 4: The colors for the Pentomino pieces.

Symbols The symbolic repesentations for the shapes are: P (2), X (3), T (4), Z (5), W (6), U (7), N (8), F (9), Y (10). 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). The 1-symbol is reserved for an empty tile.

Gripper The gripper can only move one position at a step and can move over pieces, but is not allowed to leave the boundaries of the board. The gripper coordinates are defined as $\{(x, y) : x \in [0, W], y \in [0, H]\}$.

References The Incremental Algorithm (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.

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
  end if
end for
  return $\mathcal{D}$

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.

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]*

Vocabulary Overall, the property values and sentence templates lead to a small vocabulary of 37 words:

- 9 shapes: P, X, T, Z, W, U, N, F, Y
- 6 colors: red, green, blue, yellow, brown, purple
- 6 position words: left, right, top, bottom, center (which are combined to e.g., right center or top left)
- 12 template words: take, the, piece, at, yes, no, this, way, go, a, bit, more
- 4 special words: <s>, <e>, <pad>, <unk>

The maximal sentence length is 12.

A.2 Task Details

To create a task, we first place the target piece on a board. Then, we sample uniformly random from all possible pieces and place them until the wanted number of pieces is reached (we experiment with 2 to 8 pieces on a board). If a piece cannot be placed after a certain amount of tries, then we re-sample a piece and try again. The coordinates are chosen at random uniform from the coordinates that fall into an area of the symbolic description. We never set a piece into the center, because that is the location where the gripper is initially located. In this way, we construct 100 training boards (or 1 evaluation board respectively) for each number of pieces (2-8). To ensure that a board scene in the training split cannot be aligned with a target piece, we create 3 extra tasks for a single board by choosing extra targets (when fewer than 4 pieces are on a board, then we create a task for each piece). For evaluation, we only create a single task for each target piece symbol.

A.3 Guide Details

Agent Parameters: $602,447$

feature_dims	128
normalize_images	True
shared_lstm	True
enable_critic_lstm	False
n_lstm_layers	1
lstm_hidden_size	128

Table 5: Policy arguments for the the RecurrentPPO agent

Policy Architecture We instantiate the actor-critic PPO agent with an architecture defined by `pi=[64, 64]`, `vf=[64, 64]` meaning that the actor is a 2-layer feedforward network with 64 parameters per layer. The critic has the same architecture, but does not share the weights with the actor.

Vision Encoder The visual encoder is a convolutional neural network (CNN) with 4 layers that maps the visual observations $v_t \in \mathbb{R}^{21\times21\times5}$ into a 128-dimensional features vector $\tilde{v} \in \mathbb{R}$. We consecutively apply four blocks of `(nn.Conv2d(),nn.BatchNorm2d(),nn.ReLU())` with same padding where the kernel size is 3×3, except for the first blocke where we set the kernel size to 1×1. After the fourth block we apply a `nn.AdaptiveMaxPool2d((1, 1))` layer from PyTorch v1.13.0 (Paszke et al. 2019) to collapse the spatial dimensions of the feature maps.

Learning Algorithm We use the RecurrentPPO implementation from StableBaselines-Contrib v1.8.0 (Raffin et al. 2021) with the hyper-parameters in Table 6 (and the defaults otherwise).

A.4 Experiment Details

We trained the agents simultaneously on 8 GeForce GTX 1080 Ti (11GB) where each of them consumed about 4GB of GPU memory. The training for the 36 configurations took around 144 hours in total (about $4h$ for the 1 million steps

learning_rate	3e-4
clip_range	0.2
gamma	0.99
gae_lambda	0.95
ent_coef	0.0
vf_coef	0.5
max_grad_norm	0.5
lr_init	3e-4
n_steps	128
batch_size	128
num_epochs	10

Table 6: RecurrentPPO hyperparameters

each). The random seeds were set to 49184, 98506 or 92999 respectively. As the evaluation criteria on the testings tasks we chose success rate which indicates the relative number of episodes (in a rollout or in a test split) where the agent selected the correct piece:

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

References

Albrecht, S. V.; and Stone, P. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artif. Intell.*, 258: 66–95.

Chane-Sane, E.; Schmid, C.; and Laptev, I. 2021. Goal-Conditioned Reinforcement Learning with Imagined Subgoals. In Meila, M.; and Zhang, T., eds., *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*, 1430–1440. PMLR.

Chang, C.; Mu, N.; Wu, J.; Pan, L.; and Xu, H. 2022. E-MAPP: Efficient Multi-Agent Reinforcement Learning with Parallel Program Guidance. In *NeurIPS*.

Chevalier-Boisvert, M.; Bahdanau, D.; Lahlou, S.; Willems, L.; Saharia, C.; Nguyen, T. H.; and Bengio, Y. 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.

Clark, H. H. 1996. *Using Language*. 'Using' Linguistic Books. Cambridge: Cambridge University Press. ISBN 978-0-521-56158-7.

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

Co-Reyes, J. D.; Gupta, A.; Sanjeev, S.; Altieri, N.; Andreas, J.; DeNero, J.; Abbeel, P.; and Levine, S. 2019. Guiding Policies with Language via Meta-Learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Dale, R.; and Reiter, E. 1995. Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. *Cogn. Sci.*, 19(2): 233–263.

Davidesco, I.; Laurent, E.; Valk, H.; West, T.; Milne, C.; Poeppel, D.; and Dikker, S. 2023. The Temporal Dynamics of Brain-to-Brain Synchrony Between Students and Teachers Predict Learning Outcomes. *Psychological Science*, 34(5): 633–643.

E, H.; Niu, P.; Chen, Z.; and Song, M. 2019. A Novel Bidirectional Interrelated Model for Joint Intent Detection and Slot Filling. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 5467–5471. Association for Computational Linguistics.

Fried, D.; Hu, R.; Cirik, V.; Rohrbach, A.; Andreas, J.; Morency, L.; Berg-Kirkpatrick, T.; Saenko, K.; Klein, D.; and Darrell, T. 2018. Speaker-Follower Models for Vision-and-Language Navigation. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 3318–3329.

Gao, X.; Gao, Q.; Gong, R.; Lin, K.; Thattai, G.; and Sukhatme, G. S. 2022. DialFRED: Dialogue-Enabled Agents for Embodied Instruction Following. *IEEE Robotics Autom. Lett.*, 7(4): 10049–10056.

Ghosh, A.; Tschiatschek, S.; Mahdavi, H.; and Singla, A. 2020. Towards Deployment of Robust Cooperative AI Agents: An Algorithmic Framework for Learning Adaptive Policies. In Seghrouchni, A. E. F.; Sukthankar, G.; An, B.; and Yorke-Smith, N., eds., *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020*, 447–455. International Foundation for Autonomous Agents and Multiagent Systems.

Golomb, S. W. 1996. *Polyominoes: Puzzles, Patterns, Problems, and Packings*. Princeton University Press. ISBN 0691024448.

Gu, J.; Stefani, E.; Wu, Q.; Thomason, J.; and Wang, X. 2022. Vision-and-Language Navigation: A Survey of Tasks, Methods, and Future Directions. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 7606–7623. Association for Computational Linguistics.

Gürtler, N.; Büchler, D.; and Martius, G. 2021. Hierarchical Reinforcement Learning with Timed Subgoals. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 21732–21743.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Hu, B.; Zhao, C.; Zhang, P.; Zhou, Z.; Yang, Y.; Xu, Z.; and Liu, B. 2023. Enabling Intelligent Interactions between an Agent and an LLM: A Reinforcement Learning Approach. *CoRR*, abs/2306.03604.

Huang, S.; Lipovetzky, N.; and Cohn, T. 2023. A Reminder of its Brittleness: Language Reward Shaping May Hinder Learning for Instruction Following Agents. *CoRR*, abs/2305.16621.

Jeon, J.; Kim, W.; Jung, W.; and Sung, Y. 2022. MASER: Multi-Agent Reinforcement Learning with Subgoals Generated from Experience Replay Buffer. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 10041–10052. PMLR.

Jurgenson, T.; and Tamar, A. 2023. Goal-Conditioned Supervised Learning with Sub-Goal Prediction. *CoRR*, abs/2305.10171.

Kolb, B.; Lang, L.; Bartsch, H.; Gansekoele, A.; Koopmanschap, R.; Romor, L.; Speck, D.; Mul, M.; and Bruni, E. 2019. Learning to Request Guidance in Emergent Communication. *CoRR*, abs/1912.05525.

Lee, G. T.; and Kim, K. J. 2023. A Controllable Agent by Subgoals in Path Planning Using Goal-Conditioned Reinforcement Learning. *IEEE Access*, 11: 33812–33825.

Liu, Y.; Li, Y.; Xu, X.; Dou, Y.; and Liu, D. 2022. Heterogeneous Skill Learning for Multi-agent Tasks. In *NeurIPS*.

Lowe, R.; Foerster, J. N.; Boureau, Y.; Pineau, J.; and Dauphin, Y. N. 2019. On the Pitfalls of Measuring Emergent Communication. In Elkind, E.; Veloso, M.; Agmon, N.; and Taylor, M. E., eds., *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, 693–701. International Foundation for Autonomous Agents and Multiagent Systems.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M. A.; Fidjeland, A.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nat.*, 518(7540): 529–533.

Moon, S.; Kottur, S.; Crook, P. A.; De, A.; Poddar, S.; Levin, T.; Whitney, D.; Difranco, D.; Beirami, A.; Cho, E.; Subba, R.; and Geramifard, A. 2020. Situated and Interactive Multimodal Conversations. In Scott, D.; Bel, N.; and Zong, C., eds., *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, 1103–1121. International Committee on Computational Linguistics.

Mordatch, I.; and Abbeel, P. 2018. Emergence of Grounded Compositional Language in Multi-Agent Populations. In McIlraith, S. A.; and Weinberger, K. Q., eds., *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*, 1495–1502. AAAI Press.

Mu, J.; Zhong, V.; Raileanu, R.; Jiang, M.; Goodman, N. D.; Rocktäschel, T.; and Grefenstette, E. 2022. Improving Intrinsic Exploration with Language Abstractions. *CoRR*, abs/2202.08938.

Mul, M.; Bouchacourt, D.; and Bruni, E. 2019. Mastering emergent language: learning to guide in simulated navigation. *CoRR*, abs/1908.05135.

Nguyen, K.; Dey, D.; Brockett, C.; and Dolan, B. 2019. Vision-Based Navigation With Language-Based Assistance via Imitation Learning With Indirect Intervention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 12527–12537. Computer Vision Foundation / IEEE.

Nguyen, K.; and III, H. D. 2019. Help, Anna! Visual Navigation with Natural Multimodal Assistance via Retrospective Curiosity-Encouraging Imitation Learning. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *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*, 684–695. Association for Computational Linguistics.

Padmakumar, A.; Thomason, J.; Shrivastava, A.; Lange, P.; Narayan-Chen, A.; Gella, S.; Piramuthu, R.; Tür, G.; and Hakkani-Tür, D. 2022. TEACh: Task-Driven Embodied Agents That Chat. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 2017–2025. AAAI Press.

Park, J. S.; O'Brien, J. C.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative Agents: Interactive Simulacra of Human Behavior. *CoRR*, abs/2304.03442.

Pashevich, A.; Schmid, C.; and Sun, C. 2021. Episodic Transformer for Vision-and-Language Navigation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 15922–15932. IEEE.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E. Z.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 8024–8035.

Pertsch, K.; Rybkin, O.; Ebert, F.; Zhou, S.; Jayaraman, D.; Finn, C.; and Levine, S. 2020. Long-Horizon Visual Planning with Goal-Conditioned Hierarchical Predictors. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; and Dormann, N. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268): 1–8.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347.

Suhr, A.; and Artzi, Y. 2022. Continual Learning for Instruction Following from Realtime Feedback. *CoRR*, abs/2212.09710.

Sun, H.; Zhuang, Y.; Kong, L.; Dai, B.; and Zhang, C. 2023. AdaPlanner: Adaptive Planning from Feedback with Language Models. *CoRR*, abs/2305.16653.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. The MIT Press, second edition.

Thomason, J.; Murray, M.; Cakmak, M.; and Zettlemoyer, L. 2019. Vision-and-Dialog Navigation. *CoRR*, abs/1907.04957.

van Deemter, K. 2016. *Computational Models of Referring*, chapter 4.6. The MIT Press. ISBN 9780262034555.

Wang, W. Z.; Shih, A.; Xie, A.; and Sadigh, D. 2021. Influencing Towards Stable Multi-Agent Interactions. In Faust, A.; Hsu, D.; and Neumann, G., eds., *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, 1132–1143. PMLR.

Xie, A.; Losey, D. P.; Tolsma, R.; Finn, C.; and Sadigh, D. 2020. Learning Latent Representations to Influence Multi-Agent Interaction. In Kober, J.; Ramos, F.; and Tomlin, C. J., eds., *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, volume 155 of *Proceedings of Machine Learning Research*, 575–588. PMLR.

Yang, M.; Zhao, J.; Hu, X.; Zhou, W.; Zhu, J.; and Li, H. 2022. LDSA: Learning Dynamic Subtask Assignment in Cooperative Multi-Agent Reinforcement Learning. In *NeurIPS*.

Zarrieß, S.; Hough, J.; Kennington, C.; Manuvinakurike, R.; DeVault, D.; Fernández, R.; and Schlangen, D. 2016. PentoRef: A Corpus of Spoken References in Task-oriented Dialogues. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 125–131. Portorož, Slovenia: European Language Resources Association (ELRA).