EReLELA: Exploration in Reinforcement Learning via Emergent Language Abstractions

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Instruction-following from prompts in Natural Languages (NLs) is an important benchmark for Human-AI collaboration. Training Embodied AI agents for instruction-following with Reinforcement Learning (RL) poses a strong exploration challenge. Previous works have shown that NL-based state abstractions can help address the exploitation versus exploration trade-off in RL. However, NLs descriptions are not always readily available and are expensive to collect. We therefore propose to use the Emergent Communication paradigm, where artificial agents are free to learn an emergent language (EL) via referential games, to bridge this gap. ELs constitute cheap and readily-available abstractions, as they are the result of an unsupervised learning approach. In this paper, we investigate (i) how EL-based state abstractions compare to NL-based ones for RL in hard-exploration, procedurally-generated environments, and (ii) how properties of the referential games used to learn ELs impact the quality of the RL exploration and learning. Results indicate that the EL-guided agent, namely EReLELA, achieves similar performance as its NL-based counterparts without its limitations. Our work shows that Embodied RL agents can leverage unsupervised emergent abstractions to greatly improve their exploration skills in sparse reward settings, thus opening new research avenues between Embodied AI and Emergent Communication.

1 Introduction

Natural Languages (NLs) have some properties, such as compositionality and recursive syntax, that allow us to talk about infinite meanings while only using a finite number of words (or even letters, or phonemes...). In other words, it enables us to be as expressive as one might needs. However, it may be interesting sometimes to use language to abstract away from the details and only focus on the essence of a specific experience, or a specific sensory stimulus. Thus, even though NLs can sometimes be used with high expressiveness, they also can work as abstractions. For instance, using a unique utterance to refer to a lot of semantically-similar but (visually) different situations, such as the one presented in Figure 1 where the utterance 'one can see a purple key and a green ball' can refer to many of the first-person perspective of the embodied agent, irrespective of the actual perspective under which each object is seen.

Tam et al. [61] referred to that aspect as compacting/clustering a state/observation space, which is in effect segmenting it into a set of less-detailed but more-meaningful sub-spaces. We employ the term meaningful with respect the task that the embodied agent is possibly trained for. For instance, if the task consists of picking and placing objects, then it is meaningful for utterances to contain information about objects and places, but not so much to contain information about other agents in the environment, if any. In this paradigm, Tam et al. [61] and Mu et al. [51] provided some arguments towards the compacting/clustering assumption of NLs, as they used NLs oracle to build an abstraction

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

over a 3D and 2D environments. They relied upon state-of-the-art exploration algorithms, such as Random Network Distillation (RND - Burda et al. [9]) and Never-Give-Up (NGU - Badia et al. [1]), which can be difficult to deploy.

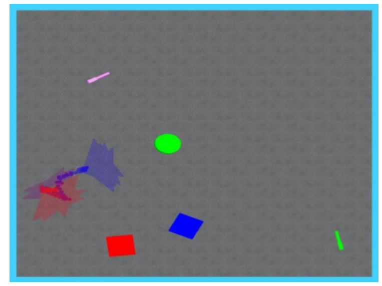

Figure 1: Top-view visualization of a wall-free 3D environment with different objects (e.g. red and blue cubes, purple and green keys, and green ball) showing the trajectory (from blue to red dots) of a randomly-walking embodied agent, with first-person perspectives highlighted at relevant timesteps using colored cones - showing the agent's viewpoint direction when a new utterance is used to describe the first-person perspective using an oracle speaking in NL.

Thus, in this work, we aim to simplify the process of using languages as abstractions and address the limitation of using NLs, as they are expensive to harvest and not necessarily the most meaningful abstraction for any given task. Indeed, instead of state-of-the-art exploration algorithms, we show that simpler count-based approaches combined with language abstraction can be leveraged for hard-exploration tasks. And, in order to remove the reliance on NLs, we look at the field of Emergent Communication (EC) [41, 7] which have shown that artificial languages, that we refer to as emergent languages (ELs), can emerge through unsupervised learning algorithms, such as Referential Games and variants [19], with structure and properties similar to NLs. Our experimental evidences show that ELs, acquired over an embodied agent's observations in an online fashion and in parallel of its training, can be leveraged for hard-exploration tasks. We investigate what are the properties of NLs and ELs in terms of their abstraction building abilities by proposing a novel metric entitled Compactness Ambiguity Metric (CAM). Measures show that ELs abstractions are aligned but not similar to NLs in terms of the abstractions they perform, as the Emergent Communication context successfully picks up on the meaningful features of the environment. Indeed, EReLELA's abstractions reflect colors in the *MultiRoom-N7-S4* environment which only features coloured, unlocked doors, but no distracting objects, or shapes in the *KeyCorridor-S3-R2* environment where it is important to pickup a relevant key, among other distractingly-shaped objects, and to open the locked door-shaped object.

We continue by reviewing EC and RL backgrounds and notations in Section 2. After detailing our method in Section 3, we present experimental results on procedurally-generated, hard-exploration task from the MiniGrid [15] benchmarks in Section 4. Finally, we discuss in Section 5 the results presented in light of some related works and highlight possible future works.

2 Background & Notation

We provide details on our Reinforcement Learning (RL) settings and count-based exploration methods in Section 2.1.Then, we review Emergent Communication in Section 2.2.

2.1 Exploration vs Exploitation in Reinforcement Learning

An RL agent interacts with an environment in order to learn a mapping from states to actions that maximises its reward signal. Initially, both the reward signal and the dynamics of the environment, i.e. the impact that the agent actions may have on the environment, are unknown to the agent. It must explore the environment and gather information, but, all the while it is exploring, it cannot exploit the best strategy that it has found so far to maximise the currently-known reward signal. This dilemma is known as the Exploration-vs-Exploitation trade-off of RL.This dilemma is only the start of the rabbit hole, as it can even get worse. Indeed, in sparse reward environments, the reward signal is mainly zero most of the time. This context makes it very difficult for RL agents to learn anything, because RL algorithms derive feedback (i.e. gradients to update their parameters) from the reward signal that they observe from the environment.It is usually referred to as extrinsic, in order to differentiate it from an intrinsic reward signal. As the extrinsic reward is mostly zero, RL agents must exploit another signal to derive information about the currently-unknown environment. This other signal can be found in relation to the observation/state space, as RL agents can learn to seek novelty or surprise around the observation/state space and attempt to manipulate it efficiently by choosing relevant actions. Focusing on this novelty, RL agents can harvest an intrinsic reward signal, in the sense that RL agents are building it and giving it to themself. Note that this intrinsic reward signal is very different from the extrinsic reward signal, because it does not inform about the task that RL agents need to perform in the environment. Ideally, though, it provides a graded and dense signal that the RL agent can use to start learning anything about the environment. This is inspired by intrinsic motivation in psychology [53]. Exploration driven by curiosity/novelty might be an important way for children to grow and learn. Here, we focus on novelty, but the intrinsic rewards could be correlated with e.g. impact [54], surprise [9] or familiarity of the state. The intrinsic reward signal is only a proxy for RL agents to start to make progress into learning about the environment and eventually, hopefully encounter some non-zero extrinsic reward signal along the way. It provides a denser reward signal that can guide RL agents into learning internal representations about the environment's dynamic so that, whenever some extrinsic reward are encountered along the way, then they can efficiently bind their previously-learned representations to those recently-encountered extrinsic rewards.

Formally, we study a single agent in a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, T, \mathcal{R}, \gamma)$, referring to, respectively, the set of states, the set of actions, the transition function $T : \mathcal{S} \times \mathcal{A} \rightarrow P(\mathcal{S})$ which provides the probability distribution of the next state given a current state and action, the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow r$, and the discount factor $\gamma \in [0, 1]$. The agent is modelled with a stochastic policy $\pi : \mathcal{S} \rightarrow P(\mathcal{A})$ from which actions are sampled at every time step of an episode of finite time horizon T. The agent's goal is to learn a policy which maximises its discounted expected return at time t, defined in equation 1. We further define $\mathcal{R} = \lambda_{\text{ext}}\mathcal{R}^{\text{ext}} + \lambda_{\text{int}}\mathcal{R}^{\text{int}}$ as the weighted sum of the extrinsic and intrinsic reward functions, respectively, $\mathcal{R}^{\text{ext}}, \mathcal{R}^{\text{int}}$, with weights $\lambda_{\text{ext}}, \lambda_{int}$. Indeed, while the extrinsic reward is provided by the environment, we assume that for any tuple (s_t, a_t, s_{t+1}) we can compute an intrinsic reward.

$$R_t = \mathbb{E}_{\substack{s_{t+k+1} \sim T(s_{t+k}, a_{t+k}) \\ a_{t+k+1} \sim \pi(s_{t+k+1})}}\Big[\sum_{k=0}^{T} \gamma^k R(s_{t+k+1}, a_{t+k+1})\Big] \quad (1)$$

Stanton and Clune [58] identifies two categories of exploration strategies, to wit *across-training*, where novelty of states, for instance, is evaluated in relation to all prior training RL episodes, and *intra-life*, where it is evaluated solely in relation of the current RL episode. And, historically, we can identify two types of intrinsic motivation exploration depending on how the intrinsic reward is computed, either relying on count-based or prediction-based methods. Prediction-based methods fit into the *across-training* category and count-based methods can actually fit in both categories but they have mainly been instantiated in the literature as *across-training* methods after extension of *intra-life* core mechanisms. As our proposed architecture EReLELA fit into the category of count-based methods, we detail them further.In the context of an intrinsic reward signal correlated with surprise, then it is necessary to quantify how much of surprise each observation/state provides. Intuitively, we can count how many times a given observation/state has been encountered and derive from that count our intrinsic reward. The reward would guide the RL agent to prefer rarely visited/observed states compared to common states. This is referred to as the count-based exploration method. Count-based exploration method were originally only applicable to tabular RL where the state space is discrete and it is easy to compare states together. When dealing with continuous or high-dimensional state spaces, such method is not practical. Thus, Bellemare et al. [3] proposed (and extended in Ostrovski et al. [52]) a pseudo-count approach which was derived from increasingly more efficient density models, and they showed success in applying it to image-based exploration environments from Atari 2600 benchmark, such as *Montezuma's Revenge*, *Private Eye*, and *Venture*. We provide more relevant details in Appendix B.

Nevertheless, hard-exploration task involving procedurally-generated environments are notoriously difficult for count-based exploration methods. Indeed, when states are procedurally-generated, almost all states will be showing 'novel' features, most times irrespectively of whether it is relevant to the task or not. It will follow that their state (pseudo-)count will always be low and therefore the RL agent will get feedback towards reaching all of them indefinitely, but if every state is 'novel' then there is nothing to guide the agent in any specific direction that would entail to good exploration.

2.2 Emergent Communication

Emergent Communication is at the interface of language grounding and language emergence. While language emergence raises the question of how to make artificial languages emerge, possibly with similar properties to NLs, such as compositionality [2, 24, 45, 55], language grounding is concerned with the ability to ground the meaning of (natural) language utterances into some sensory processes, e.g. the visual modality. On one hand, the compositionality of ELs has been shown to further the learnability of said languages [38, 57, 8, 45] and, on the other hand, the compositionality of NLs promises to increase the generalisation ability of the artificial agent that would be able to rely on them as a grounding signal, as it has been found to produce learned representations that generalise, when measured in terms of the data-efficiency of subsequent transfer and/or curriculum learning [27, 49, 50, 33]. Yet, emerging languages are far from being 'natural-like' protolanguages [40, 10, 11], and the questions of how to constraint them to a specific semantic or a specific syntax remain open problems. Nevertheless, some sufficient conditions can be found to further the emergence of compositional languages and generalising learned representations [40, 43, 17, 5, 24, 39, 12, 21].

The backbone of the field rests on games that emphasise the functionality of languages, namely, the ability to efficiently communicate and coordinate between agents. The first instance of such an environment is the *Signaling Game* or *Referential Game (RG)* by Lewis [44], 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 observation of the message by choosing one of the *actions* available to it in order to perform the 'best' *action* given the observed *state* depending on the notion of 'best' *action* being defined by the interests common to both players. In RGs, typically, the listener action is to discriminate between a target stimulus, observed by the speaker and prompting its message generation, and some other distractor stimuli. Distractor stimuli are selected using a distractor sampling scheme, which has been shown to impact the resulting EL [42, 43]. The listener must discriminate correctly while relying solely on the speaker's message. The latter defined the discriminative variant, as opposed to the generative variant where the listener agent must reconstruct/-generate the whole target stimulus (usually played with symbolic stimuli). Visual (discriminative) RGs have been shown to be well-suited for unsupervised representation learning, either by competing with state-of-the-art self-supervised learning approaches on downstream classification tasks [22], or because they have been found to further some forms of disentanglement [28, 35, 14, 46] in learned representations [65, 18]. Such properties can enable "better up-stream performance"[63], greater sample-efficiency, and some form of (systematic) generalization [48, 26, 59]. Thus, this paper aims to investigate visual discriminative RGs as auxiliary tasks for RL agents.

3 Method

In this section, following the acknowledgement of a gap in terms of evaluating the abstractions that different languages perform over different state/observation space, we start by introducing in Section 3.1 our Compactness Ambiguity Metric (CAM) that attempts to fill in that gap.Then, in Section 3.2, we present the EReLELA architecture that leverages EL abstractions in an *intra-life* count-based exploration scheme for RL agents.

3.1 Compactness Ambiguity Metric

In order to measure qualities related to the kind of abstraction that a language performs over stimuli, we propose to rely on the temporal aspects of embodied agent's trajectories in a given environment. We build over the following intuition, represented in Figure 2: we consider two possible languages grounded into the first-person viewpoint of an embodied agent situated in a 3D environment populated with objects of different shapes and colors. On one hand, we have the Blue language, which is only concerned about blue objects and its utterances only describe that they are of color blue when they are, while, on the other hand, we have the Color language, which is describing the color of all visible objects. Inherently, those two languages expose different semantics about the world, and therefore they perform different abstractions. We aim to build a metric that captures how different the semantics they expose are. To do so, we propose to arrange their respective utterances when prompted with the very same agent's trajectories into different timespan-focused buckets towards building an histogram. These timespan-focused buckets reflect $\delta(u)$ the number of consecutive timesteps $(t_k)_{k\in[k_{\text{start}},k_{\text{start}}+\delta(u)]}$ for which a specific utterance u would be uttered by a speaker of each language when prompted with the stimuli in those timesteps. We will refer to these are compactness counts. For instance the Blue language's utterance 'I see a blue object' at the beginning of the trajectory occupies twice as more consecutive timesteps as the same utterance coming from a Color language speaker (or, its compactness count in the Blue language is twice its compactness count in the Color language). Therefore, in the case of the Blue language, this utterance would increment the medium-length bucket, while it would increment the short-length bucket in the case of Color language histogram. It ensues that the histograms of timespan-focused buckets captures semantics exposed by each language, and we will therefore refer to the resulting histogram as the histogram of semantic-clustering timespans. As the toy example highlights, the histograms of semantic-clustering timespans will differ from one language to another depending on the semantics each language expose or, in other words, depending on the abstractions they perform. This is the first intuition on which the Compactness Ambiguity metric is built.

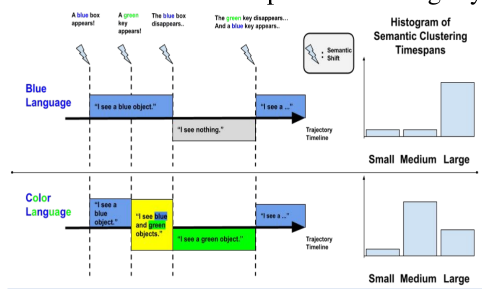

Figure 2: Toy example illustration of how different languages expose different semantics over the same observed trajectory of stimuli, and that the discrepancy in exposed semantics can be captured by an histogram of semantic-clustering timespans.

Formally, we define $\mathcal{L}$ as the set of all possible languages over vocabulary V with maximum sentence length L, such that for any language $l \in \mathcal{L}$ we denote $\text{Sp}_l : \mathcal{S} \to l$ as a speaker agent or oracle that maps any state/observation $s \in \mathcal{S}$ to a caption or utterance $u \in l$. Thus, we can now consider N buckets whose related timespans $(T_i)_{i \in [1,N]}$ are sampled relative to the maximal length T of a trajectory in the given environment, and the histogram of semantic-clustering timespans that they induce.

Then, the other intuition on which the metric is built is made evident by considering the expressivity or, its inverse, the ambiguity, of a given language l, defined as $\mathcal{E}_l = \frac{\#\text{unique utterances}}{\#\text{unique stimuli}}$ with $\#$ the set cardinality operator. Dealing with stimuli being states/observations of a (randomly walking) embodied agent, gathered into a dataset $\mathcal{D}$, the number of unique stimuli cannot be estimated reliably when dealing with complex, continuous stimuli. Thus, the best we can rely on is a measure of relative expressivity over a dataset, that we define as $\mathcal{RE}_l(\mathcal{D}) = \frac{\#\text{unique utterances}}{\#\text{stimuli}} = \frac{\#\text{Sp}_l(\mathcal{D})}{|\mathcal{D}|}$, with $|\cdot|$ being the size operator over collections (differing from sets in the sense that they allow duplicates). In those terms, the relative expressivity is maximised if and only if (i) $\#\mathcal{D} = |\mathcal{D}|$, and (ii) Sp_l is a bijection over $\mathcal{D}$. On the other hand, considering that a language l performs an abstraction over $\mathcal{D}$ is tantamount to some stimuli $(s, s') \in \mathcal{D}^2$ sharing the same utterance $u = \text{Sp}_l(s) = \text{Sp}_l(s')$, i.e. consisting of a hash collision, meaning that the mapping Sp_l from $\mathcal{D}$ to l woud not be injective (and therefore not bijective). Incidentally, the relative expressivity $\mathcal{RE}_l(\mathcal{D})$ cannot be maximised, leading to the language l being ambiguous over $\mathcal{D}$. In this consideration, we can see that the ambiguity of a language (over a given dataset) can be impacted by either the extent to which an abstraction is performed (meaning that most colliding states/observations are of consecutive timesteps) or the extent to which the dataset is redundant (meaning $\#\mathcal{D} << |\mathcal{D}|$). Therefore it is important that our proposed Compactness Ambiguity Metric is built to focus on sources of ambiguities that are the result of consecutive-timesteps states colliding, more than sources of ambiguities that are the result of redundancy in the given dataset.

$$\forall i \in [1, N],\ T_i = 1 + \lceil \lambda_i \cdot \mathcal{RA}_l(\mathcal{D}) \rceil \tag{2}$$

$$\forall i \in [1, N],\ T'_i = 1 + \lceil \lambda_i \cdot T \rceil \tag{3}$$

$$\forall i \in [1, N],\ CA(\mathcal{D})_{T_i} = \sum_{u \in l} \frac{\#\delta_{\mathcal{D}}^{\geq T_i}(u)}{\#\delta_{\mathcal{D}}(u)} \tag{4}$$

Yet, in its currently proposed form, it is impacted by the amount of redundancy in the dataset. In order to reduce this dependence, we propose to bake some invariance to redudancy-induced ambiguity into the timespan-focused buckets. To this end, for a given language l and dataset $\mathcal{D}$, we define the buckets' related timespans in relation to the relative ambiguity $\mathcal{RA}_l(\mathcal{D}) = \frac{1}{\mathcal{RE}_l(\mathcal{D})} = \frac{|\mathcal{D}|}{\#\text{Sp}_l(\mathcal{D})}$, as shown in equation 2 with $\lambda_i \in [0, 1]\ s.t.\ \forall (j,k),\ j < k \implies \lambda_j < \lambda_k$, and $\lceil \cdot \rceil$ being the ceiling operator. This is in lieu of defining them in relation to the maximal length T of an agent's trajectory in the environment, as shown in equation 3. More specifically, let us first acknowledge decomposition of relative ambiguity over two independent quantities, one for each of its sources being either abstraction or redundancy, such that $\mathcal{RA}_l = \mathcal{RA}_l^{\text{redundancy}} + \mathcal{RA}_l^{\text{abstract}}$. Then note that the relative ambiguity is equal to the mean number of consecutive timesteps, or compactness count, for which a given utterance would be used when the unique utterances are uniformly distributed over the dataset $\mathcal{D}$. Thus, in the metric, we propose to absorb variations of relative ambiguity due to redundancy by changing the metric's bucket setup, from Equation 3 to Equation 2. Doing so, it is true that the metric's bucket setup will also vary when the abstraction-induced relative ambiguity varies, we remark that the metric would not build invariance to this source of relative ambiguity since it is taken into accounts when sorting out the different unique utterances into their relevant bucket, based on the maximal number of consecutive timesteps in which they occur, as shown in equation 4 with $\delta_{\mathcal{D}} : l \rightarrow 2^{\mathbb{N}}$ is the compactness count function that associates each utterances $u \in l$ to its related set of compactness counts over dataset $\mathcal{D}$, i.e. the set that contains numbers of consecutive timesteps for which $u \in l$ was uttered by Sp_l, each time it was uttered without being uttered in the previous timestep. For instance, if we consider $u \in l$ such that $\text{Sp}_l^{-1}(u) = \{s_{t_1}, s_{t_1+1}, s_{t_1+2}, s_{t_2}\}$, with $(t_1, t_2) \in [0, T]^2$ such that $t_2 > t_1 + 3$, then $\delta_{\text{D}}(u) = \{3, 1\}$ because u occurred 2 non-consecutive times over $\mathcal{D}$ and those occurrences lasted for, respectively, 3 and 1 consecutive timesteps, i.e. for compactness counts of 3 and 1. The superscript $\geq T_i$ in $\delta_{\mathcal{D}}^{\geq T_i}$ implies filtering of the output set based on compactness counts being greater or equal to T_i. We provide in appendix C an analysis of the sensitivity of our proposed metric, and in appendix E.1 experimental results that ascertain the internal validity of our proposed metric, we consider a 3D room environment of MiniWorld [15], filled with 5 different, randomly-placed objects, as shown in a top-view perspective in Figure 1.

3.2 EReLELA Architecture

This section details the EReLELA architecture, which stands for Exploration in Reinforcement Learning via Emergent Language Abstractions. As a count-based exploration method, we present here its *intra-life* core mechanism, where intrinsic reward signals are derived from novelty at the level of language utterances describing the current observation/state. It relies on a hashing-like function (cf. Appendix B), which takes the form of the speaker agent of a referential game (RG), to turn continuous and high-dimensional observations/states into discrete, variable-length sequences of tokens. EReLELA is built around an RL agent augmented with an unsupervised auxiliary task, a (discriminative, here, or generative) RG, following the UNREAL architecture from Jaderberg et al. [31], as shown in Figure 3.

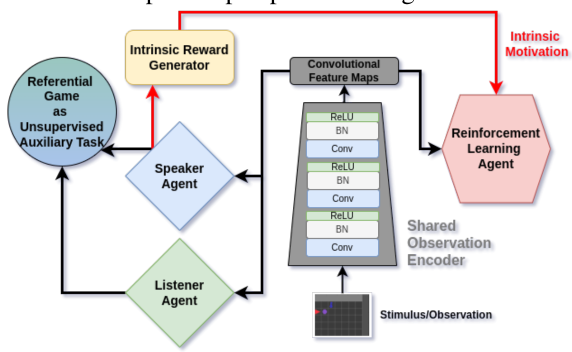

Figure 3: EReLELA architecture consisting of a stimulus/observation encoder shared between an RL agent and the speaker and listener agents of a RG, framed as an unsupervised auxiliary task [31]. The language utterances outputted by the RG speaker agent are used in a count-based exploration method to generate intrinsic rewards for the RL agent.

We train the RG agents in a descriptive, discriminative RG with $K = 256$ distractors, every $T_{RG} = 32768$ gathered RL observations, on a dataset $\mathcal{D}_{RG}$ consisting of the most recent $|\mathcal{D}_{RG}| = 8192$ observations, among which 2048 are held-out for validation/testing-purpose, over a maximum of $N_{RG-epoch} = 32$ epochs or until they reach a validation/testing RG accuracy greater than a given threshold $acc_{RG-thresh} = 90\%$. Our preliminary experiments in Appendices D.1 and D.2 show, respectively, that increasing the RG accuracy threshold $acc_{RG-thresh}$ increases the sample-efficiency of the EL-guided RL agent, and that the number of distractors $K \in [15, 128, 256]$ is critical (even more so than the distractor sampling scheme - which we set to be uniform unless specified otherwise), and that it correlates positively with the performance of the RL agent. More specific details about the RG and its agents' architectures can be found in Appendices F and G and our open-source implementation[1].

4 Experiments

Agents Our RL agent is optimized using the R2D2 algorithm from [34] with the Adam optimizer Kingma and Ba [36]. Importantly, as it aims to maximise the weighted sum of the extrinsic and intrinsic reward functions following equation 1, throughout this paper, we use $\lambda_{int} = 0.1$ and $\lambda_{ext} = 10.0$ in order to make sure that the agent pursues the external goal once the exploration of the environment has highlighted it. Further details about the RL agent can be found in Appendix F. For our RG agents, we consider optimization using either the Impatient-Only or the LazImpa loss function from Rita et al. [56], but the latter is adapted to the context of a Straight-Through Gumbel-Softmax (STGS) communication channel [25, 21], as detailed in Appendix G.1, and we refer to it as STGS-LazImpa. Indeed, the LazImpa loss function has been shown to induce Zipf's Law of

[1] HIDDEN_FOR_REVIEW_PURPOSE

Abbreviation (ZLA) in the ELs. Thus, we can investigate in the following experiments how does **structural** similarity between NLs and ELs affect the kind of abstractions they perform, as well as the resulting RL agent. Further details about the RG in EReLELA can be found in Appendix G.

Environments. After having considered in our preliminary experiments (cf. Appendix E.4) the 2D environment *MultiRoom-N7-S4*, we propose below experiments in the more challenging *KeyCorridor-S3-R2* environment from MiniGrid [15]. Indeed, it involves complex object manipulations, such as (distractors) object pickup/drop and door unlocking, which requires first picking up the relevantly-colored key object.

Natural Language Oracles. Our implementation of a NL oracle is simply describing the visible objects in terms of their colour and shape attributes, from left to right on the agent's perspective, whilst also taking into account object occlusions. For instance, around the end of the trajectory presented in Figure 1, the green key would be occluded by the blue cube, therefore the NL oracle would provide the description 'blue cube red cube' alone. We also implement colour-specific and shape-specific language oracles, which consists of filtering out from the NL oracle's utterance the information that each of those language abstract away, i.e. removing any shape-related word in the case of the colour-specific language, and vice-versa.

Hypotheses. We seek to validate the following hypotheses. Firstly, we consider whether NL abstractions can help for hard-exploration in RL with a simple count-based approach **(H1)**, and refer to the relevant agent using NL abstractions to compute intrinsic rewards as NLA. We carry on with the hypothesis that ELs can be used similarly **(H2)**, and we investigate to what extent do ELs compare to NLs in terms of abstraction. We would expect ELs to perform more meaningful abstractions than NLs **(H3)**, in the sense that their abstractions would be more aligned with the relevant features of a given environment.

Evaluation. We employ 3 random seeds for each agent. We evaluate (H1) and (H2) using both the success rate and the manipulation count, in the hard-exploration task of *KeyCorridor-S3-R2*. The manipulation count is a per-episode counter incremented each time an object is successfully picked up or dropped by the RL agent over the course of each episode. In order to evaluate both (H3.1) and (H3.2), we use the CAM to measure the kind of abstractions performed by ELs, and compare those measures with those of the oracles' languages that we previously studied. We report the CAM distances between ELs and the NL, Color language, and Shape language oracles, which is computed as an euclidean distance in $\mathbb{R}^6$ by considering the $N = 6$ CAM scores for each timespans/thresholds as vectors in this space. As we remarked that an agent's skillfullness at the task would induce very different trajectories (e.g. in *MultiRoom-N7-S4*, staying in the first room and only ever seeing the first door, for an unskillfull agent, as opposed to visiting multiple rooms and observing multiple colored-doors, for a skillfull agent), we compute the oracle languages CAM scores on the exact same trajectories than used to compute each EL's CAM scores.

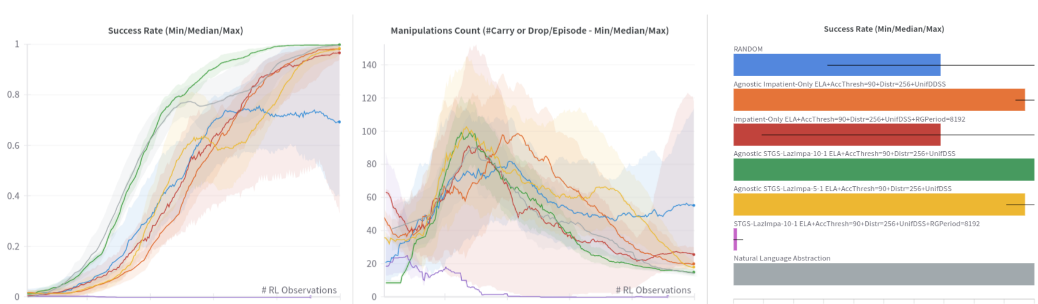

Figure 4: Success rate learning curve (left), computed as running averages over 1024 episodes each time (i.e. 32 in parallel, as there are 32 actors, over 32 running average steps), and barplot (right), along with per-episode manipulation count (middle) in *KeyCorridor-S3-R2* from MiniGrid [15], for different agents: (i) the *Natural Language Abstraction* agent (NLA) refers to using the NL oracle to compute intrinsic reward, (ii) the *STGS-LazImpa-β_1-β_2 EReLELA* agents with $\beta_1 = 5$ (agnostic only) or $\beta_1 = 10$ (shared and agnostic), and $\beta_2 = 1$, (iii) the *Impatient-Only EReLELA* agents (shared and agnostic), and (iv) the *RANDOM* agent referring to an ablated version of EReLELA without RG training.

4.1 EReLELA learns Systematic Navigational & Manipulative Exploration Skills from Scratch

We present in Figure 4 both the success rate of the different agents (as line plot through learning -left-, or barplot at the end of learning -right-), and the per-episode manipulation count (middle). From the fact that both the NLA and EReLELA agent performance converges higher or close to 80% of success rate (except the STGS-LazImpa-10-1), we validate hypotheses (H1) and (H2), meaning that it is possible to learn systematic exploration skills from both NL or EL abstractions with a simple count-based exploration method, in 2D environments (cf. further evidence in Appendix D.1 with the *MultiRoom-S7-R4* environment). This result puts into perspective the directions of previous literature designing complex exploration algorithms [9, 1].

The sample-efficiency is better for NLA than it is for most EL-based agents, except the Agnostic STGS-LazImpa-10-1 agent, possibly because of the fact that ELs are learned online in parallel of the RL training, as opposed to the case of NLA which makes use of a ready-to-use oracle. Concerning the most-sample-efficient Agnostic STGS-LazImpa-10-1 agent, we interpret its success to be the result of benefiting from both a language structure ascribing to the ZLA and a performed abstraction that is more optimal than NL oracle's ones, because it is learned from the stimuli themselves.

Among the different Agnostic EReLELA agents, the final performance are not statistically-significantly distinguishable, meaning that learning systematic exploration skills with EReLELA can be done with some robustness to the anecdotical differences in qualities of the different ELs. On the other hand, the shared/non-agnostic EReLELA agents's performance are statistically-significantly distinguishable from each other and from their agnostic versions, achieving lower performance or even failing to learn anything in the case of the STGS-LazImpa-10-1 EReLELA agent. We interpret these results as being caused by some kind of interference between the RG training and the RL training, preventing any valuable representations from being learned in the shared observation encoder (cf. Figure 3), thus warranting the need for future works to investigate whether a synergy can be achieved.

Finally, acknowledging the RANDOM agent, which is the ablated version of EReLELA without RG training, enabling still a median performance around 70% of success rate, we recall the Random Network Distillation approach from Burda et al. [9], for they both share a randomly initialised networked from which feedback is harvested to guide an RL agent. Thus, even more so in a 2D environment, this ablated version is not to be confused with a lower-bound baseline but rather an interesting ablation that enables us to show the impact of the RG training, increasing the sample-efficiency and final performance of the resulting RL agent.

4.2 EReLELA learns Meaningful Abstractions

Regarding hypothesis (H3), we show in Figure 5 the CAM distances between the different agent's ELs and the natural, colour-specific, and shape-specific languages. We recall that in the *KeyCorridor-S3-R2* environment, the most important feature is object shape as the agent must pickup a key from

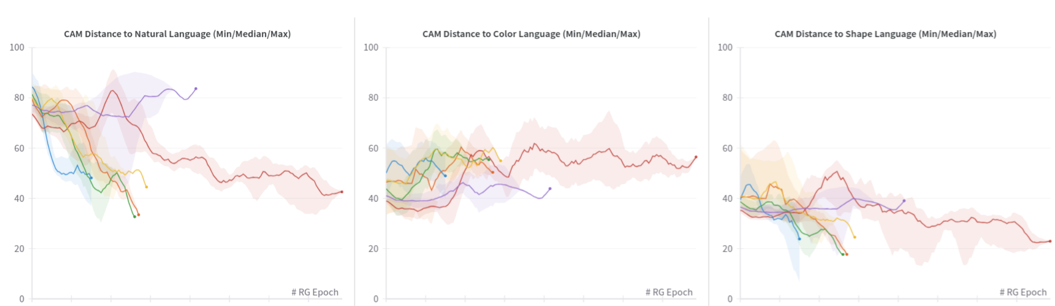

Figure 5: CAM distances to NL (left), Color language (middle), and Shape language (right), for ELs brought about in *KeyCorridor-S3-R2* from MiniGrid [15], with different agents: (i) the *STGS-LazImpa-β_1-β_2 EReLELA* agents with $\beta_1 = 5$ (agnostic only) or $\beta_1 = 10$ (shared and agnostic), and $\beta_2 = 1$, (ii) the *Impatient-Only EReLELA* agents (shared and agnostic), and (iii) the *RANDOM* agent referring to an ablated version of EReLELA without RG training.

all other distractor objects and then use it to unlock the locked door. Thus, as we observe that most ELs' abstractions are closer to the shape-specific language than the others, we conclude that EReLELA learns meaningful abstractions, thus validating hypothesis (H3) (cf. Appendix E.3 for further evidence in the context of *MultiRoom-N7-S4*). Further, we remark that the failing STGS-LazImpa-10-1 EReLELA agent is indeed failing because its EL's abstractions are not highlighting shape features. When considering the shared/non-agnostic agents only, we can see that they require many more RG training epochs, meaning that they reach the accuracy threshold less often than their agnostic counterparts. We take this as further evidence for our interpretation that there might be interference between the RL objective and the RG objective.

We note that abstractions from ELs brought about in the contexts of the *Agnostic STGS-LazImpa* agents and the *Agnostic Impatient-Only* agents are the closest to that of the shape-specific language ones, and their evolution throughout learning are similar. Yet, the *Agnostic STGS-LazImpa* agents achieves statistically-significantly better sample-efficiency (cf. Figure 7). We interpret this as being caused by the ZLA structure of the ELs in the context of the *Agnostic STGS-LazImpa* agents, thus showing that NL-like structure is impacting the kind of abstractions being performed in ways that are yet to be unveiled by future works.

Limitations. With regards to the external validity of EReLELA, we acknowledge that the current work only addresses a 2D environment and therefore, despite being procedurally-generated, it presents less challenges to count-based exploration methods than in the context of 3D procedurally-generated environments. Although we provide some results in Appendix E.3 showing that EReLELA is able to learn meaningful abstractions in a 3D environment, we leave it to future work to ascertain the external validity of EReLELA by testing it in a procedurally-generated 3D environment that pose purely-navigational or navigational and manipulative exploration challenges.

5 Discussion

We investigated the compacting/clustering hypothesis for ELs, questioning how do NLs and ELs compare in terms of the abstractions they perform over state/observation spaces. To answer this question, we proposed a novel metric entitled Compactness Ambiguity Metric (CAM), for which we analysed the sensitivity and performed internal validation.

We then leveraged this metric to show that ELs abstractions are more meaningful than NLs ones, as the Emergent Communication context successfully picks up on the meaningful features of the environment.

Then, we have proposed the **Exploration in Reinforcement Learning via Emergent Languages Abstractions (EReLELA)** agent, which leverages ELs abstractions to generate intrinsic motivation rewards for an RL agent to learn systematic exploration skills. Our experimental evidences showed the performance of EReLELA in procedurally-generated, hard-exploration 2D environments from MiniGrid [15].

Moreover, in the parallel optimization of the RG players, we evidenced how the STGS-LazImpa loss function, which induces EL to abide by ZLA like most NLs, impacts the kind of abstraction being performed compared to baseline Impatient-Only loss function, and yields better sample-efficiency for the RL agent training.

Future work ought to investigate different loss functions and distractor sampling schemes, especially if playing discriminative RGs like here, as we expect, for instance, that sampling distractors more contrastively, e.g. like in Choi et al. [17], may induce the emergence of more complete, and therefore more meaningful ELs. By complete, we mean that the ELs would still be abstracting away details but also capturing more information about the underlying structure of the stimuli space, e.g. capturing both colour- and shape-related information of visible objects. In this light, we would also expect generative RGs to propose a possibly different picture that is worth investigating.

While we leave it to subsequent work to investigate the external validity of EReLELA and whether it transfers similarly well to 3D environments, our results open the door to a new application of the principles of Emergent Communication and ELs towards influencing/shaping the learned representations and behaviours of Embodied AI agents trained with RL.

References

[1] A. P. Badia, P. Sprechmann, A. Vitvitskyi, D. Guo, B. Piot, S. Kapturowski, O. Tieleman, M. Arjovsky, A. Pritzel, A. Bolt, et al. Never give up: Learning directed exploration strategies. In *International Conference on Learning Representations*, 2019.

[2] M. Baroni. Linguistic generalization and compositionality in modern artificial neural networks. mar 2019. URL `http://arxiv.org/abs/1904.00157`.

[3] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[4] L. Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

[5] B. Bogin, M. Geva, and J. Berant. Emergence of Communication in an Interactive World with Consistent Speakers. sep 2018. URL `http://arxiv.org/abs/1809.00549`.

[6] D. Bouchacourt and M. Baroni. How agents see things: On visual representations in an emergent language game. aug 2018. URL `http://arxiv.org/abs/1808.10696`.

[7] N. Brandizzi. Towards more human-like AI communication: A review of emergent communication research. Aug. 2023.

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

[9] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros. Large-Scale Study of Curiosity-Driven Learning. aug 2018. URL `http://arxiv.org/abs/1808.04355`.

[10] R. Chaabouni, E. Kharitonov, E. Dupoux, and M. Baroni. Anti-efficient encoding in emergent communication. *NeurIPS*, may 2019. URL `http://arxiv.org/abs/1905.12561`.

[11] R. Chaabouni, E. Kharitonov, A. Lazaric, E. Dupoux, and M. Baroni. Word-order biases in deep-agent emergent communication. may 2019. URL `http://arxiv.org/abs/1905.12330`.

[12] R. Chaabouni, E. Kharitonov, D. Bouchacourt, E. Dupoux, and M. Baroni. Compositionality and Generalization in Emergent Languages. apr 2020. URL `http://arxiv.org/abs/2004.09124`.

[13] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.

[14] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating sources of disentanglement in VAEs, 2018.

[15] M. Chevalier-Boisvert, B. Dai, M. Towers, R. de Lazcano, L. Willems, S. Lahlou, S. Pal, P. S. Castro, and J. Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831, 2023.

[16] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[17] E. Choi, A. Lazaridou, and N. de Freitas. Compositional Obverter Communication Learning From Raw Visual Input. apr 2018. URL `http://arxiv.org/abs/1804.02341`.

[18] K. Denamganaï, S. Missaoui, and J. A. Walker. Visual referential games further the emergence of disentangled representations. *arXiv preprint arXiv:2304.14511*, 2023.

[19] K. Denamganaï and J. A. Walker. Referentialgym: A nomenclature and framework for language emergence & grounding in (visual) referential games. *4th NeurIPS Workshop on Emergent Communication*, 2020.

[20] K. Denamganaï and J. A. Walker. Referentialgym: A framework for language emergence & grounding in (visual) referential games. *4th NeurIPS Workshop on Emergent Communication*, 2020.

[21] K. Denamganaï and J. A. Walker. On (emergent) systematic generalisation and compositionality in visual referential games with straight-through gumbel-softmax estimator. *4th NeurIPS Workshop on Emergent Communication*, 2020.

[22] R. Dessi, E. Kharitonov, and M. Baroni. Interpretable agent communication from scratch (with a generic visual processor emerging on the side). May 2021.

[23] T. Eccles, Y. Bachrach, G. Lever, A. Lazaridou, and T. Graepel. Biases for emergent communication in multi-agent reinforcement learning. Dec. 2019.

[24] S. Guo, Y. Ren, S. Havrylov, S. Frank, I. Titov, and K. Smith. The emergence of compositional languages for numeric concepts through iterated learning in neural agents. *arXiv preprint arXiv:1910.05291*, 2019.

[25] S. Havrylov and I. Titov. Emergence of Language with Multi-agent Games: Learning to Communicate with Sequences of Symbols. may 2017. URL `http://arxiv.org/abs/1705.11192`.

[26] I. Higgins, A. Pal, A. Rusu, L. Matthey, C. Burgess, A. Pritzel, M. Botvinick, C. Blundell, and A. Lerchner. DARLA: Improving Zero-Shot Transfer in Reinforcement Learning. URL `https://arxiv.org/pdf/1707.08475.pdf`.

[27] I. Higgins, N. Sonnerat, L. Matthey, A. Pal, C. P. Burgess, M. Botvinick, D. Hassabis, and A. Lerchner. SCAN: Learning Abstract Hierarchical Compositional Visual Concepts. jul 2017. URL `http://arxiv.org/abs/1707.03389`.

[28] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner. Towards a Definition of Disentangled Representations. dec 2018. URL `http://arxiv.org/abs/1812.02230`.

[29] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[30] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. Van Hasselt, and D. Silver. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*, 2018.

[31] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2016.

[32] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. A. Ortega, D. Strouse, J. Z. Leibo, and N. De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. *arXiv preprint arXiv:1810.08647*, 2018.

[33] Y. Jiang, S. Gu, K. Murphy, and C. Finn. Language as an Abstraction for Hierarchical Deep Reinforcement Learning. jun 2019. URL `http://arxiv.org/abs/1906.07343`.

[34] S. Kapturowski, G. Ostrovski, J. Quan, R. Munos, and W. Dabney. Recurrent experience replay in distributed reinforcement learning. In *International conference on learning representations*, 2018.

[35] H. Kim and A. Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[36] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[38] S. Kirby. Learning, bottlenecks and the evolution of recursive syntax. 2002.

[39] T. Korbak, J. Zubek, Ł. Kuciński, P. Miłoś, and J. Rączaszek-Leonardi. Developmentally motivated emergence of compositional communication via template transfer. oct 2019. URL http://arxiv.org/abs/1910.06079.

[40] S. Kottur, J. M. F. Moura, S. Lee, and D. Batra. Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog. jun 2017. URL http://arxiv.org/abs/1706.08502.

[41] A. Lazaridou and M. Baroni. Emergent Multi-Agent communication in the deep learning era. June 2020.

[42] A. Lazaridou, A. Peysakhovich, and M. Baroni. Multi-Agent Cooperation and the Emergence of (Natural) Language. dec 2016. URL http://arxiv.org/abs/1612.07182.

[43] A. Lazaridou, K. M. Hermann, K. Tuyls, and S. Clark. Emergence of Linguistic Communication from Referential Games with Symbolic and Pixel Input. apr 2018. URL http://arxiv.org/abs/1804.03984.

[44] D. Lewis. Convention: A philosophical study. 1969.

[45] F. Li and M. Bowling. Ease-of-Teaching and Language Structure from Emergent Communication. jun 2019. URL http://arxiv.org/abs/1906.02403.

[46] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem. A sober look at the unsupervised learning of disentangled representations and their evaluation. Oct. 2020.

[47] R. Lowe, J. Foerster, Y.-L. Boureau, J. Pineau, and Y. Dauphin. On the Pitfalls of Measuring Emergent Communication. mar 2019. URL http://arxiv.org/abs/1903.05168.

[48] M. L. Montero, C. J. Ludwig, R. P. Costa, G. Malhotra, and J. Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qbH974jKUVy.

[49] I. Mordatch and P. Abbeel. Emergence of Grounded Compositional Language in Multi-Agent Populations. URL https://arxiv.org/pdf/1703.04908.pdf.

[50] K. Moritz Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. M. Czarnecki, M. Jaderberg, D. Teplyashin, M. Wainwright, C. Apps, D. Hassabis, P. Blunsom, and D. London. Grounded Language Learning in a Simulated 3D World. URL https://arxiv.org/pdf/1706.06551.pdf.

[51] J. Mu, V. Zhong, R. Raileanu, M. Jiang, N. Goodman, T. Rocktäschel, and E. Grefenstette. Improving intrinsic exploration with language abstractions. *Advances in Neural Information Processing Systems*, 35:33947–33960, 2022.

[52] G. Ostrovski, M. G. Bellemare, A. Oord, and R. Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR, 2017.

[53] P.-Y. Oudeyer and F. Kaplan. How can we define intrinsic motivation? In *the 8th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*. Lund University Cognitive Studies, Lund: LUCS, Brighton, 2008.

[54] R. Raileanu and T. Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. In *International Conference on Learning Representations*, 2019.

[55] Y. Ren, S. Guo, M. Labeau, S. B. Cohen, and S. Kirby. Compositional Languages Emerge in a Neural Iterated Learning Model. feb 2020. URL http://arxiv.org/abs/2002.01365.

[56] M. Rita, R. Chaabouni, and E. Dupoux. " lazimpa": Lazy and impatient neural agents learn to communicate efficiently. *arXiv preprint arXiv:2010.01878*, 2020.

[57] K. Smith, S. Kirby, H. B. A. Life, and U. 2003. Iterated learning: A framework for the emergence of language. *Artificial Life*, 9(4):371–389, 2003. URL https://www.mitpressjournals.org/doi/abs/10.1162/106454603322694825.

[58] C. Stanton and J. Clune. Deep curiosity search: Intra-life exploration can improve performance on challenging deep reinforcement learning problems. *arXiv preprint arXiv:1806.00553*, 2018.

[59] X. Steenbrugge, S. Leroux, T. Verbelen, and B. Dhoedt. Improving generalization for abstract reasoning tasks using disentangled feature representations. Nov. 2018.

[60] U. Strauss, P. Grzybek, and G. Altmann. *Word length and word frequency*. Springer, 2007.

[61] A. Tam, N. Rabinowitz, A. Lampinen, N. A. Roy, S. Chan, D. Strouse, J. Wang, A. Banino, and F. Hill. Semantic exploration from language abstractions and pretrained representations. *Advances in Neural Information Processing Systems*, 35:25377–25389, 2022.

[62] H. Tang, R. Houthooft, D. Foote, A. Stooke, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. Exploration: A study of count-based exploration for deep reinforcement learning. arxiv e-prints, page. *arXiv preprint arXiv:1611.04717*, 2016.

[63] S. van Steenkiste, F. Locatello, J. Schmidhuber, and O. Bachem. Are disentangled representations helpful for abstract visual reasoning? May 2019.

[64] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016.

[65] Z. Xu, M. Niethammer, and C. Raffel. Compositional generalization in unsupervised compositional representation learning: A study on disentanglement and emergent language. Oct. 2022.

[66] G. K. Zipf. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books, 2016.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Contribution/Claim # 1, i.e. a comparison between emergent and natural languages with respect to the kind of abstractions they perform, is substantiated in Section E.1, where we verify the internal validity of the metric we propose for quantitative comparison, and Section E.2 where measures using our proposed metrics on different natural or emergent languages are presented and discussed. Contribution/Claim # 2, i.e. simple count-based exploration methods guided by natural or emergent language abstractions are helpful for exploration in reinforcement learning over hard-exploration, procedurally-generated environments, is substantiated in Section E.3.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations at the end of Section 4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our only theoretical results is found in Appendix C with the full set of assumptions and a complete and correct proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All the information needed to reproduce the main experimental results and appendices experimental results are discussed both in Sections 3 or 4 for critical (and new) hyperparameters, and in Appendices G and F for hyperparameters introduced in previous works.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
    (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
    (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
    (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The open-access code contains a README.md file with sufficient instructions to faithfully reproduce the main experimental results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the information needed to reproduce the main experimental results and appendices experimental results are discussed both in Sections 3 or 4 for critical (and newly-introduced) hyperparameters, and in Appendices G and F for hyperparameters introduced in previous works.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: All plots (barplots or line plots) contains in the title the type of information about the statistical significance of the experiments (i.e. min/median/max, meaning that the shaded area reflect the min and max values of the distribution while the bar or line reflects the median of the distribution).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section F contains sufficient information on the computer resources needed to reproduce the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform in every respect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper contains a Broader Impact discussion in Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does release data or models that have any risk for misuses.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

Justification: Apart from the environments from MiniGrid [15], the paper does not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve experiments with human subjects nor crowdsourcing.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

A Broader impact

No technology is safe from being used for malicious purposes, which equally applies to our research. However, we view many of the ethical concerns surrounding research to be mitigated in the present case. These include data-related concerns such as fair use or issues surrounding use of human subjects, given that our data consists solely of simulations.

With regards to the ethical aspects related to its inclusion in the field of Artificial Intelligence, we argue that our work aims to have positive outcomes on the development of human-machine interfaces since we investigate, among other things, alignment of emergent languages with natural-like languages.

The current state of our work does not allow extrapolation towards negative outcomes. We believe that this work is of benefit to the research community of reinforcement learning, language emergence and grounding, in their current state.

B Further details on Count-Based Exploration

Another approach to counting states from continuous and/or high-dimensional state spaces is by relying on hashing functions, so that states become tractable. Indeed, Tang et al. [62] have shown that a generalisation of classical counting techniques through hashing can provide an appropriate signal for exploration in continuous and/or high-dimensional environments where informed exploration is required. In effect, they proposed to discretise the state space $\mathcal{S}$ with a hash function $\phi : \mathcal{S} \rightarrow \mathbb{Z}^k$, with $k \in \mathbb{N} \setminus \{0\}$, to derive an exploration bonus of the form $r^+(s) = \frac{\beta}{\sqrt{n(\phi(s))}}$ where $\beta \in \mathbb{R}^+$ is a bonus coefficient and $n(.)$ is a count initialised at zero for the whole range of ϕ and updated at each step t of the RL loop by increasing by 1 the count $n(\phi(s_t))$ related to the current observation/state s_t. Performance is dependent on the hash function ϕ, and especially in terms of granularity of the discretisation it induces. Indeed, it would be desirable that the 'similar' states result in hashing collisions while the 'distant' states would not. To this end, they propose to use locality-sensitive hashing (LSH) such as SimHash [13], resulting in the following:

$$\phi(s) = \operatorname{sgn}(Ag(s)) \in \{-1, 1\}^k, \tag{5}$$

where sgn is the sign function, $A \in \mathbb{R}^{k \times D}$ is a matrix with each entry drawn i.i.d. from a standard Gaussian distribution, and $g : S \rightarrow \mathbb{R}^D$ is an optional preprocessing function. Note that increasing k leads to higher granularity and therefore decreases the number of hashing collisions. Tang et al. [62] reports great results on the Atari 2600 benchmarks, both with and without a learnable g that is modelled as the encoder of an autoencoder (AE).

C Sensitivity Analisys of the Compactness Ambiguity Metric

Based on derivative-based local sensitivity analysis, we propose an intuitive proof of our claim that defining timespans in relation to the relative ambiguity reduces the sensibility to variations induced by redundancy-based ambiguity in the resulting metric, compared to defining timespans in relation to the the maximal length T of an agent's trajectory in the environment. To do so, we assume:

(i) that there exists two differentiable function $f_i.f'_i$ such that for all $i \in [1, N]$, we have $CA(\mathcal{D})_{T_i} = f_i(\mathcal{D}, \mathcal{RA}_l^{\text{redundancy}}, \mathcal{RA}_l^{\text{abstract}})$ when T_i is defined according to Equation 2, and respectively with f'_i when using T'_i from Equation 3, and

(ii) that their partial derivatives with respect to T_i or T'_i are negative. Indeed, T_i and T'_i are involved into filtering operations reducing the value of the numerator in Equation 4, therefore any increase of their values would result in decreasing the overall metric output, which implies that their partial derivatives with f_i and f'_i must be negative.

With those assumptions, we show that f_i's sensitivity to redundancy-induced ambiguity $\mathcal{RA}_l^{\text{redundancy}}$ is less than that of f'_i:

Proof.

$$\frac{\partial f_i}{\partial \mathcal{RA}_l^{\text{redundancy}}} = \frac{\partial f_i}{\partial CC_{\mathcal{D}}} \cdot \frac{\partial CC_{\mathcal{D}}}{\partial \mathcal{RA}_l^{\text{redundancy}}} + \frac{\partial f_i}{\partial T_i} \cdot \frac{\partial T_i}{\partial \mathcal{RA}_l^{\text{redundancy}}}$$

$$\text{(from Assump. (i) about } f_i\text{)}$$

$$\iff \frac{\partial f_i}{\partial \mathcal{RA}_l^{\text{redundancy}}} = \frac{\partial f'_i}{\partial \mathcal{RA}_l^{\text{redundancy}}} + \frac{\partial f_i}{\partial T_i} \cdot \frac{\partial T_i}{\partial \mathcal{RA}_l^{\text{redundancy}}} \qquad \text{(from Assump. (i) about } f'_i\text{)}$$

$$\iff \frac{\partial f_i}{\partial \mathcal{RA}_l^{\text{redundancy}}} = \frac{\partial f'_i}{\partial \mathcal{RA}_l^{\text{redundancy}}} + \frac{\partial f_i}{\partial T_i} \cdot \lambda_i$$

$$\implies \left|\frac{\partial f_i}{\partial \mathcal{RA}_l^{\text{redundancy}}}\right| \leq \left|\frac{\partial f'_i}{\partial \mathcal{RA}_l^{\text{redundancy}}}\right| \qquad \text{(since } \tfrac{\partial f_i}{\partial T_i} \cdot \lambda_i \leq 0 \text{ from Assump. (ii))}$$

□

D Preliminary Experiments

D.1 Impact of Referential Game Accuracy

In this experiments, we investigate whether the RG accuracy impacts the RL agent training, in the context of the *MultiRoom-N7-S4* environment from *MiniGrid* [15], with an RL sampling budget of $1M$ observations.

Hypothesis. We seek to validate the following hypotheses, **(PH1)** : the sample-efficiency of the RL agent is dependant on the quality of the RG players, as parameterised by the $acc_{RG-thresh}$ hyperparameter.

Evaluation. We report both the success rate and the coverage count in the hard-exploration task of *MultiRoom-N7-S4*. To compute the coverage count, we overlay a grid of tiles over the environment's possible locations/cells of the agents and we count the number of different tiles visited by the RL agent over the course of each episode. We use 3 random seeds for each agent. In order to evaluate the impact of the RG accuracy strictly in terms of the kind of abstractions that are being performed by the resulting EL, we use the *Impatient-Only* loss function (removing the impact of the hyperparameter of the scheduling function $\alpha(\cdot)$ from the *Lazy* term of the *STGS-LazImpa* loss function), and we employ an **agnostic** version of our proposed EReLELA agent, i.e. **without sharing the observation encoder between the RG players and the RL agent**. We present results for two different RG accuracy threshold $acc_{RG-thresh} = 60\%$ (green) or $acc_{RG-thresh} = 80\%$ (red), and compare against, as an upper bound the *Natural Language Abstraction* agent (blue), which refers to using the NL oracle to compute intrinsic reward, and, as a lower bound an ablated version of EReLELA without RG training (orange).

Results. We present results in Figure 6. We observe statistically significant differences between the performances (in terms of success rate, cf. Figure 6(left)) of the two EReLELA agents with $acc_{RG-thresh} = 60\%$ or $acc_{RG-thresh} = 80\%$, thus validating hypothesis (PH1). We observe that higher RG accuracy threshold lead to higher sample-efficiency.

As a sanity check, we plot the results of the ablated EReLELA agent without RG training, and we were expecting it to perform poorer than any other agent since the quality of its RG players is the lowest, at chance level. Yet, we observe that it performs on par with the best $acc_{RG-thresh} = 80\%$-EReLELA agent. While puzzling, we propose a possible explanation in the observation that the test-time relative expressivity of the ablated agent is higher than that of the least-performing, $acc_{RG-thresh} = 60\%$-EReLELA agent, and on par with that of the best-performing, $acc_{RG-thresh} = 80\%$-EReLELA agent, at the beginning of the RL agent training process. Thus, we interpret this as follows: the randomly-initialised ablated agent's EL is possibly performing an abstraction over the observation

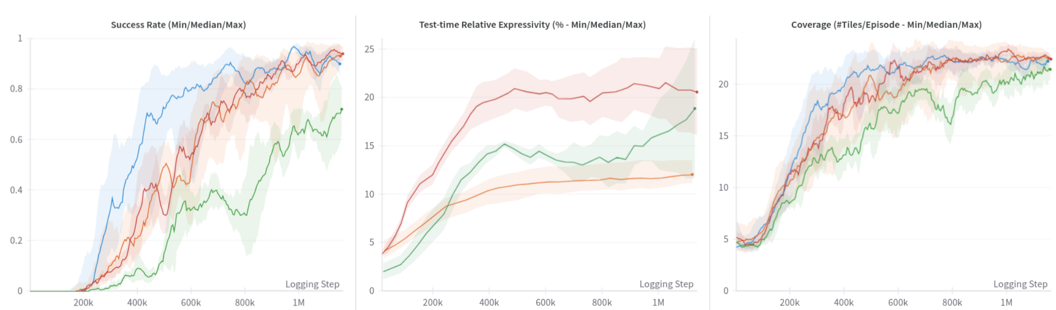

Figure 6: Success rate (left), test-time relative expressivity (middle), and per-episode coverage count (right) in *MultiRoom-N7-S4* from MiniGrid [15], computed as running averages over 256 episodes each time (i.e. 32 in parallel, as there are 32 actors, over 8 running average steps), for different agents: (i) the *Natural Language Abstraction* agent (blue) refers to using the NL oracle to compute intrinsic reward, the *Agnostic Impatient-Only EReLELA* agent refers to our proposed architecture **without sharing the observation encoder between the RG players and the RL agent**, using the Impatient-Only loss function to optimize the RG players, with an RG accuracy threshold $acc_{RG-thresh} = 60\%$ (ii - green) or $acc_{RG-thresh} = 80\%$ (iii - red), and (iv) an ablated version without RG training (orange).

space that is good-enough for the RL agent to start learning exploration skills, the same way the random network in the context of the RND agent from Burda et al. [9] probably does, and increasing the quality of the RG players may only be a sufficient condition to increasing the sample-efficiency of the EL-guided RL agent.

D.2 Impact of Referential Game Distractors

In this experiments, we investigate whether the RG's number of distractors K and distractor sampling scheme impacts the RL agent training, in the context of the *KeyCorridor-S3-R2* environment from *MiniGrid* [15], with an RL sampling budget of $1M$ observations.

Hypothesis. We seek to validate the following hypotheses, **(PH2)** : the sample-efficiency of the RL agent is dependant on the number of distractors K and the distractor sampling scheme.

Evaluation. We report the success rate in the hard-exploration task of *KeyCorridor-S3-R2*. We use 3 random seeds for each agent. Like previously, we use the *Impatient-Only* loss function (to remove the impact of the hyperparameter of the scheduling function $\alpha(\cdot)$ from the *Lazy* term of the *STGS-LazImpa* loss function), and we employ an **agnostic** version of our proposed EReLELA agent, i.e. **without sharing the observation encoder between the RG players and the RL agent**. We present results for three different number of distractors $K \in [15, 128, 256]$ and two different sampling scheme between *UnifDSS* corresponding to uniformly sampling distractors over the whole training dataset, or *Sim*50*DSS* corresponding to sampling distractors 50% of the time from the same RL episode than the current target stimulus is from and, the rest of the time following *UnifDSS*. Following results in Appendix D.1, we set the RG accuracy threshold $acc_{RG-thresh} \in [80\%, 90\%]$.

Results. We present results in Figure 7. We observe statistically significant differences between the performances of the different EReLELA agents, thus validating hypothesis (PH2). Our results show that (i) the number of distractors K is the most impactful parameter and it correlates positively with the resulting performance, irrespective of the distractor sampling scheme used, and, indeed, (ii) while the *Sim*50*DSS* seems to provide better performance than *UnifDSS* for low numbers of distractors $K = 15$, although not statistically-significantly, the table is turned when considering high number of distractors $K = 256$ where the *UnifDSS* yields statistically significantly better performance than the *Sim*50*DSS*.

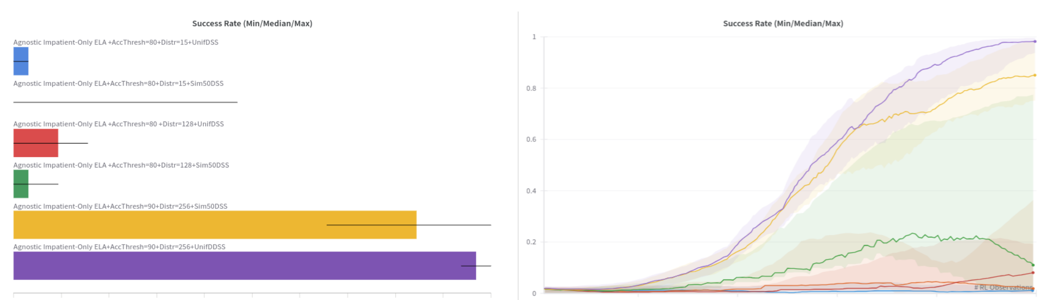

Figure 7: Final success rate barplot (left) and success rate throughout learning (right) in *KeyCorridor-S3-R2* from MiniGrid [15], computed as running averages over 1024 episodes each time (i.e. 32 in parallel, as there are 32 actors, over 32 running average steps), for the *Agnostic Impatient-Only EReLELA* agent, which refers to our proposed architecture **without sharing the observation encoder between the RG players and the RL agent**, using the Impatient-Only loss function to optimize the RG players, with different number of distractors K and distractors sampling schemes: with RG accuracy threshold $acc_{RG-thresh} = 80\%$, (i) $K = 15$ and *UnifDSS* or Sim50DSS, (ii) $K = 1128$ and *UnifDSS* or Sim50DSS, or with RG accuracy threshold $acc_{RG-thresh} = 90\%$, (iii) $K = 256$ and *UnifDSS* or Sim50DSS.

E Further Experiments

E.1 Experiment #1: CAM Metric Internal Validity

Environment. We consider a 3D room environment of MiniWorld [15], where the agent's observation is egocentric, as a first-person viewpoint. The room is filled with 5 different, randomly-placed objects, with different shapes (among ball, box or key) and colours (among). The dimensions simulate a 12 by 5 meters room, like shown in a top-view perspective in Figure 1.

Hypothesis. In this experiments, we seek to validate two hypotheses, **(H1.1)** : the Compactness Ambiguity Metric captures something that is related to the kind of abstraction a language performs, and **(H1.2)** : the Compactness Ambiguity Metric allows a graduated comparison of different kind of abstractions being performed, meaning that it allows discrimination between different kind of abstractions.

Evaluation. In order to compute the metric, we use 5 seeds to gather random walk trajectories in our environment, for each language. In order to evaluate (H1.1), we propose to measure a language that is built to present no meaningful abstractions and we expect the measure to be close to null. We build a language that performs no meaningful abstraction from the natural language oracles by shuffling its utterances over the set of agent trajectories that are used to compute the metric, meaning that the mapping between temporally-sensitive stimuli and linguistic utterances is rendered completely random.

Then, in order to evaluate (H1.2), we show experimental evidences that the metric allows qualitative discrimination between the different languages built above from the natural language oracles, which are build to perform different kind of abstractions.

Results. We present results of the metric with $N = 6$ timespans in Figure 8, for $\lambda_0 = 0.0306125$, $\lambda_1 = 0.06125$, $\lambda_2 = 0.125$, $\lambda_3 = 0.25$, $\lambda_4 = 0.5$ and $\lambda_5 = 0.75$. As the shuffled (natural) language measure is almost null on all timespans/thresholds, we validate hypothesis (H1.1).

We observe that we can qualitatively discriminate between each evaluated language's measures since the histograms are statistically different. Moreover, language abstractions scores are inversely correlated with the amount of information being abstracted away, i.e. attribute-value-specific languages' abstraction score lower than colour/shape-specific languages abstraction, which score lower than natural language abstractions. Thus, we can see that the metric is graduated and that the graduation follows the amount of abstraction being performed by each language. This allows us to validate hypothesis (H1.2).

Figure 8: Interval validity measures of Compactness Ambiguity Metric for $N = 6$ timespans/thresholds, with $\lambda_0 = 0.0306125$, $\lambda_1 = 0.06125$, $\lambda_2 = 0.125$, $\lambda_3 = 0.25$, $\lambda_4 = 0.5$ and $\lambda_5 = 0.75$, for different languages built to perform different kind of abstraction. We can qualitatively discriminate between each languages, and validate that the shuffled (natural) language's meaningless abstraction scores almost null.

E.2 Experiment #2: Qualities of Emergent Languages Abstractions in 3D environment

In this experiment, we investigate what kind of abstractions do ELs perform over a 3D environment, in comparison to some natural languages abstractions, as detailed at the beginning of Section 4. For further precision, we also implement attribute-value-specific language oracles with the same filtering approach. For instance, for the green value on the colour attribute, we would obtain a green-only language oracle whose utterances could be 'EoS' if no visible object is green, or 'green green' if there are two green objects visible in the agent's observation. We consider the same 3D room environment of MiniWorld [15] as in Section E.1, i.e. the agent's observation is egocentric, as a first-person viewpoint and the room is filled with 5 different, randomly-placed objects, with different shapes (among ball, box or key) and colours (among). The dimensions simulate a 12 by 5 meters room, like shown in a top-view perspective in Figure 1.

Hypothesis. We seek to validate the following hypotheses, **(H2.1)** : ELs build meaningful abstractions, and **(H2.2)** : ELs brought about using the STGS-LazImpa loss function (type II) perform more meaningful abstractions than Impatient-Only baseline (type I).

Evaluation. In order to make the CAM measures, we use 5 seeds to gather random walk trajectories in our environment, for each language. In order to evaluate both (H2.1) and (H2.2), we use the CAM to measure the kind of abstractions performed by ELs brought about in the two different EReLELA settings, with Impatient-Only or STGS-LazImpa losses, and compare those measures with those of the oracles' languages that we previously studied.

Results. We present results of the metric with $N = 6$ timespans in Figure 9. We observe statistically significant differences between ELs of type I and II, with type I's abstraction being similar to a Blue-specific language's abstraction (timespans $0 - 4$) or a Ball-specific language's abstraction (timespans $1 - 3$), and type II's abstraction not really resembling any of the oracle languages' abstractions, but still being meaningful with scores increasing along with the length of the considered timespans. Thus, we validate hypothesis (H2.1), but cannot conclude on hypothesis (H2.2), unless we consider that CAM scores related to longer timespans are more meaningful, for instance.

E.3 Experiment #3: Learning Purely-Navigational Systematic Exploration Skills from Scratch

In the following, we present an experiment in the *MultiRoom-N7-S4* environment from *MiniGrid* [15], which is possibly less challenging than *KeyCorridor-S3-R2*, presented in the Section 4, for it does not involve as many complex object manipulation (e.g. only open/close doors, no unlocking of doors – which requires the corresponding key to be firstly picked up – nor pickup/drop keys or other objects as distractors), but still poses a **purely-navigational** hard-exploration challenge. We report results on the **agnostic** version of our proposed EReLELA architecture, that is to say **without sharing the observation encoder between both RG players and the RL agent**, in order to guard ourselves against the impact of possible confounders found in multi-task optimization, such as possible

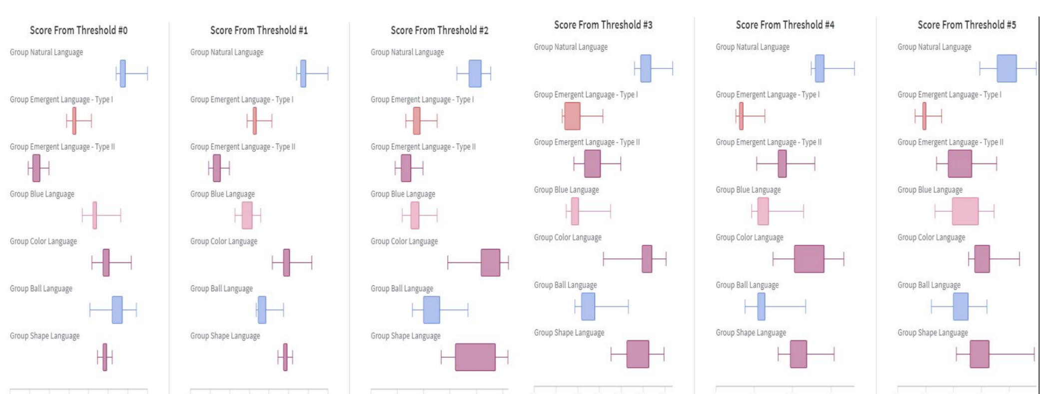

Figure 9: Measures of Compactness Ambiguity Metric for $N = 6$ timespans/thresholds, with $\lambda_0 = 0.0306125$, $\lambda_1 = 0.06125$, $\lambda_2 = 0.125$, $\lambda_3 = 0.25$, $\lambda_4 = 0.5$ and $\lambda_5 = 0.75$, comparing ELs (Type I and II) with different oracles' languages built to perform different kind of abstraction.

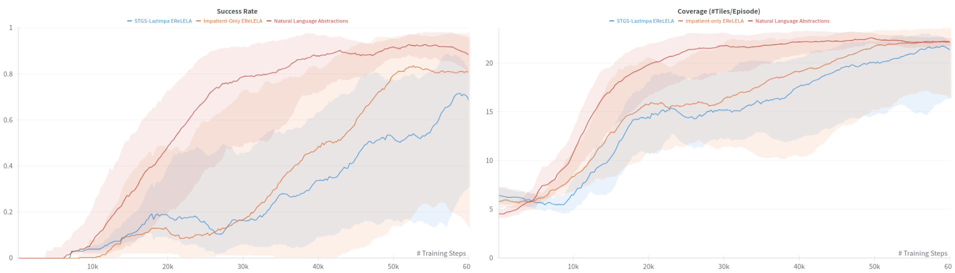

Figure 10: Success rate (left) and per-episode coverage count (right) in *MultiRoom-N7-S4* from MiniGrid [15], computed as running averages over 1024 episodes each time (i.e. 32 in parallel, as there are 32 actors, over 32 running average steps), for different agents: (i) the *Natural Language Abstraction* agent (NLA) refers to using the NL oracle to compute intrinsic reward, (ii) the *STGS-LazImpa EReLELA* agent refers to our proposed architecture, EReLELA, using the STGS-LazImpa loss function to optimize the RG players, and (iii) the *Impatient-Only EReLELA* agent refers to the same architecture without the lazy-speaker loss to optimize the RG players.

interference between the RL-objective-induced gradients and the RG-training-induced gradients. We use an RG accuracy threshold $acc_{RG-thresh} = 65\%$ and a number of training distractors $K = 3$ (like at testing/validation time).

Hypotheses. We consider whether NL abstractions can help for a purely-navigational hard-exploration task in RL with a count-based approach **(H3.0)**, and refer to the relevant agent using NL abstractions to compute intrinsic rewards as NLA. Then, we make the hypothesis that ELs can be used similarly **(H3.1)**, and we investigate to what extent do ELs compare to NLs in terms of abstraction performed, in this purely-navigational task. In the case of (H3.1) being verified, we would expect ELs to perform similar abstractions as NLs **(H3.2)**.

Evaluation. We evaluate (H3.0) and (H3.1) using both the success rate and the coverage count.To compute the coverage count, we overlay a grid of tiles over the environment's possible locations/cells of the agents and we count the number of different tiles visited by the RL agent over the course of each episode. To evaluate (H3.2), we compute the CAM scores of both the ELs and the oracles' natural, color-specific, and shape-specific languages. As we remarked that an agent's skillfullness at the task would induce very different trajectories (e.g. in *MultiRoom-N7-S4*, staying in the first room and only ever seeing the first door, for an unskillfull agent, as opposed to visiting multiple rooms and observing multiple colored-doors, for a skillfull agent), we compute the oracle languages CAM scores on the exact same trajectories than used to compute each EL's CAM scores.

Results. We present in Figure 10(left) the success rate of the different agents, and the per-episode coverage count in Figure 10(right).From the fact that both the NLA and EReLELA agent performance converges higher or close to 80% of success rate, we validate hypotheses (H0) and (H3.1), in the context of the *MultiRoom-N7-S4* environment. We remark that the sample-efficiency is slightly better for NLA than it is for EL-based agents, possibly because of the fact that ELs are learned online in parallel of the RL training, as opposed to the case of NLA which makes use of a ready-to-use oracle. Among the two EReLELA agents, the learning curves are not statistically-significantly

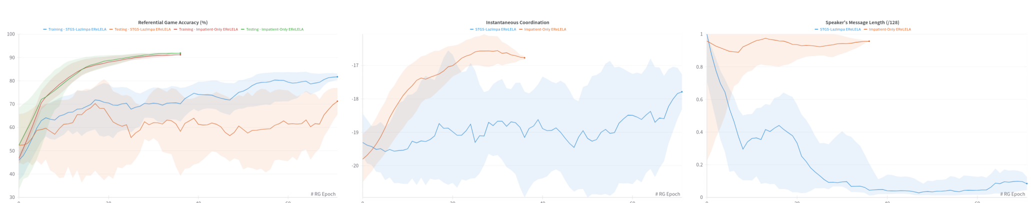

Figure 11: Performance and qualities of the ELs brought about in the context of both (i) the *STGS-LazImpa EReLELA* agent, and (ii) the *Impatient-Only EReLELA* agent, with respect to both the training- and validation/testing-time RG accuracy (left), the validation/test-time Instantaneous Coordination [32, 47, 23](middle), and the validation/testing-time length of the speaker's messages (as a ratio over the max sentence length $L = 128$ - right).

Figure 12: Comparison of Compactness Ambiguity Metric scores for $N = 6$ timespans/thresholds, with $\lambda_0 = 0.0306125$, $\lambda_1 = 0.06125$, $\lambda_2 = 0.125$, $\lambda_3 = 0.25$, $\lambda_4 = 0.5$ and $\lambda_5 = 0.75$, between the abstractions performed by ELs brought about in the context of both (i) the *STGS-LazImpa EReLELA* agent (in green, first rows) and (ii) the *Impatient-Only EReLELA* agent (in purple, bottom rows), and the abstractions performed by the natural, colour-specific, and shape-specific languages, computed on the very same agent trajectories.

distinguishable, meaning that learning systematic exploration skills with EReLELA can be done with some robustness to the anecdotical differences in qualities of the different ELs due to using different optimization losses. Indeed, we also report in Figure 11 both the training- and validation/testing-time RG accuracies (on the left), the validation/testing-time Instantaneous Coordination (in the middle – Jaques et al. [32], Lowe et al. [47], Eccles et al. [23]), and the validation/testing-time length of the RG speaker's messages (on the right), showing that the ELs brought about in the two different contexts perform differently in terms of their RG objective and have different qualities, but these discrepancies do not seem to impact the RL agents learning equally well from the different abstractions they perform (as evidenced in the next paragraph).

Next, with regards to hypothesis (H3.2), we investigate whether the two contexts bring about ELs that perform different abstractions, and how do these relate to the abstractions performed by natural, colour-specific, and shape-specific languages, by showing in Figure 12 their CAM scores. We observe that both contexts result in ELs performing abstractions similar or better than colour-specific languages, which is to be expected as (door) colours are the most salient features of the environment. Indeed, the only two shapes or objects visible are 'wall' and 'door', whereas there are more than 7 different colours of interest. In the context of the Impatient-Only EReLELA agent, the EL's abstractions are scoring very similarly to NL abstractions, as we consider longer timespans (from timespans #2 to #5). We could hypothesise that without the lazy-ness constraint the speaker agent may be given enough capacity to compress/express information pertaining to the location of visible objects, as this information is the only one that is captured by the NL oracle but not captured by the shape- and colour-specific languages.

E.4 Experiment #4: Quantifying RL Agents' Learning Progress?

In the context of RGs, the speed at which a language emerges (in terms of sampled observations, or number of games played) may possibly remain constant, when the data and the player architectures are fixed. Thus, when the data changes, the rate of language emergence may change too. Incidentally, we are entitled to ponder whether some properties of the data, which here are RL trajectories, would influence the rate of language emergence and how?

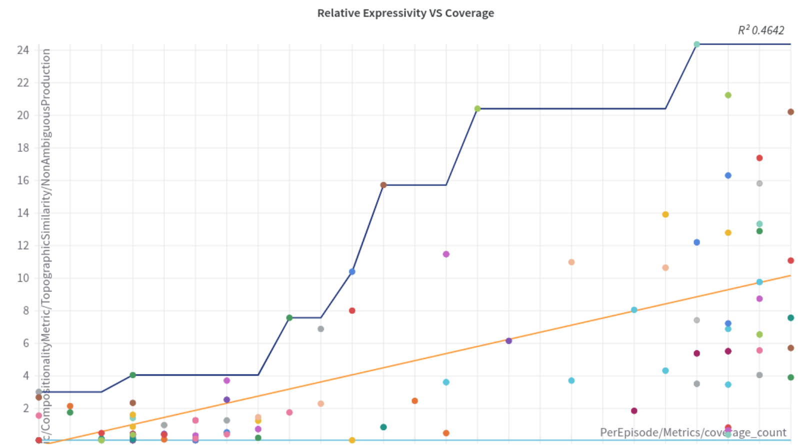

Figure 13: Relative expressivity of the EL as a function of the per-episode coverage of the RL agent, at the end of training, over multiple runs with different hyperparameters during a W&B Sweep [4].

Hypothesis. We hypothesise that as the RL agent gets more skillful, the expressivity of the emergent language increases **(H4.1)**. Indeed, at each RG training epoch, the size of the dataset is fixed, and as the stimuli gets more diverse when the RL agent gets more skillful at exploring, the RG training will prompt the EL to increase its expressivity.

Evaluation. To verify our hypothesis, we propose to measure the skillfullness of the RL agent in terms of exploration using the per-episode coverage count metric, and we measure the expressivity of the EL via the test-time (Relative) Expressivity after each RG training epoch.

Results. We present results in Figure 13, that show the (relative) expressivity of the ELs does exhibit variations throughout the learning process of the RL agent. And, if we perform a regression analysis with each runs in terms of the per-episode coverage count of the RL agent on the x-axis and the expressivity of the ELs on the y-axis, we obtain a high coefficient of determination between the two metrics, $R^2 = 0.4642$. Thus, we conclude that the (relative) expressivity of the ELs in EReLELA can provide a way to quantify the progress of the RL agent, at least when it comes to exploration skills.

Limitations. Exploration skills translates directly into diversity of the stimuli being observed, and therefore it prompts any RG players to increase the expressivity of their communication protocol, but it is remains to be seen whether this effect is valid in any environment. For instance, it is unclear whether a skillfull player in any other video game would induce the same effect on the diversity of the stimuli encountered. Thus, it is worth investigating whether this correlation holds for other genre of environments and skills, which we leave to future works.

F Agent Architecture

The ERELELA architecture is made up of three differentiable agents, the language-conditioned RL agent and the two RG agents (speaker and listener). Each agent contains at least a visual/observation encoder module that can be shared between agents.Both RG agents contain a language module that is not shared. The *listener* agent additionally incorporates a third decision module that combines the outputs of the other two modules. The RL agent similarly incorporates a third decision module with the addition that this third module contains a recurrent network, acting as core memory module for the agent. Using the Straight-Through Gumbel-Softmax (STGS) approach in the communication channel of the RG, the *speaker* agent is prompted to produce the output string of symbols with a *Start-of-Sentence* symbol and the visual module's output as an initial hidden state while the *listener* agent consumes the string of symbols with the null vector as the initial hidden state. In the following subsections, we detail each module architecture in depth.

Visual Module. The visual module $f(\cdot)$ consists of the *Shared Observation Encoder*, which can be shared between all the different agents.The former consists of three blocks of convolutional layers of sizes $8, 4, 3$ with strides $4, 3, 1$, each followed by a 2D batch normalization layer and a ReLU non-linear activation function. The two first convolutional layers have 32 filters, whilst the last one has 64. The bias parameters of the convolutional layers are not used, as it is common when using batch normalisation layers. Inputs are stimuli consisting of RGB frames of the environment resized to 64×64.

Language Module. The language module $g(\cdot)$ consists of some learned Embedding followed by either a one-layer GRU network [16] in the case of the RL agent, or a one-layer LSTM network [29] in the case of the RG agents. In the context of the *listener* agent, the input message $m = (m_i)_{i\in[1,L]}$ (produced by the *speaker* agent) is represented as a string of one-hot encoded vectors of dimension $|V|$ and embedded in an embedding space of dimension 64 via a learned Embedding. The output of the *listener* agent's language module, $g^l(\cdot)$, is the last hidden state of the RNN layer, $h^l_L = g^L(m_L, h^l_{L-1})$. In the context of the *speaker* agent's language module $g^S(\cdot)$, the output is the message $m = (m_i)_{i\in[1,L]}$ consisting of one-hot encoded vectors of dimension $|V|$, which are sampled using the STGS approach from a categorical distribution $Cat(p_i)$ where $p_i = Softmax(\nu(h^s_i))$, provided ν is an affine transformation and $h^s_i = g^s(m_{i-1}, h^s_{i-1})$. $h^s_0 = f(s_t)$ is the output of the visual module, given the target stimulus s_t.

Decision Module. From the RL agent to the RG's listener agent, the decision module are very different since their outputs are either, respectively, in the action space $\mathcal{A}$ or the space of distributions over $K+1$ stimuli (i.e. discriminating between distractors and target stimuli). For the RL agent, the decision module takes as input a concatenated vector comprising the output of visual module, after it has been procesed by a 3-layer fully-connected network with 256, 128 and 64 hidden units with ReLU non-linear activation functions, and some other information relevant to the RL context (e.g. previous reward and previous action selected, following the recipe in Kapturowski et al. [34]). The resulting concatenated vector is then fed to the core memory module, a one-layer LSTM network [29] with 1024 hidden units, which feeds into the advantage and value heads of a 1-layer dueling network [64].

In the case of the RG's listener agent, similarly to Havrylov and Titov [25], the decision module builds a probability distribution over a set of $K+1$ stimuli/images $(s_0, ..., s_K)$, consisting of K distractor stimuli and the target stimulus, provided in a random order, given a message m using the scalar product:

$$p((d_i)_{i\in[0,K]} | (s_i)_{i\in[0,K]}; m) = Softmax\Big((h^l_L \cdot f(s_i)^T)_{i\in[0,K]}\Big). \tag{6}$$

Regarding optimization of the RL agent, table 1 highlights the hyperparameters used for the off-policy RL algorithm, R2D2[34]. More details can be found, for reproducibility purposes, in our open-source implementation at HIDDEN-FOR-REVIEW-PURPOSES.

Each run can be done on less than 2Gb of VRAM, and the amount of training time for a run, with e.g. one NVIDIA GTX1080 Ti, is between 24 and 48 hours depending on the architecture (e.g. shared or agnostic).

Table 1: Hyper-parameter values relevant to R2D2 in the EReLELA architecture presented. All missing parameters follow the ones in Ape-X [30].

R2D2	
Number of actors	32
Actor update interval	1 env. step
Sequence unroll length	20
Sequence length overlap	10
Sequence burn-in length	10
N-steps return	3
Replay buffer size	1×10^4 obs.
Priority exponent	0.9
Importance sampling exponent	0.6
Discount γ	0.98
Minibatch size	64
Optimizer	Adam [36]
Learning rate	6.25×10^{-5}
Adam ϵ	10^{-12}
Target network update interval	2500 updates
Value function rescaling	None

G On the Referential Game in EReLELA

We follow the nomenclature proposed in Denamganaï and Walker [20] and focus on a *descriptive object-centric (partially-observable) 2-players/$L = 10$-signal/$N = 0$-round/K-distractor* RG variant.

The descriptiveness implies that the target stimulus may not be passed to the listener agent, but instead replaced with a descriptive distractor. In effect, the listener agent's decision module therefore outputs a $K+2$-logit distribution where the $K+2$-th logit represents the meaning/prediction that a descriptive distractor has been introduced and none of the $K+1$ stimuli is the target stimulus that the speaker agent was 'talking' about. The addition is made following Denamganaï et al. [18] as a learnable logit value, $logit_{no-target}$, it is an extra parameter of the model. In this case the decision module output is no longer as specified in Equation 6, but rather as follows:

$$p((d_i)_{i\in[0,K+1]}|(s_i)_{i\in[0,K]}; m) = Softmax\Big((h^l_L \cdot f(s_i)^T)_{i\in[0,K]} \cup \{logit_{no-target}\}\Big). \quad (7)$$

The descriptiveneness is ideal but not necessary in order to employ the listener agent as a predicate function for the hindsight experience replay scheme. Thus, in the main results of the paper, we present the version without descriptiveness.

The object-centrism is achieved via application of data augmentation schemes before feeding stimuli to any RG agent, following Dessi et al. [22] but using Gaussian Blur transformation alone, as it was found sufficient in practice.

We optimize the RG agents with either the Impatient-Only STGS loss and the STGS-LazImpa loss.

In the remainder of this section, we detail the STGS-LazImpa loss that we employed to optimize the referential game agents.

G.1 STGS-LazImpa Loss

Emergent languages rarely bears the core properties of natural languages [40, 6, 43, 12], such as Zipf's law of Abbreviation (ZLA). In the context of natural languages, this is an empirical law which states that the more frequent a word is, the shorter it tends to be [66, 60]. Rita et al. [56] proposed LazImpa in order to make emergent languages follow ZLA.

To do so, Lazimpa adds to the speaker and listener agents some constraints to make the speaker lazy and the listener impatient. Thus, denoting those constraints as $\mathcal{L}_{STGS-lazy}$ and $\mathcal{L}_{impatient}$, we obtain the STGS-LazImpa loss as follows:

$$\mathcal{L}_{STGS-LazImpa}(m, (s_i)_{i\in[0,K]}) = \mathcal{L}_{STGS-lazy}(m) + \mathcal{L}_{impatient}(m, (s_i)_{i\in[0,K]}). \tag{8}$$

In the following, we detail those two constraints.

Lazy Speaker. The Lazy Speaker agent has the same architecture as common speakers. The 'Laziness' is originally implemented as a cost on the length of the message m directly applied to the loss, of the following form:

$$\mathcal{L}_{lazy}(m) = \alpha(acc) \cdot |m| \tag{9}$$

where acc represents the current accuracy estimates of the referential games being played, and α is a scheduling function as follows: α : accuracy $\in [0, 1] \mapsto \frac{\text{accuracy}^{\beta_1}}{\beta_2}$, with $(\beta_1, \beta_2) = (45, 10)$. It is aimed to adaptively penalize depending on the message length. Since the lazyness loss is not differentiable, they ought to employ a REINFORCE-based algorithm for the purpose of credit assignement of the speaker agent.

In this work, we use the STGS communication channel, which has been shown to be more sample-efficient than REINFORCE-based algorithms [25], but it requires the loss functions to be differentiable. Therefore, we modify the lazyness loss by taking inspiration from the variational autoencoders (VAE) literature [37].

The length of the speaker's message is controlled by the appearance of the EoS token, wherever it appears during the message generation process that is where the message is complete and its length is fixed. Symbols of the message at each position are sampled from a distribution over all the tokens in the vocabulary that the listener agent outputs. Let (W_l) be this distribution over all tokens $w \in V$ at position $l \in [1, L]$, such that $\forall l \in [1, L]$, $m_l \sim (W_l)$. We devise the lazyness loss as a Kullbach-Leibler divergence $D_{KL}(\cdot|\cdot)$ between these distribution and the distribution (W_{EoS}) which attributes all its weight on the EoS token. Thus, we dissuade the listener agent from outputting distributions over tokens that deviate too much from the EoS-focused distribution (W_{EoS}), at each position l with varying coefficients $\beta(l)$. The coefficient function $\beta : [1, L] \to \mathbb{R}$ must be monotically increasing. We obtain our STGS-lazyness loss as follows:

$$\mathcal{L}_{STGS-lazy}(m) = \alpha(acc) \cdot \sum_{l\in[1,L]} \beta(l) D_{KL}\Big((W_{EoS})|(W_l)\Big) \tag{10}$$

Impatient Listener. Our implementation of the Impatient Listener agent follows the original work of Rita et al. [56]: it is designed to guess the target stimulus as soon as possible, rather than solely upon reading the EoS token at the end of the speaker's message m. Thus, following Equation 6, the Impatient Listener agent outputs a probability distribution over a set of $K + 1$ stimuli $(s_0, ..., s_K)$ for all sub-parts/prefixes of the message $m = (m_1, ..., m_l)_{l\in[1,L]} = (m_{\leq l})_{l\in[1,L]}$:

$$\forall l \in [1, L],\ \ p((\mathbf{d_i^{\leq l}})_{\mathbf{i\in[0,K]}}|(s_i)_{i\in[0,K]}; \mathbf{m^{\leq l}}) = Softmax\Big((\mathbf{h_{\leq l}} \cdot f(s_i)^T)_{i\in[0,K]}\Big), \tag{11}$$

where $\mathbf{h_{\leq l}}$ is the hidden state/output of the recurrent network in the language module after consuming tokens of the message from position 1 to position l included.

Thus, we obtain a sequence of L probability distributions, which can each be contrasted, using the loss of the user's choice, against the target distribution (D_{target}) attributing all its weights on the decision d_{target} where the target stimulus was presented to the listener agent. Here, we employ Havrylov and Titov [25]'s Hinge loss. Denoting it as $\mathbb{L}(\cdot)$, we obtain the impatient loss as follows:

$$\mathcal{L}_{impatient/\mathbb{L}}(m, (s_i)_{i\in[0,K]}) = \frac{1}{L} \sum_{l\in[1,L]} \mathbb{L}((d^{\leq l}_{i\in[0,K]}, (D_{target})). \tag{12}$$