SocialAI: Benchmarking Socio-Cognitive Abilities in Deep Reinforcement Learning Agents

Grgur Kovač, Rémy Portelas, Katja Hofmann, Pierre-Yves Oudeyer

SocialAI: Benchmarking Socio-Cognitive Abilities in Deep Reinforcement Learning Agents

Grgur Kovač[*†]
Inria (FR)

Rémy Portelas[*†]
Inria (FR)

Katja Hofmann
Microsoft Research (UK)

Pierre-Yves Oudeyer
Inria (FR)

Abstract

Building embodied autonomous agents capable of participating in social interactions with humans is one of the main challenges in AI. Within the Deep Reinforcement Learning (DRL) field, this objective motivated multiple works on embodied language use. However, current approaches focus on language as a communication tool in very simplified and non-diverse social situations: the "naturalness" of language is reduced to the concept of high vocabulary size and variability. In this paper, we argue that aiming towards human-level AI requires a broader set of key social skills: 1) language use in complex and variable social contexts; 2) beyond language, complex embodied communication in multimodal settings within constantly evolving social worlds. We explain how concepts from cognitive sciences could help AI to draw a roadmap towards human-like intelligence, with a focus on its social dimensions. As a first step, we propose to expand current research to a broader set of core social skills. To do this, we present *SocialAI*, a benchmark to assess the acquisition of social skills of DRL agents using multiple grid-world environments featuring other (scripted) social agents. We then study the limits of a recent SOTA DRL approach when tested on *SocialAI* and discuss important next steps towards proficient social agents. Videos and code are available at `https://sites.google.com/view/socialai`.

1 Introduction

How do human children manage to reach the social and cognitive complexity of human adults? For Vygotsky, a soviet scholar from the 1920s, a main driver for this path towards "higher-level" cognition are socio-cultural interactions with other human beings [Vygotsky and Cole, 1978]. For him, many high-level cognitive functions a child develops first appear at the social level and then develop at the individual level. This leap from interpersonal processes to intrapersonal processes is referred to as *internalization*. Vygotsky's theories influenced multiple works within cognitive science [Clark, 1996, Hutchins, 1996], primatology [Tomasello, 1999] and the developmental robotics branch of AI [Billard and Dautenhahn, 1998, Brooks et al., 2002, Cangelosi et al., 2010, Mirolli and Parisi, 2011].

Another influential perspective on child development are Jean Piaget's foundational theories of cognitive development [Piaget, 1963]. For Piaget, the child is a solitary thinker. While he acknowledged that social context can assist development, for him cognitive maturation happens mainly through the child's solitary exploration of their world. The child is a "little scientist" deciding which experiments to perform to challenge its assumptions and improve its representation of the world.

This Piagetian view on development is well aligned with mainstream Deep Reinforcement Learning research, which mainly focuses on sensorimotor development, through navigation and object

---

[*]Equal contribution
[†]Email grgur.kovac@inria.fr & remy.portelas@inria.fr

manipulation problems rather than language based social interactions [Mnih et al., 2015, Lillicrap et al., 2016, Andrychowicz et al., 2017]. On the other hand, the study of language has been mostly separated from DRL, into the field of Natural Language Processing (NLP), which is mainly focused on learning (disembodied) language models for text comprehension and/or generation (e.g. using large text corpora as in Brown et al. [2020]).

In the last few years however, recent advances in both DRL and NLP made the Machine Learning community reconsider experiments with language based interactions [Luketina et al., 2019, Bender and Koller, 2020]. Text-based exploratory games have been leveraged to study the capacities of autonomous agents to properly navigate through language in abstract worlds [Côté et al., 2018, Prabhumoye et al., 2020, Ammanabrolu et al., 2020]. While these environments allow meaningful abstractions, they neglect the importance of embodiment for language learning, which has long been identified as an essential component for proper language understanding and grounding [Cangelosi et al., 2010, Bisk et al., 2020]. Following this view, many works attempted to use DRL to train embodied agents to leverage language, often in the form of language-guided RL agents [Chevalier-Boisvert et al., 2018a, Colas et al., 2020a, Hill et al., 2020a, Akakzia et al., 2021] and Embodied visual Question Answering (EQA) [Das et al., 2017, Gordon et al., 2018], and more recently on interactive question production and answering [Abramson et al., 2020]. Language use has also been studied in Multi-agent emergent communication settings, both in embodied and disembodied scenarios [Mordatch and Abbeel, 2018, Jaques et al., 2019, Lowe et al., 2020, Woodward et al., 2020].

One criticism that could be made over aforementioned works in light of Vygotsky's theory is the simplicity of the "social interactions" and language-use situations that are considered: in language-conditioned works, the interaction is merely just the agent receiving its goal as natural language within a simple and rigid interaction protocol [Luketina et al., 2019]. In EQA, language-conditioned agents only need to first navigate and then produce simple one or two words answers. And because of the complexity of multi-agent training, studies on emergent communication mostly consider simplistic language (e.g. communication bits) and tasks.

To catalyze research on building proficient social agents, we propose to identify a richer set of socio-cognitive skills than those currently considered in most of the DRL and NLP literature. We organize this set along 3 dimensions. Proficient social agents must be able to master *intertwined multimodality*, i.e. coordinating multimodal actions based on multimodal observations. They should also be able to build an (explicit or implicit) *theory of mind*, i.e. inferring other's mental state, e.g. beliefs, desires, emotions, etc. Lastly, they should be able to learn diverse and complex *pragmatic frames*, i.e. social interaction protocols described as "verbal or non-verbal patterns of goal-oriented behaviors that evolve over repeated interactions between learners and teachers" Bruner [1985].

Based on these target socio-cognitive skills, we present *SocialAI*, a set of grid-world environments as a first step to foster research in this direction (see fig. 1). To study complex social scenarios in reasonable computational time, we consider single-agent learning among scripted agents (a.k.a. Non-Player-Characters or NPCs) and use low-dimensional observation and action spaces. We also use templated language, enabling to emphasize the under-studied challenges of dealing with more complex and diverse social and pragmatic situations. To showcase the relevance of *SocialAI*, we study the failure case of a current SOTA DRL approach on this benchmark through detailed case studies.

Social agents are not objects. Although social peers could be seen as merely complex interactive objects, we argue they are in essence quite different. Social agents (e.g. humans) can have very complex and changing internal states, including intents, moods, knowledge states, preferences, emotions, etc. The resulting set of possible interactions with peers (social affordances) is essentially different than those with objects (classical affordances). In cognitive science, an affordance refers to what things or events in the environment afford to an organism [de Carvalho, 2020]. A flat surface can afford "walking-on" to an agent, while a peer can afford "obtaining directions from". The latter is a social affordance, which may require a social system and conventions (e.g. politeness), implying that social peers have complex internal states and the ability to reciprocate. Successful interaction might also be conditioned on the peer's mood, requiring communication adjustments.

Training an agent for such social interactions most likely requires drastically different methods – e.g. different architectural biases – than classical object-manipulation training. In *SocialAI* we simulate such social peers using scripted NPCs. We argue that studying isolated social scenarios featuring NPCs in tractable environments is a promising first step towards designing proficient social agents.

Grounding language in social interactions. In AI, *natural language* often refers to the ability of an agent to use a large vocabulary and complex grammar. We argue that this is but one dimension of the *naturalness* of language. Another, often overlooked, dimension of this *naturalness* refers to language grounding, i.e. the ability to anchor the meaning of language in physical, pragmatic and social situations [Steels, 2007]. The large literature on language grounding has so far mostly focused on grounding language into physical action [Cangelosi et al., 2010, Chevalier-Boisvert et al., 2018a, Colas et al., 2020a]: here the meanings of sentences refer to actions to be made in interaction with objects (e.g. "Grasp the blue box"). However, natural language as used by humans is also strongly grounded in social contexts: not only the interpretation of language requires understanding the social context (e.g. taking into account intents or beliefs of others), but meanings can refer to social actions (e.g. "Help your friend to learn his dance lesson"). Here, an important aspect of language naturalness refers to the diversity of kinds of pragmatic social situations in which it is grounded: the work presented here aims at making steps in this direction.

Main contributions:

- An outline of a set of core socio-cognitive skills necessary to enable artificial agents to efficiently act and learn in a social world.
- *SocialAI*, a set of grid-world environments including complex social situations with scripted NPCs to benchmark the capacity of DRL agents to learn socio-cognitive skills organized across several dimensions.
- Performance assessment of a SOTA Deep RL approach on *SocialAI* and analysis of its failure using multiple case studies.

2 Related Work

An extended version of the following related work is available in appendix A.

Earlier calls for socially proficient agents This work aims to connect the recent DRL & NLP literature to the older developmental robotics field [Asada et al., 2009, Cangelosi and Schlesinger, 2014], which studies how to leverage knowledge from the cognitive development of human babies into embodied robots. Within this field, multiple calls for developing the social intelligence of autonomous agents have already been formulated [Billard and Dautenhahn, 1999, Lindblom and Ziemke, 2003, Mirolli and Parisi, 2011]. Vygotsky's emphasis on the importance of social interactions for learning is probably what led Bruner to conceptualize the notion of pragmatic frames [Bruner, 1985], which has later been reused to theorize language development [Rohlfing et al., 2016]. In this work we intent to further motivate the relevance of this notion to catalyse progress in Deep RL and AI.

Human-Robot Interaction Interactions with knowledgeable human teachers is a well studied form of social interaction. Many works within the Human-Robot Interaction (HRI) and the Interactive Learning field studied how to provide interactive teaching signals to their agents, e.g. providing instructions [Grizou et al., 2014], demonstrations [Argall et al., 2009], or corrective advice [Celemin and Ruiz-del Solar, 2015]. In [Vollmer et al., 2016], authors review this field and acknowledge a lack of diversity and complexity in the set of studied pragmatic frames. Echoing this observation, the *SocialAI* benchmark invites to study a broader set of social situations, e.g. requiring agents to both move and speak, and even to learn to interact in a *diversity* of pragmatic frames. Catalysing research on DRL and social skills seems even more relevant now that many application-oriented works are beginning to leverage RL and DRL into real world humanoid social robots [Akalin and Loutfi, 2021].

Recent works on language grounded DRL Building on NLP, developmental robotics, and goal-conditioned DRL [Colas et al., 2020b], many recent works presented embodied DRL-based agents able to process language [Luketina et al., 2019]. Most of this research concerns language conditioned agents in *instruction following* scenarios (e.g. "go to the red bed"), and present various solutions to speed-up learning, such as auxiliary tasks Hermann et al. [2017], pre-trained language models [Hill et al., 2020b], demonstrations [Fu et al., 2019, Lynch and Sermanet, 2020], or descriptive feedbacks [Colas et al., 2020a, Nguyen et al., 2021]. In Embodied Visual Question Answering works, agents are conditioned on questions, requiring them to navigate within environments and then produce an answer ("what color is the bed ?") [Das et al., 2017, Gordon et al., 2018]. Compared to these works, *SocialAI* aims to enlarge the set of considered scenarios by studying how to ground and produce language within diverse forms of social interactions among embodied social peers.

Closely related to our work, Abramson et al. [2020] present experiments on a simulated 3D environment designed for multi-agent interactive scenarios between embodied multi-modal agents and human players. Authors propose novel ways to leverage human demonstrations to bootstrap learning. Because of the complexity of their setup (imitation learning, 3D environments, pixel-based, human in the loop, ...), only two now-common social interaction scenarios are studied: question answering (i.e. EQA) and instruction following. The novelty of their work is that these questions/instructions are alternatively produced and tackled by learning agents. *SocialAI* focuses on a lighter experimental pipeline (2D grid-world, symbolic pixels, no humans) to enable the study of a broader range of social scenarios, requiring multi-steps conversations and interactions with multiple (scripted) agents.

Benchmarks on embodied agents and language   Benchmarks featuring language and agents embodied in physical worlds already exists, however many of them only consider the aforementioned instruction-following [Chevalier-Boisvert et al., 2018a, Misra et al., 2018, Ruis et al., 2020] and question-answering [Das et al., 2017, Gordon et al., 2018] scenarios. In between disembodied NLP testbeds [Wang et al., 2018, Zadeh et al., 2019] and previous embodied benchmarks is the LIGHT environment [Urbanek et al., 2019], a multiplayer text adventure game allowing to study social settings requiring complex dialogue production [Ammanabrolu et al., 2020, Prabhumoye et al., 2020]. Instead of the virtual embodiment of text-worlds, *SocialAI* tackles the arguably harder and richer setting of egocentric embodiment among embodied social peers. Within the Multi-Agent RL field, Mordatch and Abbeel [2018] propose embodied environments to study the emergence of grounded compositional language. Here language is merely a discrete set of abstract symbols used one at a time (per step). While symbol negociation is an interesting social situation to study, we leave it to future work and consider scenarios in which agents must enter an already existing social world (using non-trivial language). In Jaques et al. [2019], authors present Multi-Agent social dilemna environments requiring the emergence of cooperative behaviors through *non-verbal* communication. We consider both non-verbal (e.g. gaze following) and language-based communication.

3   Social skills for socially competent agents

Social skills have been extensively studied in cognitive science [Riggio, 1986, Beauchamp and Anderson, 2010] and developmental robotics [Cangelosi et al., 2010]. Based on these works, this section identifies a set of core social skills when aiming to train socially competent artificial agents.

1 - Intertwinded multimodality   refers to the ability to interact and use multiple modalities (verbal and non-verbal) in a coordinated manner. A proficient agent should be able to act using both primitive actions (moving) and language actions (speaking), and to process both visual and language observations of social peers. Importantly, socially competent agents must be able to interact using multiple modalities in an *intertwined* fashion. By intertwined multimodality we refer to the ability to adapt its multimodal interaction sequence, rather than following a pre-established progression of modalities. For example, in EQA[Das et al., 2017], the progression is always as follows: 1) a question is given to the agent at the beginning of the episode, 2) the agent moves through the environment to gather information, and 3) upon finding an answer it responds (in language) and the episode ends. By the term *intertwined multimodality* we aim to emphasize that the modalities often interchange and the question of "when to use which modality" is non-trivial, e.g. sometimes the relevant information can be obtained by *asking* for it and sometimes by *looking* for it.

2 - Theory of Mind(ToM)   refers to the ability of an agent to attribute to others and itself mental states, including beliefs, intents, desires, emotions and knowledge [Wellman, 1992, Flavell, 1999]. An agent that has ToM perceives other participants as *minds* like itself. This enables the agent to theorise about other's intents, knowledge, lack of knowledge etc. Here we outline some, of many, different perspectives of ToM to better demonstrate how ToM is essential for human social interactions:

Inferring intents: the agent is able to infer, based on verbal or non-verbal cues, what others will do or want to do, e.g. that some social peers are liars/trustworthy.

False belief: the agent understands that someone's belief (including its own) can be faulty [Baillargeon et al., 2010].

Imitating or emulating social peer's behaviour: agent can imitate a behaviour or a goal seen in a social peer, e.g. upon observing a peer cut onions the agent is able to cut the onions himself, either with the same movement or with its own strategy.

3 - Pragmatic frames refer to the regular patterns characterizing the unfolding of possible social interactions (equivalent to an interaction protocol or a grammar of social interactions). Pragmatic frames simplify learning by providing a stable structure to social interactions. An example of a pragmatic frame are the turn taking games. By playing those games a child extracts the rule of each participant having his "turn". It can then generalize this rule to a conversation where it understands that it shouldn't speak while someone else is speaking. We propose to outline several facets of pragmatic frames that proficient social agents should be able to master:

Learning a pragmatic frame The agent is able to learn a frame through social interactions, without it being manually hand-coded (e.g. as in instruction-following scenarios). Through rich social interactions (e.g. dialogues) with one or several peers, the agent should be able to infer the structure of the interaction pattern (frame), extract potential instructions, and leverage them appropriately.

Teaching frames are a specific type of pragmatic frames involving a teacher explicitly teaching a certain content via a *slot*. A slot refers to the place in the interaction sequence holding the variable learning content. A parent teaching a child words with the help of a picture book is one such teaching frame. Upon seeing a picture with a dog a parent might point to the dog, say "Look, it's a dog", and establish eye contact to verify that the child understood the message. Upon, however, seeing a picture of a cat he might say "Look, it's a cat". Here "dog" and "cat" are learning contents and the slot is the location of those words in the sequence ("Look, it's a *<slot>*"). A socially competent agent should be able to learn such a frame and extract the *learning content* from it.

Roles - An agent is able to not only understand the relevance of various participants for achieving a shared goal but also learn about the others' role just from playing its own. For example, in a setting where one agent opens the door to enable another agent to exit the room, the *exiting* agent should be able to learn what the role of opening the door consists of. This exiting agent should then be able to open the door for another agent with little or no additional training. The social interaction described above consists of one frame viewed from two different perspectives corresponding to two different roles. Socially proficient agents should be able to easily learn this whole frame by experiencing it just from their own perspective.

Diversity - the agent can learn many different frames and differentiate between them. Furthermore, the agent can reuse those frames in new situations and even negotiate and construct new ones.

Frame changes - an agent is able to detect and adjust to a change of the current pragmatic frame. For example, while playing football we are able to participate in *small talk* with another player.

4   The SocialAI 1.0 Benchmark

To catalyse research on developing socially proficient autonomous agents, we present SocialAI 1.0 (see fig. 1), a set of grid-world environments designed to challenge learners on important aspects of the core social skills mentioned in sec. 3. In this section we briefly present each environment (see app. B for details) and highlight how they require various subsets of the aforementioned core social skills.

Common components The key design principle of *SocialAI* environments is to allow the study of complex social situations in reasonable computational time. As such, we consider single-room grid-world environments (8x8 grid), based on minigrid [Chevalier-Boisvert et al., 2018b](Apache License). The learning agent can both navigate using discrete actions (e.g. turn left/right, go forward, toggle) and use template-based language generation (environment-dependent). As observations, the agent receives a partial 7x7 agent-centric symbolic pixel grid (see highlighted cells in fig. 1), with 4 dimensions per cell (type, color, status, orientation). Additionally, the agent receives the history of observed language outputs from NPCs preceded by the NPC's name (ex. "John: go to the red door). A positive reward is given only upon successful completion of the social scenario (discounted by time taken).
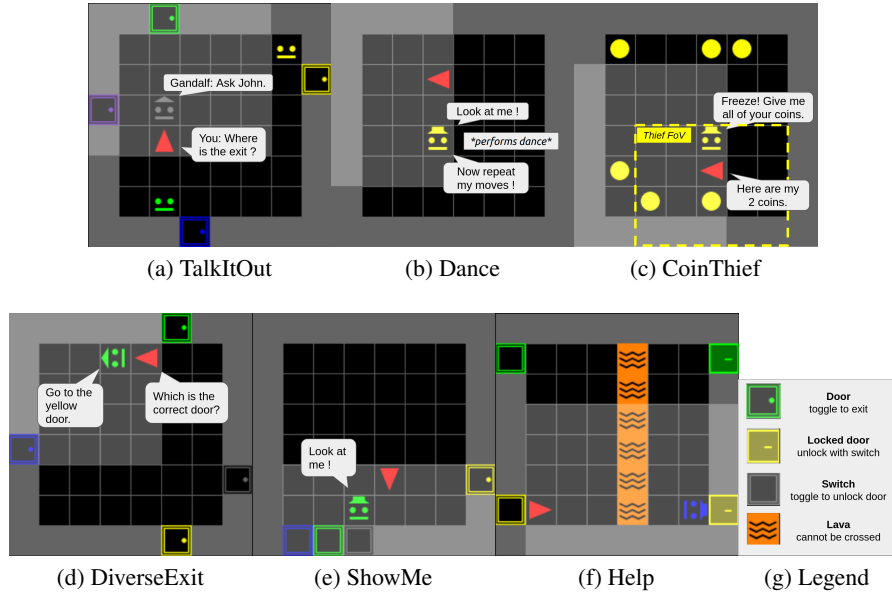
Figure 1: **SocialAI 1.0** is composed of multiple grid-world environments featuring scripted NPCs. Solving this benchmark requires to train socially proficient Deep Reinforcement Learning agents.

In the following description of environments, unless stated otherwise, the agent, all objects and all NPCs are spawned randomly for each new episode. Each description highlights the socio-cognitive skills required to solve the environment (see table 1 for a recapitulating overview).

TalkItOut - The agent has to exit the room using one of the four doors (by uttering "Open Sesame" in front of it). The environment features a wizard and two guides (one lying, one trustworthy). To find out which door is the correct one the agent has to ask the trustworthy guide for directions, and to find out which guide is trustworthy it has to query the wizard (which requires a preliminary politeness formula: "Hello, how are you?"). Solving TalkItOut requires mastering *intertwined multimodality*, basic *Theory of Mind* (infering ill-intentions), and a basic *pragmatic frame* (the agent must stand near NPCs to interact with them).

Dance - A NPC demonstrates a 3-steps dance pattern (randomly generated for each episode) and then asks the agent to reproduce this dance. Each dance step is composed of a movement action and, half of the time, of an utterance. To solve *Dance*, agent must reproduce the full dance sequence. Multiple trials are authorized. Only trials performed after the NPC completed his dance are recorded. This requires the agent to be able to infer that the NPC is setting up a *teaching pragmatic frame* ("Look at me" + `do_dance_performance` + "Now repeat my moves"), requiring the agent to *imitate* a social peer, process *multi-modal observations* and produce *multi-modal actions*.

CoinThief - In a room containing 6 coins, a thief NPC spawns near the agent, and utters that the agent must give "all of its coins". To obtain a positive reward, the agent must give (using language) exactly the number of coins *that the thief can see* (the thief field of view is a 5x5 square, i.e. a smaller version than the agent's). This requires *Theory of Mind* as the agent must understand that the thief holds *false belief* over the agent's total number of coins and must infer how many coins he actually sees.

ShowMe - The agent has to exit the room through the locked door. To unlock the door it has to press the correct button, and to find out which button is the correct one it has to look at the NPC. The NPC waits for the agent to establish eye contact, then presses the correct button and exits the room. Solving ShowMe requires that the agent infers the *teaching pragmatic frame* and imitates the NPC's goals (pressing a button, and exiting the room).

DiverseExit - The agent has to exit the room using the correct door (one out of four). To find out which door is the correct one it has to ask the NPC. There are twelve different NPCs which can be present in the environment (each episode a random one is chosen). Each NPC prefers to be asked

6

(using language) for directions differently (e.g. by standing close, by poking him, etc), i.e. via a different pragmatic frame. To solve DiverseExit the agent has to learn the diversity of frames and, most importantly, which one to use with which NPC.

Help - the environment consists of two roles (the Exiter and the Helper), one played by the agent and another by the NPC. The Exiter is placed on the right side of the room and has to exit the room using one of the two doors The doors are locked and each has a corresponding unlocking switch on the left wall. The episode ends without reward if both switches are pressed. The Helper, placed on the left side of the room, has to press the switch unlocking the door by which the agent wants to exit. The agent is trained in the Exiter role, but tested in the Helper role. To solve Help the agent needs to learn about both roles just from training as the Exiter. i.e. learn the full pragmatic frame just from seeing its own perspective of it.

SocialEnv - In this meta-environment, which contains all previous ones, we consider a multi-task setup, in which the agent is facing a randomly drawn environment, i.e. it has to infer what is the current social scenario he is spawned in (using pragmatic information collected through interaction). Mastering this environment requires to be proficient in *all of the core social skills* we proposed.

Table 1: List of core socio-cognitive skills required in each environment.

Social Skills \SocialAI Envs	TalkItOut	Dance	CoinThief	DiverseExit	ShowMe	Help	SocialEnv
Intertwined m.m.	++	++	+	++	++		++
ToM - inferring intent	++	+	+	++	+	++	++
ToM - false belief			++				++
ToM - imitating peers		++	+			+	++
ToM-joint attention				++	+	+	++
P. Frames - Diversity	+	+	+	+	++	+	++
P. Frames -Teaching		+			+		++
P. Frames - Roles						++	++

5 Experiments and Results

To showcase the relevance of *SocialAI* as a testbed to assess the socio-cognitive abilities of DRL agents, and to provide initial target baselines to outperform, we test a recent DRL architecture on our environments. Through global performance assessment and multiple case-studies, we demonstrate that this agent essentially fails to learn due to the *social complexity* of *SocialAI*'s environments.

Baselines   Our main baseline is a PPO-trained [Schulman et al., 2017] DRL architecture proposed in Hui et al. [2020]. We chose this model as it was designed for language-conditioned navigation in grid worlds, which is similar to our setup (although in our case language input is not fixed but varies along interactions). We modify the original architecture to be Multi-Headed, since our agent has to both navigate and talk, and thereafter name the resulting condition *PPO*. We also consider a variation of this baseline trained with additional intrinsic exploration bonuses (*PPO+Explo*). We consider two different exploration bonuses to reward the discovery of either new utterances or new visual objects. In each environments, we determined empirically the optimal set of exploration bonus (visual only, utterance only, or both), and only report results for the best configuration. Finally, as a lower-baseline, we consider an ablated, non-social version of our PPO agent, from which observation inputs emanating from NPCs are removed (*Unsocial PPO*). See appendix C.1 for details.

Overall results   For each condition, 16 seeded runs of 30 Millions environment steps are performed on each environments. Performance is defined as the percentage of episodes that were solved (success rate). Post-training performance results are gathered in Table 2. All considered agents essentially fail to learn, on all environments. In *ShowMe*, *DiverseExit* and *Help*, both PPO and PPO with exploration bonus (PPO+Explo) performance are not statistically significantly different from Unsocial PPO, our lower-baseline agent that doesn't observe the NPC ($p > 0.5$ in all cases, using a post-training Welch's

Table 2: Success rates (mean ± std. dev.) of considered baselines on *SocialAI* after 30 Millions environment steps (on a fixed test set of 500 environments). Our DRL agents fails to learn.

Env \Cond	PPO	PPO + Explo	Unsocial PPO
TalkItOut	0.25 ± 0.01	0.12 ± 0.03	0.25 ± 0.01
Dance	0.03 ± 0.01	0.03 ± 0.01	0.01 ± 0.0
CoinThief	0.45 ± 0.08	0.47 ± 0.04	0.38 ± 0.02
DiverseExit	0.25 ± 0.02	0.25 ± 0.01	0.24 ± 0.01
ShowMe	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Help	0.12 ± 0.05	0.11 ± 0.04	0.15 ± 0.06
SocialEnv	0.06 ± 0.02	0.08 ± 0.02	N/A

t-test). This implies that our agents are not able to leverage NPC-related inputs, i.e. they are not socially proficient. On both *TalkItOut* and *DiverseExit*, PPO agents converge to a local optimum of 25% success rate, which corresponds to ignoring the NPC and going to any door.

To better understand why our agents are failing to learn (and as a sanity check for our implementations), we present additional performance analysis of three environment-specific case studies highlighting different social skills categories of sec. 3: *TalkItOut* (Intertwined Multimodality), *CoinThief* (Theory of Mind), and *Help* (Pragmatic Frames).

Case study - TalkItOut *TalkItOut* is challenging because the agent has to master a non-trivial progression of modalities. To talk with an NPC, apart from the language modality, both vision, and primitive actions have to be used to move close to the NPC (which is mandatory for communication). Furthermore, the agent has to learn to infer, from verbal-cues of the dialogue with the wizard, which guide is the ill-intended one (i.e. a facet of ToM). For this experiment we construct an ablation environment where the ill-intended NPC is removed, which greatly reduces the social complexity as 1) all NPCs are now well-intended, and 2) dialogue with only the trustworthy guide is sufficient to solve the task.

Results - Figure 2a shows the training success rates of all our baselines. We can see that, in both environments, the PPO condition gets stuck at 25% success rate, i.e. the local optimum of ignoring the NPC and going to a random door. Adding exploration bonus (PPO+Explo condition) enables the agent to overcome this local optimum, however only in the ablation environment does this result in solving the task. This shows that the social complexity introduced by the lying guide is too challenging for our conditions. These experiments suggest that our agents lack sufficient biases for both mastering intertwined multimodal interactions and inferring different intents of social peers.

Case-study - CoinThief To assess whether it is the social complexity of the CoinThief environment that prevents our agents to learn high-performing policies, we consider a simplified version of the environment in which coins visible to the NPC have a different visual encoding from other coins. This modification removes the need to infer the NPC's field of view, i.e. the correct number of coins can be given to the NPC without any form of social awareness.

Results - Performance curves for our PPO variants on CoinThief and on the simplified CoinThief (with *coin tags* for NPC-visible coins) are shown in figure 2b. For both PPO and PPO+Explo, statistically significant improvements are obtained on the simplified environment w.r.t. vanilla CoinThief ($p < 0.001$): both approaches respectively reach a final performance of 0.81 and 0.75 (not statistically significantly different, $p = 0.07$).

Case study - Help The Help environment aims to test the ability of the agent to learn about the other's role from training only on its own i.e. to learn the whole pragmatic frame just from seeing its own perspective on it. We train the agent to achieve a shared goal on one role and then evaluate in a zero-shot manner on the other.

Results - Figure 2c shows the training success rates for the agent in the Exiter role. The horizontal dotted lines depict the performance of the same final agents on the Helper role (depicted by the same colors). We can see that in training the Exiter role is easily solved, reaching almost perfect success

8

rate in less than two million environment steps. Furthermore, we can see that the agent with the exploration bonus (PPO+Explo) is able to solve the task faster. When the same agents are evaluated in the Helper role the performance drastically drops ($\leq 15\%$ success rate). Qualitative analysis shows that this non-zero success rate on Helper role is due to agents acting as if in the Exiter role, which sometimes make them press the switch due to the stochastic nature of the PPO action sampling. The agent doesn't show any implication of understanding that the roles have been reversed. These unsurprising results outline the inability of standard RL techniques to transfer the knowledge about the task to the opposite role. The agent only learns its perspective of the pragmatic frame and not the frame itself. It doesn't understand that its goal is shared with the NPC.
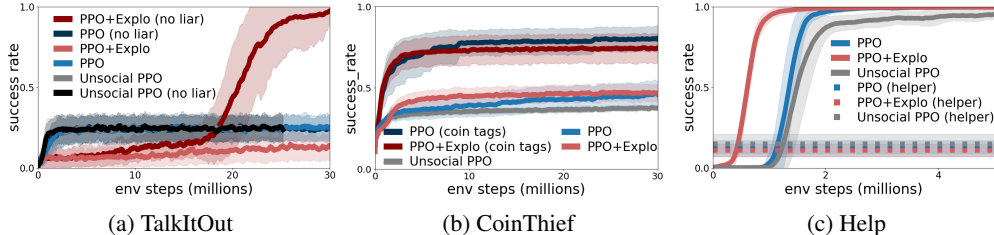

(a) TalkItOut (b) CoinThief (c) Help

Figure 2: Evolution of success rates along training in three environment specific case-studies. Mean and std. deviation plotted, 16 seeds per condition.

6   Conclusion And Discussion

In this work we classified and described a first set of core socio-cognitive skills needed to obtain socially proficient autonomous agents. We then presented *SocialAI*– an open-source testbed to assess the social skills of DRL learners, which leverages the computational simplicity of grid-world environments to enable the study of complex social situations. We then studied how a current SOTA DRL approach was unable to solve *SocialAI*. By analyzing the failure cases of this approach through multiple case studies, we were able to highlight the relevance of our benchmark as a tool to catalyze future research on socially proficient DRL agents.

A need for architectural biases. This work suggests that architectural improvements are needed for DRL agents to learn to behave appropriately in multimodal social environments. One avenue towards this is to endow agents with mechanisms enabling to learn models of others' minds, which has been identified in cognitive neuroscience works as a key ingredient of human social proficiency [Vélez and Gweon, 2020]. Some ideas have already been formulated regarding how to enable agents to master theory of mind, such as by using a meta-learning approach (through the observation and modeling of populations of agents) [Rabinowitz et al., 2018], or by leveraging inverse RL [Jara-Ettinger, 2019]. This also points to the general open-question of what parts of biases need to be "innate", and what others could be learned through practicing diverse social interaction games in the lifetime of an agent.

Limitations. What we present in this work is *SocialAI version 1.0*, i.e. we expect this benchmark to evolve along the development of better learning architectures for agents learning in social worlds. As such, multiple interesting improvements over the current version of the benchmark could be considered. We could design NPCs with more elaborated internal states, e.g. by making them more adaptive to the learner's behavior. While we consider environments with fixed sets of pragmatic frames, another interesting avenue is to design environments with *emergent* pragmatic frames, i.e. pragmatic frames that are negotiated between participants (a crucial component lacking from Human Robot Interaction methods [Vollmer et al., 2016]).

Broader impact Decision-making Machine Learning systems are more and more present in our everyday lives. In this work, we propose a fundamental research to catalyze the development of autonomous agents able to properly understand and act in a social world. Ultimately, this has the potential to simplify the alignment of machine behaviors to our human needs by easing communication.

9

Acknowledgments

References

L. S. Vygotsky and Michael Cole. *Mind in society : the development of higher psychological processes*. Harvard University Press Cambridge, 1978. ISBN 0674576284 0674576292.

Andy Clark. *Being There: Putting Brain, Body, and World Together Again*. MIT Press, Cambridge, MA, USA, 1st edition, 1996. ISBN 0262032406.

Edwin Hutchins. *Cognition in the Wild (Bradford Books)*. The MIT Press, September 1996. ISBN 0262581469. URL http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20{&}path=ASIN/0262581469.

Michael Tomasello. *The Cultural Origins of Human Cognition*. Harvard University Press, 1999. ISBN 9780674000704. URL http://www.jstor.org/stable/j.ctvjsf4jc.

Aude Billard and Kerstin Dautenhahn. Grounding communication in autonomous robots: An experimental study. *Robotics and Autonomous Systems*, 24(1):71 – 79, 1998. ISSN 0921-8890. doi: https://doi.org/10.1016/S0921-8890(98)00023-2. URL http://www.sciencedirect.com/science/article/pii/S0921889098000232. Scientific Methods in Mobile Robotics.

Rodney Brooks, Cynthia Breazeal, Matthew Marjanovic, Brian Scassellati, and Matthew Williamson. The cog project: Building a humanoid robot. *Lecture Notes in Artificial Intelligence*, 1562, 03 2002. doi: 10.1007/3-540-48834-0_5.

Angelo Cangelosi, Giorgio Metta, Gerhard Sagerer, Stefano Nolfi, Chrystopher Nehaniv, Kerstin Fischer, Jun Tani, Tony Belpaeme, Giulio Sandini, Francesco Nori, et al. Integration of action and language knowledge: A roadmap for developmental robotics. *IEEE Transactions on Autonomous Mental Development*, 2(3):167–195, 2010.

Marco Mirolli and Domenico Parisi. Towards a vygotskyan cognitive robotics: The role of language as a cognitive tool. *New Ideas in Psychology*, 29(3):298–311, 2011. ISSN 0732-118X. doi: https://doi.org/10.1016/j.newideapsych.2009.07.001. URL https://www.sciencedirect.com/science/article/pii/S0732118X09000348. Special Issue: Cognitive Robotics and Reevaluation of Piaget Concept of Egocentrism.

Jean Piaget. The origins of intelligence in children. *W W Norton & Co*, 1963.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *ICLR*, 2016.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *NeurIPS*, 2017.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *NeurIPS 2020*, 2020.

Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. A survey of reinforcement learning informed by natural language. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6309–6317, 7 2019. doi: 10.24963/ijcai.2019/880. URL `https://doi.org/10.24963/ijcai.2019/880`.

Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.463. URL `https://www.aclweb.org/anthology/2020.acl-main.463`.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew J. Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. Textworld: A learning environment for text-based games. *ArXiv*, abs/1806.11532, 2018. URL `http://arxiv.org/abs/1806.11532`.

Shrimai Prabhumoye, Margaret Li, Jack Urbanek, Emily Dinan, Douwe Kiela, Jason Weston, and Arthur Szlam. I love your chain mail! making knights smile in a fantasy game world: Open-domain goal-oriented dialogue agents, 2020.

Prithviraj Ammanabrolu, Jack Urbanek, Margaret Li, Arthur Szlam, Tim Rocktäschel, and Jason Weston. How to motivate your dragon: Teaching goal-driven agents to speak and act in fantasy worlds, 2020.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, and et al. Experience grounds language. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. doi: 10.18653/v1/2020.emnlp-main.703. URL `http://dx.doi.org/10.18653/v1/2020.emnlp-main.703`.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning, 2018a.

Cédric Colas, Tristan Karch, Nicolas Lair, Jean-Michel Dussoux, Clément Moulin-Frier, Peter F. Dominey, and Pierre-Yves Oudeyer. Language as a cognitive tool to imagine goals in curiosity driven exploration. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS 2020*, 2020a. URL `https://proceedings.neurips.cc/paper/2020/hash/274e6fcf4a583de4a81c6376f17673e7-Abstract.html`.

Felix Hill, Sona Mokra, N. Wong, and Tim Harley. Human instruction-following with deep reinforcement learning via transfer-learning from text. *ArXiv*, abs/2005.09382, 2020a.

Ahmed Akakzia, Cédric Colas, Pierre-Yves Oudeyer, Mohamed Chetouani, and Olivier Sigaud. Grounding Language to Autonomously-Acquired Skills via Goal Generation. In *ICLR 2021 - Ninth International Conference on Learning Representation*, Vienna / Virtual, Austria, May 2021. URL `https://hal.inria.fr/hal-03121146`.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. *ArXiv*, abs/1711.11543, 2017. URL `http://arxiv.org/abs/1711.11543`.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. IQA: visual question answering in interactive environments. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4089–4098. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00430. URL `http://openaccess.thecvf.com/content_cvpr_2018/html/Gordon_IQA_Visual_Question_CVPR_2018_paper.html`.

Josh Abramson, Arun Ahuja, Arthur Brussee, Federico Carnevale, Mary Cassin, Stephen Clark, Andrew Dudzik, Petko Georgiev, Aurelia Guy, Tim Harley, Felix Hill, Alden Hung, Zachary Kenton, Jessica Landon, Timothy P. Lillicrap, Kory Mathewson, Alistair Muldal, Adam Santoro, Nikolay Savinov, Vikrant Varma, Greg Wayne, Nathaniel Wong, Chen Yan, and Rui Zhu. Imitating interactive intelligence. *ArXiv*, abs/2012.05672, 2020. URL `https://arxiv.org/abs/2012.05672`.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1495–1502. AAAI Press, 2018. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17007`.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Çaglar Gülçehre, Pedro A. Ortega, DJ Strouse, Joel Z. Leibo, and Nando de Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97, pages 3040–3049. PMLR, 2019. URL `http://proceedings.mlr.press/v97/jaques19a.html`.

Ryan Lowe, Abhinav Gupta, Jakob N. Foerster, Douwe Kiela, and Joelle Pineau. On the interaction between supervision and self-play in emergent communication. In *8th International Conference on Learning Representations, ICLR 2020*, 2020. URL `https://openreview.net/forum?id=rJxGLlBtwH`.

Mark Woodward, Chelsea Finn, and Karol Hausman. Learning to interactively learn and assist. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 2535–2543. AAAI Press, 2020. URL `https://aaai.org/ojs/index.php/AAAI/article/view/5636`.

Jerome Bruner. Child's talk: Learning to use language. *Child Language Teaching and Therapy*, 1 (1):111–114, 1985. doi: 10.1177/026565908500100113. URL `https://doi.org/10.1177/026565908500100113`.

Eros Moreira de Carvalho. Social affordance. *Encyclopedia of Animal Cognition and Behavior*, 2020.

Luc Steels. The symbol grounding problem has been solved. so what's next? *Symbols, Embodiment and Meaning. Oxford University Press, Oxford, UK*, 01 2007. doi: 10.1093/acprof:oso/9780199217274.003.0012.

Minoru Asada, Koh Hosoda, Yasuo Kuniyoshi, Hiroshi Ishiguro, Toshio Inui, Yuichiro Yoshikawa, Masaki Ogino, and Chisato Yoshida. Cognitive developmental robotics: A survey. *IEEE Transactions on Autonomous Mental Development*, 1(1):12–34, 2009. doi: 10.1109/TAMD.2009.2021702.

Angelo Cangelosi and Matthew Schlesinger. *Developmental Robotics: From Babies to Robots*. The MIT Press, 2014. ISBN 0262028018.

Aude Billard and Kerstin Dautenhahn. Experiments in learning by imitation - grounding and use of communication in robotic agents. *Adaptive Behavior*, 7(3-4):415–438, 1999. doi: 10.1177/105971239900700311. URL `https://doi.org/10.1177/105971239900700311`.

Jessica Lindblom and Tom Ziemke. Social situatedness of natural and artificial intelligence: Vygotsky and beyond. *Adaptive Behavior*, 11(2):79–96, 2003. doi: 10.1177/10597123030112002. URL `https://doi.org/10.1177/10597123030112002`.

Katharina J. Rohlfing, Britta Wrede, Anna-Lisa Vollmer, and Pierre-Yves Oudeyer. An alternative to mapping a word onto a concept in language acquisition: Pragmatic frames. *Frontiers in Psychology*, 7:470, 2016. ISSN 1664-1078. doi: 10.3389/fpsyg.2016.00470. URL `https://www.frontiersin.org/article/10.3389/fpsyg.2016.00470`.

Jonathan Grizou, Iñaki Iturrate, Luis Montesano, Pierre-Yves Oudeyer, and Manuel Lopes. Interactive learning from unlabeled instructions. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, page 290–299, Arlington, Virginia, USA, 2014. AUAI Press. ISBN 9780974903910.

Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2008.10.024. URL `https://www.sciencedirect.com/science/article/pii/S0921889008001772`.

Carlos Celemin and Javier Ruiz-del Solar. Coach: Learning continuous actions from corrective advice communicated by humans. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 581–586, 2015. doi: 10.1109/ICAR.2015.7251514.

Anna-Lisa Vollmer, Britta Wrede, Katharina J. Rohlfing, and Pierre-Yves Oudeyer. Pragmatic frames for teaching and learning in human–robot interaction: Review and challenges. *Frontiers in Neurorobotics*, 10:10, 2016. ISSN 1662-5218. doi: 10.3389/fnbot.2016.00010. URL https://www.frontiersin.org/article/10.3389/fnbot.2016.00010.

Neziha Akalin and Amy Loutfi. Reinforcement learning approaches in social robotics. *Sensors*, 21(4), 2021. ISSN 1424-8220. doi: 10.3390/s21041292. URL https://www.mdpi.com/1424-8220/21/4/1292.

Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Intrinsically motivated goal-conditioned reinforcement learning: a short survey. *CoRR*, abs/2012.09830, 2020b. URL https://arxiv.org/abs/2012.09830.

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, Marcus Wainwright, Chris Apps, Demis Hassabis, and Phil Blunsom. Grounded language learning in a simulated 3d world. *CoRR*, abs/1706.06551, 2017. URL http://arxiv.org/abs/1706.06551.

Felix Hill, Sona Mokra, Nathaniel Wong, and Tim Harley. Human instruction-following with deep reinforcement learning via transfer-learning from text. *CoRR*, abs/2005.09382, 2020b. URL https://arxiv.org/abs/2005.09382.

Justin Fu, Anoop Korattikara, Sergey Levine, and Sergio Guadarrama. From language to goals: Inverse reinforcement learning for vision-based instruction following. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=r1lq1hRqYQ.

Corey Lynch and Pierre Sermanet. Grounding language in play. *CoRR*, abs/2005.07648, 2020. URL https://arxiv.org/abs/2005.07648.

Khanh Nguyen, Dipendra Misra, Robert E. Schapire, Miroslav Dudík, and Patrick Shafto. Interactive learning from activity description. *CoRR*, abs/2102.07024, 2021. URL https://arxiv.org/abs/2102.07024.

Dipendra Kumar Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. Mapping instructions to actions in 3d environments with visual goal prediction. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2667–2678. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1287. URL https://doi.org/10.18653/v1/d18-1287.

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. A benchmark for systematic generalization in grounded language understanding. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19861–19872. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/e5a90182cc81e12ab5e72d66e0b46fe3-Paper.pdf.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://www.aclweb.org/anthology/W18-5446.

Amir Zadeh, Michael Chan, Paul Pu Liang, Edmund Tong, and Louis-Philippe Morency. Social-iq: A question answering benchmark for artificial social intelligence. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8799–8809, 2019. doi: 10.1109/CVPR.2019.00901.

Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. Learning to speak and act in a fantasy text adventure game. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 673–683. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1062. URL https://doi.org/10.18653/v1/D19-1062.

Ronald Riggio. Assessment of basic social skills. *Journal of Personality and Social Psychology*, 51: 649–660, 09 1986. doi: 10.1037/0022-3514.51.3.649.

Miriam Beauchamp and Vicki Anderson. Social: An integrative framework for the development of social skills. *Psychological bulletin*, 136:39–64, 01 2010. doi: 10.1037/a0017768.

H. M. Wellman. *The child's theory of mind*. The MIT Press, 1992.

John H. Flavell. Cognitive development: Children's knowledge about the mind. *Annual Review of Psychology*, 50(1):21–45, 1999. doi: 10.1146/annurev.psych.50.1.21. URL `https://doi.org/10.1146/annurev.psych.50.1.21`.

Renée Baillargeon, Rose M. Scott, and Zijing He. False-belief understanding in infants. *Trends in Cognitive Sciences*, 14(3):110–118, 2010. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2009.12.006. URL `https://www.sciencedirect.com/science/article/pii/S1364661309002885`.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. `https://github.com/maximecb/gym-minigrid`, 2018b.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017. URL `http://arxiv.org/abs/1707.06347`.

David Yu-Tung Hui, Maxime Chevalier-Boisvert, Dzmitry Bahdanau, and Yoshua Bengio. Babyai 1.1, 2020.

Natalia Vélez and Hyowon Gweon. Learning from other minds: An optimistic critique of reinforcement learning models of social learning. *PsyArXiv*, Aug 2020. doi: 10.31234/osf.io/q4bxr. URL `psyarxiv.com/q4bxr`.

Neil C. Rabinowitz, Frank Perbet, H. Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. Machine theory of mind. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4215–4224. PMLR, 2018. URL `http://proceedings.mlr.press/v80/rabinowitz18a.html`.

Julian Jara-Ettinger. Theory of mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 29:105–110, 2019. ISSN 2352-1546. doi: https://doi.org/10.1016/j.cobeha.2019.04.010. URL `https://www.sciencedirect.com/science/article/pii/S2352154618302055`. Artificial Intelligence.

Daniel H Grollman and Aude Billard. Donut as i do: Learning from failed demonstrations. In *2011 IEEE International Conference on Robotics and Automation*, pages 3804–3809, 2011. doi: 10.1109/ICRA.2011.5979757.

Dzmitry Bahdanau, Felix Hill, Jan Leike, Edward Hughes, Seyed Arian Hosseini, Pushmeet Kohli, and Edward Grefenstette. Learning to understand goal specifications by modelling reward. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=H1xsSjC9Ym`.

Felix Hill, Andrew Lampinen, Rosalia Schneider, Stephen Clark, Matthew Botvinick, James L. McClelland, and Adam Santoro. Environmental drivers of systematicity and generalization in a situated agent. In *International Conference on Learning Representations*, 2020c. URL `https://openreview.net/forum?id=SklGryBtwr`.

Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Joshua B. Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration, 2020.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1988–1997. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.215. URL `https://doi.org/10.1109/CVPR.2017.215`.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. {ALFW}orld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=0IOX0YcCdTn`.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2067–2075. JMLR.org, 2015. URL `http://proceedings.mlr.press/v37/chung15.html`.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. *CoRR*, abs/1709.07871, 2017. URL `http://arxiv.org/abs/1709.07871`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy P. Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. *ArXiv*, abs/1810.02274, 2018. URL `http://arxiv.org/abs/1810.02274`.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Exploration: A study of count-based exploration for deep reinforcement learning, 2017.

A Supplementary Material: Extended Related Work

Earlier calls for socially proficient agents   This work aims to connect the recent DRL & NLP literature to the older developmental robotics field [Asada et al., 2009, Cangelosi and Schlesinger, 2014], which studies how to leverage knowledge from the cognitive development of human babies into embodied robots. Within this field, multiple calls for developing the social intelligence of autonomous agents have already been formulated [Billard and Dautenhahn, 1999, Lindblom and Ziemke, 2003, Mirolli and Parisi, 2011]. This emphasis on the importance of social interactions for learning is probably what led Bruner to conceptualize the notion of pragmatic frames [Bruner, 1985], which has later been reused for example as as conceptual tool to theorize language development [Rohlfing et al., 2016]. In this work we intent to further motivate the relevance of this notion to enable further progress in Deep RL and AI.

Human-Robot Interaction   Interactions with knowledgeable human teachers is a well studied form of social interaction. Many works within the Human-Robot Interaction (HRI) and the Interactive learning field studied how to provide interactive teaching signals to their agents, e.g. providing instructions [Grizou et al., 2014], demonstrations [Argall et al., 2009, Grollman and Billard, 2011], or corrective advice [Celemin and Ruiz-del Solar, 2015]. In [Vollmer et al., 2016], authors review this field, showing that many of the considered interaction protocols can be reduced to a restricted set of pragmatic frames. They note that most of these works consider single rigid pragmatic frames. Echoing this observation, the SocialAI benchmark invites to study a broader set of social situations, e.g. requiring agents to both move and speak, and even to learn to interact in a *diversity* of pragmatic frames. Catalysing research on DRL and social skills seems even more relevant now that many application-oriented works are beginning to leverage RL and DRL into real world humanoid social robots [Akalin and Loutfi, 2021].

Recent works on language grounded DRL   Building on NLP, developmental robotics, and previous works on classical goal-conditioned DRL [Colas et al., 2020b], a renewed interest emerged towards the development of embodied autonomous agents able to process language [Luketina et al., 2019]. Most approaches were proposed to design language conditioned agents in instruction following scenarios (e.g. "go to the red bed"). In Hermann et al. [2017], authors train a DRL model from pixels in a 3D world by augmenting instruction following with auxiliary tasks (language prediction and temporal autoencoding), enabling their agent to follow object-relative instructions ("pick the red object next to the green object"). Multiple other approaches were studied to ease learning: leveraging pre-trained language models [Hill et al., 2020b], demonstrations [Fu et al., 2019, Lynch and Sermanet, 2020] (from which rewards can be learned [Bahdanau et al., 2019]), or using descriptive feedbacks [Nguyen et al., 2021] (which has also been tested in combination with imagining new goals [Colas et al., 2020b]). Hill et al. [2020c] try to assess to which extent vanilla language conditioned agents are able to perform systematic generalization (combining known concepts/skills into new ways). In Embodied Visual Question Answering works, agents are conditionned on questions, requiring them to navigate within environments and then produce an answer ("what color is the bed ?") [Das et al., 2017, Gordon et al., 2018]. Compared to these previous works, SocialAI aims to enlarge the set of considered scenarios by studying how language conditioned agents are able to ground and produce language within diverse forms of social interactions among embodied social peers.

Closely related to the present work is Abramson et al. [2020], which is an invitation to focus research on building embodied multi-modal agents suited for human-robot interactions in the real-world. Towards this goal, authors propose a series of experiments on a simulated 3D playroom environment designed for multi-agent interactive scenarios featuring both autonomous agents and/or human players. The main focus of the paper is in proposing novel ways to leverage human demonstrations to bootstrap agents performance and allow meaningful interactive sessions with humans (as scaffolding a randomly acting agent is a tedious journey). Because of the complexity of their considered experiments (imitation learning, 3D environments, pixel-based, human in the loop, ...), their work only considers the two now-common social interaction scenarios: Visual Question Answering and instruction following. The novelty of their setup is that these questions/instructions are alternatively produced or tackled by learning agents in an interactive fashion. In SocialAI, we focus on a lighter experimental pipeline (2D grid-world, low dimensional symbolic pixels, no humans) such that we are able to study a broader range of social scenarios, requiring multi-steps conversations and interactions with multiple (scripted) agents within a single episode.

Benchmarks on embodied agents and language Multiple benchmarks featuring language and embodied agents already exists. The BabyAI [Chevalier-Boisvert et al., 2018a] and gSCAN benchmarks [Ruis et al., 2020] test language conditioned agents on grid-world environments. BabyAI focuses on assessing the sample efficiency of tested agents while gSCAN targets systematic generalization. Misra et al. [2018] extends this type of benchmark to 3D environments. In contrary to SocialAI, these benchmarks do not consider multi-modal action spaces, i.e. agents do not produce language. Besides, they only consider a single rigid social interaction protocol: instruction following.

Related to instruction following benchmarks are testbeds for embodied visual question answering [Gordon et al., 2018, Das et al., 2017]. Here agents are conditioned on questions: they must navigate within an environment to collect information and produce an answer (i.e. a one or two word output).

Puig et al. [2020] propose a new benchmark to test social perception in machine learning models. Learning agents must infer the intent of a scripted agent in a 3D world (using a single demo) to better collaborate towards a similar goal in a new environment. Here again, despite being novel and relevant, only a single social interaction is considered: cooperation towards a common goal, which in that case does not require language use nor understanding.

In between classical disembodied NLP testbeds [Johnson et al., 2017, Wang et al., 2018, Zadeh et al., 2019] and previously discussed embodied language benchmarks is the LIGHT environment [Urbanek et al., 2019], A multiplayer text adventure game allowing to study social settings requiring complex dialogue production [Ammanabrolu et al., 2020, Prabhumoye et al., 2020]. While they consider a text world, i.e. a virtual embodiment, the SocialAI benchmark tackles the arguably harder and richer setting of egocentric embodiment among embodied social peers. Text worlds have also been used in combination with an embodied environment to demonstrate how language-based planning (in text worlds) can benefit instruction following [Shridhar et al., 2021].

Within the Multi Agent Reinforcement Learning field, Mordatch and Abbeel [2018] propose embodied environments to study the emergence of grounded compositional language. Here language is merely a discrete set of abstract symbols that can only be used one at a time (per step) and whose meanings must be negotiated by agents. While symbol negotiation is an intersting social situation to study, we leave it to future work and consider scenarios in which agents must enter an already existing social world (using non-trivial language). In Jaques et al. [2019], authors present Multi Agent social dilemna environments requiring the emergence of cooperative behaviors through communication. In this work communication is stricly non-verbal, while we consider both non-verbal communication (e.g. gaze following) and language based communication.

B Environment details

B.1 Action space

The action space of the environment consists of two modalities (*primitive actions* and *language*) which results in a 3D discrete action vector.

The first dimension corresponds to the primitive action modality, which are identical to actions available in minigrid ([Chevalier-Boisvert et al., 2018b], from which our code is based on. It consists of 7 actions (turn left, turn right, move forward, pickup, drop, toggle, done).

In all the environments *pickup* and *drop* actions do not do anything and *done* terminates the episode. We kept these actions as we intend to use them in future iterations of the benchmark. In TalkItOut, Dance, and CoinThief *toggle* terminates the episode with 0 reward. In DiverseExit, ShowMe and Help *toggle* opens doors and presses buttons. In DiverseExit, it can also be used to *poke* the NPC.

In Dance and CoinThief, only a subset of 3 primitive actions are available (to simplify these environments): *turn left*, *turn right* and *move forward*.

In SocialEnv, all actions behave as in the original environment that is sampled for each new episode.

The second and third dimensions regard the language modality. The second dimension selects a template and the third a noun. The full grammar for each environments is shown in table 3. In SocialEnv all grammars are merged to a single big one containing all templates and nouns used in all single environments.

Table 3: Template-based grammars for all the environments

Templates

Action	Template			
	TalkItOut, DiverseExit	CoinThief	Dance, ShowMe, Help	SocialEnv
0	Where is <noun>	Here is <noun>	Move your <noun>	Where is <noun>
1	Open <noun>		Shake your <noun>	Open <noun>
2	Which is <noun>			Close <noun>
3	How are <noun>			How are <noun>
4				Move your <noun>
5				Shake your <noun>
6				Here is <noun>
7				Which is <noun>

Nouns

Action	Noun				
	TalkItOut	DiverseExit	CoinThief	Dance, ShowMe, Help	SocialEnv
0	sesame	sesame	1	body	sesame
1	the exit	the exit	2	head	the exit
2	the wall	the correct door	3		the wall
3	you	you	4		the floor
4	the ceiling	the ceiling	5		the ceiling
5	the window	the window	6		the window
6	the entrance	the entrance			the entrance
7	the closet	the closet			the closet
8	the drawer	the drawer			the drawer
9	the fridge	the fridge			the fridge
10	oven	oven			oven
11	the lamp	the lamp			the lamp
12	the trash can	the trash can			the trash can
13	the chair	the chair			the chair
14	the bed	the bed			the bed
15	the sofa	the sofa			the sofa
16					the correct door
16					1
17					2
18					3
19					4
20					5
21					6
22					body
23					head

Both modalities can also be undefined, i.e. no action is taken in the undefined modality. Examples of such actions are shown in table 4.

B.2 Observation space

The multimodal observation space consists of the *vision* modality and the *language* modality.

The *vision* modality is manifested as a *7x7* grid displaying the space in front of the agent (shown as highlighted grids in figure 1). Each location of this grid is encoded as 4 integers for the object type, color, status and orientation (used for NPCs). *status* is used to refer to object states (e.g. door is *open*)

Table 4: Examples of various actions in the environment. Second and third dimension must both either be underfined or not.

Action	description
(1, -, -)	moves left without speaking
(1, 1, 5)	moves left and utters "Open the window"
(-, 1, 5)	doesn't move but utters "Open the window"
(-, -, -)	nothing happens

or, if the object is an NPC, it is used to inform about the NPC type (e.g. *wizard* NPC. For example, a blue wizard NPC facing down will be encoded as $(11, 2, 0, 1)$ and a blue guide NPC facing up will be encoded as $(11, 2, 1, 3)$.

The *language* modality is represented as a string containing the currently heard utterances, i.e. utterances uttered by NPCs next to the agent, and their names (ex. "John: go to the green door"). In case of silence, an "empty indicator" symbol is used.

As it is often more convenient to concatenate all the utterances heard, to simplify the implementation of the agent, the implementation of the environment also supports giving the full history of heard utterances with the "empty indicator" symbols removed as additional information.

B.3 The reward

In all of our environments the extrinsic reward is given upon completing the task. The reward is calculated by the following equation:

$$r_{extr} = 1.0 - 0.9 * \frac{t}{t_{max}} \tag{1}$$

, where t is the number of steps the agent made in the environment and t_{max} is the maximum allowed number of steps.

B.4 TalkItOut

This environment consists of three NPCs and four doors, and the goal of the agent is to exit the room using the correct (one out of four) door in $t_{max} = 100$ steps. The agent can find out which door is the correct one by asking the true guide. To find out which guide is the correct one, the agent has to ask the wizard. Before talking to any NPC, the agent has to stand in front of it and introduce himself by saying "How are you?". Upon finding out which door is the correct one, the agent has to stand in front of it and utter "Open sesame". Then the episode ends, and the reward is given.

If the agent executes *done*, *toggle* or utters "Open sesame" in front of the wrong door the episode ends with no reward.

An example of a dialogue that might appear in a successful episode is shown in table 5

For each episode the colors of doors and NPCs are selected randomly from a set of six and the names of the two guides are selected randomly from a set of two (Jack, John), i.e. in one episode either Jack or John will be the truth speaking guide and the other will be the lying guide. Furthermore, the grid width and height are randomized from the minimal size of 5 up to 8 and the NPCs and the agent are placed randomly inside (omitting locations in front of doors).

Required social skills   In the remainer of this section we use the TalkItOut environment to provide an in-depth example of the detailed list of social skills required in one of our environments (we revert to more straightforward descriptions for all subsequent environment descriptions).

Intertwined multimodality - To solve TalkItOut the agent must use both modalities both in the action and in the observation space. Furthermore, this multimodality is intertwined because the progression in which the modalities are used is non-trivial. To discuss this notion further, let's imagine an example of instruction following. The progression of modalities here is trivial because the agent always *listens* for the command first and then *looks* and *moves/acts* to complete the task. Another good example is

Table 5: An example of a successful episode in the TalkItOut environemnt

True guide: John
Correct door color: blue

agent goes to the wizard
Agent: How are you?
Wizard: I am fine.
Agent: Where is the exit?
Wizard: Ask John.
agent goes to one guide
Agent: How are you?
Jack: I am fine.
Agent: Where is the exit?
Jack: Go to the red door.
agent goes to the other guide
Agent: How are you?
John: I am fine.
Agent: Where is the exit?
John: Go to the blue door.
agent goes to the blue door
Agent: Open sesame

embodied question answering. Here the agent again always first *listens* to the question, then *looks* and *moves* in the environment to finally, at the end, *speak* the answer.

In our environment, however, the agent must always choose which modality to use based on the current state. Furthermore, it will often be required to switch between modalities many times. For example, to talk to an NPC the agent first *looks* to find the NPC, then it *moves* to the NPC, finally the agent *speaks* to it and *listens* to the response. This progression is then used, if needed, for other NPCs, and finally a similar one used to go to the correct door and open it. Furthermore, depending on the current configuration of environment, the progression can also be different. Usually, after finding out the correct door the agent needs to *look* for it and *move* to it to *speak* the password, but if the true guide is already next to the correct door only *looking* for the door and *speaking* the password is required.

Theory of Mind - Since the agent must be able to infer good or bad intentions of other NPCs, a basic form of ToM is needed. Primarily, the agent needs to infer that the wizard is well-intended, wants to help, and is therefore trustworthy. Using the inferred trust in the wizard, it is possible to infer the good intentions of the true guide, and likewise the bad intentions of the false guide.

On the other hand, as the false guide chooses which false direction to give each time asked, it is also possible to infer its ill-intentions by asking him many times in the same episode and observing this inconsistency. If an NPC gives different answers for the same question in the same episode, then it is evident its intentions are bad.

Pragmatic frames - Pragmatic frames were not the focus of this environment, and are studied in more detail in other environments, they are present in this environment only in a simple form. To talk with an NPC the agent needs to stand next to it and introduce itself by saying "how are you", and to get an answer the agent needs to ask "where is the exit". These simple rules (a.k.a. social conventions) are pragmatic frames, i.e. grammars describing possible and impossible interactions. For example, it is impossible to communicate if you are far and get directions if you ask "Where is the floor". The agent needs to be able to extract these rules and use them in relation to all NPCs.

B.5 Dance

A Dancer NPC demonstrates a 3-steps dance pattern (randomly generated for each episode) and then asks the agent to reproduce this dance. Each dance step is composed of a primitive action, randomly selected among rotating left, right, or moving forward. 50% of the time, a randomly selected utterance among 4 possible ones (see table 3) is also performed simultaneously with the primitive action. In

the first step of each episode, the NPC utters "Look at me!". It then performs the dance in the next 3 steps. Finally, at the fifth step, the NPC utters "Now repeat my moves", and starts to record the agents' actions. In contrary to TalkItOut, the agent does not need to be close to the NPC to interact with it (i.e. both are "shouting"). To solve *Dance*, the agent must reproduce the full dance sequence. Multiple trials are authorized within the $t_{max} = 20$ steps of an episode. Only trials performed after the NPC completed his dance are recorded.

The Dance environment requires the agent to be able to infer that the NPC is setting up a *teaching pragmatic frame* ("Look at me!" + do_dance_performance + "Now repeat my moves!"), requiring the agent to *imitate* a social peer, process *multi-modal observations* and produce *multi-modal actions*.

B.6 CoinThief

In a room containing 6 coins (randomly placed), a Thief NPC spawns near the agent, i.e. in one of the 4 cells adjacent to the agent (selected randomly for each new episode). At step 0, the Thief NPC utters "Freeze! Give me all your coins!". The agent can "give coins" by uttering "here is $< nb >$", with $< nb >$ ranging from 0 to 6 (see table 3. Note that the agent does not need to collect coins by navigating within the environment, it only has to utter. To obtain a positive reward, the agent must give exactly the number of coins *that the thief can see*. The thief field of view is a 5x5 square, i.e. a smaller version than the agent's. In addition of its initial orientation facing the agent, the thief also "look around" in another direction, either left or right (selected at random for each episode). Episodes are aborted without reward if the agent use the *move forward* action (the thief wants the agent not to move), or if the maximum number of steps ($t_{max} = 20$) is reached. Solving the CoinThief environment requires *Theory of Mind* as the agent must understand that the thief holds *false belief* over the agent's total number of coins and must infer how many coins he actually sees. To infer how many coins the thief sees, the agent must learn the thief's field of view and use memory to remember the thief's two view directions.

B.7 DiverseExit

The goal of the agent is to exit the room using one of four doors in $t_{max} = 50$ steps. One NPC is present in the environment. Colors and the initial positions of the NPC and the doors are randomized each episode.

To find out which door is the correct one, the agent has to ask the NPC. To talk to the NPC, the agent has to introduce himself by saying one of two possible *introductory utterances* ("Where is the exit?" or "Which is the correct door?").

There are 12 different NPC types, one of which is randomly selected to be present in the episode. Each of the 12 NPCs prefer to be introduced to differently, to be more precise, when the agent utters one of the two *introductory utterances* for the first time in the episode, the *introductory configuration* is saved. The *introductory configuration* is manifested as the tuple of the following four elements: (is the agent next to the NPC, was the NPC poked, is eye contact established, which *introductory utterance* was used). Each NPC must be asked with its preferred *introductory configuration*. This enables us to create twelve different NPC and their corresponding *introductory configurations*. Those twelve configurations are listed in the table 6.

If the *introductory configuration* is the one corresponding to the present NPC, the NPC will give the agent the directions (ex. "go to the green door") every time they establish eye contact. However, if the *introductory configuration* was not the right one, the NPC will not give the directions in this episode (a once saved introductory state cannot be overwritten in the same episode).

To solve TeachingGames the agent must learn a large diversity of different frames (12). Furthermore, it must learn to differentiate between them and infer which frame to use with which NPC.

B.8 ShowMe

The goal of the agent is to exit the room through the door placed at the top wall of the environment in $t_{max} = 100$ steps. At the beginning of the episode the door is locked and can be unlocked by activating the correct switch (one out of three) on the bottom wall. The switches can be activated using the *toggle* action, however once a switch is activated it cannot be deactivated. This means that

Table 6: Twelve *introductory configurations* corresponding to twelve possible different NPCs in DiverseExit.

npc_type	is next to the NPC	was the NPC poked	eye contact	introductory utterance used
0	next to	poked	Yes	"Where is the exit"
1	next to	not poked	Yes	"Where is the exit"
2	not next to	not poked	Yes	"Where is the exit"
3	next to	poked	Yes	"Which is the correct door"
4	next to	not poked	Yes	"Which is the correct door"
5	not next to	not poked	Yes	"Which is the correct door"
6	next to	poked	No	"Where is the exit"
7	next to	not poked	No	"Where is the exit"
8	not next to	not poked	No	"Where is the exit"
9	next to	poked	No	"Which is the correct door"
10	next to	not poked	No	"Which is the correct door"
11	not next to	not poked	No	"Which is the correct door"

the agent must press the correct switch from the first try. The information about which switch is the correct one can be inferred by looking at the NPC. Once eye contact is established with the NPC, the NPC will say "Look at me" and proceed to press the switch and exit the room. After this, the switches are reset and the door is locked once again. As the switch doesn't change whether it's activated or not (it looks the same) the agent must infer what switch was pressed by looking at the NPCs movement and infer which switch was activated. The reward is given once both the NPC and the agent have left the room, i.e. if the agent leaves the room before the NPC no reward is given.

To solve ShowMe, the agent must learn to imitate the NPCs goals (pressing the correct switch) from its behavior (a facet of ToM). Furthermore, it has to infer the *Teaching pragmatic frame* where the *slot* is the pressed button.

B.9 Help

The environment consists of two roles: The Exiter and the Helper. The shared goal of both the participants it for the Exiter to exit the environment in $t_{max} = 20$ steps. It can do so using any of the two doors on the right wall of the environment. At the beginning of the episode both doors are locked and can be unlocked by pressing the corresponding switch on the left wall. The environment is separated by a wall of lava in the middle, disabling movement from the left to the right side of the room. The Exiter is placed on the right side of the environment (next to the doors) and the Helper on the left side (next to the switches). As the episode ends without reward if both switches are pressed, the two participants have to agree on which door to use.

The purpose of this environment is to train the agent in the Exiter role and test in the Helper role.

When the agent is in the Exiter role (training phase), the NPC is in the Helper role. Then the NPC acts as follows. It moves towards the switch corresponding to the door that is the closest to the agent. Once in front of the switch, it looks at the agent and waits for eye contact. Once eye contact has been established, the NPC activates the switch. The agent, therefore, needs to learn to choose a door and confirm this choice by establishing eye contact.

When the agent is in the Helper role (testing phase), the NPC is in the role of the Exiter. Then the NPC chooses a door and moves in front of it. Once there, it looks at the agent and waits for eye contact. Once eye contact has been established, the NPC attempts to exit the room by the door.

To solve Help, the agent must learn the whole pragmatic frame just from seeing its own perspective of it. It must learn to infer the shared goal and which actions from both the agent and the NPC lead to the achievement of this goal.

B.10 SocialEnv

SocialEnv is a *meta-environment*, i.e. it is a multi-task environment in which, for each new episode, the agent is randomly spawned into one of the 6 previously discussed environments. The agent's

Table 7: Training hyperparameters

Hyperparameter	value
learning rate	$1e^4$
GAE λ	0.99
clip ϵ	$1e^5$
batch size	1280
γ	0.99
recurrence	10
epochs	4

Table 8: Exploration bonus hyperparameters

Hyperparameter	value						
	TalkItOut	Dance	CoinThief	DiverseExit	ShowMe	Help	SocialEnv
type	lang	vision	vision	lang	vision	vision	vision
T	0.6	0.6	0.6	0.6	0.6	0.6	0.6
C	7	3	2	20	3	3	2
M	50	50	50	50	50	50	50

grammar is a set of all previous grammars (see table 3). t_{max} is set to the original t_{max} of each environment.

Solving SocialEnv requires to infer what is the current social scenario (i.e. the current environnment) the agent is spawned in. This can be achieved by leveraging pragmatic information collected through interaction, i.e. differentiating environments from social interaction footprints. For instance, a proficient agent could reliably detect it is in the TalkItOut environment by observing that there are 3 NPCs (1 wizard-type and 2 guide-type). Given that this environment detection is mastered, the agent still has to be proficient in *all of the core social skills* we proposed, to be able to solve each environment.

C   Experimental details

C.1   Baselines details

PPO baseline   In this work we use a PPO-trained [Schulman et al., 2017] DRL architecture initially designed for the BabyAI benchmark [Chevalier-Boisvert et al., 2018a]. The policy design was improved in a follow-up paper by Hui et al. [2020] (more precisely, we use their *original_endpool_res* model). See figure 3 for a visualization of the complete architecture. First, symbolic pixel grid observations are fed into two convolutional layers [LeCun et al., 1989, Krizhevsky et al., 2012] (3x3 filter, stride and padding set to 1), while dialogue inputs are processed using a Gated Recurrent Unit layer [Chung et al., 2015]. The resulting image and language embeddings are combined using two FiLM attention layers [Perez et al., 2017]. Max pooling is performed on the resulting combined embedding before being fed into an LSTM [Hochreiter and Schmidhuber, 1997] with a $128D$ memory vector. The LSTM embedding is then used as input for the navigation action head, which is a two-layered fully-connected network with tanh activations and has an 8D output (i.e. 7 navigation actions and no_op action).

In order for our agent to be able to both move and talk, we add to this architecture a talking action head, which is composed of three networks. All of them are two-layered, fully-connected networks with tanh activations, and take the LSTM's embedding as input. The first one is used as a switch: it has a one-dimensional output to choose whether the agent talks (output > 0.5) or not (output < 0.5). If the agent talks, the two other networks are used to respectively sample the template and the word.

Note that the textual input given to the agent consists of the full dialogue history (without the "empty string" indicator) as we found it works better than giving only current utterances.
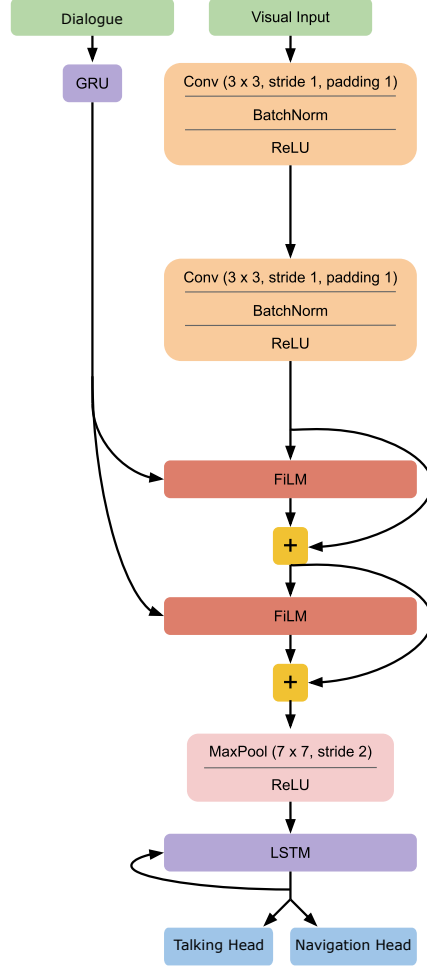
Figure 3: Our Multi-Headed PPO baseline DRL agent. Architecture visualization is a modified version of the one made by Hui et al. [2020]. We perform two modifications: 1) Instead of fixed instruction inputs our model is fed with NPC's language outputs (if the agent is near an NPC), and 2) We add a language action head, as our agent can both navigate and talk.

Exploration bonus   The exploration bonus we use is inspired by recent works in intrinsically motivated exploration [Pathak et al., 2017, Savinov et al., 2018, Tang et al., 2017]. These intrinsic rewards estimate the novelty of the currently observed state and add the novelty based bonus to the extrinsic reward. The novelty is estimated by counting various aspects of the state. We make our reward episodic by resetting the counts at the end of each episode.

In this work we study two different techniques for computing the exploration bonus (counting), and we use the one that was more suitable for a given environment. Which reward was used for which environment and the corresponding hyperparameters are visible in table 8. The two different techniques are: language-based and vision-based.

In the language-based intrinsic reward we count how many times was each utterance observed and compute an additional bonus based on the following equation:

$$r_{intr} = tanh(\frac{C}{(N(s_{lang}) + 1)^M) * T)})$$

(2)

, where M, C, and T are hyperparameters and $N(s_{lang})$ is the number of times the utterance s_{lang} was observed during this episode. In the current version of the environment the agent cannot hear his own utterances and the NPCs speak only when spoken to. Therefore, this exploration bonus can be seen as analogous to social influence [Jaques et al., 2019] in the language modality, as the reward is given upon *making the NPC respond*.

In the vision-based intrinsic reward, we reward the agent for seeing a new encoding. An encoding is the 4D representation of a grid (object_type, color, additional_information, orientation). At each step, a set of encountered encodings is created by removing the duplicates, and then the reward computed by the following equation:

$$r_{intr} = tanh(\sum_{e \in E(s)} \frac{C}{(N(e)+1)^M) * T})$$
(3)

, where M, C, and T are as in equation 2, $E(s)$ is a set of unique encodings visible in state s, and $N(e)$ is the number of times an encoding e was encountered.

These intrinsic rewards are a good example of biases that have to be discovered for training social agents.

C.2   Computational ressources

To perform our experiments, we used a slurm-based cluster. Producing our final results require to run 16 seeds of 3 different conditions on each of our 10 environments (7 environments and 3 modified environments for our case-studies), i.e. 480 seeds. Each of these experiments takes approximately 42 hours on one CPU and one 32GB Tesla V100 GPU (one GPU can serve 4 experiments in parallel), which amounts to 20160 CPU hours and 5040 GPU hours.