# On the Pitfalls of Measuring Emergent Communication

Ryan Lowe
Mila & Facebook AI Research
Montreal, QC
ryan.lowe@cs.mcgill.ca

Jakob Foerster
Facebook AI Research
Menlo Park, CA

Y-Lan Boureau
Facebook AI Research
New York, NY

Joelle Pineau
Mila & Facebook AI Research
Montreal, QC

Yann Dauphin
Google AI
Accra, Ghana

## ABSTRACT

How do we know if communication is emerging in a multi-agent system? The vast majority of recent papers on emergent communication show that adding a communication channel leads to an increase in reward or task success. This is a useful indicator, but provides only a coarse measure of the agent's learned communication abilities. As we move towards more complex environments, it becomes imperative to have a set of finer tools that allow qualitative and quantitative insights into the emergence of communication. This may be especially useful to allow humans to monitor agents' behaviour, whether for fault detection, assessing performance, or even building trust. In this paper, we examine a few intuitive existing metrics for measuring communication, and show that they can be misleading. Specifically, by training deep reinforcement learning agents to play simple matrix games augmented with a communication channel, we find a scenario where agents appear to communicate (their messages provide information about their subsequent action), and yet the messages do not impact the environment or other agent in any way. We explain this phenomenon using ablation studies and by visualizing the representations of the learned policies. We also survey some commonly used metrics for measuring emergent communication, and provide recommendations as to when these metrics should be used.

## KEYWORDS

Deep learning; Learning agent capabilities; Multi-agent learning

## 1 INTRODUCTION

Communication through language is one of the hallmarks of human intelligence; it allows us to share information efficiently between humans and coordinate on shared tasks. This is a strong motivation to develop machines that can communicate, so that they can share their knowledge with us and help us coordinate on tasks where humans alone perform poorly. One of the approaches for developing agents capable of language is through *emergent communication*. The idea is to use an environment consisting of randomly initialized agents and a dedicated communication channel, which the agents can learn to use to accomplish tasks that require coordination. This approach is motivated by a utilitarian view of communication, where language is viewed as one of many tools an agent can use towards achieving its goals, and an agent 'understands' language once it can use it to accomplish these goals [15, 45]. The question of emergent communication is also of scientific interest, as it could provide insights into the origins of human language [32].

Recently, there has been a significant revival of emergent communication research using methods from *deep reinforcement learning* (deep RL). In this line of work, researchers define a multi-agent environment where agents possess a (usually discrete and costless) communication channel, and train the agents' policies (represented by deep neural networks) to maximize the cumulative reward. In most cases, the multi-agent environment is a cooperative *referential game*, where one agent (the 'speaker') has some private information about a public observation, and must communicate it accurately to the other agent (the 'listener') [7, 9, 11, 18, 23–25]. In other works, agents navigate a simulated world with a limited field of view, and communication can help the agents coordinate to achieve a common goal [4, 13, 21, 39]. Algorithmic development has led to emergent communication in games with pixel-based inputs [4, 7, 24], in settings incorporating human language [25, 27], and even in more complex games that require inferring the belief state of the opponent, like Hanabi [13].

Despite these advances, there has been no formal study evaluating *how* we measure emergent communication. Most papers in this area show that adding a communication channel helps the agents achieve a higher reward, and attempt to understand the communication protocols qualitatively by, for example, plotting the distribution over messages in various states of the environment. This is a reasonable approach: if adding a communication channel increases the reward in an environment, then the agents are making some use of that channel. But it is useful to quantify more finely the degree of communication as it can provide more insights into agent behaviour. This may be essential for human monitoring; we want to understand why agents are making certain decisions, and understanding their language would make this significantly easier. Our aim is to develop a set of evaluation tools for detecting and measuring emergent communication that increases the robustness of emergent communication research on the way to developing agents that communicate with humans.

In this paper, we take a step in this direction by analyzing some intuitive metrics and tests that have been used in the recent literature for measuring emergent communication. In particular, we show that some of these metrics, including speaker consistency [21] and simply examining the agents' policies qualitatively, can be misleading if not properly understood. We use a simple class of matrix games augmented with a bidirectional communication channel, which we call Matrix Communication Games (MCGs), to study the emergence of communication among deep RL agents. MCGs are simple and efficient for learning, and have the appealing property that the resulting communication policies are interpretable relative to more complex gridworld environments.

To aid in our analysis, we categorize emergent communication metrics into two broad classes: those that measure *positive signaling*, which indicates that an agent is sending messages that are related in some way with its observation or action; and those that measure *positive listening*, indicating that the messages are influencing the agents' behaviour in some way. Both positive signaling and positive listening are desirable; we want to develop communicative agents that can both speak about their world and intentions, and can interpret the messages of others and respond accordingly. While intuitively it might seem that positive listening is a prerequisite for positive signaling (otherwise, why would the agent signal in the first place?), we show that, surprisingly, this is not always true for deep RL agents; it is possible for positive signaling to emerge as a byproduct of the task and policy architecture.

Our main contributions are as follows.

- We briefly survey the existing metrics for detecting emergent communication in the deep RL community, and find that the majority measure positive signaling rather than positive listening.
- We run experiments on MCGs and show that agents can exhibit positive signaling without positive listening. This lack of positive listening falsifies the hypothesis that the agents are communicating to share information to increase the reward of both agents. Thus, positive listening is an important tool for understanding the purpose of emergent communication.
- We propose a metric, the causal influence of communication (CIC), designed to directly measure positive listening. Note that a variant of CIC was developed concurrently by [21], where they showed that agents trained with causal influence as an additional reward performed better in several multi-agent settings. We adopt their terminology in this paper.
- We discuss other metrics and tests for measuring emergent communication, and make recommendations for future work in this area.[1]

## 2 BACKGROUND

### 2.1 Markov games

The MCG environment in this paper can be considered a multi-agent extension of partially observable Markov decision processes (POMDPs) called partially observable Markov games [29]. A Markov game for $N$ agents is defined by a set of states $\mathcal{S}$, $N$ sets of actions

$\mathcal{A}_1, ..., \mathcal{A}_N$ and $N$ sets of observations $O_1, ..., O_N$, one each per agent. To choose actions, each agent $i$ uses a stochastic policy $\boldsymbol{\pi}_{\theta_i} : O_i \times \mathcal{A}_i \mapsto [0, 1]$, which produces the next state according to the state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \mapsto \mathcal{S}$. Each agent $i$ obtains rewards as a function of the state and agent's action $r_i : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \mapsto \mathbb{R}$, and receives a private observation correlated with the state $o_i : \mathcal{S} \mapsto O_i$. The initial states are determined by a distribution $\rho : \mathcal{S} \mapsto [0, 1]$. Each agent $i$ aims to maximize its own total expected return $R_i = \sum_{t=0}^{T} \gamma^t r_i^t$ where $\gamma$ is a discount factor and $T$ is the time horizon.

More specifically, in the games we consider the action space $\mathcal{A}_i$ for each agent $i$ can be subdivided into disjoint *environment actions* $\mathcal{A}_i^e$, and *communication actions* $\mathcal{A}_i^m$, such that $\mathcal{A}_i^e \cup \mathcal{A}_i^m = \mathcal{A}_i$ and $\mathcal{A}_i^e \cap \mathcal{A}_i^m = \emptyset$. Environment actions are those that have a direct effect on the environment dynamics and the rewards obtained by the agent. We model each agent's communication action as a sequence of discrete symbols sent over a dedicated communication channel, which are observed by the other agents at the next time step. Communication actions do not affect the environment dynamics (other than being observed by the other agent), and incur a fixed cost $r_c \in \mathbb{R}_{<0}$. In our paper, we consider the *cheap talk* setting [12] where $r_c = 0$, and leave an examination of costly signaling (e.g. [16, 46]) to future work.

### 2.2 Policy gradient methods

Policy gradient methods are a popular choice for a variety of RL tasks. The main idea is to directly adjust the parameters $\theta$ of the policy in order to maximize the objective $J(\theta) = \mathbb{E}_{s \sim p^{\boldsymbol{\pi}}, a \sim \boldsymbol{\pi}_\theta}[R]$ by taking steps in the direction of $\nabla_\theta J(\theta)$. The gradient of the policy can be written as [40]:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s \sim p^{\boldsymbol{\pi}}, a \sim \boldsymbol{\pi}_\theta}[\nabla_\theta \log \boldsymbol{\pi}_\theta(a|s) Q^{\boldsymbol{\pi}}(s, a)], \quad (1)$$

where $p^{\boldsymbol{\pi}}$ is the state distribution, and $Q^{\boldsymbol{\pi}}(s, a)$ is an estimate of the expected value of taking action $a$ in state $s$.

The policy gradient theorem has given rise to several practical algorithms, which often differ in how they estimate $Q^{\boldsymbol{\pi}}$. In the simplest case, one can use a sample return $R^t = \sum_{i=t}^{T} \gamma^{i-t} r_i$, which leads to the REINFORCE algorithm [43]. We use the REINFORCE algorithm in our work, as it is simple, and has been by far the most popular approach among emergent communication works using deep RL [6, 9, 18, 21, 23–25, 39].

## 3 EMERGENT COMMUNICATION METRICS

### 3.1 A categorization of metrics

When analyzing metrics to measure a certain quantity, it is important to ask what that quantity actually represents. What does it mean for agents to be communicating with each other in a reinforcement learning setting? We take a pragmatic perspective, and identify two broad prerequisites for communication to occur: (1) one agent needs to produce a signal that is in some way correlated with its observation or intended action, and (2) another agent needs to update its belief or alter its behaviour after observing the signal.

We define *positive signaling* as behaviour that satisfies criterion (1), and *positive listening* as behaviour that satisfies criterion (2). To formalize these intuitions, we provide very broad definitions these terms below.

*Definition 3.1 (Positive Signaling).* Let $\bar{m} = (m_0, m_1, ..., m_T)$ be the sequence of messages sent by an agent over the course of a trajectory of length $T$, and similarly for $\bar{o} = (o_0, o_1, ..., o_t)$ and $\bar{a} = (a_0, a_1, ..., a_T)$. An RL agent exhibits positive signaling if either $\bar{m} \not\perp \bar{o}$ or $\bar{m} \not\perp \bar{a}$, i.e. if $\bar{m}$ is statistically dependent (indicated by $\not\perp$) in some way with either $\bar{a}$ or $\bar{o}$.

*Definition 3.2 (Positive Listening).* An RL agent exhibits positive listening if there exists a message generated by another agent $m \in \mathcal{A}_i^m$, for some $i \in \{1, ..., N\}$ such that $||\boldsymbol{\pi}(o, \mathbf{0}) - \boldsymbol{\pi}(o, m)||_\tau > 0$, where $\mathbf{0}$ is the 0 vector, and $|| \cdot ||_\tau$ is a distance in the space of expected trajectories followed by $\boldsymbol{\pi}$.

Evidently, these definitions are very loose, and most agents in an multi-agent environment with the capacity to communicate will satisfy them to some degree. However, we can speak of the degree or extent to which an agent exhibits positive signaling or positive listening behaviour, and measure this using metrics or tests. Thus, these terms are useful for categorizing different metrics of emergent communication.

## 3.2 What metrics are being used now?

We now conduct a brief review of the existing metrics being used in papers applying deep reinforcement learning to the problem of emergent communication, and categorize them as being indicative of positive signaling or positive listening. We focus on metrics that are the most prevalent in the recent deep RL literature to give an overview of how current research in this area is being conducted. From our review, we find that only one metric (instantaneous coordination) is explicitly designed to measure positive listening, and that it has several shortcomings. This motivates our definition of a new metric, the causal influence of communication, in Section 3.3.

*Reward and task completion.* As previously mentioned, in all papers we surveyed, the authors either measure the task completion % of their proposed algorithm (for referential games [11, 24, 25]), or show that adding a communication channel increases the agents' reward (for games where the agents interact with the world [4, 7, 39]). For referential games [9, 24], this is an adequate measure of communicative performance because these games are *non-situated.* In the terminology of [42], a non-situated environment (or simulation) is one where agents' actions consist solely of sending and receiving signals. Non-embodied agents in these environments do not have non-communicative actions that affect other objects or each other. Thus, if task success increases in this setting, it is likely because the sender agent has developed a better communication protocol, or the listener agent has become better at understanding it.

In *situated* environments [42], where agents have non-communicative actions that affect the environment and/or modify their internal state, an increase in reward is still a loose measure of both positive listening and positive signaling. If adding a communication channel leads to an increase in reward, then agents must be using that channel to transmit some kind of information that is affecting their behaviour. However, when comparing multiple algorithms on the same environment, an increase in reward may not necessarily indicate improved communication of one algorithm over another, as this increase could be simply due to the improved action policies. Thus, in this paper we advocate for more fine-grained

metrics for measuring the quality of emergent communication protocols. One of the reasons for our adoption of MCGs is that they are in some ways the simplest situated environment (agents have non-communicative interactions that affect each other's reward).

*Qualitative analysis of messages given states.* An equally common practice is to analyze the communication policies qualitatively, to interpret what the agents are saying. This is done most commonly by visualizing which messages were sent with which inputs or observations [4, 18, 21, 25, 39]. For very simple environments, such as the riddle game [14], the authors are able to make a full tree describing the agents' policies for all states in the game. However, for other games with larger state spaces, papers simply show some messages and inputs that the authors observed to have high co-occurrence. While this is a useful practice to understand the agents' behaviour, it is usually only testing positive signaling, as it does not analyze how the message changes the other agent's behaviour.

*Speaker consistency.* Some papers quantify the degree of alignment between an agent's messages and its actions [4, 21]. One such example is the *speaker consistency* (SC), proposed in [21]. The easiest way to understand the speaker consistency is as the mutual information between an agent's message and its future action:

$$SC = \sum_{a \in \mathcal{A}^e} \sum_{m \in \mathcal{A}^m} p(a, m) \log \frac{p(a, m)}{p(a)p(m)} \quad (2)$$

where the probabilities $p(a, m) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\{\text{act}=a, \text{comm}=m\}}$ are calculated empirically by averaging (message, action) co-occurrences over the $N$ episodes in each epoch.[2] We provide an algorithm for calculating SC in the Appendix.

The speaker consistency is a measure of positive signaling, as it indicates a statistical relationship between the messages and actions of an agent, and does not tell us anything about how these messages are interpreted. On the surface, SC is a useful measure of communication because it tells us how much an agent's message reduces the uncertainty about its subsequent action. Importantly, when an agent learns a deterministic policy independent of observations, the SC will be 0. Because of these appealing properties, we primarily focus on SC as our metric of comparison in our experiments.

*Context independence.* Context independence (CI), introduced by [4], is designed to measure the degree of alignment between an agent's messages and task concepts (corresponding to a particular part of the input space). It is calculated as follows:

$$\forall c: \quad m^c = \arg\max_m p_{cm}(c|m)$$

$$CI(p_{cm}, p_{mc}) = \frac{1}{|C|} \sum_c p_{mc}(m^c|c) \cdot p_{cm}(c|m^c)$$

where $C$ is the space of all concepts, $p_{cm}(c|m)$ is the conditional probability of a concept given a message, and similarly for $p_{mc}(m|c)$. This quantity relies on having a well-defined notion of 'concept' (in [4], this corresponds e.g. to the number and colour of objects), and a way of estimating $p_{cm}$ and $p_{mc}$ ([4] use an IBM model 1 [5]).

Context independence captures the same intuition as speaker consistency: if a speaker is consistently using a specific word to refer to a specific concept, then communication has most likely

---

[2]The SC metric in [21] is a normalized version of the metric shown here.

emerged. Thus, it is also a measure of positive signaling. The difference is that CI emphasizes that a single symbol should represent a certain concept or input, whereas a high speaker consistency can be obtained using a set of symbols for each input, so long as this set of symbols is (roughly) disjoint for different inputs.

*Entropy of message distribution.* Another common practice is to measure the perplexity or entropy ($H(\boldsymbol{\pi}^c)$) of the distribution over messages for different inputs [7, 11, 18]. Different papers interpret the meaning of this quantity differently, but generally if the message distribution has low entropy for a given input, then the speaker is consistently using the same message to describe that input. This differs from the speaker consistency metric, as it does not measure whether or not the speaker is using a *different* message for each input. The entropy actually measures neither positive signaling (an agent that always outputs the same message, independent of observation and action, will have low entropy), nor positive listening (it does not take into account the other agent's response to the messages), which gives it questionable utility as a metric.

*Instantaneous coordination.* Another metric considered in [21] is instantaneous coordination (IC). IC uses the same formula as for SC (Eq. 2), except it calculates the mutual information between one agent's message and the *other* agent's action, again by averaging (message, action) co-occurrences over episodes (see Appendix for algorithm). IC is a measure of positive listening; however, because of the way $p(a, m)$ is calculated, it only detects cases where a message from one agent changes the other agent's action regardless of context.

To illustrate why this is undesirable, we describe a simple matrix communication game (MCG) example (see Section 4.1 for description) where IC would fail to detect positive listening. Let us draw the entries of the payoff matrix $R \in \mathbb{R}^{2\times2}$ from a Gaussian $\mathcal{N}(0, 3)$ at each timestep (as we do in our experiments in Section 5), and fix agent 1 to always truthfully signal the action it will take. In this case, the optimal policy for agent 2 is to select the best response to agent 1's action, which it knows exactly because agent 1 always signals truthfully. Depending on the payoff matrix $R$, sometimes the best response will be to take action 1, and sometimes to take action 2. Clearly agent 2's policy exhibits significant positive listening (it changes its action depending on the message from agent 1 and the input). However, if $R$ is drawn randomly, when averaged across inputs agent 2 will take action 1 and 2 equally. The IC will then be 0 in expectation, since it calculates $p(a, m)$ by averaging over episodes and does not condition on the input $R$.

## 3.3 Causal influence of communication

We now propose a metric that more directly measures positive listening, which we call the causal influence of communication (CIC). This nomenclature follows concurrent work in [21], where the authors also propose to train agents to maximize causal influence (both of actions and of messages), with good results on a number of domains. We recommend the reader consult [21] for a more thorough examination of causal influence.

CIC measures the causal effect that one agent's message has on another agent's behaviour. In the simplest case, we can measure the effect that an agent's message has on the next action of the

---

**Algorithm 1:** One-step causal influence of communication.

**Data:** Agent policy $\boldsymbol{\pi}_1$, other agent policy $\boldsymbol{\pi}_2$, possible messages $\bar{m} = (m_0, ..., m_{M-1})$, number of test games $T$.

CIC = 0
**for** $i \in \{0, ..., T-1\}$ **do**
    Generate new state $S$, observations $O$.
    // Intervene by changing message $m_j$
    **for** $j \in \{0, ..., M-1\}$ **do**
        $p(m_j) \leftarrow \boldsymbol{\pi}_2(m_j|o_2), \quad p(a|m_j) \leftarrow \boldsymbol{\pi}_1(a|o_1, m_j)$
        $p(a, m_j) = p(a|m_j)p(m_j)$
        $p(a) = \sum_{m \in \mathcal{A}^m} p(a, m)$
        CIC += $1/T \cdot \sum_{a \in \mathcal{A}^e} p(a, m_j) \log \frac{p(a, m_j)}{p(a)p(m_j)}$
    **end**
**end**

---

opponent. We call this one-step causal influence. The one-step CIC is calculated using the mutual information between an agent's message and the other agent's action, similarly as for IC (cf. Eq 2), however the probabilities $p(a, m) = \boldsymbol{\pi}_1(a|m)\boldsymbol{\pi}_2(m)$ represent the change in the agent's ($\boldsymbol{\pi}_1$) action probability distribution when intervening to change the message $m$ spoken by the other agent ($\boldsymbol{\pi}_2$). In other words, the probabilities are normalized over each game, rather than across games. We describe in detail how to calculate the one-step CIC in Algorithm 1, and we discuss how CIC might be generalized to the multi-step case in Section 6.1. When calculating the CIC, care must be taken that we condition on all the variables that can affect the other agent's action, to avoid back-door paths [34]. In our setting this is easy, as our MCGs are not iterated (the state and reward are independent of actions and states at previous timesteps).

## 4 EXPERIMENTAL SETUP

### 4.1 Matrix Communication Games

Our work is based on a simple class of games called *matrix games*, where each agent $i$'s reward $r_i^t$ at time step $t$ is determined via lookup into a fixed payoff matrix $R_i^t \in \mathbb{R}^{|\mathcal{A}_1^e|\times\cdots\times|\mathcal{A}_N^e|}$ for each agent, indexed by the agents' actions (i.e. $r_i^t = R_i^t(a_1^t, ...a_N^t)$). We study emergent communication in an augmented version of matrix games which we call *matrix communication games* (MCGs), where agents can send discrete, costless messages over a communication channel before acting. In other words, MCGs are matrix games where $|\mathcal{A}_i^m| > 0$ for some $i \in \{0, ..., N\}$. MCGs are interesting to study because they are the simplest form of game where agents can both communicate with each other, and act in the environment. MCGs can be easily adapted to various settings (cooperative vs. non-cooperative, fully vs. partially observable), are simple and efficient for learning, and the resulting communication policies are generally interpretable. Variants of matrix games with communication have been used to model emergent signaling in the fields of economics, game theory, and evolutionary biology [12, 19, 38, 42].

In our variant of the game, two agents play an MCG in rounds. A round begins with one agent communicating, and ends when both agents have acted and received a reward, at which point another payoff $R^{t+1} = (R_1^{t+1}, R_2^{t+1})$ is selected. The agents observe both
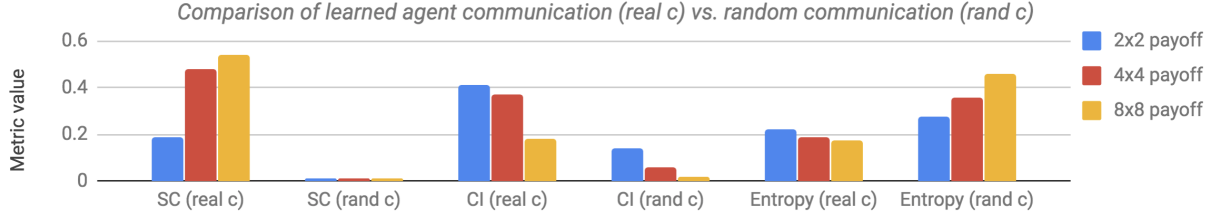
Figure 1: When comparing the speaker consistency (SC), context independence (CI), and entropy of the message policy (divided by 5) between agents who learned to signal (real $c$) and agents who message randomly (rand $c$), it appears as though the agents are communicating (note we expect a lower entropy for a communicating policy).

payoff matrices and all messages sent that round. Communication occurs in turns;[3] at each round, one agent is randomly chosen to speak first, and sends a message (a one-hot vector of length $M = |\mathcal{A}^m|$) to the other agent. That agent observes the message from the first agent, and can in turn send its own message. After this exchange, both agents act simultaneously, and receive a reward. We consider the non-iterated case, where the actions and messages from previous rounds are not observed.[4] To simplify learning, we use this reward to update both the agent's action and communication policies for that round.

For the experiments in this paper, we consider the *general-sum* case, where agents are not strictly cooperative and may have competing objectives (i.e. $R_1 \neq R_2$). Note that this does not mean the agents are strictly competitive either; the agents have some *partial common interest* in the game which permits a degree of coordination. See the Appendix for examples of matrix games. We vary the size of the payoff matrix from 2x2, to 4x4, to 8x8, and we provide a communication channel slightly larger than the number of actions ($M = 4$, 6, and 10, respectively), so that the agents can in theory learn flexible communication policies.

## 4.2 Model and learning algorithm

We train our agents using an adaptation of the REINFORCE algorithm [43]. We represent the agents' policies using a two-layer feedforward network (with parameters $\theta^n$), with separate linear output layers for the action $a$ ($\theta^a$), communication $c$ ($\theta^c$), and learned baseline $V$ ($\theta^v$). Let $\theta^a = \{\theta^a, \theta^n\}$ be all the parameters of the action network giving rise to an action policy $\pi_{\theta^a}$, and similarly for $\theta^c$ and $\theta^v$ (with outputs $\pi_{\theta^c}$ and $V$). The objective for each agent can be written as:

$$J(\theta) = J_{pol}(\theta^a) + \lambda_c J_{pol}(\theta^c) + \lambda_{ent}(J_{ent}(\theta^a) + J_{ent}(\theta^c)) + \lambda_v J_v(\theta^v),$$

where $\theta = \{\theta^n, \theta^a, \theta^c, \theta^v\}$. This can be broken down as follows: $J_{pol}(\theta^a) = \mathbb{E}_{\pi_{\theta^a}}[-\log \pi_{\theta^a}(a|o) \cdot (r - V(o))]$ is the normal REINFORCE update for the action probabilities, with the learned value function $V(o), o \in O$ as a baseline to reduce the variance. $J_{pol}(\theta^c)$ takes the same form, except $V$ only uses information available to the agent at the time it sends its message, which may not include the message of the other agent (there is no separate value function for the communication output). This value function is updated using the squared loss $J_v(\theta^v) = (r - V(o))^2$. The entropy bonus $J_{ent}(\theta)$

[3]The exact form of communication is not important for our results; we have observed similar behaviour when agents speak simultaneously, or if only one agent speaks.
[4]In Section 5.3 we show that the iterated case where agents (trained using the A2C algorithm [30]) have a memory of previous interactions produces similar behaviour.
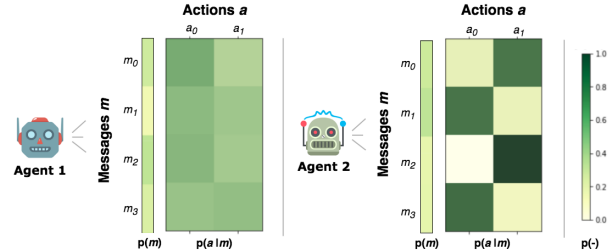


Figure 2: Visualization of example learned policies for two agents playing 2x2 MCGs ($M = 4$), averaged over 1000 test games. Agent 2 learns to clearly signal its next action.

gives a small reward based on the entropy of the agent's policies, as is common in the RL literature to encourage exploration [8, 37, 44]. Each $\lambda_i$ is a real-valued coefficient that we tune separately. We provide full hyperparameter and training details in the Appendix.

## 5 RESULTS

We now train deep RL agents to play MCGs. We find that even though our agents show strong indicators of communicating according to speaker consistency and qualitative analysis, this does not mean that the communication is useful. In fact, we show that this 'communication' occurs even if the messages are scrambled (replaced by a random message) before being observed.

## 5.1 Positive signaling with random payoffs

*Fixed R setting.* It is known that humans are able to use communication to obtain a higher reward in general-sum matrix games, such as the Prisoner's Dilemma [36]. To our knowledge, whether RL agents can learn to communicate for various MCGs remains an open question. So, we first conduct an experiment where we train two REINFORCE agents to play MCGs where the payoff $R^t$ is fixed for every timestep $t$. We vary the size of the payoff matrix and communication channel as described in Section 4.1. We find experimentally that, for every payoff matrix we tried, the agents don't learn to communicate. Instead, when there is partial common interest, agents collapse to executing a single action (even with a well-tuned $\lambda_{ent}$), and in zero-sum games they cycle between actions without communicating. Intuitively, this makes sense; the main utility in an agent learning to communicate in this setting is in reducing the other agent's uncertainty about their action. Evidently, when always playing the same payoff it is easier for these naïve

| Payoff Size | min(CIC) | Average CIC | % games with CIC < $1.02 \cdot$ min(CIC) |
|---|---|---|---|
| 2x2 | 1.386 | 1.408 ± 0.002 | 89.3 ± 0.6% |
| 4x4 | 1.792 | 1.797 ± 0.001 | 97.9 ± 0.4% |
| 8x8 | 2.303 | 2.303 ± 0.001 | 99.9 ± 0.1% |

Table 1: Causal influence values for various matrix sizes, calculated over 1000 test games. In all cases, the average CIC is very close to the minimum CIC (when changing the message has no effect on the action distribution).

| Experiments | 2x2 payoff | 4x4 payoff | 8x8 payoff |
|---|---|---|---|
| Scrambled $c$ | 0.198 ± 0.038 | 0.487 ± 0.051 | 0.597 ± 0.091 |
| Separate $c$ net | 0.028 ± 0.002 | 0.124 ± 0.011 | 0.020 ± 0.019 |
| No $c$ training | 0.171 ± 0.033 | 0.428 ± 0.025 | 0.686 ± 0.049 |
| Default | 0.202 ± 0.040 | 0.510 ± 0.094 | 0.541 ± 0.090 |

Table 2: SC values for the randomized $R$ setting. 'Scrambled $c$' is when the messages are replaced by a random message before being observed, 'Separate $c$ net' is when the action and message networks have no shared parameters, and 'No $c$ training' is when $\lambda_c = 0$.

agents to adapt to the actions of the opponent directly, rather than learning a communication protocol.

*Randomized R setting.* One way we can increase the uncertainty each agent has about the other's action (with the hope of producing emergent communication) is by *randomizing the payoffs* at each round. In our next experiment, we train two agents on an MCG where, at every round, each entry of the payoff matrix $R_i^t$ is drawn from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, with $\mu = 0$ and $\sigma^2 = 3$.

As shown in Figure 1, in this randomized setting speaker consistency emerges. The SC for the generated messages is significantly greater than for random messages, which rules out the possibility that the agents' messages are simply acting as a random public signal that the agents are using to condition their actions (as would be the case in a correlated equilibrium [3]). The CI is also higher for the generated messages, and the entropy of the message distribution is lower, both indicating that communication has emerged. We also examine the policies qualitatively in Figure 2, and find that one agent clearly learns to signal its intended action. One plausible explanation for this phenomenon is that the agents have uncertainty about the intended action of the other agent, and receiving a high reward in these games requires agents to predict the intended action of the opponent. An agent can do this in two ways: first by using the opponent's payoff matrix (and its own) to determine what action they are likely to take to achieve a high reward, and second using the opponent's message to predict their most likely action. When the payoff matrix is changing at each time step, predicting the action of the opponent conditioned solely on the payoff matrix becomes significantly harder. Thus, the agents seem to learn to leverage the communication channel to provide additional information to the other agent, which is beneficial for both agents when the game has partial common interest.
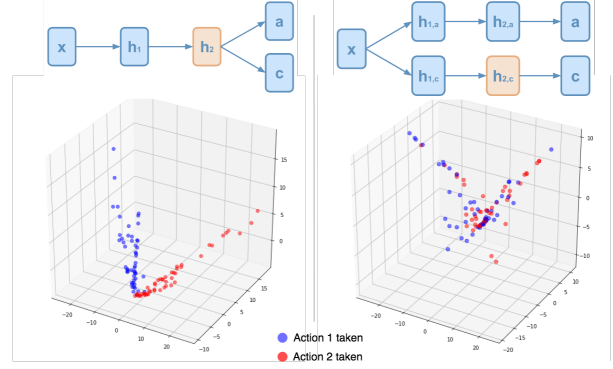


Figure 3: Activations of the last layer of the policy network for both the standard architecture (left), and when using a separate network for communication (right). Calculated on 100 random 2x2 games, and reduced to 3D using PCA.

## 5.2 Positive signaling $\not\Rightarrow$ positive listening

It turns out that, in the randomized $R$ setting, the communication has very little effect on the agents' behaviour. We show this in two ways. First, we examine the trained policies directly to see how altering the message in various games changes the resulting action selection, using the CIC metric. We calculate the CIC over 1000 test games, and show the results in Table 1. We find that, for the vast majority of games, the message sent by an agent has no effect on the action of the opponent. Thus, communication is not having a significant effect on the training process.

Second, we conduct an experiment where we train the agents in the randomized $R$ setting described above, except we *scramble* the messages received by both agents. That is, each agent outputs some message $c_i$, but we have both agents observe a different communication $c'_i$, which has no relation to $c_i$ and is drawn uniformly at random from the set of messages. Thus, there is no possibility for the agents' messages to impact the learning whatsoever. However, as shown in Table 2, the SC between the agents' action and sent message (*not* the random replacement message) is still positive and indistinguishable from the SC in the regular MCG set-up. This is convincing evidence that the correlation between actions and communications does not emerge because the message are useful, but rather as a byproduct of optimization.

*Why is the SC positive?* If the emergent communication is not useful at all, why is the SC positive? To help us understand what the policies are learning, we train agents according to the randomized $R$ setting in Section 5.1, and we plot the activations of the last hidden layer (values in the policy network after the last activation function) for 100 inputs using principal component analysis (PCA) [35] in Figure 3. This shows us our policy network's learned representations of the data [26]. When using shared features (left), the network learns to separate the inputs based on which action the agent takes. This makes sense: in order to take the appropriate action for a given input, the representations need to be linearly separable at the last layer of the network, since the action output layer has a linear decision boundary [17]. This separation of representations does not occur in the last layer of a separate communication network, which provides further evidence for this explanation.
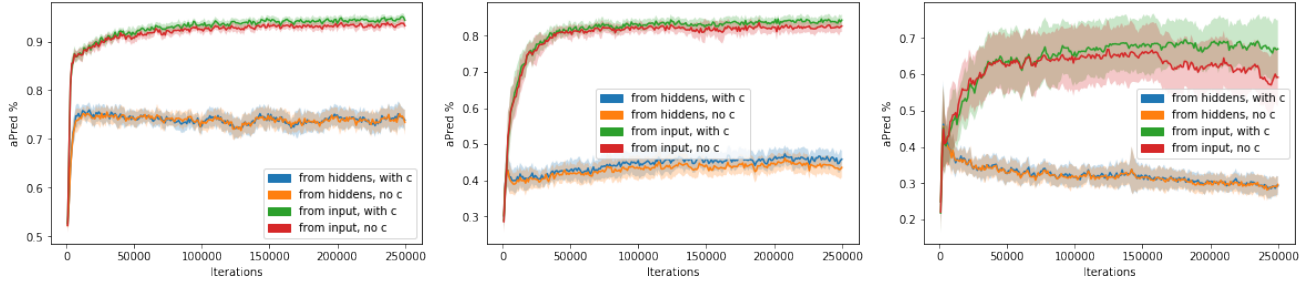
Figure 4: Accuracy in predicting the opponent's action using action classifier probes, for 2x2 payoffs (left), 4x4 payoffs (center), and 8x8 payoffs (right). 'no c' indicates that the other agent's communication was not used to predict their action, and 'from input' indicates that a separate network was trained to predict the action (rather than using a linear model on top of the last hidden layer of the policy network).

| Experiments | 2x2 payoff | 4x4 payoff | 8x8 payoff |
|---|---|---|---|
| Scrambled $c$ | 0.195 ± 0.059 | 0.127 ± 0.065 | 0.208 ± 0.067 |
| Separate $c$ net | 0.016± 0.019 | 0.019 ± 0.011 | 0.000 ± 0.000 |
| Default | 0.169 ± 0.015 | 0.130 ± 0.055 | 0.269 ± 0.076 |

Table 3: Speaker consistency values for different experiments in the iterated MCG case with A2C agents [30].

This separation of representations makes it easy for a relationship to emerge between an agent's actions and messages, even if the parameters of the communication head are completely random; since the communication output layer also has a linear decision boundary, it is likely to separate based on intended actions to some degree. Indeed, we find that SC emerges using our architecture when the communication parameters are not trained (Table 2). Further, when we re-train the agents using completely separate networks for the actions and messages, we find that the SC completely disappears (see Table 2), showing that it was indeed our choice of architecture that resulted in the emergent signaling behaviour.

*Why aren't the agents using the messages?* We would imagine that, even if the correlation between actions and messages emerged accidentally, that this might still be useful for the agents in selecting their action. After all, isn't more information about the opponent strictly better? To answer this question, we use a set of *action classifier probes*, related to the linear classifier probes for understanding hidden layers of neural networks proposed in [1]. Specifically, in the randomized $R$ setting, we train a neural network 'probe' to predict the action of the opponent in the current round, based on either part of the input (using 2-layer MLP probes) or part of the last hidden layer of the policy network (using linear probes).

The results are shown in Figure 4. We observe that removing the opponent's message from the input does not significantly reduce the accuracy in predicting the opponent's action, both when using a probe directly from the input and a probe from the last layer of the network. This suggests that the information being provided by the messages is redundant when compared to the information provided by the payoff matrix itself.

### 5.3 Results in the iterated MCG setting

The results in our paper are not limited to the non-iterated case. To show this, we run experiments on an iterated version of our environment, using the A2C algorithm [30]. We keep the policy architectures the same, except we give each agent a memory of the previous 5 rounds (the actions and the messages of both agents), which is concatenated to the input at each round. We increase the discount factor $\gamma$ to 0.9. Changing our REINFORCE algorithm (Eq. 1) to A2C requires changing the way $Q^{\pi}$ is estimated; instead of using the next reward, we use the $n$-step return (sum of $n$ next rewards), with $n = 5$ [30]. The results are shown in Table 3. We can see that the same general trend is present: there is positive SC in the randomized $R$ setting, even when scrambling the messages $c$; however, the SC disappears when a different network is used to produce the messages.

## 6 DISCUSSION

### 6.1 How general is this analysis?

*Positive signaling without positive listening.* An important question is whether the behaviours observed in this paper are specific to training our policy architecture on MCGs, and whether any of these insights can be applied in other emergent communication settings. We conjecture that this could indeed happen whenever the agent's architecture uses shared feature learning layers between the action and communication outputs; policies will always learn representations that separate the inputs based on the desired action, and this may lead to spurious correlations between an agent's messages and actions. Since sharing features is quite common in RL (e.g. [20]), it is possible that this becomes an occasional occurrence in emergent communication work.

However, our claim is not that this specific failure case will be frequently observed; rather, our goal is to highlight the importance of understanding what our metrics are measuring, and encourage emergent communication researchers to explore quantitative metrics for measuring the impact that the agents' communication is having on their behaviour (i.e. positive listening).

*Scaling causal influence.* In this paper, we focused on the one-step approximation to CIC, which only calculates the effect of an agent's message (which consists of one symbol) on the other agent's next action. While this is sufficient for the non-iterated MCG setting, as we move to more complex environments we will need to measure the effect of compositional messages on the long-term behaviour

| Metric | Positive signaling | | Positive listening | | Remarks |
|---|---|---|---|---|---|
| | Sufficient? | Necessary? | Sufficient? | Necessary? | |
| SC | ✓ | ✗ | ✗* | ✗ | Not necessary for positive signaling, as there may be no relationship between the message and subsequent action (can communicate previous observations or actions). |
| CI | ✓ | ✗ | ✗* | ✗ | More restrictive than SC, punishes formation of synonyms. |
| $H(\boldsymbol{\pi}^c)$ | ✗ | ✗ | ✗* | ✗ | Useful for diagnosing policy behaviour, but not as a metric. |
| IC | ✗ | ✗ | ✓ | ✗* | Since it is not state-dependent, IC can miss many positive listening relationships. |
| MIN | ✗ | ✗ | ✗* | ✓ | If $||W_m^1|| = 0$, then no positive listening is present. |
| $\Delta r$ | ✓ | ✗ | ✓ | ✗ | Should always be measured, strong indicator that communication is present. But less applicable when comparing the communicative behaviour of two policies |
| Qual. | ✓ | ✓ | ✗* | ✗ | Very useful for understanding agent behaviour. Can come in many forms. |
| CIC | ✗ | ✗* | ✓ | ✓ | Should always be used to determine effect communication has on agent behaviour. |

**Table 4: Summary of the metrics analyzed in this paper ($\Delta r$ = increase in reward when adding a communication channel, Qual. = qualitative analysis of messages, and MIN = message input norm, detailed in Section 6.2). Asterisks (\*) mark relationships we have shown experimentally (Sections 5 and 6.2) or via counterexample (Section 3). See text for a more detailed explanation.**

of all other agents in the environment. In this case, calculating CIC naïvely using Algorithm 1 will be computationally expensive. However, we can make this efficient via *sampling*; rather than iterating over all possible messages and all agents, the CIC could be calculated by iterating over a small set of messages sampled from the agent's communication policy, and evaluating the change in behaviour over finite time horizon for agents in some neighbourhood of the speaking agent. We leave a detailed exploration of this direction to future work.

## 6.2 Recommendations

Here we provide recommendations as to when the metrics presented in Section 3 might be used to either detect whether communication is emerging, or measure the difference in communication quality between algorithms. We also propose some other tests that could be used for this purpose. We summarize our insights in Table 4. In general, no single metric or test will tell the whole story, and we advise using several of them to illuminate the agents' behaviour.

*Detecting emergent communication.* If our goal is to detect whether communication is emerging at all, showing that adding a communication channel to a given algorithm leads to improved reward is a sufficient indicator. However, it may not be necessary; agents may obtain a similar reward by coordinating via learned convention [28], rather than communication. In other words, communication may act as an alternate pathway for optimization. Detecting useful communication is not as simple as testing if removing the communication channel at test time leads to a decrease in reward; neural networks are notoriously sensitive to their input distribution [41], and a change in this distribution (e.g. setting the messages to 0) may cause them to fail, even if the messages contain no useful information. We recommend instead using CIC and other causal metrics, discussed below.

The variant of SC explored here only measures the one-step relationship between an agent's message and subsequent action. In general, communication could influence the actions of agents further in the future than a single time step, and the language used by the agents may be compositional and temporally extended. This should be taken into account as we move towards more complex environments. In general, we acknowledge that speaker consistency

is useful from the perspective of detecting positive signaling, but we reiterate that the observed relationships may be spurious.

There are other aspects of the environment that an agent could learn to signal about: an agent might send a message to get another agent to perform an action, to share an observation it has made, or to reveal the sequence of actions it has taken in the past. New metrics need to be developed to evaluate these possibilities. We recommend researchers evaluate the quantities that are relevant for their environment or task. The crucial point is that, if these quantities are measured by observing the agent's behaviour without causal intervention, detecting positive listening is difficult, as evidenced by our experiments in Section 5. If a relationship is observed between an agent's messages and some quantity in the environment, we recommend researchers *investigate the causal relationship* between these variables, by intervening to change in turn the environmental quantity and the agent's message, and observing the impact on the other quantity over a number of episodes [10].

One way to tell conclusively that there is no positive listening is to look at the weight matrix of the first layer of the policy network ($W^1$), specifically the part that comes from the message inputs of the other agents ($W_m^1$). If the norm of this part of the weight matrix (the *message input norm*, MIN) is 0, then clearly no positive listening is present, as the messages from the other agents cannot affect a given agent's behaviour. However, just because $||W_m^1|| > 0$, does not mean positive listening is present; in our experiments on MCGs, we found that this norm was of similar magnitude to the norm of the weights from the payoff matrix.

*Measuring improvement in communication.* How should we judge the quality of a learned communication protocol in a multi-agent environment? Of course, this depends on the environment and the objectives of the researcher. Often, researchers may want to show that their algorithm exhibits a new kind of communication (e.g. verbal agreements, compositional language, or deception). In these cases, it makes sense to use metrics targeted at measuring the phenomenon in question. If the goal is to develop compositional communication, as has been the case for several recent emergent communication papers [4, 18, 31], then it is perhaps sufficient to evaluate using metrics designed to measure compositionality, such

as context independence [4]. These metrics will have to be developed on a case-by-case basis, depending on the type of communication under investigation.

There may also be cases where we simply want to show that the learned communication protocol for a proposed algorithm has a larger effect on agent behaviour than for previous algorithms. Here, a variant of CIC should be used that measures the impact of communication on the long-term behaviour of the other agents. Another test one could run is measuring the difference in reward for each algorithm with and without communication. This should be done by training each algorithm from scratch with and without communication, rather than removing the communication channel at test time, to avoid the problems of distributional shift mentioned earlier in this section. Of course, since these metrics may be exploitable, it is important to benchmark against a range of metrics and tests to avoid overfitting.

## 7 CONCLUSION

In this paper, we have highlighted some of the challenges of measuring emergent communication, namely that existing metrics for detecting positive signaling do not necessarily indicate the presence of positive listening. In addition to measuring the reward and studying agent behaviour qualitatively, we advocate for investigating the causal relationship between the agents' messages and other variables in the environment by performing interventions. This gives rise to a family of metrics, including the CIC described here, which evaluates the impact of an agent's message on another agent's next action. An open question is how our insights can be applied to evaluating agent communication with humans. Indeed, while there have been some methods proposed for understanding agent communication by 'translating' it to human language [2], this remains an open problem, and we leave an examination of this direction to future work.

## ACKNOWLEDGEMENTS

## REFERENCES

[1] Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644* (2016).
[2] Jacob Andreas, Anca Dragan, and Dan Klein. 2017. Translating neuralese. *arXiv preprint arXiv:1704.06960* (2017).
[3] Robert J Aumann. 1974. Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics* 1, 1 (1974), 67–96.
[4] Ben Bogin, Mor Geva, and Jonathan Berant. 2018. Emergence of communication in an interactive world with consistent speakers. *arXiv preprint arXiv:1809.00549* (2018).
[5] Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics* 19, 2 (1993), 263–311.
[6] Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z Leibo, Karl Tuyls, and Stephen Clark. 2018. Emergent Communication through Negotiation. *arXiv preprint arXiv:1804.03980* (2018).

[7] Edward Choi, Angeliki Lazaridou, and Nando de Freitas. 2018. Compositional Obverter Communication Learning From Raw Visual Input. *arXiv preprint arXiv:1804.02341* (2018).
[8] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*. 4299–4307.
[9] Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. 2017. Learning cooperative visual dialog agents with deep reinforcement learning. *arXiv preprint arXiv:1703.06585* (2017).
[10] Tom Everitt, Pedro A Ortega, Elizabeth Barnes, and Shane Legg. 2019. Understanding Agent Incentives using Causal Influence Diagrams, Part I: Single Action Settings. *arXiv preprint arXiv:1902.09980* (2019).
[11] Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. 2017. Emergent Communication in a Multi-Modal, Multi-Step Referential Game. *arXiv preprint arXiv:1705.10369* (2017).
[12] Joseph Farrell and Matthew Rabin. 1996. Cheap talk. *Journal of Economic perspectives* 10, 3 (1996), 103–118.
[13] Jakob Foerster, Francis Song, Edward Hughes, Neil Burch, Iain Dunning, Shimon Whiteson, Matthew Botvinick, and Michael Bowling. 2018. Bayesian action decoding for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1811.01458* (2018).
[14] Jakob N. Foerster, Yannis M. Assael1, Nando de Freitas, and Shimon Whiteson. 2016. Learning to Communicate to Solve Riddles with Deep Distributed Recurrent Q-Networks. (2016).
[15] Jon Gauthier and Igor Mordatch. 2016. A paradigm for situated and goal-driven language learning. *arXiv preprint arXiv:1610.03585* (2016).
[16] Herbert Gintis, Eric Alden Smith, and Samuel Bowles. 2001. Costly signaling and cooperation. *Journal of theoretical biology* 213, 1 (2001), 103–119.
[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*.
[18] Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: learning to communicate with sequences of symbols. In *Advances in Neural Information Processing Systems*. 2149–2159.
[19] Simon M Huttegger, Brian Skyrms, Rory Smead, and Kevin JS Zollman. 2010. Evolutionary dynamics of Lewis signaling games: signaling systems vs. partial pooling. *Synthese* 172, 1 (2010), 177.
[20] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. 2016. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397* (2016).
[21] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A Ortega, DJ Strouse, Joel Z Leibo, and Nando de Freitas. 2018. Intrinsic Social Motivation via Causal Influence in Multi-Agent RL. *arXiv preprint arXiv:1810.08647* (2018).
[22] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[23] Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. 2017. Natural Language Does Not Emerge'Naturally'in Multi-Agent Dialog. *arXiv preprint arXiv:1706.08502* (2017).
[24] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984* (2018).
[25] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2016. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182* (2016).
[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.
[27] Jason Lee, Kyunghyun Cho, Jason Weston, and Douwe Kiela. 2017. Emergent translation in multi-agent communication. *arXiv preprint arXiv:1710.06922* (2017).
[28] Adam Lerer and Alexander Peysakhovich. 2018. Learning Social Conventions in Markov Games. *arXiv preprint arXiv:1806.10071* (2018).
[29] Michael L Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the eleventh international conference on machine learning*, Vol. 157. 157–163.
[30] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*. 1928–1937.
[31] Igor Mordatch and Pieter Abbeel. 2017. Emergence of Grounded Compositional Language in Multi-Agent Populations. *arXiv preprint arXiv:1703.04908* (2017).
[32] Martin A Nowak and David C Krakauer. 1999. The evolution of language. *Proceedings of the National Academy of Sciences* 96, 14 (1999), 8028–8033.
[33] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).
[34] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. 2016. *Causal inference in statistics: a primer*. John Wiley & Sons.

[35] Karl Pearson. 1901. Principal components analysis. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 6, 2 (1901), 559.

[36] David Sally. 1995. Conversation and cooperation in social dilemmas: A meta-analysis of experiments from 1958 to 1992. *Rationality and society* 7, 1 (1995), 58–92.

[37] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International Conference on Machine Learning*. 1889–1897.

[38] John M Smith. 1991. Honest signalling: The Philip Sidney game. *Animal Behaviour* (1991).

[39] Sainbayar Sukhbaatar, Rob Fergus, et al. 2016. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*. 2244–2252.

[40] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*.

[41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[42] Kyle Wagner, James A Reggia, Juan Uriagereka, and Gerald S Wilkinson. 2003. Progress in the simulation of emergent communication and language. *Adaptive Behavior* 11, 1 (2003), 37–69.

[43] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.

[44] Ronald J Williams and Jing Peng. 1991. Function optimization using connectionist reinforcement learning algorithms. *Connection Science* 3, 3 (1991), 241–268.

[45] Ludwig Wittgenstein. 1953. Philosophical investigations. (1953).

[46] Amotz Zahavi. 1975. Mate selection—a selection for a handicap. *Journal of theoretical Biology* 53, 1 (1975), 205–214.

# A ADDITIONAL ALGORITHMS

For clarity, we provide three additional algorithms related to our paper here. Algorithm 2 describes a single iteration of a matrix communication game (MCG), described in Section 4.1. Algorithms 3 and 4 describe the calculation of the speaker consistency (SC) and instantaneous coordination (IC), respectively, for both agents. The difference between Algorithms 3 and 4 is how the joint probabilities are calculated: for SC, $p(a, m)$ refers to the probability that agent 1 sends message $m$ and then takes action $a$, while for IC it refers to the probability that agent 1 sends message $m$ and then *agent 2* takes action $a$.

---

**Algorithm 2:** EnvStep (A single round of an MCG with turn-based communication.)

**Data:** Agent policies $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$, state $S$, payoff matrices $R = (R_1, R_2)$ and observations $O = (o_1, o_2)$.
// Randomly select agent $j$ to go first, $j \in \{1, 2\}$.
$j \sim$ Bernoulli$(0.5) + 1$
// Agent $j$ sends a message $m_j$ given observation $o_j$.
$m_j \leftarrow \boldsymbol{\pi}_j(m|o_j)$
// Agent $1 - j$ sends a message $m_{1-j}$ given observation $o_{1-j}$ and agent $j$'s message.
$m_{1-j} \leftarrow \boldsymbol{\pi}_{1-j}(m|o_{1-j}, m_j)$
// Both agents act simultaneously.
$a_1 \leftarrow \boldsymbol{\pi}_1(a|o_1, m_1, m_2), \quad a_2 \leftarrow \boldsymbol{\pi}_2(a|o_2, m_1, m_2)$
// Agents receive reward
$r_1 \leftarrow R_1[a_1, a_2], \quad r_2 \leftarrow R_2[a_1, a_2]$
**return** $(m_1, m_2), (a_1, a_2), (r_1, r_2)$

---

**Algorithm 3:** Speaker consistency (SC).

**Data:** Agent policy $\boldsymbol{\pi}_1$, other agent policy $\boldsymbol{\pi}_2$, possible messages $\bar{m} = (m_0, ..., m_{M-1})$, number of test games $T$.
$SC_1 = 0, SC_2 = 0$
// Initialize matrices $P$ for tracking (message, action) co-occurrences.
$P_1 = 0^{|m| \times |a|}, P_2 = 0^{|m| \times |a|}$
**for** $i \in \{0, ..., T-1\}$ **do**
  Generate new state $S$, observations $O$, payoffs $R$.
  // Get agents messages and actions.
  $(m_1, m_2), (a_1, a_2), (r_1, r_2) \leftarrow$ EnvStep$(S, O, R)$
  // Update co-occurrence matrices
  $P_1[m_1, a_1] += 1$
  $P_2[m_2, a_2] += 1$
**end**
**for** $j \in \{1, 2\}$ **do**
  // Calculate probabilities from co-occurrence matrix.
  $p_j(a, m) = P_j/T$
  $p_j(a) = \sum_{m \in \mathcal{A}^m} P_j(a, m)$
  $p_j(m) = \sum_{a \in \mathcal{A}^e} P_j(a, m)$
  // Calculate mutual information.
  $SC_j = \sum_{a \in \mathcal{A}^e} \sum_{m \in \mathcal{A}^m} p_j(a, m) \log \frac{p_j(a, m)}{p_j(a)p_j(m)}$
**end**
**return** $SC_1, SC_2$

---

**Algorithm 4:** Instantaneous coordination (IC).

**Data:** Agent policy $\boldsymbol{\pi}_1$, other agent policy $\boldsymbol{\pi}_2$, possible messages $\bar{m} = (m_0, ..., m_{M-1})$, number of test games $T$.
$IC_1 = 0, IC_2 = 0$
// Initialize matrices $P$ for tracking (message, action) co-occurrences.
$P_1 = 0^{|m| \times |a|}, P_2 = 0^{|m| \times |a|}$
**for** $i \in \{0, ..., T-1\}$ **do**
  Generate new state $S$, observations $O$, payoffs $R$.
  // Get agents messages and actions.
  $(m_1, m_2), (a_1, a_2), (r_1, r_2) \leftarrow$ EnvStep$(S, O, R)$
  // Update co-occurrence matrices. Difference with SC is index of $m, a$.
  $P_1[m_1, a_2] += 1$
  $P_2[m_2, a_1] += 1$
**end**
**for** $j \in \{1, 2\}$ **do**
  // Calculate probabilities from co-occurrence matrix.
  $p_j(a, m) = P_j/T$
  $p_j(a) = \sum_{m \in \mathcal{A}^m} P_j(a, m)$
  $p_j(m) = \sum_{a \in \mathcal{A}^e} P_j(a, m)$
  // Calculate mutual information.
  $IC_j = \sum_{a \in \mathcal{A}^e} \sum_{m \in \mathcal{A}^m} p_j(a, m) \log \frac{p_j(a, m)}{p_j(a)p_j(m)}$
**end**
**return** $IC_1, IC_2$

# B ADDITIONAL RESULTS

## B.1 Fixed matrix SC results

In Table 5, we show the results of measuring the speaker consistency for various fixed matrices. We have observed that positive signaling does not emerge for fixed payoffs, even in games where it would make sense as a coordination device (e.g. the Battle of the Sexes). Since the agents are always playing the same payoff, they are able to adapt to the action of the opponent, and find a deterministic equilibrium (or cycle through actions for competitive games).

| Experiment | Reward | SC |
|---|---|---|
| Prisoner's Dilemma | 1.001 ± 0.000 | 0.000 ± 0.000 |
| Battle of the Sexes | 1.500 ± 0.316 | 0.000 ± 0.000 |
| Matching Pennies | 0.000 ± 0.512 | 0.011 ± 0.007 |
| Random 2x2 payoff | 1.286 ± 1.804 | 0.000 ± 0.000 |
| Random 4x4 payoff | 2.896 ± 1.721 | 0.001 ± 0.003 |
| Random 8x8 payoff | 3.997 ± 1.353 | 0.015 ± 0.025 |

**Table 5: Average reward and speaker consistency for some popular MCGs (see Table 9 for payoffs), as well as 20 different random fixed payoffs.**

## B.2 Reward and SC results on iterated MCGs

The results in our paper are not limited to the non-iterated case. To show this, we run experiments on an iterated version of our environment, using the A2C algorithm [30]. We keep the policy architectures the same, except we give each agent a memory of the previous 5 rounds (the actions and the messages of both agents), which is concatenated to the input at each round. We increase the value of the discount factor $\gamma$ to 0.9, from 0. Changing our REINFORCE algorithm with a learned baseline to A2C requires changing the way $Q^\pi$ is estimated; instead of using the next reward, we use the $n$-step return, with $n = 5$ (see [30] for details). The results are shown in 5. We can see that the same general trend is present: there is positive SC in the randomized $R$ setting, even when scrambling the messages $c$; however, the SC disappears when a different network is used to produce the messages.

## B.3 Full metric results for randomized $R$

In Table 6, we show results for the speaker consistency, policy entropy, and context independence in the randomized $R$ setting. In Table 7 we show the reward and speaker consistency evaluated over the last 10,000 games of training for the ablation studies performed in the paper.

## B.4 Result plots

In Figure 6, we plot the SC for various matrix sizes in the randomized $R$ setting.

# C HYPERPARAMETER DETAILS

We provide a table of the hyperparameter values below. For all the plots and tables in our paper, unless otherwise specified, we average across 10 random seeds, and plot twice the standard error.

For tables where the numbers are calculated on test games after the policy has been trained, we test for 1000 games per seed, for a total of 10,000 games. The exception for this is when measuring the causal influence in Table 1, where we use 5 random seeds.

# D EXAMPLES OF MATRIX GAMES

In Table 9, we provide three examples of popular matrix games. Generally, agents playing this game will act simultaneously, and receive a reward according to the payoff matrix. In general, the agents receive different rewards from each other. We augment these games with a communication channel, where agents take turns sending discrete one-hot vectors to each other. For most of the experiments in our paper, the payoff matrices are randomly generated according to a Gaussian distribution with mean 0 and variance 3.
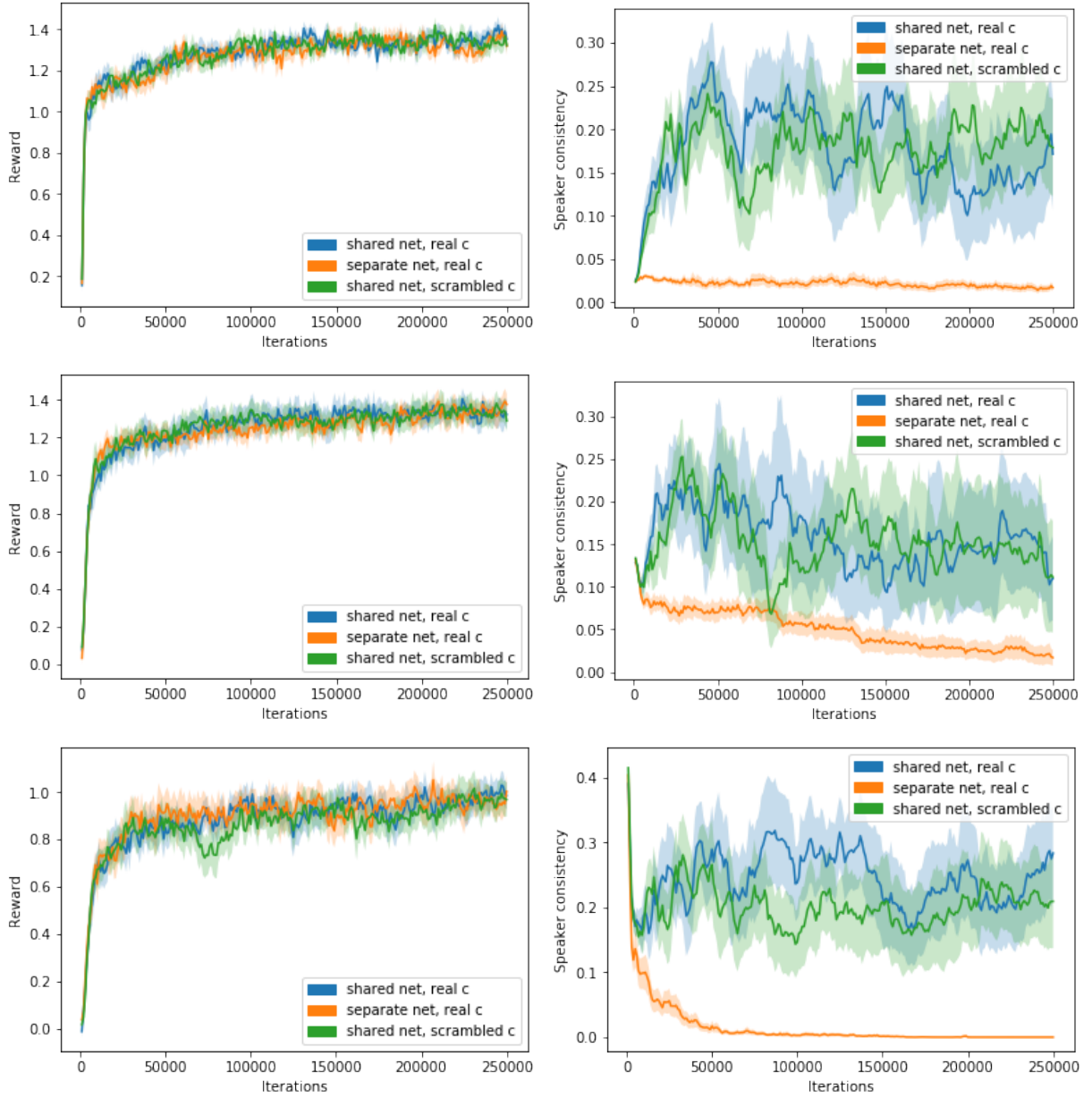
**Figure 5: Results for the iterated setting using A2C. Plots of the reward (left) and speaker consistency (right), for the 2x2 (top), 4x4 (middle), and 8x8 (bottom) sized payoff matrix.**

| Metrics | $2 \times 2$ payoffs | | $4 \times 4$ payoffs | | $8 \times 8$ payoffs | |
|---|---|---|---|---|---|---|
| | Real $c$ | Random $c$ | Real $c$ | Random $c$ | Real $c$ | Random $c$ |
| SC | $0.187 \pm 0.032$ | $0.001 \pm 0.001$ | $0.476 \pm 0.063$ | $0.008 \pm 0.001$ | $0.541 \pm 0.055$ | $0.016 \pm 0.001$ |
| $H(\boldsymbol{\pi}^c)$ | $1.116 \pm 0.044$ | $1.386 \pm 0.001$ | $0.934 \pm 0.087$ | $1.789 \pm 0.001$ | $0.855 \pm 0.080$ | $2.298 \pm 0.001$ |
| CI | $0.408 \pm 0.044$ | $0.137 \pm 0.001$ | $0.369 \pm 0.040$ | $0.057 \pm 0.001$ | $0.182 \pm 0.016$ | $0.021 \pm 0.001$ |

**Table 6: Various metrics calculated in the random $R$ setting described in Section 5.1. Results are calculated on 1000 test games over 10 random seeds. Averages are shown along with the standard error.**

| Training | 2x2 payoff | | 4x4 payoff | | 8x8 payoff | |
|---|---|---|---|---|---|---|
| setting | Reward | SC | Reward | SC | Reward | SC |
| Scrambled $c$ | $1.42 \pm 0.08$ | $0.20 \pm 0.08$ | $1.60 \pm 0.07$ | $0.49 \pm 0.11$ | $1.13 \pm 0.11$ | $0.60 \pm 0.19$ |
| Separate $c$ net | $1.41 \pm 0.08$ | $0.03 \pm 0.004$ | $1.58 \pm 0.12$ | $0.12 \pm 0.02$ | $1.12 \pm 0.13$ | $0.02 \pm 0.03$ |
| No $c$ training | $1.39 \pm 0.08$ | $0.17 \pm 0.07$ | $1.65 \pm 0.13$ | $0.43 \pm 0.05$ | $1.07 \pm 0.15$ | $0.69 \pm 0.10$ |
| Default | $1.43 \pm 0.07$ | $0.20 \pm 0.08$ | $1.63 \pm 0.09$ | $0.51 \pm 0.19$ | $1.12 \pm 0.09$ | $0.54 \pm 0.18$ |

Table 7: The reward and speaker consistency values for different training settings across various matrix sizes. 'Scrambled $c$' is where the messages are replaced by a random message before being observed, 'Separate $c$ net' is where the action and message networks have no shared parameters, and 'No $c$ training' is where $\lambda_{comm} = 0$. Error shown is twice the standard error.
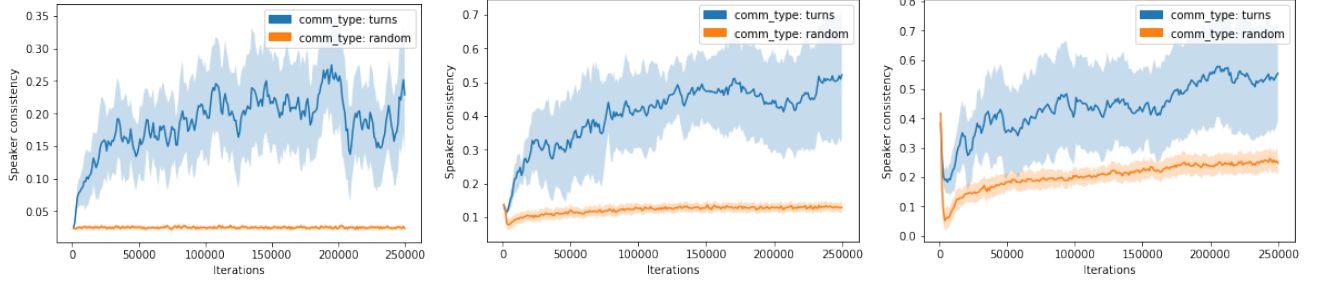


Figure 6: When given the ability to communicate, agents' messages provide information about their subsequent action (from left to right: with 2x2, 4x4, and 8x8 payoff matrices). However, agents don't do this if the message is randomly generated, as would be the case in a correlated equilibrium.

| Hyperparameter | Value |
|---|---|
| $\lambda_{comm}$ | 0.1 |
| $\lambda_{ent}$ | 0.01 |
| $\lambda_v$ | 0.1 |
| Learning rate | 0.005 |
| Batch size | 64 |
| Reward discount ($\gamma$) | 0 (not iterated) |
| Number of hidden units in policy network | (40, 60, 100) for matrix sizes of (2x2, 4x4, 8x8) |
| Number of episodes | 250,000 |
| Optimizer | ADAM [22] with default params |

Table 8: Hyperparameters

| | Opera | Fight |
|---|---|---|
| Opera | 2, 1 | 0, 0 |
| Fight | 0, 0 | 1, 2 |

(a) Battle of the sexes

| | Stag | Rabbits |
|---|---|---|
| Stag | 1, -1 | -1, 1 |
| Rabbits | -1, 1 | 1, -1 |

(b) Matching Pennies

| | Cooperate | Defect |
|---|---|---|
| Cooperate | 3, 3 | 0, 4 |
| Defect | 4, 0 | 1, 1 |

(c) Prisoner's Dilemma

Table 9: Three popular matrix games, to which we add a communication channel. The tuple of rewards ($r_1$, $r_2$) indicates the rewards for agent 1 and agent 2, respectively.