META-REFERENTIAL GAMES TO LEARN COMPOSITIONAL LEARNING BEHAVIOURS

Anonymous authors
Paper under double-blind review

ABSTRACT

Human beings use compositionality to generalise from past to novel experiences, assuming that past experiences can be decomposed into fundamental atomic components that can be recombined in novel ways. We frame this as the ability to learn to generalise compositionally, and refer to behaviours making use of this ability as compositional learning behaviours (CLBs). Learning CLBs requires the resolution of a binding problem (BP). While it is another feat of intelligence that human beings perform with ease, it is not the case for artificial agents. Thus, in order to build artificial agents able to collaborate with human beings, we develop a novel benchmark to investigate agents' abilities to exhibit CLBs by solving a domain-agnostic version of the BP. Taking inspiration from the Emergent Communication, we propose a meta-learning extension of referential games, entitled Meta-Referential Games, to support our benchmark, the Symbolic Behaviour Benchmark (S2B). Baseline results and error analysis show that the S2B is a compelling challenge that we hope will spur the research community to develop more capable artificial agents.

1 INTRODUCTION

Defining compositional behaviours (CBs) as "the ability to generalise from combinations of **trained-on** atomic components to novel re-combinations of those very same components", we can define compositional learning behaviours (CLBs) as "the ability to generalise **in an online fashion** from a few combinations of never-before-seen atomic components to novel re-combinations of those very same components". We employ the term online here to imply a few-shot learning context (Vinyals et al., 2016; Mishra et al., 2018) that demands that agents learn from, and then leverage some novel information, both over the course of a single lifespan, or episode, in our case of few-shot meta-RL (see Beck et al. (2023) for a review of meta-RL). Thus, in this paper, we investigate artificial agents' abilities for CLBs, which involve a few-shot learning aspect that is not present in CBs. For an intuitive comparison, benchmarks that tests for CBs alone ask whether trained-and-now-frozen agents can generalise to novel combinations of those same **familiar** atomic components they have been training on. This is different from what a benchmark testing for CLBs would ask, to wit, whether trained-and-now-frozen agents can generalise to novel combinations of **never-before-seen** atomic components, provided some warm-up exposition rounds to those novel atomic components. Such a benchmark instantiates a meta-learning challenge where tested agents can only be successful if they have learned to **learn to generalise compositionally**, whereas CB-testing benchmark, like SCAN (Lake & Baroni, 2018) and gSCAN (Ruis et al., 2020), can be solved by agents that have only learned to generalise compositionally. Thus, in order to prompt agents to acquire the **skill of learning to generalise compositionally**, we rely on the instantiation of a theoretically-infinite distribution of different atomic components, and the underlying semantic structure they belong to, so that at each training episode our agents are prompted with novel atomic components and novel underlying semantic structure.

Compositional Learning Behaviours as Symbolic Behaviours. Santoro et al. (2021) states that a symbolic entity does not exist in an objective sense but solely in relation to an "*interpreter who treats it as such*", and it ensues that there exists a set of behaviours, i.e. *symbolic behaviours*, that are consequences of agents engaging with symbols. Thus, in order to evaluate artificial agents in terms of their ability to collaborate with humans, we can use the presence or absence of symbolic behaviours. Among the different characteristic of symbolic behaviours, this work will primarily focus on the receptivity and constructivity aspects. Receptivity aspects amount to the ability to receive

new symbolic conventions in an online fashion. For instance, when a child introduces an adult to their toys' names, the adults are able to discriminate between those new names upon the next usage. Constructivity aspects amount to the ability to form new symbolic conventions in an online fashion. For instance, when facing novel situations that require collaborations, two human teammates can come up with novel referring expressions to easily discriminate between different events occurring. Both aspects refer to abilities that support collaboration. Thus, this paper develops a benchmark to evaluate agents' abilities in receptive and constructive behaviours, with a primary focus on CLBs.

Binding Problem & Meta-Learning. Following Greff et al. (2020), we refer to the binding problem (BP) as the challenges in "dynamically and flexibly bind[/re-use] information that is distributed throughout the [architecture]" of some artificial agents (modelled with artificial neural networks here). We note that there is an inherent BP that requires solving for agents to exhibit CLBs. Indeed, over the course of a single episode (as opposed to a whole training process, in the case of CBs), agents must dynamically identify/segregate the component values from the observation of multiple stimuli, timestep after timestep, and then bind/(re-)use/(re-)combine this information (hopefully stored in some memory component of their architecture) in order to respond correctly to novel stimuli.Solving the BP instantiated in such a context, i.e. re-using previously-acquired information in ways that serve the current situation, is another feat of intelligence that human beings perform with ease, on the contrary to current state-of-the-art artificial agents. Thus, our benchmark must emphasise testing agents' abilities to exhibit CLBs by solving a version of the BP. Moreover, we argue for a domain-agnostic BP, i.e. not grounded in a specific modality such as vision or audio, as doing so would limit the external validity of the test. We aim for as few assumptions as possible to be made about the nature of the BP we instantiate (Chollet, 2019). This is crucial to motivate the form of the stimuli we employ, and we will further detail this in Section 3.2.

Language Grounding & Emergence. In order to test the quality of some symbolic behaviours, our proposed benchmark needs to query the semantics that agents (*the interpreters*) may extract from their experience, and it must be able to do so in a referential fashion (e.g. being able to query to what extent a given experience is referred to as, for instance, 'the sight of a red tomato'), similarly to most language grounding benchmarks. Subsequently, acknowledging that the simplest form of collaboration is maybe the exchange of information, i.e. communication, via a given code, or language, we argue that the benchmark must therefore also allow agents to manipulate this code/language that they use to communicate. This property is known as the metalinguistic/reflexive function of languages (Jakobson, 1960). It is mainly investigated in the current deep learning era within the field of Emergent Communication (Lazaridou & Baroni (2020), and see Brandizzi (2023) and Denamganaï & Walker (2020a) for further reviews), via the use of variants of the referential games (RGs) (Lewis, 1969). Thus, we take inspiration from the RG framework, where (i) the language domain represents a semantic domain that can be probed and queried, and (ii) the reflexive function of language is indeed addressed. Then, in order to instantiate different BPs at each episode, we propose a meta-learning extension to RGs, entitled Meta-Referential Games, and use this framework to build our benchmark. It results in our proposed Symbolic Behaviour Benchmark (S2B), which has the potential to test for many aspects of symbolic behaviours.

After review of the background (Section 2) , we will present our contributions as follows: we propose the Symbolic Behaviour Benchmark to enables evaluation of symbolic behaviours in Section 3, presenting the Symbolic Continuous Stimulus (SCS) representation scheme which is able to instantiate a BP, on the contrary to common symbolic representations (Section 3.2), and our Meta-Referential Games framework, a meta-learning extension to RGs (Section 3.1) ; then we provide baseline results and error analysis in Section 4 for state-of-the-art RL agents and Large Language Models (LLMs - Brown et al. (2020), showing that our benchmark is a compelling challenge that we hope will spur the research community.

2 BACKGROUND

The first instance of an environment with a primary focus on efficient communication is the *signaling game* or *referential game* (RG) by Lewis (1969), where a speaker agent is asked to send a message to the listener agent, based on the *state/stimulus* of the world that it observed. The listener agent then acts upon the observed message by choosing one of the *actions* available to it. Both players' goals are aligned (it features *pure coordination/common interests*), with the aim of performing the

'best' *action* given the observed *state*. In the recent deep learning era, many variants of the RG have appeared (Lazaridou & Baroni, 2020). Following the nomenclature proposed in Denamganaï & Walker (2020b), Figure 1 illustrates in the general case a *discriminative* 2*-players / L-signal / N-round / K-distractors / descriptive / object-centric* variant, where the speaker receives a stimulus and communicates with the listener (up to N back-and-forth using messages of at most L tokens each), who additionally receives a set of $K + 1$ stimuli (potentially including a semantically-similar stimulus as the speaker, referred to as an object-centric stimulus). The task is for the listener to determine, via communication with the speaker, whether any of its observed stimuli match the speaker's. We highlight here features of RGs that will be relevant to how S2B is built, and then provide formalism used throughout the paper. The **number of communication rounds** N characterises (i) whether the listener agent can send messages back to the speaker agent and (ii) how many communication rounds can be expected before the listener agent is finally tasked to decide on an action. The basic (discriminative) *RG* is **stimulus-centric**, which assumes that both agents would be somehow embodied in the same body, and they are tasked to discriminate between given stimuli, that are the results of one single perception 'system'. On the other hand, Choi et al. (2018) introduced an **object-centric** variant which incorporates the issues that stem from the difference of embodiment (which has been later re-introduced under the name *Concept game* by Mu & Goodman (2021)). The agents must discriminate between objects (or scenes) independently of the viewpoint from which they may experience them. In the object-centric variant, the game is more about bridging the gap between each other's cognition rather than just finding a common language. The adjective 'object-centric' is used to qualify a stimulus that is different from another but actually present the same meaning (e.g. same object, but seen under a different viewpoint). Following the last communication round, the listener outputs a decision (D_i^L in Figure 2) about whether any of the stimulus it is observing matches the one (or a semantically similar one, in object-centric RGs) experienced by the speaker, and if so its action index must represent the index of the stimulus it identifies as being the same. The **descriptive** variant allows for none of the stimuli to be the same as the target one, therefore the action of index 0 is required for success. The agent's ability to make the correct decision over multiple RGs is referred to as RG accuracy.

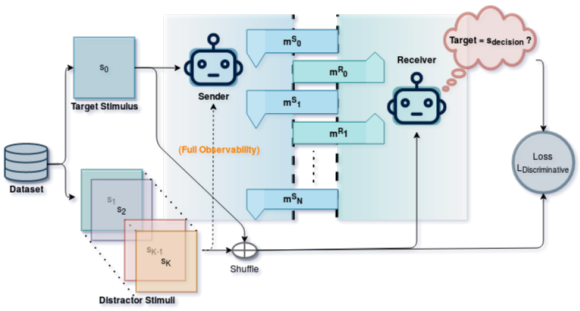


Figure 1: Illustration of a *discriminative* 2*-players / L-signal / N-round* variant of a *RG*.

Compositionality, Disentanglement & Systematicity. Compositionality is a phenomenon that human beings are able to identify and leverage thanks to the assumption that reality can be decomposed over a set of "disentangle[d,] underlying factors of variations" (Bengio, 2012), and our experience is a noisy, entangled translation of this factorised reality. This assumption is critical to the field of unsupervised learning of disentangled representations (Locatello et al., 2020) that aims to find "manifold learning algorithms" (Bengio, 2012), such as variational autoencoders (VAEs (Kingma & Welling, 2013)), with the particularity that the latent encoding space would consist of disentangled latent variables (see Higgins et al. (2018) for a formal definition). As a concept, compositionality has been the focus of many definition attempts. For instance, it can be defined as "the algebraic capacity to understand and produce novel combinations from known components"(Loula et al. (2018) referring to Montague (1970)) or as the property according to which "the meaning of a complex expression is a function of the meaning of its immediate syntactic parts and the way in which they are combined" (Krifka, 2001). Although difficult to define, the commmunity seems to agree on the fact that it would enable learning agents to exhibit systematic generalisation abilities (also referred to as combinatorial generalisation (Battaglia et al., 2018)). While often studied in relation to languages, it is usually defined with a focus on behaviours. In this paper, we will refer to (linguistic) compositionality when considering languages, and interchangeably compositional behaviours or systematicity to refer to "the ability to entertain a given thought implies the ability to entertain thoughts with semantically related contents"(Fodor & Pylyshyn, 1988).

Compositionality can be difficult to measure. Brighton & Kirby (2006)'s *topographic similarity* (**topsim**) which is acknowledged by the research community as the main quantitative metric (Lazaridou et al., 2018; Guo et al., 2019; Słowik et al., 2020; Chaabouni et al., 2020; Ren et al., 2020).

Recently, taking inspiration from disentanglement metrics, Chaabouni et al. (2020) proposed the **posdis** (positional disentanglement) and **bosdis** (bag-of-symbols disentanglement) metrics, that have been shown to be differently 'opinionated' when it comes to what kind of compositionality they capture. As hinted at by Choi et al. (2018); Chaabouni et al. (2020) and Dessi et al. (2021), compositionality and disentanglement appears to be two sides of the same coin, in as much as emergent languages are discrete and sequentially-constrained unsupervisedly-learned representations. In Section 3.2, we bridge further compositional language emergence and unsupervised learning of disentangled representations by asking *what would an ideally-disentangled latent space look like?* to build our proposed benchmark.

Richness of the Stimuli & Systematicity. Chaabouni et al. (2020) found that compositionality is not necessary to bring about systematicity, as shown by the fact that non-compositional languages wielded by symbolic (generative) RG players were enough to support success in zero-shot compositional tests (ZSCTs). They found that the emergence of a posdis-compositional language was a sufficient condition for systematicity to emerge. Finally, they found a necessary condition to foster systematicity, that we will refer to as richness of stimuli condition (Chaa-RSC). It was framed as (i) having a large stimulus space $|I| = {i_{val}}^{i_{attr}}$, where i_{attr} is the number of attributes/factor dimensions, and i_{val} is the number of possible values on each attribute/factor dimension, and (ii) making sure that it is densely sampled during training, in order to guarantee that different values on different factor dimensions have been experienced together. In a similar fashion, Hill et al. (2019) also propose a richness of stimuli condition (Hill-RSC) that was framed as a data augmentation-like regularizer caused by the egocentric viewpoint of the studied embodied agent. In effect, the diversity of viewpoint allowing the embodied agent to observe over many perspectives the same and unique semantical meaning allows a form of contrastive learning that promotes the agent's systematicity.

3 Symbolic Behaviour Benchmark

We present a version of the S2B that focuses on evaluating receptive and constructive behaviour traits[1]. This evaluation relies on a single task built around 2-players multi-agent RL (MARL) episodes. Each episode consists of a series of RGs (cf. lines 11 and 17 in Alg. 5 calling Alg. 3). We denote one such episode as a meta-RG and detail it in Section 3.1. Each RG in a meta-RG follows the formalism laid out in Section 2. The only difference is that both players speak simultaneously, rather than turn-by-turn. Consequently, they observe their partner's message upon the next RL step. Each RG consists of $N + 2$ RL steps, where N is the *number of communication rounds* available to the players (cf. Section 2).

At each RL step, each player observe a stimulus and a message coming from their partner. Throughout the first $N + 1$ steps, these stimuli remain constant. They are *object-centric*, and may be similar or different from one player to the other. We recall that the common goal of RG players is to figure out whether they observe similar stimuli or not. Stimuli are represented using the the Symbolic Continuous Stimulus (SCS) representation, which we detail in Section 3.2.

From those observations, each player acts from different actions spaces. Each player's action space is dependent on their role in the RG, as the speaker or the listener. In the first N steps, both players' action space only allow them to send messages to their partner. Then, at step $N + 1$, the listener player must decide whether they observe a similar stimulus than the speaker. Thus, the listener's action space only allows the decision-related actions. And, the speaker's action space only allows a *no-operation* (NO-OP) action. In practice, the environment provides the players with masks that identify valid actions. If the players still choose invalid actions, the environment simply ignores them.

Finally, after the listener has provided their decision, step $N + 2$ provides feedback to the listener player. This feedback consist of two elements. First, the environment reward is non-null. And, secondly, the listener's stimulus is the same that the speaker was observing throughout the current RG (cf. line 12 and 18 in Alg. 5). Note, that it is the exact stimulus rather than an object-centric sample. In Figure 4, we present SCS-represented stimuli, observed by a speaker over the course of a typical episode.

[1] HIDDEN_FOR_REVIEW_PURPOSE

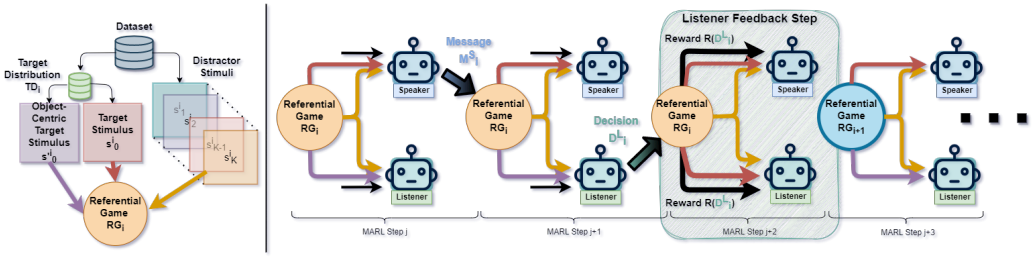


Figure 2: Left: Sampling of the necessary components to create the i-th RG (RG_i) of a meta-RG. The target stimulus (red) and the object-centric target stimulus (purple) are both sampled from the Target Distribution TD_i, a set of O different stimuli representing the same latent semantic meaning. The latter set and a set of K distractor stimuli (orange) are both sampled from a dataset of SCS-represented stimuli (**Dataset**), which is instantiated from the current episode's symbolic space, whose semantic structure is sampled out of the meta-distribution of available semantic structure over N_{dim}-dimensioned symbolic spaces. Right: Illustration of the resulting meta-RG with a focus on the i-th RG RG_i. The speaker agent receives at each step the target stimulus s_0^i and distractor stimuli $(s_k^i)_{k\in[1;K]}$, while the listener agent receives an object-centric version of the target stimulus s'^i_0 or a distractor stimulus (randomly sampled), and other distractor stimuli $(s_k^i)_{k\in[1;K]}$, with the exception of the **Listener Feedback step** where the listener agent receives feedback in the form of the exact target stimulus s_0^i. The Listener Feedback step takes place after the listener agent has provided a decision D_i^L about whether the target meaning is observed or not and in which stimuli is it instantiated, guided by the vocabulary-permutated message M_i^S from the speaker agent.

3.1 META-REFERENTIAL GAMES

A Meta-RG consists of a series of RGs, played over specific stimuli. These stimuli are sampled from a given N_{dim}-dimensioned symbolic space. To be successful, players must learn to communicate about stimuli using initially-ungrounded symbols. They must ground symbols into the symbolic space they observe. But each new episode brings about a new communication channel and a new symbolic space. The symbolic space is renewed by changing its semantic structure, which we detail further in Section 3.2. And the communication channel has its symbols being randomly permutated, as detailed below. Players of a Meta-RG must learn to adapt to new stimuli and new communication channels. This renewal mechanism, in a meta-learning way, defines a distribution of adaptation tasks. Each meta-RG evaluates the ability of the players to adapt.

The series of RGs that a Meta-RG consists of can be separated over two phases: a supporting and querying/ZSCT phase. The supporting phase helps players learn the new semantics of the symbolic space. Players must also agree on communication conventions by grounding the newly randomized symbols. They prepare for the querying/ZSCT phase, during which ZSCT-purposed RGs are played. They rely on target stimuli consisting of new combinations of the supporting stimuli's atomic components. Focusing on novel combinations of never-before-seen components, prior to the current episode, is the main advantage of Meta-RGs over RGs. Indeed, this focus allows evaluation of players' ability to learn to generalise compositionally.

Players that mastered the skill of **learning to generalise compositionally** will exhibit high RG accuracy during the querying/ZSCT phase. But those who lacks the (meta-)learning/adaptation part will not. Algorithms 4 and 5 (in Appendix A) contrast how a common RG differ from a meta-RG. Indeed, the supporting phase of a Meta-RG does not involve updating the parameters/weights of the learning agents. This is in keeping with the meta-learning framework of the few-shot learning kind. Its instantiation in the context of Meta-RG becomes striking when comparing the positions and dependencies of lines 21 in Alg. 5 and 6 in Alg. 4.

The supporting phase lasts until all possible atomic component values on each latent dimension have been shown for at least S shots (cf. lines $3-7$ in Alg. 5). Thus, it will amount to at least S different target stimuli being shown. That is to say at least S supporting-phase RGs. And, at most, there will actually be less supporting-phase RGs than the number of possible supporting-purposed stimuli in the current episode's symbolic space/dataset. This is due to the focus on familiarising players with atomic component values rather than stimuli themselves. On the otherhand, the querying/ZSCT phase

will present all the testing-purposed stimuli. We emphasise again that all RGs from both phases are played without the agents' parameters changing in-between. Indeed, learning CLBs involve agents adapting in an online/few-shot learning setting. The semantic structure of the symbolic space is randomly sampled at the beginning of each episode (cf. lines $2-3$ in Alg. 5). The reward function proposed to both agents is null except on the $N+2$-th step. It will be $+1$ if the listener decided correctly or, during the querying phase only, -2 if incorrect (cf. line 21 in Alg. 5).

Stimulus Representation. Meta-RG players must be able to deal with stimuli from N_{dim}-dimensioned symbolic spaces of varying semantic structures. Thus, it is necessary that stimulus representation shapes remain constant from one semantic structure to another. This is not the case for the common one-/multi-hot-encoded representation scheme. Thus, we propose the Symbolic Continuous Stimulus (SCS) representation scheme, detailed in Section 3.2. Thanks to its *shape invariance property*, once a number of latent/factor dimension N_{dim} is choosen, it allows generation of many different semantically structured symbolic spaces while maintaining a consistent stimulus representation shape. Figure 2 highlights the structure of an episode, and its reliance on differently semantically structured N_{dim}-dimensioned symbolic spaces.

Vocabulary Permutation. We remark that only changing the semantic structure of the symbolic space is not sufficient to force MARL agents to adapt in each episode. Indeed, they can learn to cheat by relying on an episode-invariant (and therefore independent of the instantiated semantic structure) emergent language (EL). This cheating EL consists of encoding the continuous values of the SCS representation like an analog-to-digital converter would. It would map a fine-enough partition of the SCS range onto a fixed vocabulary in a bijective fashion (see Appendix D for more details). Therefore, to prevent the MARL agents from relying on such a cheating EL, we employ a vocabulary permutation scheme (Cope & Schoots, 2021) that samples at the beginning of each episode a random permutation of the vocabulary symbols (cf. line 1 in Alg. 2). This approach is bears some similarity with the Other-Play algorithm from Hu et al. (2020).

Richness of the Stimulus. We further bridge the gap between Hill-RSC and Chaa-RSC by allowing the **number of object-centric samples** O and the **number of shots** S to be parameterized in the benchmark. S represents the minimal number of times any given atomic component value may be observed throughout the course of an episode. Intuitively, throughout their lifespan, an embodied observer may only observe a given component a limited number of times (e.g. considering the value 'blue', on the latent/factor dimension 'color', being observed once within a 'blue car' stimulus, and another time within a 'blue cup' stimulus). These parameters allow experimenters to account for both the Chaa-RSC's sampling density of the different stimulus components and Hill-RSC's diversity of viewpoints.

3.2 Symbolic Continuous Stimulus representation

Building about successes of the field of unsupervised learning of disentangled representations (Higgins et al., 2018), to the question *what would an ideally-disentangled latent space look like?*, we propose the Symbolic Continuous Stimulus (SCS) representation and provide numerical evidence of it in Appendix E.2. It is continuous and relying on Gaussian kernels, and it has the particularity of enabling the representation of stimuli sampled from differently semantically structured symbolic spaces while maintaining the same representation shape (later referred as the *shape invariance property*), as opposed to the one-/multi-hot encoded (OHE/MHE) vector representation commonly used when dealing with symbolic spaces.

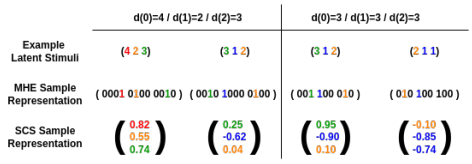


Figure 3: OHE/MHE and SCS representations of example latent stimuli for two differently-structured symbolic spaces with $N_{dim}=3$, i.e. on the left for $d(0)=4$, $d(1)=2$, $d(2)=3$, and on the right for $d(0)=3$, $d(1)=3$, $d(2)=3$. Note the shape invariance property of the SCS representation, as its shape remains unchanged by the change in semantic structure of the symbolic space, on the contrary to the OHE/MHE representations.

While the SCS representation is inspired by vectors sampled from VAE's latent spaces, this representation is not learned and is not aimed to help the agent performing its task. It is solely meant to make it possible to define a distribution over infinitely many semantic/symbolic spaces, while instantiating

a BP for the agent to resolve. Indeed, contrary to OHE/MHE representation, observation of one stimulus is not sufficient to derive the nature of the underlying semantic space that the current episode instantiates (cf. Figure 3 for comparison). Rather, it is only via a kernel density estimation on multiple samples (over multiple timesteps) that the semantic space's nature can be inferred, thus requiring the agent to segregated and (re)combine information that is distributed over multiple observations. In other words, the benchmark instantiates a domain-agnostic BP. We provide in Appendix E.1 some numerical evidence to the fact that the SCS representation differentiates itself from the OHE/MHE representation because it instantiates a BP. Deriving the SCS representation from an idealised VAE's latent encoding of stimuli of any domain makes it a domain-agnostic representation, which is an advantage compared to previous benchmark because domain-specific information can therefore not be leveraged to solve the benchmark.

In details, the semantic structure of an N_{dim}-dimensioned symbolic space is the tuple $(d(i))_{i\in[1;N_{dim}]}$ where N_{dim} is the number of latent/factor dimensions, $d(i)$ is the **number of possible symbolic values** for each latent/factor dimension i. Stimuli in the SCS representation are vectors sampled from the continuous space $[-1,+1]^{N_{dim}}$. In comparison, stimuli in the OHE/MHE representation are vectors from the discrete space $\{0,1\}^{d_{OHE}}$ where $d_{OHE} = \Sigma_{i=1}^{N_{dim}} d(i)$ depends on the $d(i)$'s. Note that SCS-represented stimuli have a shape that does not depend on the $d(i)$'s values, this is the *shape invariance property* of the SCS representation (see Figure 3 for illustration).

In the SCS representation, the $d(i)$'s do not shape the stimuli but only the semantic structure, i.e. representation and semantics are disentangled from each other. The $d(i)$'s shape the semantic by enforcing, for each factor dimension i, a partitionaing of the $[-1,+1]$ range into $d(i)$ value sections. Each partition corresponds to one of the $d(i)$ symbolic values available on the i-th factor dimension. Having explained how to build the SCS representation sampling space, we now describe how to sample stimuli from it. It starts with instantiating a specific latent meaning/symbol, embodied by latent values $l(i)$ on each factor dimension i, such that $l(i) \in [1; d(i)]$. Then, the i-th entry of the stimulus is populated with a sample from a corresponding Gaussian distribution over the $l(i)$-th partition of the $[-1,+1$ range. It is denoted as $g_{l(i)} \sim \mathcal{N}(\mu_{l(i)}, \sigma_{l(i)})$, where $\mu_{l(i)}$ is the mean of the Gaussian distribution, uniformly sampled to fall within the range of the $l(i)$-th partition, and $\sigma_{l(i)}$ is the standard deviation of the Gaussian distribution, uniformly sampled over the range $[\frac{2}{12d(i)}, \frac{2}{6d(i)}]$. $\mu_{l(i)}$ and $\sigma_{l(i)}$ are sampled in order to guarantee (i) that the scale of the Gaussian distribution is large enough, but (ii) not larger than the size of the partition section it should fit in. Figure 4 shows an example of such instantiation of the different Gaussian distributions over each factor dimensions' $[-1,+1]$ range.

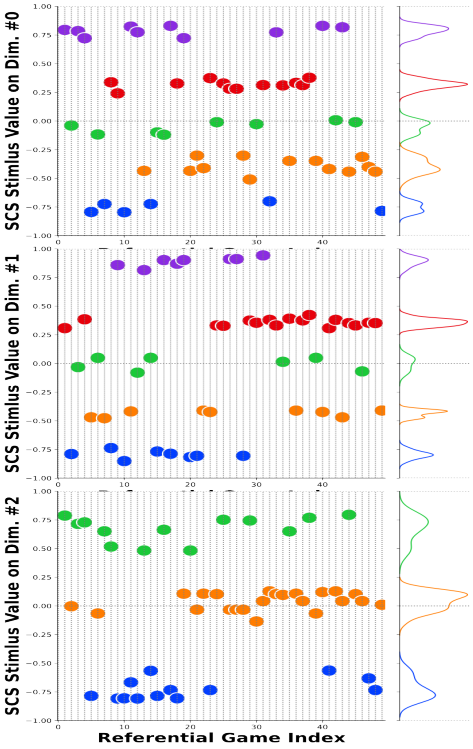


Figure 4: Visualisation of the SCS-represented stimuli (column) observed by the speaker agent at each RG over the course of one meta-RG, with $N_{dim} = 3$ and $d(0) = 5$, $d(1) = 5$, $d(2) = 3$. The supporting phase lasted for 19 RGs. For each factor dimension $i \in [0; 2]$, we present on the right side of each plot the kernel density estimations of the Gaussian kernels $\mathcal{N}(\mu_{l(i)}, \sigma_{l(i)})$ of each latent value available on that factor dimension $l(i) \in [1; d(i)]$. Colours of dots, used to represent the sampled value $g_{l(i)}$, imply the latent value $l(i)$'s Gaussian kernel from which said continuous value was sampled. For each factor dimension, there is no overlap between the different latent values' Gaussian kernels.

4 Experiments

Agent Architecture. The architectures of the RL agents that we consider are detailed in Appendix C. Optimization is performed via an R2D2 algorithm(Kapturowski et al., 2018) augmented with both the

Value Decomposition Network (Sunehag et al., 2017) and the *Simplified Action Decoder* approach (Hu & Foerster, 2019). As preliminary results showed poor performance, we follow Hill et al. (2020) and add an auxiliary reconstruction task to promote agents learning to use their core memory module. It consists of a mean squared-error between the stimuli observed at a given time step and a prediction conditioned on the current state of the core memory module after processing the current stimuli.

4.1 Learning CLBs is Out-Of-Reach to State-of-the-Art MARL

Playing a meta-RG, the speaker aims at each episode to make emerge a new language (constructivity) and the listener aims to acquire it (receptivity) as fast as possible, before the querying-phase of the episode comes around. Critically, we assume that both agents must perform in accordance with the principles of CLBs as it is the only resolution approach. Indeed, there is no success without a generalizing and easy-to-learn EL, or, in other words, a (linguistically) compositional EL (Brighton & Kirby, 2001; Brighton, 2002). Thus, we investigate whether agents are able to coordinate to learn to perform CLBs from scratch, which is tantamount to learning receptivity and constructivity aspects of CLBs in parallel.

Table 1: Meta-RG ZSCT and Ease-of-Acquisition (EoA) ZSCT accuracies and linguistic compositionality measures ($\% \pm$ s.t.d.) for the multi-agent context after a sampling budget of $500k$. The last column shows linguist results when evaluating the Posdis-Speaker (PS).

	Shots		PS
Metric	$S=1$	$S=2$	
$Acc_{\text{ZSCT}} \uparrow$	53.6 ± 4.7	51.6 ± 2.2	N/A
$Acc_{\text{EoA}} \uparrow$	50.6 ± 8.8	50.6 ± 5.8	N/A
topsim $\uparrow$	29.6 ± 16.8	21.3 ± 16.6	96.7 ± 0
posdis $\uparrow$	23.7 ± 20.8	13.8 ± 12.8	92.0 ± 0
bosdis $\uparrow$	25.6 ± 22.9	19.1 ± 17.5	11.6 ± 0

Evaluation & Results. We report the performance and compositionality of the behaviours in the multi-agent context in Table 1, on 3 random seeds of an LSTM-based model in the task with $N_{dim} = 3$, $V_{min} = 2, V_{max} = 5$, $O = 4$, and $S = 1$ or 2. As we assume no success without emergence of a (linguistically) compositional language, we measure the linguistic compositionality profile of the emerging languages by, firstly, freezing the speaker agent's internal state (i.e. LSTM's hidden and cell states) at the end of an episode and query what would be its subsequent utterances for all stimuli in the latest episode's dataset (see Figure 2), and then compute the different compositionality metrics on this collection of utterances. We compare the compositionality profile of the ELs to that of a compositional language, in the sense of the **posdis** compositionality metric (Chaabouni et al., 2020) (see Figure 7 and Table 6 in Appendix C.2). This language is produced by a fixed, rule-based agent that we will refer to as the Posdis-Speaker (PS). Similarly, after the latest episode ends and the speaker agent's internal state is frozen, we evaluate the EoA of the emerging languages by training a **new, non-meta/common listener agent** for 512 epochs on the latest episode's dataset with the frozen speaker agent using a *descriptive-only/object-centric* **common** RG and report its ZSCT accuracy (see Algorithm 3).Table 1 shows Acc_{ZSCT} being around chance-level (50%), thus the meta-RL agents fail to coordinate together, despite the simplicity of the setting, meaning that learning CLBs from scratch is currently out-of-reach to state-of-the-art MARL agents, and therefore show the importance of our benchmark. As the linguistic compositionality measures are very low compared to the PS agent, and since the chance-leveled Acc_{EoA} implies that the emerging languages are not easy to learn, it leads us to think that the poor MARL performance is due to the lack of compositional language emergence.

4.2 Single-Agent Listener-Focused RL Context

Seeing that the multi-agent benchmark is out of reach to state-of-the-art cooperative MARL agents, we investigate a simplification along two axises. Firstly, we simplify to a single-agent RL problem by instantiating a fixed, rule-based agent as the speaker, which should remove any issues related to agents learning in parallel to coordinate. Secondly, we use the Posdis-Speaker agent, which should remove any issues related to the emergence of assumed-necessary compositional languages, which corresponds to the constructivity aspects of CLBs. These simplifications allow us to focus our investigation on the receptivity aspects of CLBs, which relates to the ability from the listener agent to acquire and leverage a newly-encountered compositional language at each episode.

4.2.1 Symbol-Manipulation Induction Biases are Valuable

Table 2: Meta-RG ZSCT accuracies ($\% \pm$ s.t.d.).

	LSTM	ESBN	DCEM
$Acc_{\text{ZSCT}} \uparrow$	86.0 ± 0.1	89.4 ± 2.8	81.9 ± 0.6

Firstly, in the simplest setting of $O = 1$ and $S = 1$, we hypothesise that symbol-manipulation biases, such as efficient memory-addressing mechanism (e.g. attention) and greater algorithm-learning abilities (e.g. explicit memory), should improve performance, and propose to test the Emergent Symbol Binding Network (ESBN) (Webb et al., 2020), the Dual-Coding Episodic Memory (DCEM) (Hill et al., 2020) and compare to baseline LSTM (Hochreiter & Schmidhuber, 1997).

Evaluation & Results. We report in Table 2 the final ZSCT accuracies in the setting of $N_{dim} = 3$, $V_{min} = 2$, $V_{max} = 3$, with a sampling budget of $10M$ observations and 3 random seeds per architecture. LSTM performing better than DCEM is presumably due to the difficulty of the latter in learning to use its complex memory scheme (preliminary experiments involving a Differentiable Neural Computer (DNC - Graves et al. (2016)), on which the DCEM is built, show it struggling to learn to use its memory compared to LSTM - cf Appendix E.3). On the other hand, we interpret the best performance of the ESBN as being due to it being built over the LSTM, thus allowing its complex memory scheme to be bypassed until it becomes useful. We validate our hypothesis but carry on experimenting with the simpler LSTM model in order to facilitate analysis.

4.3 Receptivity Aspects of CLBs Can Be Learned Sub-Optimally

Table 3: Meta-RG ZSCT accuracies ($\% \pm$ s.t.d.).

	Shots		
Samples	$S = 1$	$S = 2$	$S = 4$
$O = 1$	62.2 ± 3.7	73.5 ± 2.4	75.0 ± 2.3
$O = 4$	62.8 ± 0.8	62.6 ± 1.7	60.2 ± 2.2
$O = 16$	64.9 ± 1.7	62.0 ± 2.0	61.8 ± 2.1

Hypotheses. The SCS representation instantiates a BP even when $O = 1$ (cf. Appendix E.1), and we suppose that when O increases the BP's complexity increases.Thus, it would stand to reason to expect performance to decrease when O increases (Hyp. 1). On the other hand, we would expect that increasing S would provide the learning agent with a denser sampling (in order to fulfill Chaa-RSC (ii)) , and thus performance is expected to increase as S increases (Hyp. 2). Indeed, increasing S amounts to giving more opportunities for the agents to estimate each Gaussian, thus relaxing the instantiated BP's complexity.

Evaluation & Results. We report in table 3 ZSCT accuracies on LSTM-based models (6 random seeds per settings) with $N_{dim} = 3$ and $V_{min} = 2, V_{max} = 5$. The chance threshold is 50%. When $S = 1$, increasing O is surprisingly correlated with non-significant increases in performance/systematicity. On the otherhand, when $S > 1$, accuracy distributions stay similar or decrease while O increases. Thus, overall, Hyp. 1 tends to be validated. Regarding Hyp. 2, when $O = 1$, increasing S (and with it the density of the sampling of the input space, i.e. Chaa-RSC (ii)) correlates with increases in systematicity. Thus, despite the difference of settings between common RG, in Chaabouni et al. (2020), and meta-RG here, we retrieve a similar result that Chaa-RSC promotes systematicity. On the other hand, our results show a statistically significant distinction between BPs of complexity associated with $O > 1$ and those associated with $O = 1$. Indeed, when $O > 1$, our results contradict Hyp.2 since accuracy distributions remain the same or decrease when S increases. Acknowledging the LSTMs' notorious difficulty with integrating/binding information from past to present inputs over long dependencies, we explain these results based on the fact that increasing S also increases the length of each RL episode, thus the 'algorithm' learned by LSTM-based agents might fail to adequately estimate Gaussian kernel densities associated with each component value.

4.4 LLMs perform below chance-level on S2B as Listener

Hypotheses. With LLMs being trained on curated human conversations, we investigate whether they have acquired skills in terms of receptivity aspects of CLBs. Thus, we test them without fine-tuning in the same listener-focused RL setting as in Section 4.3, with the same set of hypotheses. Our prompts are presented in Appendix B, but note that in this experiment we only make use of the listener prompt, as the speaker agent is played by our PS rule-based agent.

Evaluation & Results. We report in table 4 ZSCT accuracies on Mixtral-8x7B-Instruct-v0.1 (Mixtral) and OpenAI GPT-4o-mini (GPT) models, with $N_{dim} = 3$ and $V_{min} = 2, V_{max} = 5$. Over all contexts ($O = 1$ or 4 and $S = 1$ or 2), we observe poor performance (below chance level at 50%) from both

tested LLMs, thus providing very clear signal that the benchmark is challenging, even in this simpler listener-focused RL setting, with low (and therefore simpler) values of (N_dim, V_min, V_max).

In more details, firstly, we observe statistically non-significant increases of performance when S is increased from 1 to 2, for both tested LLMs, which comfort Chaa-RSC but the lack of statistical significance prevent us from drawing any conclusion. Then, as O is increased from 1 to 4, we observe a surprising increase of performance, especially marked for GPT-4o-mini, albeit far from being statistically non-significant.

Table 4: Meta-RG ZSCT accuracies (mean $\% \pm$ s.t.d.) for Mixtral-8x7B-Instruct-v0.1 (Mixtral) and OpenAI GPT-4o-mini (GPT), with $N_{\text{dim}} = 3$, $V_{min} = 2$, $V_{\text{max}} = 5$, $O = 1$ and $S = 1$ or 2. Evaluation is performed over 5 seeds per setting and 64 Meta-RG episodes per seed, using the HuggingFace Text-Generation Inference API or OpenAI API, with Structured Outputs.

	$O = 1$		$O = 4$	
	$S = 1$	$S = 2$	$S = 1$	$S = 2$
Mixtral	45.6 ± 10.5	49.3 ± 13.7	48.6 ± 17.6	49.9 ± 7.9
GPT	33.1 ± 11.4	36.8 ± 12.7	39.8 ± 11.6	42.9 ± 3.8

We assume that these below-chance level results are, at least in part, caused by the well-studied limitation of LLMs when dealing with numerical values (Lee et al., 2023; Shen et al., 2023). We leave it to future works to investigate whether enhanced embeddings of numerical data (e.g. McLeish et al. (2024); Schwartz et al. (2024)) enable greater performance.

5 Discussion

Compositional Behaviours vs CLBs. The learning of compositional behaviours (CBs) is one of the central study in language grounding with benchmarks like SCAN (Lake & Baroni, 2018) and gSCAN (Ruis et al., 2020), as well as in the subfield of Emergent Communication (see Brandizzi (2023); Boldt & Mortensen (2023) for reviews), but none investigates nor allow testing for CLBs. Thus, our benchmark aims to fill in this gap. At a high level, SCAN and gSCAN are build to evaluate compositional behaviours (CBs) only, because the (supervised-learning) agents tackling those benchmarks are only facing a single, fixed underlying semantic structure. Thus, they get to learn what atomic components compose this single, fixed underlying semantic structure over multiple 'lives' (over multiple update iterations of the agent's parameter weights). For an intuitive comparison, SCAN and gSCAN benchmarks ask whether trained-and-now-frozen agents can generalise to novel combinations of **those same familiar atomic components they have been training on**. This is different from our proposed S2B, which intuitively asks whether trained-and-now-frozen agents can generalise to novel combinations of **never-before-seen atomic components, provided some warm-up exposition rounds to those atomic components**. Warm-up exposition rounds do not involve update of the agent's parameters. Thus, S2B instantiates meta-learning challenges where the agents can only be successful if they have learned to **learn to generalise compositionally**, whereas SCAN and gSCAN can be solved by agents that have only learned to **generalise compositionally**. Indeed, S2B instantiates a theoretically-infinite distribution of different atomic components by way of controlling the underlying semantic structure they belong to. Contrary to SCAN and gSCAN, agents tackling the S2B are facing an infinity of always-changing underlying semantic structure. S2B is a meta-learning benchmark, whereas SCAN and gSCAN are not, in principle.

That being said, SCAN and gSCAN can be updated to become meta-learning benchmarks, which is what Lake (2019) and Lake & Baroni (2023) did to train their agents. Indeed, without making the nuance, Lake (2019) and Lake & Baroni (2023) actually use CLBs as a training paradigm, where a meta-learning extension of the sequence-to-sequence learning setting (i.e. CLB training) is shown to enable human-like systematic CBs at test-time. Contrary to our work, they evaluate AI's abilities towards SCAN-specific CBs after SCAN-specific CLBs training. We propose to go further because we propose to train for CLBs and test for CLBs too, because we argue that CBs are actually not useful in open-ended contexts. We refer to open-ended contexts as real-world situations where agents would encounter more diverse semantic structure than just the single one that they have been trained on. We argue that CLBs are very useful in open-ended contexts. Moreover, even though Lake (2019) and Lake & Baroni (2023) extended SCAN to be used as a meta-learning benchmark, we argue that there is a missing element, which is the instantiation of a novel binding problem (BP) in each task. Indeed, their extension made use of an OHE/MHE representation which we have argued in Section 3.2 and

Appendix E.1 that it does not instantiate a BP, contrary to the SCS representation which we use in our proposed S2B. Given their demonstration of the potential of CLBs, we leverage our proposed Meta-RG framework to propose a domain-agnostic CLB-focused benchmark for evaluation of CLBs abilities themselves, in order to address novel research questions around CLBs.

Symbolic Behaviours & Binding Problem. Following Santoro et al. (2021)'s definition of symbolic behaviours, our benchmark is the first specifically-principled benchmark to evaluate systematically artificial agents's abilities towards any symbolic behaviours. Similarly, while most challenging benchmark instantiates a version of the BP, as described by Greff et al. (2020), there is currently no principled benchmark that specifically investigates whether BP can be solved by artificial agents. Thus, not only does our benchmark fill that other gap, but it also instantiate a domain-agnostic version of the BP, which is critical in order to ascertain the external validity of conclusions that may be drawn from it. Indeed, domain-agnosticity guards us against confounders that could make the task solvable without fully solving the BP, e.g. by gaming some domain-specific aspects (Chollet, 2019).

Limitations. Our experiments only evaluated state-of-the-art RL models and LLMs in the simplest configuration of our benchmark. We leave it to future works to investigate more complex configurations and evaluate other classes of models, such as neuro-symbolic models (Yu et al., 2023) or LLMs augmented with prompting methods that further reasoning abilities (Wei et al., 2022) and/or abilities to experiment (Lampinen et al., 2022) and correct themselves (Shinn et al., 2023).

In summary, we have proposed a novel benchmark to investigate artificial agents abilities at learning CLBs, by casting the problem of learning CLBs as a meta-reinforcement learning problem. It uses our proposed extension to RGs, entitled Meta-Referential Games, which contains an instantiation of a domain-agnostic BP. We provided baseline results for both the multi-agent tasks and the single-agent listener-focused tasks of learning CLBs in the context of our proposed benchmark. Our analysis of the behaviours in the multi-agent context highlighted the complexity for the speaker agent to invent a compositional language. But, when the language is already compositional, then a learning listener is able to acquire it and coordinate, albeit sub-optimally, with a rule-based speaker, in some of the simplest settings of our benchmark. Symbol-manipulation induction biases were found to be valuable, but, overall, our results show that our proposed benchmark is currently out of reach for current state-of-the-art artificial agents, as further exemplified by LLMs performing even below chance-level when instantiated as a listener and paired with our rule-based speaker. Thus, we hope our benchmark will spur the research community towards developing more capable artificial agents.

REFERENCES

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. 2018. URL https://arxiv.org/pdf/1806.01261.pdf.

Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*, 2023.

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, 27:17–37, 2012.

Brendon Boldt and David R Mortensen. A review of the applications of deep learning-based emergent communication. *Transactions on Machine Learning Research*, 2023.

Nicolo' Brandizzi. Towards more human-like AI communication: A review of emergent communication research. August 2023.

Henry Brighton. Compositional syntax from cultural transmission. *MIT Press*, Artificial, 2002. URL https://www.mitpressjournals.org/doi/abs/10.1162/106454602753694756.

Henry Brighton and Simon Kirby. The survival of the smallest: Stability conditions for the cultural evolution of compositional language. In *European Conference on Artificial Life*, pp. 592–601. Springer, 2001.

Henry Brighton and Simon Kirby. Understanding Linguistic Evolution by Visualizing the Emergence of Topographic Mappings. *Artificial Life*, 12(2):229–242, jan 2006. ISSN 1064-5462. doi: 10.1162/artl.2006.12.2.229. URL http://www.mitpressjournals.org/doi/10.1162/artl.2006.12.2.229.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and Generalization in Emergent Languages. apr 2020. URL http://arxiv.org/abs/2004.09124.

Ricky T Q Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in VAEs, 2018.

Edward Choi, Angeliki Lazaridou, and Nando de Freitas. Compositional Obverter Communication Learning From Raw Visual Input. apr 2018. URL http://arxiv.org/abs/1804.02341.

François Chollet. On the Measure of Intelligence. Technical report, 2019.

Dylan Cope and Nandi Schoots. Learning to communicate with strangers via channel randomisation methods. *arXiv preprint arXiv:2104.09557*, 2021.

Kevin Denamganaï and James A. Walker. Referentialgym: A nomenclature and framework for language emergence & grounding in (visual) referential games. *4th NeurIPS Workshop on Emergent Communication*, 2020a.

Kevin Denamganaï and James Alfred Walker. Referentialgym: A framework for language emergence & grounding in (visual) referential games. *4th NeurIPS Workshop on Emergent Communication*, 2020b.

Roberto Dessi, Eugene Kharitonov, and Marco Baroni. Interpretable agent communication from scratch (with a generic visual processor emerging on the side). May 2021.

Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626): 471–476, 2016.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

Shangmin Guo, Yi Ren, Serhii Havrylov, Stella Frank, Ivan Titov, and Kenny Smith. The emergence of compositional languages for numeric concepts through iterated learning in neural agents. *arXiv preprint arXiv:1910.05291*, 2019.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a Definition of Disentangled Representations. dec 2018. URL `http://arxiv.org/abs/1812.02230`.

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. October 2019.

Felix Hill, Olivier Tieleman, Tamara von Glehn, Nathaniel Wong, Hamza Merzic, and Stephen Clark DeepMind. Grounded language learning fast and slow. Technical report, 2020.

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

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*, 2018.

Hengyuan Hu and Jakob N Foerster. Simplified action decoder for deep multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2019.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.

Roman Jakobson. Linguistics and poetics. In *Style in language*, pp. 350–377. MA: MIT Press, 1960.

Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *International conference on learning representations*, 2018.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

D P Kingma and M Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Manfred Krifka. Compositionality. *The MIT encyclopedia of the cognitive sciences*, pp. 152–153, 2001.

Brenden M Lake. Compositional generalization through meta sequence-to-sequence learning. *Advances in neural information processing systems*, 32, 2019.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *35th International Conference on Machine Learning, ICML 2018*, 7:4487–4499, oct 2018. URL `http://arxiv.org/abs/1711.00350`.

Brenden M Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, pp. 1–7, 2023.

Andrew K Lampinen, Nicholas Roy, Ishita Dasgupta, Stephanie CY Chan, Allison Tam, James Mcclelland, Chen Yan, Adam Santoro, Neil C Rabinowitz, Jane Wang, et al. Tell me why! explanations support learning relational and causal structure. In *International Conference on Machine Learning*, pp. 11868–11890. PMLR, 2022.

Angeliki Lazaridou and Marco Baroni. Emergent Multi-Agent communication in the deep learning era. June 2020.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of Linguistic Communication from Referential Games with Symbolic and Pixel Input. apr 2018. URL http://arxiv.org/abs/1804.03984.

Nayoung Lee, Kartik Sreenivasan, Jason D Lee, Kangwook Lee, and Dimitris Papailiopoulos. Teaching arithmetic to small transformers. *arXiv preprint arXiv:2307.03381*, 2023.

David Lewis. Convention: A philosophical study. 1969.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. A sober look at the unsupervised learning of disentangled representations and their evaluation. October 2020.

João Loula, Marco Baroni, and Brenden M. Lake. Rearranging the Familiar: Testing Compositional Generalization in Recurrent Networks. jul 2018. URL http://arxiv.org/abs/1807.07545.

Sean McLeish, Arpit Bansal, Alex Stein, Neel Jain, John Kirchenbauer, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, Jonas Geiping, Avi Schwarzschild, et al. Transformers can do arithmetic with the right embeddings. *arXiv preprint arXiv:2405.17399*, 2024.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Richard Montague. Universal grammar. *Theoria*, 36(3):373–398, 1970.

Jesse Mu and Noah Goodman. Emergent communication of generalizations. *Advances in Neural Information Processing Systems*, 34:17994–18007, 2021.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. Compositional Languages Emerge in a Neural Iterated Learning Model. feb 2020. URL http://arxiv.org/abs/2002.01365.

Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the F-Statistic loss, 2018.

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. A benchmark for systematic generalization in grounded language understanding. March 2020.

Adam Santoro, Andrew Lampinen, Kory Mathewson, Timothy Lillicrap, and David Raposo. Symbolic behaviour in artificial intelligence. *arXiv preprint arXiv:2102.03406*, 2021.

Eli Schwartz, Leshem Choshen, Joseph Shtok, Sivan Doveh, Leonid Karlinsky, and Assaf Arbelle. Numerologic: Number encoding for enhanced llms' numerical reasoning. *arXiv preprint arXiv:2404.00459*, 2024.

Ruoqi Shen, Sébastien Bubeck, Ronen Eldan, Yin Tat Lee, Yuanzhi Li, and Yi Zhang. Positional description matters for transformers arithmetic. *arXiv preprint arXiv:2311.14737*, 2023.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning.(2023). *arXiv preprint cs.AI/2303.11366*, 2023.

Agnieszka Słowik, Abhinav Gupta, William L. Hamilton, Mateja Jamnik, Sean B. Holden, and Christopher Pal. Exploring Structural Inductive Biases in Emergent Communication. feb 2020. URL http://arxiv.org/abs/2002.01335.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. volume 29, 2016.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pp. 1995–2003. PMLR, 2016.

Taylor Whittington Webb, Ishan Sinha, and Jonathan Cohen. Emergent symbols through binding in external memory. In *International Conference on Learning Representations*, 2020.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Dongran Yu, Bo Yang, Dayou Liu, Hui Wang, and Shirui Pan. A survey on neural-symbolic learning systems. *Neural Networks*, 2023.

A On Algorithmic Details of Meta-Referential Games

In this section, we detail algorithmically how Meta-Referential Games differ from common RGs. We start by presenting in Algorithm 4 an overview of the common RGs, taking place inside a common supervised learning loop, where we highlight the following:

(i) preparation of the data on which the referential game is played (highlighted in green),

(ii) elements pertaining to the **supervised learning loop** (highlighted in red).

Helper functions are detailed in Algorithm 1, 2 and 3. Next, we can now show in greater and contrastive details the Meta-Referential Game algorithm in Algorithm 5, where we highlight the following:

(i) preparation of the data on which the referential game is played (highlighted in green),

(ii) elements pertaining to the **meta-learning loop** (highlighted in blue).

(iii) elements pertaining to setup of a Meta-Referential Game (highlighted in red).

Algorithm 1: Helper function : DataPrep

Given :

- a target stimuli s_0,
- a dataset of stimuli Dataset,
- O : Number of Object-Centric samples in each Target Distribution over stimuli $TD(\cdot)$.
- K : Number of distractor stimuli to provide to the listener agent.
- FullObs : Boolean defining whether the speaker agent has full (or partial) observation.
- DescrRatio : Descriptive ratio in the range $[0, 1]$ defining how often the listener agent is observing the same semantic as the speaker agent.

$s'_0, D^{Target} \leftarrow s_0, 0$;
if *random*$(0, 1) > DescrRatio$ **then**
  `/* Exclude target stimulus from listener's observation: */`
  $s'_0 \sim$ Dataset $- TD(s_0)$;
  $D^{Target} \leftarrow K + 1$;
end
else if $O > 1$ **then**
  Sample an Object-Centric distractor $s'_0 \sim TD(s_0)$;
end
Sample K distractor stimuli from Dataset $- TD(s_0)$: $(s_i)_{i\in[1,K]} \sim$ Dataset $- TD(s_0)$;
$Obs_{\text{Speaker}} \leftarrow \{s_0\}$; **if** *FullObs* **then**
  $Obs_{\text{Speaker}} \leftarrow \{s_0\} \cup \{s_i | \forall i \in [1, K]\}$;
end
$Obs_{\text{Listener}} \leftarrow \{s'_0\} \cup \{s_i | \forall i \in [1, K]\}$;
`/* Shuffle listener observations and update index of target decision: */`
$Obs_{\text{Listener}}, D^{Target} \leftarrow Shuffle(Obs_{\text{Listener}}, D^{Target})$;
Output : $Obs_{\text{Speaker}}, Obs_{\text{Listener}}, D^{Target}$;

Algorithm 2: Helper function : MetaRGDatasetPreparation

Given :

- V : Vocabulary (finite set of tokens available),
- N_{dim} : Number of attribute/factor dimensions in the symbolic spaces,
- V_{min} : Minimum number of possible values on each attribute/factor dimensions in the symbolic spaces,
- V_{max} : Maximum number of possible values on each attribute/factor dimensions in the symbolic spaces,

Initialise random permutation of vocabulary: $V' \leftarrow RandomPerm(V)$
Sample semantic structure: $(d(i))_{i\in[1,N_{\text{dim}}]} \sim \mathcal{U}(V_{min}; V_{max})^{N_{\text{dim}}}$;
Generate symbolic space/dataset $D((d(i))_{i\in[1,N_{\text{dim}}]})$;
Split dataset into supporting set D^{support} and querying set D^{query} ($((d(i))_{i\in[1,N_{\text{dim}}]})$ is omitted for readability);
Output : $V', D((d(i))_{i\in[1,N_{\text{dim}}]}), D^{\text{support}}, D^{\text{query}}$;

Algorithm 3: Helper function : PlayRG

Given :

- Speaker and Listener agents,
- Set of speaker observations Obs_{Speaker},
- Set of listener observations Obs_{Listener},
- N : Number of communication rounds to play,
- L : Maximum length of each message,
- V : Vocabulary (finite set of tokens available),

Compute message $M^S = \text{Speaker}(Obs_{\text{Speaker}}|\emptyset)$;
Initialise Communication Channel History: CommH $\leftarrow [M^S]$;
for *round* $= 0, N$ **do**
  Compute Listener's reply $M^L_{\text{round}}, \_ = \text{Listener}(Obs_{\text{Listener}}|\text{CommH})$;
  CommH $\leftarrow$ CommH $+ [M^L_{\text{round}}]$;
  Compute Speaker's reply $M^S_{\text{round}} = \text{Speaker}(Obs_{\text{Speaker}}|\text{CommH})$;
  CommH $\leftarrow$ CommH $+ [M^S_{\text{round}}]$;
end
Compute listener decision $\_, D^L = \text{Listener}(Obs_{\text{Listener}}|\text{CommH})$;
Output : Listener's decision D^L, Communication Channel History CommH;

Algorithm 4: Common Referential Game inside a Common Supervised Learning Loop

Given :

- a dataset of stimuli $Dataset$,
- a set of hyperparameters defining the RG:
  - O : Number of Object-Centric samples in each Target Distribution over stimuli $TD(\cdot)$.
  - N : Number of communication rounds to play.
  - L : Maximum length of each message.
  - V : Vocabulary (finite set of tokens available).
  - K : Number of distractor stimuli to provide to the listener agent.
  - FullObs : Boolean defining whether the speaker agent has full (or partial) observation.
  - DescrRatio : Descriptive ratio in the range $[0, 1]$ defining how often the listener agent is observing the same semantic as the speaker agent.
  - $\mathcal{L}$: Loss function to use in the agents update.

Initialize :

- Speaker$(\cdot)$ and Listener$(\cdot)$ agents.

Systematically split $Dataset$ into training and testing dataset, D^{train} and D^{test};
for $epoch = 1, N_{epoch}$ **do**
  for *target stimulus* $s_0 \in D^{train}$ **do**
    /* **Preparation of observations and target decision:** */
    $Obs_{\text{Speaker}}, Obs_{\text{Listener}}, D^{Target} \leftarrow DataPrep(\text{Dataset}, s_0, O, K, \text{FullObs}, \text{DescrRatio})$
    /* **Play Referential Game:** */
    $D^L, \_ = \text{PlayRG}(\text{Speaker}, \text{Listener}, Obs_{\text{Speaker}}, Obs_{\text{Listener}}, N, L, V)$;
    /* **Supervised Learning Parameters Update on Training Stimulus Only:** */
    Update both speaker and listener agents' parameters using the loss $\mathcal{L}(D^{Target}, D^L)$;
  end
  Initialise ZSCT accuracy: $Acc_{\text{ZSCT}} \leftarrow 0$;
  for *target stimulus* $s_0 \in D^{test}$ **do**
    /* **Preparation of observations and target decision:** */
    $Obs_{\text{Speaker}}, Obs_{\text{Listener}}, D^{Target} \leftarrow DataPrep(\text{Dataset}, s_0, O, K, \text{FullObs}, \text{DescrRatio})$
    /* **Play Referential Game:** */
    $D^L, \_ = \text{PlayRG}(\text{Speaker}, \text{Listener}, Obs_{\text{Speaker}}, Obs_{\text{Listener}}, N, L, V)$;
    /* **Update ZSCT Accuracy:** */
    $Acc_{\text{ZSCT}} \leftarrow \text{Update}(Acc_{\text{ZSCT}}, D^{Target}, D^L)$;
  end
end

Algorithm 5: Meta-Referential Game inside a Meta-Learning Loop

Given :

- $N_{episode}$, N_{dim} : Number of episodes, and number of attribute/factor dimensions,
- S : Minimum number of Shots over which each possible value on each attribute/factor dimension ought to be observed by the agents (as part of a target stimulus).
- V_{min}, V_{max} : Minimum and maximum number of possible values on each attribute/factor dimensions in the symbolic spaces,
- $TSS(\mathcal{D}, \mathcal{S}, S)$: Target stimulus sampling function which samples from dataset $\mathcal{D}$, given a set of previously sampled stimuli $\mathcal{S}$, while maximising the likelihood that each possible value on each attribute/factor dimension are sampled at least S times.
- a set of hyperparameters defining the RG:
  - O : Number of Object-Centric samples in each Target Distribution over stimuli $TD(\cdot)$.
  - N : Number of communication rounds to play.
  - L : Maximum length of each message.
  - V : Vocabulary (finite set of tokens available).
  - K : Number of distractor stimuli to provide to the listener agent.
  - FullObs : Boolean defining whether the speaker agent has full (or partial) observation.
  - DescrRatio : Descriptive ratio in the range $[0, 1]$ defining how often the listener agent is observing the same semantic as the speaker agent.

Initialize :

- Speaker$(\cdot)$ and Listener$(\cdot)$ agents.

for $episode = 1, N_{episode}$ **do**

/* **Preparation of the symbolic space/dataset:** */

$V', D_{\text{episode}}, D^{\text{support}}_{\text{episode}}, D^{\text{query}}_{\text{episode}} \leftarrow MetaRGDatasetPreparation(V, N_{\text{dim}}, V_{\text{min}}, V_{\text{max}})$;

Initialise set of sampled supporting stimuli: $\mathcal{S}^{\text{support}} \leftarrow \emptyset$;

repeat

Sample training-purposed target stimulus $s^i_0 \sim TSS(D^{\text{support}}_{\text{episode}}, \mathcal{S}^{\text{support}}, S)$

$\mathcal{S}^{\text{support}} \leftarrow \mathcal{S}^{\text{support}} \cup \{s^i_0\}$; $i \leftarrow i + 1$;

until *all values on each attribute/factor dimension have been instantiated at least* S *times*;

Initialise RG index: $i \leftarrow 0$;

/* **Supporting Phase:** */

for *target stimulus* $s^i_0 \in \mathcal{S}^{support}$ **do**

$Obs^i_{\text{Speaker}}, Obs^i_{\text{Listener}}, D^{Target}_i \leftarrow DataPrep(D^{\text{support}}_{\text{episode}}, s^i_0, O, K, \text{FullObs}, \text{DescrRatio})$;

$D^L_i, CommH_i = \text{PlayRG}(\text{Speaker}, \text{Listener}, Obs^i_{\text{Speaker}}, Obs^i_{\text{Listener}}, N, L, V')$;

$\_, \_ = \text{Listener}(Obs^i_{Speaker} | CommH_i)$; /* **Listener-Feedback Step** */

end

/* **Querying/ZSCT Phase:** */

Initialise ZSCT accuracy: $Acc_{\text{ZSCT}} \leftarrow 0$;

for *target stimulus* $s^i_0 \in D^{query}_{episode}$ **do**

$Obs^i_{\text{Speaker}}, Obs^i_{\text{Listener}}, D^{Target}_i \leftarrow DataPrep(D_{\text{episode}}, s^i_0, O, K, \text{FullObs}, \text{DescrRatio})$;

$D^L_i, CommH_i = \text{PlayRG}(\text{Speaker}, \text{Listener}, Obs^i_{\text{Speaker}}, Obs^i_{\text{Listener}}, N, L, V')$;

$\_, \_ = \text{Listener}(Obs^i_{Speaker} | CommH_i)$; /* **Listener-Feedback Step** */

/* **Update ZSCT Accuracy:** */

$Acc_{\text{ZSCT}} \leftarrow \text{Update}(Acc_{\text{ZSCT}}, D^{Target}_i, D^L_i)$; $i \leftarrow i + 1$;

end

/* **Meta-Learning Parameters Update on Whole Episode:** */

Update both agents using rewards $R_i = \begin{cases} 1 & \text{if } D^{Target}_i == D^L_i \\ 0 & \text{otherwise, during supporting phase;} \\ -2 & \text{otherwise, during querying phase} \end{cases}$

end

B PROMPTS FOR LANGUAGE MODELS

Speaker & Listener - Detailed Prompt Examples

Speaker Prompt - RG #0 - Step 1

```
1  You and your partner are playing a sequence of referential games. You are the speaker .
2  In the first phase, you will get accounted with the atomic components of the possible observations. Then, the game counter will restart, and you will be tested with new observations, combining the same atomic components in novel ways.
3  At each game, each of you observes a stimulus, which represents a latent meaning, and your common goal is to figure out whether you are observing different or similar latent meanings. You can communicate with your partner using the communication channel. The communication channel is made up of 10 symbols that you can combine together to form a sentence of maximum length 5. Beware that symbol 0 is grounded already. It is the end-of-message symbol. It means that any symbol that comes after it will be ignored and regularised into symbol 0.
4  From one game to the next, you should aim to be consistent so that your partner can figure out the code that you are using to communicate and decrypt messages towards fulfilling your common goal.
5
6  Starting game #0, this is the new stimulus: [-0.469 0.682 -0.532].
7
8  You are an expert in the matter. Given the information above, answer the following question(s) to the best of your abilities .
9
10 Question #1: Do you think your partner understands your messages?
11 Answer either 0.:'Yes' or 1.:'No'.
12
13 Question #2: What message should you send to your partner to better coordinate together towards fulfilling your common goal?
14 The message is made up of 5 symbols, each of which can be filled with one of the 10 vocab symbols. For example: [0, 0, 0, 0, 0].
15 This question corresponds to 5 implicit questions, one for each of the 5 symbols of the message. Thus, each possible answer id is between 0 and 9, corresponding to one of the 10 vocab symbols.
```

Listener Prompt - RG #0 - Step 1

```
You and your partner are playing a sequence of referential games. You are the listener.
In the first phase, you will get accounted with the atomic components of the possible observations. Then, the game counter will restart, and you will be tested with new observations, combining the same atomic components in novel ways.
At each game, each of you observes a stimulus, which represents a latent meaning, and your common goal is to figure out whether you are observing different or similar latent meanings. To help you do so, your partner can send you messages using the communication channel , which is made up of 10 symbols that can be combined together to form a sentence of maximum length 5.
Beware that symbol 0 is grounded already. It is the end-of-message symbol. It means that any symbol that comes after it will be ignored and regularised into symbol 0.

Starting game #0, this is the new stimulus: [0.346 -0.524 0.868].

You are an expert in the matter. Given the information above, answer the following question(s) to the best of your abilities .

Question #1: Are you observing a stimulus representing the same latent meaning as the stimulus observed by your partner?
Answer either 0.:'Yes' or 1.:'No'.

Question #2: What message should you send your partner to better coordinate with them towards fulfilling your common goal?
The message is made up of 5 symbols, each of which can be filled with one of the 10 vocab symbols. For example: [1, 9, 2, 6, 5].
This question corresponds to 5 implicit questions, one for each of the 5 symbols of the message. Thus, each possible answer id is between 0 and 9, corresponding to one of the 10 vocab symbols.
```

Figure 5: Detailed prompts for the speaker and listener agents at the start of a Meta-Referential Game (RG #0), starting with an explanation of the context and the rules (lines 1-4), followed by information about the (previous and) current game(s) (line 6 - cf. Figure 6 - RG #1 Step 1 for more details presenting previous games). The prompt is ended (from line 8) with two multi-choice questions referring to the two actions that agents may perform, i.e. providing a decision and a message that is to be communicated to the other player at the next step. **Note that both of those actions are not necessary for both players, i.e. the decision is only necessary for the listener, and the message is only necessary for the speaker in the experiments presented here.** Our proposed benchmark incorporates the setting where the listener agent is allowed to communicate back to the speaker, but we leave investigation of this more complex setting to subsequent works. RL actions that are not necessary are simply ignored by the benchmark RL environment.

Speaker & Listener - Partial Prompt Examples

Speaker Prompt - RG #0 - Step 2

```
You and your partner are playing a sequence of referential games. You are the speaker .
...
At game #0, you are observing stimulus: [-0.469 0.682 -0.532].
You have sent the following message:  [[3, 3, 3, 3, 3]].

You are an expert in the matter. Given the information above, answer the following question(s) to the best of your abilities .
...
```

Listener Prompt - RG #0 - Step 2

```
You and your partner are playing a sequence of referential games. You are the listener.
...
At game #0, you are observing stimulus: [0.346 -0.524 0.868].
Your partner has sent you the following message:  [9, 9, 9, 9, 9].

You are an expert in the matter. Given the information above, answer the following question(s) to the best of your abilities .
...
```

Speaker Prompt - RG #0 - Step 3: Listener Feedback Step

```
...
At game #0, you have observed stimulus: [-0.469 0.682 -0.532].
You have sent the following message:  [[3, 3, 3, 3, 3]].
Your partner has decided that both of you were observing different latent meanings.
This was correct.  You and your partner have won game #0.
At the end of game #0, here is a special step where your partner is being shown the exact stimulus that you observe.

...
```

Listener Prompt - RG #0 - Step 3: Listener Feedback Step

```
...
At game #0, you have observed stimulus: [0.346 -0.524 0.868].
Your partner had sent you the following message:  [9, 9, 9, 9, 9].
You have decided that both of you were observing different latent meanings.
This was correct.  You and your partner have won game #0.
At the end of game #0, here is a special step where you are given an opportunity to sync with your partner:  this is the exact stimulus that your partner observes: [-0.469 0.682 -0.532].

...
```

Speaker Prompt - RG #1 - Step 1

```
...
At game #0, you have observed stimulus: [-0.469 0.682 -0.532].You have sent the following message:  [[3, 3, 3, 3, 3]].Your partner has decided that both of you were observing different latent meanings.This was correct.  You and your partner have won game #0.
At the end of game #0, there was a special step where your partner was being shown the exact stimulus that you observed.

Starting game #1, this is the new stimulus: [0.102 -0.049 0.638].

...
```

Listener Prompt - RG #1 - Step 1

```
...
At game #0, you have observed stimulus: [0.346 -0.524 0.868].
Your partner had sent you the following message:  [9, 9, 9, 9, 9].
You have decided that both of you were observing different latent meanings.
This was correct.  You and your partner have won game #0.
In game #0, this is the exact stimulus that your partner was observing:  [-0.469 0.682 -0.532].

Starting game #1, this is the new stimulus: [0.912 -0.550 -0.091].

...
```

Figure 6: Partial prompts for the speaker and listener agents, from step 2 of RG #0 to step 1 of RG #1, highlighting information changes from one step to the next. **Note the effect of the vocabulary permutation scheme transforming the message send by the speaker into a vocabulary-permutated version of the message observed by the listener, mainly at step 2.** Step 3 corresponds to the **Listener Feedback step** where the listener is presented with the exact stimulus that the speaker was observing during the current RG.

Table 5: Hyper-parameters values used in R2D2, with LSTM or DNC as the core memory module. All missing parameters follow the ones in Ape-X (Horgan et al., 2018).

R2D2	
Number of actors	32
Actor parameter update interval	1 environment step
Sequence unroll length	20
Sequence length overlap	10
Sequence burn-in length	10
N-steps return	3
Replay buffer size	5×10^4 observations
Priority exponent	0.9
Importance sampling exponent	0.6
Discount γ	0.997
Minibatch size	32
Optimizer	Adam (Kingma & Ba, 2014)
Optimizer settings	learning rate $= 6.25 \times 10^{-5}$, $\epsilon = 10^{-12}$
Target network update interval	2500 updates
Value function rescaling	None

Core Memory Module			
LSTM (Hochreiter & Schmidhuber, 1997)		DNC (Graves et al., 2016)	
Number of layers	2	LSTM-controller settings	2 hidden layers of size 128
Hidden layer size	256, 128	Memory settings	128 slots of size 32
Activation function	ReLU	Read/write heads	2 reading ; 1 writing

C AGENT ARCHITECTURE & TRAINING

The baseline RL agents that we consider use a 3-layer fully-connected network with 512, 256, and finally 128 hidden units, with ReLU activations, with the stimulus being fed as input. The output is then concatenated with the message coming from the other agent in a OHE/MHE representation, mainly, as well as all other information necessary for the agent to identify the current step, i.e. the previous reward value (either $+1$ and 0 during the training phase or $+1$ and -2 during testing phase), its previous action in one-hot encoding, an OHE/MHE-represented index of the communication round (out of N possible values), an OHE/MHE-represented index of the agent's role (speaker or listener) in the current game, an OHE/MHE-represented index of the current phase (either 'training' or 'testing'), an OHE/MHE representation of the previous RG's result (either success or failure), the previous RG's reward, and an OHE/MHE mask over the action space, clarifying which actions are available to the agent in the current step. The resulting concatenated vector is processed by another 3-layer fully-connected network with 512, 256, and 256 hidden units, and ReLU activations, and then fed to the core memory module, which is here a 2-layers LSTM (Hochreiter & Schmidhuber, 1997) with 256 and 128 hidden units, which feeds into the advantage and value heads of a 1-layer dueling network (Wang et al., 2016).

Table 5 highlights the hyperparameters used for the learning agent architecture and the learning algorithm, R2D2(Kapturowski et al., 2018). More details can be found, for reproducibility purposes, in our open-source implementation at HIDDEN_FOR_REVIEW_PURPOSE.

Training was performed for each run on 1 NVIDIA GTX1080 Ti, and the average amount of training time for a run is 18 hours for LSTM-based models, 40 hours for ESBN-based models, and 52 hours for DCEM-based models.

C.1 ESBN & DCEM

The ESBN-based and DCEM-based models that we consider have the same architectures and parameters than in their respective original work from Webb et al. (2020) and Hill et al. (2020), with the exception of the stimuli encoding networks, which are similar to the LSTM-based model.

C.2 RULE-BASED SPEAKER AGENT

The rule-based speaker agents used in the single-agent task, where only the listener agent is a learning agent, speaks a compositional language in the sense of the posdis metric (Chaabouni et al., 2020), as presented in Table 6 for $N_{dim} = 3$, a maximum sentence length of $L = 4$, and vocabulary size $|V| >= max_i d(i) = 5$, assuming a semantical space such that $\forall i \in [1, 3], d(i) = 5$.

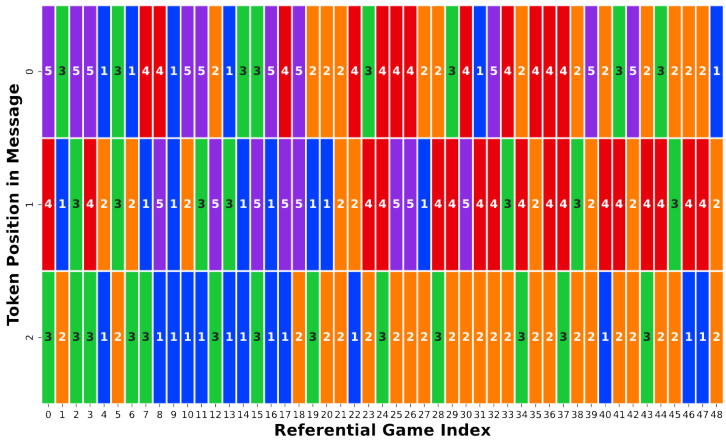


Figure 7: Visualisation on each column of the messages sent by the posdis-compositional rule-based speaker agent over the course of the episode presented in Figure 4. Colours are encoding the information of the token index, as a visual cue.

D CHEATING LANGUAGE

The agents can develop a cheating language, cheating in the sense that it could be episode/task-invariant (and thus semantic structure invariant). This emerging cheating language would encode the continuous values of the SCS representation like an analog-to-digital converter would, by mapping a fine-enough partition of the $[-1, +1]$ range onto the vocabulary in a bijective fashion.

For instance, for a vocabulary size $\|V\| = 10$, each symbol can be unequivocally mapped onto $\frac{2}{10}$-th increments over $[-1, +1]$, and, by communicating N_{dim} symbols (assuming $N_{dim} \leq L$), the speaker agents can communicate to the listener the (digitized) continuous value on each dimension i of the SCS-represented stimulus. If $max_j d(j) \leq \|V\|$ then the cheating language is expressive-enough for the speaker agent to digitize all possible stimulus without solving the binding problem, i.e. without inferring the semantic structure. Similarly, it is expressive-enough for the listener agent to convert the spoken utterances to continuous/analog-like values over the $[-1, +1]$ range, thus enabling the listener agent to skirt the binding problem when trying to discriminate the target stimulus from the different stimuli it observes.

Table 6: Examples of the latent stimulus to language utterance mapping of the posdis-compositional rule-based speaker agent. Note that token 0 is the EoS token.

Latent Dims			Comp. Language
#1	#2	#3	Tokens
0	1	2	1, 2, 3, 0
1	3	4	2, 4, 5, 0
2	5	0	3, 6, 1, 0
3	1	2	4, 2, 3, 0
4	3	4	5, 4, 5, 0

E FURTHER EXPERIMENTS:

E.1 ON THE BP INSTANTIATED BY THE SCS REPRESENTATION

Hypothesis. The SCS representation differs from the OHE/MHE one primarily in terms of the binding problem (Greff et al., 2020) that the former instantiates while the latter does not. Indeed, the semantic structure can only be inferred after observing multiple SCS-represented stimuli. We hypothesised that it is via the *dynamic binding of information* extracted from each observations that

an estimation of a density distribution over each dimension i's $[-1, +1]$ range can be performed. And, estimating such density distribution is tantamount to estimating the number of likely gaussian distributions that partition each $[-1, +1]$ range.

Evaluation. Towards highlighting that there is a binding problem taking place, we show results of baseline RL agents (similar to main experiments in Section 4) evaluated on a simple single-agent recall task. The Recall task structure borrows from few-shot learning tasks as it presents over 2 shots all the stimuli of the instantiated symbolic space (not to be confused with the case for Meta-RG where all the latent/factor dimensions' values are being presented over S shots – Meta-RGs do not necessarily sample the whole instantiated symbolic space at each episode, but the Recall task does). Each shot consists of a series of recall games, one for each stimulus that can be sampled from an $N_{dim} = 3$-dimensioned symbolic space. The semantic structure $(d(i))_{i \in [1; N_{dim}]}$ of the symbolic space is randomly sampled at the beginning of each episode, i.e. $d(i) \sim \mathcal{U}(2; 5)$, where $\mathcal{U}(2; 5)$ is the uniform discrete distribution over the integers in $[2; 5]$, and the number of object-centric samples is $O = 1$, in order to remove any confounder from object-centrism.

Each recall game consists of two steps: in the first step, a stimulus is presented to the RL agent, and only a *no-operation* (NO-OP) action is made available, while, on the second step, the agent is asked to infer/recall the **discrete** $l(i)$ **latent value** (as opposed to the representation of it that it observed, either in the SCS or OHE/MHE form) that the previously-presented stimulus had instantiated, on a given i-th dimension, where value i for the current game is uniformly sampled from $\mathcal{U}(1; N_{dim})$ at the beginning of each game. The value of i is communicated to the agent via the observation on this second step of different stimulus that in the first step: it is a zeroed out stimulus with the exception of a 1 on the i-th dimension on which the inference/recall must be performed when using SCS representation, or over all the OHE/MHE dimensions that can encode a value for the i-th latent factor/attribute when using the OHE/MHE representation. On the second step, the agent's available action space now consists of discrete actions over the range $[1; max_j d(j)]$, where $max_j d(j)$ is a hyperparameter of the task representing the maximum number of latent values for any latent/factor dimension. In our experiments, $max_j d(j) = 5$. While the agent is rewarded at each game for recalling correctly, we only focus on the performance over the games of the second shot, i.e. on the games where the agent has theoretically received enough information to infer the density distribution over each dimension i's $[-1, +1]$ range. Indeed, observing the whole symbolic space once (on the first shot) is sufficient (albeit not necessary, specifically in the case of the OHE/MHE representation).

Results. Figure 8 details the recall accuracy over all the games of the second shot of our baseline RL agent throughout learning. There is a large gap of asymptotic performance depending on whether the Recall task is evaluated using OHE/MHE or SCS representations. We attribute the poor performance in the SCS context to the instantiation of a BP. We note again that during those experiments the number of object-centric samples was kept at $O = 1$, thus emphasising that the BP is solely depending on the use of the SCS representation and does not require object-centrism.

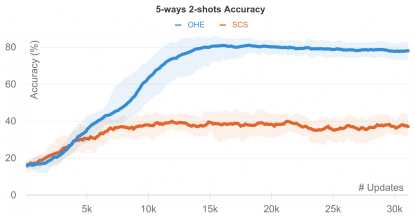


Figure 8: 5-ways 2-shots accuracies on the Recall task with different stimulus representation (OHE:blue ; SCS; orange).

E.2 On the ideally-disentangled-ness of the SCS representation

In this section, we verify our hypothesis that the SCS representation yields ideally-disentangled stimuli. We report on the **FactorVAE Score** Kim & Mnih (2018), the Mutual Information Gap (**MIG**) Chen et al. (2018), and the **Modularity Score** Ridgeway & Mozer (2018) as they have been shown to be part of the metrics that correlate the least among each other (Locatello et al., 2020), thus representing different desiderata/definitions for disentanglement. We report on the $N_{dim} = 3$-dimensioned symbolic spaces with $\forall j, d(j) = 5$ and $O = 5$. The measurements are of 100.0%, 94.8, and 98.9% for, respectivily, the FactorVAE Score, the MIG, and the Modularity Score, thus validating our design hypothesis about the SCS representation. We remark that the MIG and Modularity Score are sensitive to the number of object-centric samples O, which can be seen decreasing the measurements as low as 64.4% and 66.6% for $O = 1$. The FactorVAE Score is not affected, possibly due to its reliance on a deterministic classifier.

E.3 Auxiliary Reconstruction Loss

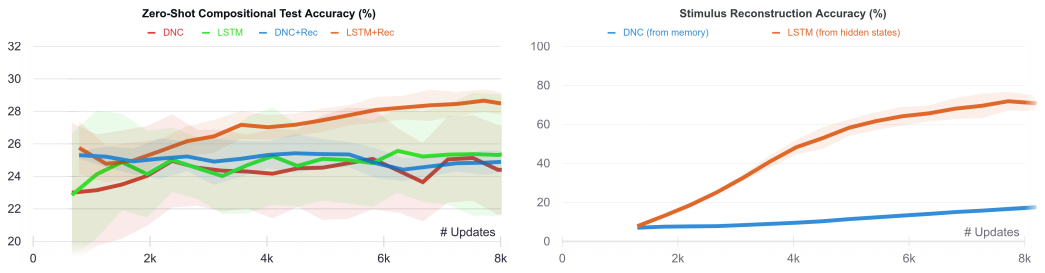
In the following, we investigate and compare the performance when using an LSTM (Hochreiter & Schmidhuber, 1997) or a Differentiable Neural Computer (DNC) (Graves et al., 2016) as core memory module, with or without the auxiliary reconstruction loss inspired from Hill et al. (2020).

In the case of the LSTM, the prediction network of the reconstruction loss takes as input the LSTM hidden states, while in the case of the DNC, the input is the memory. Figure 9b shows the stimulus reconstruction accuracies for both architectures, highlighting a greater data-efficiency (and resulting asymptotic performance in the current observation budget) of the LSTM-based architecture, compared to the DNC-based one.

Figure 9a shows the 4-ways (3 distractors descriptive meta-RGs) ZSCT accuracies of the different agents throughout learning. The ZSCT accuracy is the accuracy over querying-/testing-purpose stimuli only, after the agent has observed for two consecutive times (i.e. $S = 2$) the supportive training-purpose stimuli for the current episode. The DNC-based architecture has difficulty learning how to use its memory, even with the use of the auxiliary reconstruction loss, and therefore it utterly fails to reach better-than-chance ZSCT accuracies. On the otherhand, the LSTM-based architecture is fairly successful on the auxiliary reconstruction task, but it is not sufficient for training on the main task to really take-off. As expected from the fact that the benchmark instantiates a binding problem that requires relational responding, our results hint at the fact that the ability to use memory towards deriving valuable relations between stimuli seen at different time-steps is primordial. Indeed, only the agent that has the ability to use its memory element towards recalling stimuli starts to perform at a better-than-chance level. Thus, the auxiliary reconstruction loss is an important element to drive some success on the task, but it is also clearly not sufficient, and the rather poor results that we achieved using these baseline agents indicates that new inductive biases must be investigated to be able to solve the problem posed in our proposed benchmark.

F Broader Impact

No technology is safe from being used for malicious purposes, which equally applies to our research. However, aiming to develop artificial agents that relies on the same symbolic behaviours and the same social assumptions (e.g. using CLBs) than human beings is aiming to reduce misunderstanding between human and machines. Thus, the current work is targeting benevolent applications. Subsequent works around the benchmark that we propose are prompted to focus on emerging protocols in general (not just posdis-compositional languages), while still aiming to provide a better understanding of artificial agent's symbolic behaviour biases and differences, especially when compared to human beings, thus aiming to guard against possible misunderstandings and misaligned behaviours. The current state of this work does not allow discussion of potential negative societal impact.



(a) (b)

Figure 9: **(a):** 4-ways (3 distractors) zero-shot compositional test accuracies of different architectures. 5 seeds for architectures with DNC and LSTM, and 2 seeds for runs with DNC+Rec and LSTM+Rec, where the auxiliary reconstruction loss is used. **(b):** Stimulus reconstruction accuracies for the architectures augmented with the auxiliary reconstruction task. Accuracies are computed on binary values corresponding to each stimulus' latent dimension's reconstructed value being close enough to the ground truth value, with a threshold of 0.05 on each dimension, which correspond to a deviation tolerance of 2.5% since the range in which SCS stimuli are instantiated is $[-1, 1]$.