Infinite use of finite means: 
Zero-Shot Generalization using Compositional Emergent Protocols

Rishi Hazra*, Sonu Dixit*, Sayambhu Sen
Indian Institute of Science, Bangalore
{rishihazra, sonudixit, sayambhusen}@iisc.ac.in,

Abstract

Human language has been described as a system that makes use of finite means to express an unlimited array of thoughts. Of particular interest is the aspect of compositionality, whereby, the meaning of a compound language expression can be deduced from the meaning of its constituent parts. If artificial agents can develop compositional communication protocols akin to human language, they can be made to seamlessly generalize to unseen combinations. However, the real question is, how do we induce compositionality in emergent communication? Studies have recognized the role of curiosity in enabling linguistic development in children. It is this same intrinsic urge that drives us to master complex tasks with decreasing amounts of explicit reward. In this paper, we seek to use this intrinsic feedback in inducing a systematic and unambiguous protolanguage in artificial agents. We show how these rewards can be leveraged in training agents to induce compositionality in absence of any external feedback. Additionally, we introduce gComm, an environment for investigating grounded language acquisition in 2D-grid environments. Using this, we demonstrate how compositionality can enable agents to not only interact with unseen objects but also transfer skills from one task to another in a zero-shot setting: *Can an agent, trained to ‘pull’ and ‘push twice’, ‘pull twice’?*

1 Introduction

As humans, we can refer to novel (composite) concepts via a systematic combination of simpler words, thus, allowing us to convey an arbitrary set of concepts using a limited vocabulary. This property of natural language is known as compositionality. In the recent past, there has been a great deal of research in the field of emergent language in artificial agents interacting in simulated environments [Kirby, 2001; Havrylov and Titov, 2017]. However, the real question here is, to what extent do these evolved protocols resemble natural language? Recent studies have revealed the following about emergent languages: (i) they do not conform to Zipf’s Law of Abbreviation1 [Chaabouni et al., 2019]; (ii) communication protocols either do not follow compositionality patterns of natural language [Kottur et al., 2017] or are not always interpretable [Lowe et al., 2019]; (iii) emerged protocols are sensitive to experimental conditions [Lazaridou et al., 2018].

While it has been shown that compositionality is not crucial to achieving generalization, more compositional protocols always have a higher zero-shot performance [Ren et al., 2020]. With regard to emergent communication, so far, existing works on compositionality are limited to analyzing simple referential games [Lewis, 1969], where a speaker communicates the input (object’s shape and color) to a stationary listener which, then, tries to classify the reconstructed messages from a list of classes [Kottur et al., 2017; Li and Bowling, 2019]. These games do not involve world state manipulation and generally comprise elementary inputs with limited attributes, thus, restricting the scope of language usage. Moreover, studies have demonstrated that compositionality is not driven naturally in neural agents [Kottur et al., 2017], and that, it is easier to converge on a holistic (unambiguous but not fully systematic) protocol, rather than a fully compositional one, during training [Ren et al., 2020].

An intelligent agent must have the ability to master a continuous flow of new tasks. To that end, we intend to push the boundaries of compositionality to a more challenging and realistic multi-task settings, arguing that it can also support the acquisition of more complex repertoire of skills (performing a *pull twice* task when it has been trained to *pull, push and push twice*), in addition to generalizing over novel composition of object properties (pushing *red square* when it has been trained to push a *red circle* and a *blue square*). We propose an intrinsic reward based framework to encourage more compositional protocols and aid the learning process. Additionally, we introduce a communication environment called grounded Comm (gComm)2 which provides a platform for investigating grounded language acquisition in agents. 3

---

1The law states that more frequent words tend to be shorter.
2demos: https://sites.google.com/view/compositional-comm
3gComm: https://github.com/SonuDixit/gComm
2 Related Work

Emergent communication has been studied in the past from the perspective of language evolution [Tieleman et al., 2019], multi-agent cooperation [Gupta et al., 2020], strategy development [Gupta and Dukkipati, 2019] and shaping behavioral policies [Lowe et al., 2017] among others. A community of differently specialized robots, while performing a given task, should not only interact amongst themselves but also occasionally with a human counterpart. As such, more recently, the emergent protolanguages are being investigated to find synergies with natural language [Lazaridou et al., 2020].

Generalization lies at the heart of artificial intelligence, which to a large extent remains unsolved. Through the use of language, agents can discover concepts they were unaware of, that is critical to solving a planning problem [Colas et al., 2020]. While work on incorporating compositionality into emergent languages is still in its early stages, certain works have studied it by using different paradigms of training. [Mordatch and Abbeel, 2018] proposed to use limited vocabulary as a means to achieve composition, by using a penalty for larger vocabulary size. A similar approach in [Chaabouni et al., 2020] proposed a limited channel capacity as a sufficient condition to achieve the same. Yet another approach proposed an evolution-driven framework to train agents in an iterated learning fashion [Ren et al., 2020], originally conceptualized in [Kirby, 2001].

Extrinsic rewards may sometimes prove to be unsuccessful, especially for sparse reward setting. MOTIVATED BY Humann behavior, existing works [Gopnik et al., 2001; Baldassarre and Mirolli, 2013] have proposed to use intrinsic rewards for improving the agent’s ability to create rich state representations with limited feedback from the environment. However, intrinsic rewards have not received much attention when it comes to artificial acquisition of language. The goal of the listener is to choose optimal actions according to a policy $\pi : (O, m_{t+1}) \rightarrow \Delta(A)$, to maximize its long-term reward $R = \sum_{t=1}^{\infty} \gamma^t r(t)$. Here, $\gamma$ is the discount factor and $\Delta$ is the transition function $\Delta : S \times A \rightarrow S$. The environment generates a 0-1 (sparse) reward, i.e., the listener gets a reward of $r = 1$ if it achieves the specified task, otherwise $r = 0$. However, the listener has no information about either the task or the target specifics and relies on the speaker for the same. The semantics of this communication is not fixed, rather, it evolves during the course of training. Moreover, the transmitted messages do not affect the environment dynamics and are only observable to the listener. Real-life applications can range from robotic assistants to exploring hazardous territories for space exploration/defense purposes (for instance, a drone communicating with ground-based vehicles using human instructions).

3 Preliminaries

3.1 Problem Setup with Emergent Communication

We analyze a typical signalling game [Lewis, 1969], comprising a stationary Speaker-Bot (speaker) and a mobile Listener-Bot (listener), by modelling it in form of a Markov Decision Process specified by the tuple $(S, O, A, r, T, \gamma)$. Given a set of all possible environment states $S \subseteq \mathbb{R}^{d_{S}}$, $O$, $A$ and $r$ are the observation space, action space and the reward respectively, for the listener. At the beginning of each round, the speaker receives a natural language instruction (push a red circle) and communicates the same using discrete messages $m_{t+1} \in \{0, 1\}^{d_{m}}$, sampled from a message space $M$, to the listener over a communication channel ($(\text{these constitute the channel capacity, } |C| = \epsilon_{d_{m}}^{d_{m}}$). At each step $t$, the listener receives an observation $a^{(t)} \in A$, comprising the 2D grid-view and the received messages $m_{t+1}$, and takes an action $a^{(t)} \in A$. The goal of the listener is to choose optimal actions according to a policy $\pi : (O, m_{t+1}) \rightarrow \Delta(A)$, to maximize its long-term reward $R = \sum_{t=1}^{\infty} \gamma^t r(t)$. Here, $\gamma$ is the discount factor and $\Delta$ is the transition function $\Delta : S \times A \rightarrow S$. The environment generates a 0-1 (sparse) reward, i.e., the listener gets a reward of $r = 1$ if it achieves the specified task, otherwise $r = 0$. However, the listener has no information about either the task or the target specifics and relies on the speaker for the same. The semantics of this communication is not fixed, rather, it evolves during the course of training. Moreover, the transmitted messages do not affect the environment dynamics and are only observable to the listener. Real-life applications can range from robotic assistants to exploring hazardous territories for space exploration/defense purposes (for instance, a drone communicating with ground-based vehicles using human instructions).

3.2 Compositionality

The principle of compositionality as stated in [Kamp and Partee, 1995] is: “The meaning of a complex expression is a function of the meanings of its parts and of their syntactic mode of combination”. In the past, compositionality has been exploited to refer to previously unseen composite objects [Kottur et al., 2017]. However, the notion of compositionality as a necessary condition for generalization has been contested. It was demonstrated that generalization can emerge even by training the speaker model with a large input space [Chaabouni et al., 2020]. Nevertheless, there is a growing consensus that having a higher compositionality leads to better zero-shot learning [Ren et al., 2020], and thus, is a sufficient condition for generalization.

We use topographic similarity (topsim) [Brighton and Kirby, 2006] as a measure of compositionality. Given a language $L(.) : C \rightarrow M$, where $C$ is the set of concepts and $M$ is the set of messages, we define two pairwise distance measures: (i) in the concept space $\Delta_{C}^{ij} = d_{C}(c_{i}, c_{j})$; (ii) in the message space $\Delta_{M}^{ij} = d_{M}(m_{i}, m_{j})$. Topsim is then defined as the correlation coefficient calculated between $\Delta_{C}^{ij}$ and $\Delta_{M}^{ij}$. Following standard practise, we use hamming distance and minimum edit distance as our distance measures for concepts and messages, respectively.

4 Approach

4.1 Environment Description

A crucial step towards studying language acquisition in agents is to endow them with the ability to communicate. 

\footnote{We define concepts as the ‘task’ + ‘object’ specifications}
Figure 1: Model Description: (1) Natural Language Instruction generated; (2) Parser converts it into \( \{\text{VERB}, \{\text{ADJ}_i\}_{i=1}^3, \text{NOUN}\} \) form; (3) Speaker encodes the parsed input to a set of one-hot encoded messages \( m^{\text{in}} \in \{0, 1\}^{d_m} \) and transmits it to the listener, (4) The grid information is processed using the Grid Encoder to output \( G_i \in \mathbb{R}^{d_g \times 4 \times 4} \); (5) Attention weights \( \alpha_i \) are obtained for each cell by taking the dot product of the messages (projected to \( \mathbb{R}^{1 \times d_g} \)) and each cell encoding \( G_i \in \mathbb{R}^{d_g} \) of the Grid Encoder output ; (6) The Master policy switches between subpolicies based on the incoming message; (7) Together with the Master Policy and the selected subpolicy, the listener executes the tasks.

the same time, an agent must rely on a robust human-machine interface so that it can learn from sophisticated human instructions. The proposed environment, gComm, augments both the aforementioned features in a 2D-grid environment, using a pair of bots, a stationary speaker and a mobile listener, that process the language instruction and the grid-view, respectively. More importantly, gComm provides several tools for studying different forms of communication with meaning grounded in the states of the grid world.

In our experiments, we use a 4 \times 4 grid. Cells in the grid contain objects characterized by certain attributes like shape, size, color and weight. These objects can either be the target object or the distractor objects. Distractors have either the same color or the same shape (or both) as that of the target. In our experiments, we keep the number of distractors fixed (= 2) as the languages can be sensitive to their count and distribution [Lazaridou et al., 2018]. The listener and the objects may spawn at any random location on the grid. Given an instruction, it is first processed using a parser to \( \{\text{VERB}, \{\text{ADJ}_i\}_{i=1}^3, \text{NOUN}\} \) and then fed to the speaker. The speaker transmits the same using a set of one-hot encoded messages to the listener which, then, processes the grid representation and the received messages to achieve the given task. The grid input can either be an image input of the complete grid or a \( \{0, 1\}^{d_{grid} \times 4 \times 4} \) vector array, where each cell is represented using a \( d_{grid} \)-dimensional encoding. In our experiments, we use the latter representation. Details about gComm are provided in Appendix B.

4.2 Model Description
Recall that the listener has access to the grid representation, but not the target or the task-specifics. The speaker receives the instruction and relies on the listener to accomplish the said tasks. The speaker uses a single-layer LSTM followed by a linear layer to map the concept input to a hidden representation \( \mathbb{R}^{n_m \times d_h} \). From this representation, a set of one-hot encoded messages \( m^{\text{in}} \in \{0, 1\}^{d_m} \) are sampled (during training) using Categorical sampling, which are then concatenated and transmitted over the communication channel. Here, \( d_m \) is the dimension of the one-hot message \( m_i \). The number of messages \( n_m \) is set to \( \|\{\text{VERB}, \{\text{ADJ}_i\}_{i=1}^3, \text{NOUN}\}\| \). During evaluation, sampling is replaced with an arg max(.) operation. We use the Straight Through trick [Jang et al., 2017] to retain differentiability. At each step, the grid input is mapped in the Grid Encoder to an output \( G_i \in \mathbb{R}^{d_g \times 4 \times 4} \), using a \( 1 \times 1 \) convolution network. The received (concatenated) messages are projected to \( z \in \mathbb{R}^{1 \times d_g} \) using a linear layer. Next, we compute the attention weights \( \alpha_i \) for each grid cell by taking a normalized dot product between \( z \) and \( G_i \). A weighted combination is then fed to the sub-policy networks. Note that the messages are generated only at the beginning of the episode.

The listener learns to (i) identify the target object in the context of distractors, (ii) interact with the target object by executing a task specified by the speaker. As shown in Figure 1, we use a hierarchical-RL framework [Sutton et al., 1999] for training. There are two sub-policies corresponding to the PUSH and the PULL tasks. In each round, the master policy selects either sub-policies using the received (concatenated) messages\(^5\). Note, that the both PUSH & PULL tasks require the listener to walk to the target object, hence, the WALK task is executed using either of these sub-policies chosen randomly (provided, the master policy takes a `Null’ action). As shown in Figure 4.2, the sub-policies have a shared input, which includes the grid encoder and the attention network. The whole framework is trained end-to-end using REINFORCE algorithm [Williams, 1992].

In order to induce a more efficient training, we keep a measure of the Learning Progress (LP) of the listener for all tasks on a held-out set, where LP for task \( i \) is given as \( \text{LP}_i = |r_i - \mu_i| \). Here, \( \mu_i \) denotes the running mean of rewards for task \( i \). The tasks are sampled from a Categorical distribution with probabilities \( p(i) = \frac{|\text{LP}_i|}{\sum_j |\text{LP}_j|} \) and, consequently, episodes corresponding to the sampled tasks are generated. This way, the listener can keep track of goals that are already learned, or can insist on goals that are currently too hard. In the beginning, LP is initialized by training three independent models on a small number of episodes, corresponding to three different tasks (WALK, PUSH, PULL). Hyperparameter details are provided in Appendix A.3.

4.3 Inducing Compositionality
We would ideally want the concept to message mapping to be injective (one-to-one), i.e. \( \forall c, \hat{c} \in \mathcal{C}, \mathcal{M}(c) = \mathcal{M}(\hat{c}) \implies c = \hat{c} \). In other words, distinct elements in the concept space should be mapped to distinct symbols in the message space. Furthermore, the messages in \( \mathcal{M} \) must exhibit a systematic structure to be fully compositional (for instance, in holistic languages [Ren et al., 2020], one can satisfy the injective property without being compositional). Studies on lan-

\(^{5}\text{actions spaces: master policy: } \{A, B, \text{Null}\}; \text{subpolicy } A/B: \{\text{left, right, forward, backward, push/pull}\} \)
guage evolution have proposed limiting the channel capacity of communication, and thus, the vocabulary size, as an important constraint for achieving compositionality [Nowak and Krakauer, 1999]. Indeed, recent works [Mordatch and Abbeel, 2018; Chaabouni et al., 2020] have demonstrated that by having $\frac{|C|}{|C'|} = 1$, better generalization can be achieved ( $|C|$: Channel capacity; $|C'|$: cardinality of concept set).

Yet, in the course of our experiments, on increasing $|C|$, we observed rather predictably that, with a limited channel capacity, it becomes increasingly difficult for the speaker to converge upon a consistent and unambiguous mapping from $C$ to $M$. Consequently, the listener would either ignore the information from the speaker (speaker abandoning), or may exploit the inadequate information (undercoverage) to converge on a local optimum (learning a fixed sequence of actions, thus, acquiring a small reward). Hence, it fails to provide a meaningful feedback to the speaker, thus, hampering the emergence of compositional protocols. To that end, we propose two types of intrinsic rewards to address these issues.

Undercoverage: The limited channel capacity acts as an information bottleneck, impeding the speaker’s ability to transmit, unambiguously, the complete input information. In other words, the speaker fails to map each element in the input to a distinct message in $M$. Hence, it becomes difficult for the listener to infer the decoded messages at its end. To address this issue, we formulate a notion of compositionality from recent works in disentanglement [Higgins et al., 2017]. We propose to use the Mutual Information (MI) between the concepts and the messages $I(C, M)$ as an intrinsic reward:

$$I(C, M) = H(C) - H(C|M)$$

$$= H(C) + \sum_{m} p(m) \left( \sum_{c} p(c|m) \log p(c|m) \right)$$

$$= H(C) + E_{c \sim C, m \sim M(c)} \log p(c|m)$$

7Inspired by machine translation works [Tu et al., 2016], we define coverage as a mapping from a particular concept element to its appropriate message element. Full coverage refers to a distinct mapping of the whole concept input to corresponding symbols in $M$.

Given that the training episodes are generated independent of the object specifications, $H(C)$ can be assumed to be constant. We approximate the last term using Jensen's inequality $E_{c \sim C, m \sim M(c)} \left[ \log p(c|m) \right] \geq E_{c \sim C, m \sim M(c)} \left[ \log q_{\phi}(c|m) \right]$ to obtain a lower bound for $I(C, M)$. Here, $q_{\phi}(c|m)$ is a learned discriminator module which takes the (concatenated) messages and tries to predict the concept labels (i.e. elements of $\{\text{VERB}, \{\text{ADJ}\}, \{\text{NOUN}\}\}$) and $E_{c \sim C, m \sim M(c)} \log q_{\phi}(c|m)$ is its negative cross-entropy loss. The final intrinsic reward is calculated as follows:

$$I(C, M) \geq H(C) + E_{c \sim C, m \sim M(c)} \log q_{\phi}(c|m) \quad (1)$$

Intuitively, it suggests that it should be easy to infer the concepts from the messages. Conversely, the confusion (high error) arising from the speaker’s inability to express concepts will lead to lower rewards. Note, that the reward will be highest when the conditions of full coverage and one-to-one mapping are satisfied (the discriminator will then be able to predict all the concept elements with high probability).

We add the $I(C, M)$ reward at the last step of the episode, given as: $r[-1] + \lambda_1 I(C, M)$, where $\lambda_1$ is a tunable hyperparameter. The discriminator $q_{\phi}$ is periodically trained using batches sampled from a memory buffer, where we store the pair $(c_i, m_i)$. Note, that we block the discriminator gradients to the speaker and use it merely as an auxiliary means to provide intrinsic feedback to the whole framework.

Speaker Abandoning: Existing works [Lowe et al., 2019] have shown that while training RL-agents augmented with a communication channel, it is likely that the speaker fails to influence the listener’s actions. We hypothesize that this could be due to the following: (i) information bottleneck imposed due to discretization and limited channel capacity [Kharitonov et al., 2020]; (ii) the dimensionality gap between the grid-view (high dimension) and received messages. To address this, we propose to add another intrinsic reward to maximize the mutual information between the speaker’s messages and the listener’s actions, given the grid information.

At each step, we simulate $k$ intermediate steps to sample pseudo messages $\tilde{m}$ from the message distribution $M$. Together with the original message $m$, we compute two sets of
probability values corresponding to actions of the listener: (i) \( \pi(a_t|m, G_t) \) or the probability distribution over listener’s policy conditioned on both the messages and the output of the grid encoder \( G_t \); (ii) \( p(a_t|G_t) \) or the probability distribution over the listener’s actions conditioned on just the output of the grid encoder. We then calculate the mutual information for each step as follows:

\[
I(a_t, m|G_t) = \sum_{a_t, m} p(a_t, m|G_t) \log \frac{p(a_t, m|G_t)}{p(a_t|G_t)p(m|G_t)}
\]

\[
= \sum_{a_t, m} p(m|G_t)p(a_t|m, G_t) \log \frac{p(a_t|m, G_t)}{p(a_t|G_t)}
\]

\[
= E_{m \sim \mathcal{M}}[D_{KL}(p(a_t|m, G_t)||p(a_t|G_t))]
\]

Note that \( p(m|G_t) = p(m) \) since messages and grid-view are independently processed. Here \( p(a_t|G_t) \) is obtained by marginalizing over the joint probability distribution, given as, \( \sum_{a_t, m} p(a_t, m|G_t) = \sum_{a_t} p(a_t|m, G_t)p(m) \). We use Monte Carlo approximation to replace the Expectation by sampling messages from \( \mathcal{M} \). The final reward equation for \( k \) pseudo-steps is given as:

\[
I(a_t, m|G_t) = \frac{1}{k} \sum_{m} D_{KL} [\pi(a_t|m, G_t)|| \sum_{m} \pi(a_t|m, G_t)p(m)]
\]

Maximizing Equation 2 leads to a higher speaker influence on the listener’s actions. The net reward at each step is given as: \( r_t + \lambda_3 \sum_{m} \pi(a_t|m, G_t) \), where \( \lambda_3 \) is a tunable hyperparameter. Our proposed reward differs slightly from that of [Jaques et al., 2019] on measuring social influence by repeatedly maximizing the mutual information between action pairs of distinct agents over all time-steps. In contrast, we consider a single interaction between the speaker and the listener, on a limited channel capacity, which makes it highly likely for the listener to completely abandon the speaker, instead of strategically ignoring it at certain time-steps.

5 Experiments

5.1 Generalization Splits

Given a compositional language embodied in perception of the listener: (i) the speaker should be able to refer to unseen combinations and; (ii) the listener should be able to ground the transmitted messages to the objects in its observation and interact with them in novel ways. To that end, we test our models for zero-shot generalization capabilities on the following splits.

Visual split: All episodes not containing the ‘red square’ as a target object, were used for training the model. For instance, the training set contains instructions like walk to a red circle or push a yellow square with the ‘red square’ being used as a distractor. During evaluation, we examine whether the trained model can generalize to the following instructions: walk to a red circle; push a red square; pull a red square.

Numeral split: The training set contains instructions with Push, Push Twice and Pull, whereas, test set contains Pull Twice task. Here the modifier Twice is used when the listener needs to act on a heavier object, thus requiring two units of force, i.e., the object would move only if the listener executes two consecutive pull actions. In order to preclude the listener from figuring out the weight of the objects from the size (in the grid representation), we separate the size and weight attributes, such that, the weight is fixed randomly in each episode. Therefore, it becomes imperative for the listener to depend on the speaker for the weight information. Moreover, it must infer from its training that a symbol corresponding to heavy requires twice as many actions.

5.2 Baselines

We compare our Intrinsic Speaker model with the following baselines to highlight the significance of our contributions.

Oracle Listener: For each cell, we zero-pad the grid encoding with an extra bit, and set the bit (= 1) for the cell containing the target object. This way, the listener has complete information about the target in context of the distractors. We use this baseline as our upper limit of performance.

Perfect Speaker: The speaker is represented using an Identity matrix that channels the input directly to the listener. Thus, it is perfectly compositional and helps us understand how perfect compositionality can lead to faster convergence.

![Figure 3: [Best viewed in color] Left: Comparison of Intrinsic Speaker with other baselines on a single policy module for WALK task. It can be observed that Intrinsic Speaker performs as well as the Perfect Speaker baseline. Right: Comparison of topsim metric of Intrinsic Speaker (with and without feedback) and Simple Speaker. All plots have been obtained by averaging the validation rewards obtained over 5 independent runs. [X-axis: 1 unit = 50 episodes]](image)

| Task                  | Model            | Zero-Shot Accuracy |
|-----------------------|------------------|--------------------|
| walk to a red square   | Simple Speaker   | 73.43%             |
|                       | Intrinsic Speaker| 80.24%             |
| push a red square      | Simple Speaker   | 67.17%             |
|                       | Intrinsic Speaker| 72.45%             |
| pull a red square      | Simple Speaker   | 66.80%             |
|                       | Intrinsic Speaker| 73.29%             |
| pull a red square twice| Simple Speaker   | 65.25%             |
|                       | Intrinsic Speaker| 69.77%             |

Table 1: Comparison of simple speaker and intrinsic speaker zero-shot performance on different splits (Section 5.1), showing that intrinsic feedback significantly increases the generalization efficacy.
Demonstration of Intrinsic Speaker on the numeral split for task PULL TWICE. Here, the green circle is heavy and doesn’t move on the first pull action, hence, the listener has to apply two units of force (TWICE) to pull it.

Simple Speaker: Here the speaker-listener is trained end-to-end without using the intrinsic rewards $I(C,M)$ and $I(a_t, m|G_t)$. This baseline helps in verifying the additional utility of the intrinsic rewards.

For ease of comparison, we use a single policy module (without the master policy) and train the intrinsic speaker and the baselines on a single task: WALK. In a separate study, we also highlight the utility of our hierarchical module when trained on all tasks.

6 Results

Through our experiments, we empirically demonstrate that a limited channel capacity cannot by itself induce compositionality, and that, it must be used alongside intrinsic rewards to provide additional incentives to the agents.

- As is evident from Figure 3, the proposed Intrinsic Speaker outperforms the Simple Speaker baseline in terms of both, convergence rewards and topsim score. In fact, the Intrinsic Speaker matches the performance of the Perfect Speaker, thus, showing that the emergent communication is highly compositional ($\approx 0.9$).

- The zero-shot generalization accuracy in Table 1 shows that the Intrinsic Speaker consistently outperforms the Simple Speaker on both Visual and Numeral splits.

It was observed that the symbol for ‘red square’ was a combination of symbols denoting ‘red’ and ‘square’ (see Table 2 in Appendix A). Additionally, we performed ablation experiments to investigate the characteristics of the proposed setup.

Hierarchical vs. Single Policy training: We compared our hierarchical module, trained on all tasks, with a single policy module, which performs all tasks using the same policy. As shown in Figure 5, Intrinsic speaker with a hierarchical module performs far better than its single policy counterpart.

Correlation between compositionality and Zero-shot performance: The objective of this paper is to encourage generalization to unseen combinations (in a zero-shot setting) by inducing compositionality. Therefore, it becomes imperative to establish that the two things are related, and that, compositionality leads to generalization. We plot the correlation between topsim and the zero-shot performance on the visual split. As shown in Figure 5, we get a high Pearson correlation coefficient $\rho$ of $0.75$ (correlation is statistically significant ($p < 0.01$)).

No external feedback setting: In order to test the effectiveness of intrinsic rewards in inducing compositionality, we trained the Intrinsic Speaker with no external reward from the environment. As shown in Figure 3 (right), the intrinsic rewards were alone capable of generating a topsim score of $\approx 0.6$. However, we also observed that validation performance significantly decreased in absence of the external rewards (Figure 3, left). We attribute it to the fact that the intrinsic rewards (in particular the coverage reward) are tailored towards encouraging more compositional protocols rather than helping the listener learn good exploration policies.

Attention analysis: We performed a qualitative analysis of the attention weights of the Intrinsic Speaker on episodes where it was not able to complete the task (reward = 0). In general, it was found that in most episodes, the listener was able to identify the target cell (highest attention value on the grid). We conclude that, even though more compositional protocols increases zero-shot performance, a perfect compositionality does not imply perfect generalization.

7 Conclusion

We introduced a new platform for language acquisition embodied in the agent’s perception. Using this platform, we demonstrated the role of intrinsic rewards in inducing compositionality in the communication channel and how the same can be leveraged in generalizing over a novel composition of object properties and acquire transferable skills. We believe this will foster future research in the field of conversational/interactive AI assistants.
References

[Baldassarre and Mirolli, 2013] Gianluca Baldassarre and Marco Mirolli. Intrinsically Motivated Learning Systems: An Overview, pages 1–14. Springer Berlin, 2013.

[Brighton and Kirby, 2006] Henry Brighton and Simon Kirby. Understanding linguistic evolution by visualizing the emergence of topographic mappings. Artif. Life, 2006.

[Chaabouni et al., 2019] Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Anti-efficient encoding in emergent communication. CoRR, abs/1905.12561, 2019.

[Chaabouni et al., 2020] Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. CoRR, abs/2004.09124, 2020.

[Colas et al., 2020] Cédric Colas, Tristan Karch, Nicolas Lair, Jean-Michel Dussoux, Clément Moulin-Frier, Peter Ford Dominey, and Pierre-Yves Oudeyer. Language as a cognitive tool to imagine goals in curiosity-driven exploration. ArXiv, 2020.

[Gopnik et al., 2001] Alison Gopnik, Andrew Meltzoff, and Patricia Kuhl. The Scientist in the Crib: Minds, Brains and How Children Learn, volume 189. HarperCollins, 03 2001.

[Gupta and Dukkipati, 2019] Shubham Gupta and Anbedkar Dukkipati. On voting strategies and emergent communication. CoRR, 2019.

[Gupta et al., 2020] Shubham Gupta, Rishi Hazra, and Anbedkar Dukkipati. Networked multi-agent reinforcement learning with emergent communication. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS ’20, page 1858–1860. International Foundation for Autonomous Agents and Multiagent Systems, 2020.

[Havrylov and Titov, 2017] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17, page 2146–2156, 2017.

[Higgins et al., 2017] I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In International Conference on Learning Representations (ICLR) 2017, 2017.

[Jang et al., 2017] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In International Conference on Learning Representations (ICLR) 2017, 2017.

[Jaques et al., 2019] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A. Ortega, DJ Strouse, Joel Z. Leibo, and Nando de Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning, 2019.

[Kamp and Partee, 1995] Hans Kamp and Barbara Partee. Prototype theory and compositionality. Cognition, 1995.

[Kharitonov et al., 2020] Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. Entropy minimization in emergent languages. In Proceedings of the 37th International Conference on Machine Learning, 2020.

[Kirby, 2001] S. Kirby. Spontaneous evolution of linguistic structure-an iterated learning model of the emergence of regularity and irregularity. IEEE Transactions on Evolutionary Computation, 5(2):102–110, 2001.

[Kottur et al., 2017] Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge ‘naturally’ in multi-agent dialog. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2962–2967. Association for Computational Linguistics, 2017.

[Lazaridou et al., 2018] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In International Conference on Learning Representations (ICLR) 2018, 2018.

[Lazaridou et al., 2020] Angeliki Lazaridou, Anna Potapenko, and Olivier Tieleman. Multi-agent communication meets natural language: Synergies between functional and structural language learning, 2020.

[Lewis, 1969] David K. Lewis. Convention: A Philosophical Study. Wiley-Blackwell, 1969.

[Li and Bowling, 2019] Fushan Li and Michael Bowling. Ease-of-teaching and language structure from emergent communication. In Advances in Neural Information Processing Systems, volume 32, 2019.

[Lowe et al., 2017] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017.

[Lowe et al., 2019] Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, and Yann Dauphin. On the pitfalls of measuring emergent communication. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS ’19, page 693–701. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[Mordatch and Abbeel, 2018] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, pages 1495–1502, 2018.

[Nowak and Krakauer, 1999] Martin A. Nowak and David C. Krakauer. The evolution of language. Proceedings of the National Academy of Sciences, 1999.

[Ren et al., 2020] Yi Ren, Shangmim Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. In
[Sutton et al., 1999] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 1999.

[Tieleman et al., 2019] Olivier Tieleman, Angeliki Lazari-dou, Shibl Mourad, Charles Blundell, and Doina Precup. Shaping representations through communication: community size effect in artificial learning systems. *CoRR*, 2019.

[Tu et al., 2016] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 76–85. Association for Computational Linguistics, 2016.

[Williams, 1992] Ronald J Williams. Simple statistical gradient following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992.
The appendix is divided into two parts. In the first part (Appendix A), we provide additional details about the experimental setups used and inferences drawn from them. In the second part (Appendix B), we give an overview of the gComm environment and its features.

A Additional details

A.1 Discriminator Training

To encourage compositionality, we propose to train a discriminator \( q_\phi \) to predict the concepts \( c_i \) from the generated (concatenated) messages \( m_i \). The (negative) prediction loss is used as an intrinsic reward to prevent undercoverage. The discriminator is parameterized by a neural network with parameters \( \phi \). At the beginning of each episode, we store the pair \( \langle c_i, m_i \rangle \) in a memory buffer \( \mathcal{B} \). \( q_\phi \) is periodically trained using batches sampled from \( \mathcal{B} \). A \( \text{detach}(\cdot) \) operation is applied to the messages while storing in the buffer, thus preventing the gradients from the discriminator to backpropagate to the speaker.

\[
\mathcal{L}_\phi = \mathcal{L}(c_i|x) \quad \text{for each batch in \( \mathcal{B} \)}
\]

Figure 6: Discriminator training curve using cross-entropy loss.

A weighted loss is added as a reward at the very last step of the episode i.e. \( r[-1] + \lambda_1 (\lambda_2 - \mathcal{L}_\phi) \). Here, \( \mathcal{L}_\phi \) is the discriminator loss and \( \lambda_1 \) & \( \lambda_2 \) are tunable hyperparameters. As the loss \( \mathcal{L}_\phi \) decreases, the intrinsic reward increases, thus incentivizing the speaker to not only transmit the complete input information (full coverage), but also have a disentangled representation in the message space \( \mathcal{M} \).

Derivation of Equation 1

We approximate \( p(c|m) \) by its lower bound using function approximation. Therefore, we need to minimize \( D_{KL}(p(c|m)||q_\phi(c|m)) \).

\[
D_{KL}(p(c|m)||q_\phi(c|m)) = \sum_{m\in\mathcal{M}} p(m) \sum_{c\in\mathcal{C}} p(c|m) \log \frac{p(c|m)}{q_\phi(c|m)}

= E_{c\sim\mathcal{C}, m\sim\mathcal{M}(c)} \log p(c|m)

- E_{c\sim\mathcal{C}, m\sim\mathcal{M}(c)} \log q_\phi(c|m) \geq 0
\]

| object | color | message symbol |
|--------|-------|----------------|
| circle | red   | aB             |
|        | blue  | aD             |
|        | yellow| aD             |
|        | green | aA             |
| square | red   | bB             |
|        | blue  | bD             |
|        | yellow| bC             |
|        | green | bA             |
| cylinder | red | cB             |
|         | blue  | cC             |
|         | yellow| cD             |
|         | green | cA             |

Table 2: Compositionality in messages transmitted by the speaker.

The correct concept to symbol mapping according to the developed protocol is: circle \( \mapsto \) “\( a \)”; square \( \mapsto \) “\( b \)”; cylinder \( \mapsto \) “\( c \)”; green \( \mapsto \) “\( A \)”; red \( \mapsto \) “\( B \)”; yellow \( \mapsto \) “\( C \)”; blue \( \mapsto \) “\( D \)”. It can be observed that even though the training never contained “red square” as a target object, it was transmitted correctly by the speaker using a combination of symbols for “red” and “square”. Note that some of the symbols are incorrectly represented: (i) the symbols for “yellow circle” is same as that of “blue circle”; (ii) the symbols for “blue cylinder” and “yellow cylinder” are switched.

A.2 Attention analysis:

Recall that the attention weights \( \alpha_{1:16}^{Gt} \) are obtained for each cell of the processed grid encoding \( Gt \) and the received messages projected to \( z \in \mathbb{R}^{1 \times d_2} \). Intuitively, these weights are an indication of whether the listener can locate the target object in the grid since the messages transmitted contain target attributes encoded in form of a discrete (symbolic) representation. In order to get a better understanding of the listener’s view of its grid on episodes where it failed to complete the task, we performed a qualitative analysis of these weights. We assign a colormap to the attention weights such that the shades of the cells become darker as one moves from lower attention weights to higher attention weights. As is evident from Figure 7, in most of the failed cases, the listener was able to correctly identify the target object (“red square”), yet, it was unable to complete the task: walk to the red square. This could be attributed to the fact that the exploration policy learned by the listener isn’t perfect even though the compositional protocols are.
Figure 7: Study of attention weights for Intrinsic Speaker on each cell of the grid. Darker shades represent higher attention. Task: *walk to the red square*; target: “red square”. It can be observed that the target object cell is correctly assigned the highest attention, yet the listener is unable to walk to the target location and keeps performs a repeated sequence to *forward* and *backward* actions.

### A.3 Hyperparameters

| **Speaker Model** |  |
| --- | --- |
| hidden dimension ($d_h$) | 4 |
| message dimension ($d_m$) | 4 |
| temperature parameter (Categorical sampling) | 1 |
| number of messages transmitted (for single-task/multi-task) ($n_m$) | 2/4 |
| learning rate with Adam optimizer | 4e-4 |

| **Listener Model: Grid Encoder** |  |
| --- | --- |
| output dimension ($d_G$) (for single task setup) | 12 |
| learning rate with Adam optimizer | 4e-4 |

| **Listener Model: Policy Module** |  |
| --- | --- |
| action space for single-task/multi-task setup | 4/5 |
| action space of master policy (for multi-task setup) | 3 |
| learning rate with Adam optimizer | 4e-4 |

| **Listener Model: Attention Network** |  |
| --- | --- |
| learning rate with Adam optimizer | 4e-4 |

| **Discriminator** |  |
| --- | --- |
| size of memory buffer $B$ | 500 |
| training batch size | 50 |
| number of batches sampled from $B$ for training the discriminator | 1 |
| learning rate with Adam optimizer | 1e-2 |
| period (of retrain) | 20 |
| loss function used for training | Cross-entropy loss |

| **Intrinsic Rewards** (for single-task setup) |  |
| --- | --- |
| Undercoverage reward parameter $\lambda_1$ | 0.01 |
| Undercoverage reward parameter $\lambda_2$ | 2.80 |
| Speaker Abandoning reward parameter $\lambda_3$ | 0.01 |
| number of pseudo-steps $k$ to sample messages in Speaker Abandoning reward calculation | 10 |
B grounded-Comm Environment

gComm is a step towards developing a robust platform to foster research in grounded language acquisition in a more challenging and realistic setting. It comprises a 2-d grid environment with a set of agents (a stationary speaker and a mobile listener connected via a communication channel) exposed to a continuous array of tasks in a partially observable setting. The key to solving these tasks lies in agents developing linguistic abilities and utilizing the same for efficiently exploring its environment. The speaker and listener have access to information provided in two different modalities, i.e. the speaker’s input is a natural language instruction that contains the target and task specifications and the listener’s input is its grid view. Each must rely on the other to complete the assigned task, however, the only way they can achieve the same, is to develop and use some form of communication. gComm provides several tools for studying different forms of communication and assessing their generalization performance.

Object Attributes: The gComm grid-world is populated with objects of different characteristics like shape, color, size and weight. Following is an exhaustive list of attributes of the objects in gComm:

- **Shapes:** circle, square, cylinder, diamond
- **Colors:** red, blue, yellow, green
- **Sizes:** 1, 2, 3, 4
- **Weights:** light, heavy

The weight attribute can be fixed corresponding to the object size at the beginning of training. For instance, smaller sized objects are lighter and vice versa. Alternatively, the weight can be set as an independent attribute. In the latter option, the weight is randomly fixed at the start of each episode so that the listener cannot deduce the same from the grid information, and must rely on the speaker.

B.1 Reinforcement Learning framework

Setup: In each round, a task is assigned to a stationary Speaker-Bot, the details of which (task and target information) it must share with a mobile Listener-Bot by transmitting a set of messages \( m_i \), via a communication channel. At each step, the listener agent selects an action from its action space \( A \) with the help of the received messages \( m_i \) and its local observation (grid-view) \( a_i \in O \). The environment state is updated using the transition function \( T: S \times A \rightarrow S \). The environment provides a reward to the agent at each time-step using a reward function \( r: S \times A \rightarrow \mathbb{R} \). The goal of the agent is to find a policy \( \pi : \mathcal{O} \rightarrow \Delta(A) \) that chooses optimal actions so as to maximize the expected reward, \( R = \mathbb{E}_\pi[\sum_{t=1}^{T} \gamma^t r(t)] \) where \( r(t) \) is the reward received by the agent at time-step \( t \) and \( \gamma \in (0, 1] \) is the discount factor. At the beginning of training, their semantic repertoires are empty, and the speaker and listener must converge on a systematic usage of symbols to complete the assigned tasks, thus, giving rise to an entirely original linguistic system.

Observation Space: To encourage communication, gComm provides a partially observable setting, in which, neither the speaker nor the listener has access to the complete state information. The speaker has access to the target and the task specifics through the natural language instruction, whereas, the listener has access to the grid representation. However, the listener is unaware of either the target object or the task it is supposed to perform. Hence, it must rely on the speaker to accomplish the given task. The observation space of the listener comprises (i) the grid representation; (ii) the message transmitted by the speaker.

The natural language instruction is parsed to \((\text{VERB}, \text{ADJ}, \text{NOUN})_i \) with the help of a semantic parser\(^8\). This, in turn, is converted to the following 18-d vector representation before being fed to the speaker: \( \{1, 2, 3, 4, \text{square}, \text{cylinder}, \text{circle}, \text{diamond}, r, b, y, g, \text{light}, \text{heavy}, \text{walk}, \text{push}, \text{pull}, \text{pickup}\} \). Each position represents a bit and is set or unset according to the attributes of the target object and the task. The breakdown of the vector representation is as follows: bits \([0 - 3]\): target size; bits \([4 - 7]\): target shape; bits \([8 - 11]\): target color; bits \([12 - 13]\): target weight; bits \([14 - 17]\): task specification.

The grid information can either be a image input of the whole grid or a predefined cell-wise vector representation of the grid. In the latter case, each grid cell in is specified by a 17-d vector representation given by: \( \{1, 2, 3, 4, \text{square}, \text{cylinder}, \text{...}\} \)
On similar lines as the concept representation, each position represents a bit and is set or unset according to the attributes of the object in the given cell. The breakdown of the vector representation is as follows: bits \([0 − 3]:\) object size; bits \([4 − 7]:\) object shape; bits \([8 − 11]:\) object color; bit 12: agent location (is set = 1 if agent is present in that particular cell, otherwise 0); bits \([13 − 16]:\) agent direction. For an obstacle or a wall object, all the bits are set to 1.

**Action Space:** gComm has a discrete action space which comprises eight different actions that the listener agent can perform: \{left, right, forward, backward, push, pull, pickup, drop\}. In order to execute the ‘push’, ‘pull’, and ‘pickup’ actions, the agent must navigate to the same cell as that of the object. Upon executing a pickup action, the object disappears from the grid. Conversely, an object that has been picked up can reappear in the grid only if a ‘drop’ action is executed in the same episode. Further details about task descriptions are provided in Section B.2.

**Rewards:** gComm generates a 0-1 (sparse) reward, i.e., the listener gets a reward of \(r = 1\) if it achieves the specified task, otherwise \(r = 0\).

**Communication:** Recall that the listener has incomplete information of its state space and is thus unaware of the task and the target object. In order to perform the assigned task, the listener must rely on the speaker agent for the required information. Since the only way of sharing information is via the communication channel, the speaker must learn to use the same while transmitting information. What makes it more challenging is the fact that this information uses discrete symbols, the semantics of which must be learned in a sparse reward setting, i.e., to solve the tasks, the speaker and the listener must converge upon a common protocol and use it systematically with minimal feedback at the end of each round. Refer to Section B.3 for further details about the communication types and the channel parameters.

**B.2 Task Description**

**Tasks:** The task descriptions are as follows:

- **Walk:** Walk to a target object
- **Push:** Push a target object in the forward direction.
- **Pull:** Pull a target object in the backward direction.
- **Pickup:** Pickup the target object.
- **Drop:** Drop the picked up object.

Additionally, there are modifiers associated with different verbs, for instance: **pull the red circle twice.** Here, twice is a numeral adverb and must be interpreted to mean two consecutive ‘pull’ actions. When an object is picked up, it disappears from the grid and appears only if a ‘drop’ action is executed in the subsequent time-steps. However, no two objects can overlap. It should be noted that while defining tasks, it is ensured that the target object is unique.

**Target and Distractor objects:** Cells in the grid-world are populated with objects characterized by certain attributes, which are divided into two classes: the target object and the distractor objects. The distractors either have the same color or the same shape (or both) as that of the target. Apart from these, some random objects distinct from the target, can also be sampled using a parameter \(other\_objects\_sample\_percentage\). The listener and the objects may spawn at any random location on the grid.

**Levels:** In addition to the simple grid-world environment comprising target and distractor objects, the task difficulty can be increased by generating obstacles and mazes. The agent is expected to negotiate the complex environment in a sparse reward setting. The number of obstacles and the maze density can be varied.

**Instruction generation:** Natural language instructions are generated based on predefined lexical rules and the specified vocabulary. At the beginning of training, the user specifies the kind of verb (transitive or intransitive), noun (object shape), and adjectives (object weight, size, color). Accordingly, the instructions are generated, thus, simulating a human-machine interface.

**B.3 Communication**

gComm endows the agents with the ability to communicate. This forms a crucial step in addressing the partial observability problem and encouraging language acquisition. Above all, gComm provides several tools for an in-depth analysis of different types of grounded communication protocols and their relation to the generalization performance in agents.

**Communication Channel:** The communication can be divided into two broad categories.

- **Discrete:** Discrete messages can either be binary (processed using Gumbel-Softmax [Jang et al., 2017]) or one-hot (processed using Categorical distribution)\(^9\).

\(^9\)The use of discrete latent variables render the neural network
Discrete messages are associated with a temperature parameter $\tau$.

- **Continuous**: As opposed to discrete messages, continuous signals are real-valued. Theoretically speaking, each dimension in the message can carry 32-bits of information (32-bit floating point). These messages do not pose the same kind of information bottleneck as their discrete counterpart, however, they are not as interpretable.

Apart from these, the communication channel can be utilized to compare against the following baseline implementations readily available in the gComm environment. These baselines not only enable us to investigate the efficacy of the emergent communication protocols, but also provides quantitative insights into the learned communication abilities, on similar lines as [Lowe et al., 2019].

- **Random**: In this baseline, the speaker transmits a set of random symbols to the listener to try and distract it. The listener must learn to ignore these symbols and focus only on its local observation.
- **Fixed**: In fixed communication, the speaker’s transmissions are masked with a set of ones. Intuitively, this baseline provides an idea of whether the emergent communication is being used in the context of the given task (whether the speaker actually influences the listener or just appears to do so).
- **Perfect**: This baseline provides an illusion of a perfect speaker by directly transmitting the input concept encoding, hence, acting as an upper bound for comparing the learned communication protocols.

**Channel parameters**: The communication channel is defined using the following parameters:

- **Message Length**: Length of the message vector $n_m$ sets a limit on the vocabulary size, i.e. higher the message length, larger is the vocabulary size. For instance, for discrete (binary) messages, the vocabulary size is given by $|V| = 2^{d_m}$. Here $d_m$ is the message length. Note, that a continuous message can transmit more information compared to a discrete message of the same length.
- **Information Rate**: It is defined as the number of messages $n_m$ transmitted per round of communication.

These parameters constitute the channel capacity, $|C| = c_{d_m}^{n_m}$.

**Setting**: Communication can either be modelled in form of cheap talk or costly signalling. In the latter case, each message passing bears a small penalty to encourage more economic and efficient communication protocols. Alternatively, the communication can be either unidirectional (message passing from speaker to listener only) or bidirectional (an interactive setting wherein message passing happens in either direction, i.e. the speaker and listener roles can be switched). gComm models communication in an unidirectional cheap talk setting.

**B.4 Metrics:**

In order to induce meaningful communication protocols, the speaker must transmit useful information, correlated with its input (positive signalling). At the same time, the listener must utilize the received information to alter its behavior and hence, its actions (positive listening). In alignment with the works of [Lowe et al., 2019], we incorporate the following metrics in our environment to assess the evolved communication protocols.

- **Positive signalling**: Context independence (CI) is used as an indicator of positive signalling. It captures the statistical alignment between the input concepts and the messages transmitted by the speaker and is given by:

\[
CI(p_{cm}, p_{cm}) = \frac{1}{|C|} \sum_{c} p_{cm}(c|m)p_{mc}(m|c)
\]

Both $p_{cm}(c|m)$ and $p_{mc}(m|c)$ are calculated using a translation model by saving $(m, c)$ pairs and running it in both directions. Since each concept element $c$ should be mapped to exactly one message $m$, CI will be high when the $p_{cm}(c|m)$ and $p_{mc}(m|c)$ are high.

- **Positive listening**: We use Causal Influence of Communication (CIC) of the speaker on the listener as a measure of positive listening. It is defined as the mutual information between the speaker’s message and the listener’s action $I(m, a)$. Higher the CIC, more is the speaker’s influence on the listener’s actions, thus, indicating that the listener is utilizing the received messages.

- **Compositionality**: Compositionality is measured using the topographic similarity (topsim) metric [Brighton and Kirby, 2006]. Given two pairwise distance measures, i.e. one in the concept (input) space $\Delta_c^{ij}$ and another in the message space $\Delta_M^{ij}$, topsim is defined as the correlation coefficient calculated between $\Delta_c^{ij}$ and $\Delta_M^{ij}$. Higher topsim indicates more compositional protocols.
B.5 Additional features

Lights Out: We introduce a *lights out* feature in the gComm environment through which the grid (including all its objects) is subjected to varying illuminations (Figure 10). The feature can be activated randomly in each episode and presents a challenging situation for the agent where it is required to navigate the grid using its memory of the past observation. Note that this feature is useful only when used with an image input as the grid representation.