ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240918

Channel Randomisation Methods for Zero-Shot Communication

Dylan Cope and Nandi Schoots

King's College London

Abstract. When agents learn to communicate via self-play the result is typically brittle communication strategies that only work with agents encountered during training. To alleviate this and train agents that can communicate with agents outside of their training communities we introduce two training-time interventions that apply to the messages sent between agents. These methods are: (1) *message mutation*, where messages are randomly changed; and (2) *channel permutation*, where random permutations are applied to the message space. These proposals are tested using a simple two-player sequential referential game in which the agents are given the opportunity to establish communicative conventions within a single episode. After training multiple sets of agents we analyse the performance of these agents when they are matched with a 'stranger' from another training run, i.e. their zero-shot communication performance. We find that both message mutation and channel permutation positively influence performance, and we discuss their effects.

1 Introduction

Given an environment in which multiple learning agents are rewarded for completing tasks, communicative behaviour may emerge as a means to achieve higher rewards. Broadly speaking, the study of such *Emergent Communication* (EC) investigates the circumstances that lead to communication as an instrumental strategy. Adjacent to EC is the study of Zero-shot Coordination (ZSC) in multi-agent populations, where agents are trained to work with previously unseen players [15, 23, 17, 40, 6]. This overlaps with the study of agents that can form novel teams, known as *Ad Hoc Teamwork* (AHT) [36], where it is not assumed that agents are trained in the same way.

At the conjunction of the EC and ZSC there are two important forms of zero-shot performance: communicating in novel settings with a known set of agents [5, 8, 35], and communicating with novel agents [16, 2, 3, 17], i.e. 'strangers'. In this paper, we will focus on the latter and refer to this problem as *Zero-shot Communication* learning, which can intuitively be thought of as 'learning to communicate with strangers'. Prior related work in AHT has looked at similar problems, notably, Sarratt and Jhala [34] applied AHT methodology to communication problems where a shared communication system is not given. On the other hand, research has also been conducted looking at situations involving ad hoc teams composed of agents with a prior shared communication protocol [30, 1, 27].

We start with the observation that to communicate effectively with strangers, a set of shared communicative conventions needs to be established (or perhaps, existing conventions may be enhanced with context). We will refer to such conventions as *communication protocols*. When two agents that have not previously met agree upon a communication protocol within an episode we call this *intra-episodic communication protocol establishment*. Conversely, a protocol that is agreed upon between episodes is an *inter-episodically established protocol*, or just a *fixed protocol*. Our goal is to train agents that can intra-episodically establish a communication protocol with a stranger and use this protocol to cooperate on a shared task.

A survey of the relevant literature demonstrates that when two agents are trained together, the agents typically converge on an inter-episodically fixed communication protocol [10, 31, 26]. In other words, the parameters of each agent's policy store the protocol itself, rather than implement the general skills of generating and interpreting a protocol. Depending on the use case, this may be acceptable. However, in a zero-shot communication setting, we should not expect effective cooperation unless the agents happen to learn the same fixed protocols by chance. When the space of possible protocols is large, this may be highly unlikely.

This leads to the hypothesis that such fixed protocols can be prevented by randomising across the space of protocols that agents are exposed to during training. Our first proposal is *message mutation*, where, alongside a specific set of training signals, agents should learn intra-episodic communication learning capabilities when their fixed protocol is randomly tampered with during an episode. Our second proposal is *channel permutation*, where for each episode a random permutation map over the communication symbols is defined, and each time a symbol is sent through the communication channel it is transformed according to this mapping. We demonstrate that after training in either of these schemes the agents indeed have enhanced zero-shot communication performance.

To study this we introduce a simple environment in which agents only have communicative actions. This allows us to isolate the effect of randomisation on the ability to establish a communication protocol with a stranger.

2 Preliminaries

2.1 Communication Protocols

The term "communication protocol" is used broadly and generally refers to "any agreed upon set of behaviours that facilitate communication". For our purposes, we define a communication protocol p as a mapping from a set of *subjects* X to a set of communication *symbols* Σ, i.e. $p : X \to \Sigma$. We will refer to elements of Σ as *symbols* or *messages*, and in this work, we are concerned with when X is a set of agent observations. In other words, we are concerned with situations where agents communicate observed information, rather than intentions or goals.

2.2 Domain Randomisation

Domain randomisation is a training-time technique for improving the zero-shot performance of a learning system when it is transported to a new domain [38, 29, 16], which becomes relevant when the agent can not be directly trained in the target environment. By randomising certain features of the training environment we apply pressure on a learning agent to find strategies that can adapt to changes in these features. As we are interested in zero-shot communication learning, our proposals involve introducing specific forms of randomisation into the communication channel that can achieve such results for the domain of possible communication protocols.

2.3 Decentralised POMDPs

Decentralised Partially Observable Markov Decision Processes (Dec-POMDPs) are an extension of partially observable Markov decision processes to a multi-agent setting [33, 25]. For N agents, a Dec-POMDP is defined by the following components. Firstly, a set of environment states $\mathcal{S}$ and a probability distribution over initial states $\rho : \mathcal{S} \to [0, 1]$. For each agent i there is: a set of actions $\mathcal{A}_i$, a set of observations $\mathcal{O}_i$, a function for extracting agent-dependent observations $\omega_i : \mathcal{S} \to \mathcal{O}_i$, and a reward function $r_i : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \to \mathbb{R}$. Finally, the environment dynamics are defined by a stochastic transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \to \Delta(\mathcal{S})$, where at time t with state s_t and agent actions $a_1, \dots, a_N$, the next state s_{t+1} is sampled from the distribution: $s_{t+1} \sim \mathcal{T}(s_t, a_1, \dots, a_N)$. For finite Dec-POMDPs, we also select one or more states $\mathcal{S}_T \subseteq \mathcal{S}$ as terminal states. An *episode* of a Dec-POMDP is defined as the state and action history from the initial state to a terminal state. The function generating each agent's actions is called their policy, $\pi_i : \mathcal{O}_i \to \mathcal{A}_i$. In this work we consider cooperative games where each agent receives the same reward.

2.4 Emergent Communication

Emergent communication research studies agents that learn to utilise dedicated communication channels to solve a given task [19, 13, 39]. Explicit communication can be implemented in Dec-POMDPs by connecting the action and observation spaces of agents. These are referred to as Dec-POMDP-Comms [11, 12, 33]. Often, the action space $\mathcal{A}_i$ can be expressed as a Cartesian product of environment actions $\mathcal{A}_i^e$ and communicative actions Σ, i.e. $\mathcal{A}_i = \mathcal{A}_i^e \times \Sigma$. Alternatively, if $\mathcal{A}_i^e = \mathcal{A}_i^e \cup \Sigma$, then agents must choose whether to communicate or not. For a more extensive overview of work in multi-agent reinforcement learning and communication, see the survey conducted by Zhu et al. [41].

An important consideration for our work was raised by Lowe et al. [26] in their discussion of the pitfalls of measuring the presence of emergent communication. They identified two behaviours that need to be present for communication: *positive signalling* and *positive listening*. Positive signalling is when a message is correlated with some observation or intended action, and positive listening is when an agent changes their behaviour in response to receiving such signals. The authors showed that there are circumstances under which positive signalling is learned, but positive listening is not.

Definition 1 (Positive Signalling [26]). *Given a sequence of messages* $\mathbf{m}$ *sent, observations* $\mathbf{o}$ *made, and actions* $\mathbf{a}$ *taken by an agent throughout a trajectory of length* T*, the agent is* ***positive signalling*** *if* $\mathbf{m}$ *is statistically dependent on* $\mathbf{a}$ *and/or* $\mathbf{o}$.

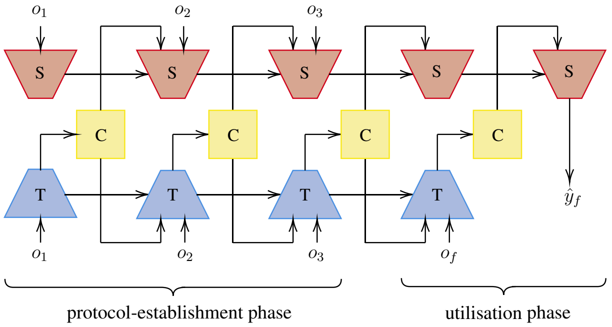


Figure 1: Diagram of the information flows through an episode where $|\mathcal{O}_E| = 3$. S represents the student's policy, T represents the teacher's policy, and C represents the communication channel. Each o_t is the input observation at time t, and o_f is the final input that is hidden from the student. The output $\hat{y}$ at the final timestep is the student's prediction of o_f. The lateral connections between S-boxes and T-boxes show the information flow through the agents' latent states.

Definition 2 (Positive Listening [26]). *Given a set of messages* Σ*, an agent i, following policy* $\pi : \mathcal{O}_i \times \Sigma \to \mathcal{A}_i$*, is* ***positive listening*** *to another agent j at time t if j has just sent message* $m_{ijt} \in \Sigma$ *to i and* $\|\pi(o_{it}, m_{ijt}) - \pi(o_{it}, m_{\varnothing})\|_\tau > 0$*, where* $\| \cdot \|_\tau$ *is a distance in the space of expected trajectories followed by* π*, and* $m_{\varnothing} \in \Sigma$ *is a special silence message, e.g. a zero-vector.*

3 Experimental Setting

In this section, we describe various aspects of our experimental setup: the environment, the communication channel, the agents, and our channel randomisation proposals.

3.1 Environment

To design a minimal environment to test the emergence of intra-episodic communication learning we identify two key phases that should be present within each episode: a *protocol establishment phase*, and a *utilisation phase*. During the first phase, the agents must have the opportunity to work together to associate observations and symbols. In the second phase, they use this protocol to communicate information. For more complex environments agents could iteratively move between these two phases, but we consider a simple situation with one cycle of this process. In our environment, there are two roles that an agent can play, which we refer to as the 'teacher' and 'student' roles. As both agents have the same architecture (described in Section 3.3), and as we will end up training agents by self-play [37] in this environment we will assume that any agent can play either of these roles.

This work is not focused on learning agents that extract useful features from high-dimensional, complex inputs (such as images), so we opt for a simple set of possible observations to be the subject of our agents' communication protocols. We call these the *environment observations*, $\mathcal{O}_E$. The set $\mathcal{O}_E$ consists of positive integers expressed as binary vectors. Specifically, given M desired classes, the environment observations are constructed as follows:

$$\mathcal{O}_E = \Big\{ (x_1, \dots, x_k) \in \mathbb{Z}_2^k \; : \; 0 < y \le M, \; \textstyle\sum_i^k 2^i x_i = y \Big\}. \quad (1)$$

Where $k = \lceil \log_2 M \rceil$. Additionally, we assign classifications to each member of $\mathcal{O}_E$ according to the number represented in binary (i.e. y in Equation 1).

Figure 1 illustrates the information flow through an episode. Each episode of the environment consists of $|\mathcal{O}_E| + 2$ timesteps. The first $0 \leq t < |\mathcal{O}_E|$ timesteps comprise the *protocol-establishment phase*. During this time, both the teacher and the student make the same environment observations o_t, and the teacher sends $s_t \in \Sigma$ to the student, i.e. the teacher may teach the student a communication protocol $p : \mathcal{O}_E \to \Sigma$. In the second phase, the *utilisation phase*, the teacher makes an observation that is hidden from the student, and the teacher must communicate this information to the student. This phase consists of two steps: one for the teacher to make an observation and produce a message, and another for the student to receive the message and make a prediction. The agents' performance in this environment is measured as the mean classification accuracy of the student's predictions of the teacher's observation in the final timestep.

The game that we use can be interpreted as a form of the Lewis signalling game [22]. Alternatively, it can be viewed as a referential game where the set of distractors coincides with the total set. Many works on observing emergent communication in referential games exist [20, 9, 7].

3.2 Communication Channel

During each episode, agents send messages to one another through a communication channel. At each timestep agents produce *utterances* as log-probabilities (logits) over a discrete set of symbols Σ. These utterances are sent through the communication channel to produce *messages*. During training, the utterance logits define a Gumbel-Softmax distribution [18, 28] that messages are sampled from. This allows for backpropagating through the communication channel, a technique used in many emergent communication works [32, 6, 4, 21]. The Gumbel-Softmax distribution requires setting a temperature parameter which we fix at 1.0 for all of our experiments (unless otherwise stated). To further encourage discretisation of the utterances we also follow [10] and inject noise into the channel (before sampling messages); we apply additive white Gaussian noise with a fixed standard deviation of 0.5. During the test evaluations, messages are constructed by computing the one-hot encoding of the argmax of the utterance logits.

3.3 Agent Architectures

Each agent's policy network takes three inputs at each timestep and produces two outputs. The inputs are: (1) a one-hot encoding of the agent's most recently sent message, (2) a one-hot encoding of the other agent's most recently sent message, and (3) an environment observation $o \in \mathcal{O}_E$. The outputs are: (1) an utterance to send as a message through a communication channel, and (2) a probability distribution over the classes of possible observations. The agents produce these actions according to their policy: an LSTM [14] Recurrent Neural Network (RNN) parameterised by θ_i, $\pi_{\theta_i} : \mathbb{R}^{|\Sigma|} \times \mathbb{R}^{|\Sigma|} \times \mathcal{O}_E \to \mathbb{R}^{|\Sigma|} \times \mathbb{R}^{|\mathcal{O}_E|}$. We will refer to the hidden state of the LSTM as the agent's *latent state*.

3.4 Message Mutation

The first of our proposals for improving zero-shot communication is *message mutation*. This is a function $f_m : \Sigma \to \Sigma$ defined:

$$f_m(s) = \begin{cases} s' & \text{if } x < p_m \\ s & \text{otherwise} \end{cases} \tag{2}$$

where $x \sim \text{Uniform}([0, 1])$ and $s' \sim \text{Uniform}(S_{mut})$.

Where p_m is the *mutation probability* and $S_{mut} \subseteq \Sigma$ is the set of possible symbols that can be chosen from. We define $S_{mut} = \{s \in \Sigma \,:\, s \notin H\}$, where H is the history of sent messages. This avoids mutations making it impossible for the teacher to produce consistent communication protocols when Σ is small.

3.5 Channel Permutation

The second of our proposals is *channel permutation*. When an episode is played with channel permutation enabled, an arbitrary bijective total function $f_{ij} : \Sigma \to \Sigma$, is created for every possible ordered pair of agents i and j. More precisely, we sample from a uniform distribution over the symmetric group $S_{|\Sigma|}$:

$$f_{ij} \sim \text{Uniform}(S_{|\Sigma|}). \tag{3}$$

Consequently, whenever agent i sends a symbol $s \in \Sigma$ to agent j, agent j receives $f_{ij}(s)$. We also investigate only permuting a subset of the symbols, i.e. for a subset size k, we uniformly sample without replacement a subset $S_{perm} \subseteq \Sigma$, and instead sample the map as follows:

$$f_{ij} \sim \text{Uniform}\big(\{f \in S_{|\Sigma|} \,:\, f(x) = x \;\forall x \notin S_{perm}\}\big). \tag{4}$$

We refer to this variant as *channel subset permutation*.

3.6 Difference between Message Mutation and Channel Permutation

We expect that channel permutation and message mutation will have somewhat different effects on the teacher, but roughly the same effect on the student. In both approaches the teacher makes utterances which may or may not be changed (mutated or permuted) to a different message. However, in message mutation, the teacher must react to changes in their protocol and adapt their behaviour in the final timestep when it needs to use the protocol. If the teacher wants to send the same message that the student received, then it needs to keep track of how the protocol was changed.

On the other hand, in channel permutation, every utterance is consistently permuted, so if in the final timestep, a teacher wants to communicate the same message as before, then they just have to make the same utterance as before. The teacher does not have to be adaptive and may converge on a specific protocol. The student on the other hand is met with many different protocols .

From this perspective, it appears that message mutation is strictly better than channel permutation, in the sense that it encourages more adaptive behaviour from both agents. However, message mutation comes with the requirement that the agents are (indirectly) rewarded for paying attention to the protocols established within the episode. Permutation does not have this requirement and can be implemented without explicit reference to a protocol.

4 Measures

Next, we introduce the measures that we use to train and evaluate our systems of agents. Suppose that at time $|\mathcal{O}_E|$ (the utilisation phase) the teacher made observation o_f of class $\mathbf{y}_f$, where $\mathbf{y}_f$ is a one-hot encoding of a class label and made utterance $\mathbf{u}_f$. At $t < |\mathcal{O}_E|$ (the protocol establishment phase) they made observations o_t of class $\mathbf{y}_t$ and made utterances $\mathbf{u}_t$. Suppose further that the student receives messages $\mathbf{m}_t$ at each timestep t and outputs $\hat{\mathbf{y}}_f$ in the final timestep.

4.1 Error Metrics

Definition 3 (Actual-Class (AC) Error)**.** *The AC error is the categorical cross-entropy (CCE) between the student's predictions and the actual class of o_f:*

$$\mathcal{L}_{AC} = CCE(\hat{\mathbf{y}}_f, \mathbf{y}_f). \tag{5}$$

Definition 4 (Student-Implied-Class (SIC) Error)**.** *The SIC error is the categorical cross-entropy between the student's predictions and the predictions that they ought to have made given the protocol established in the episode:*

$$\begin{aligned} &\mathcal{L}_{SIC} = CCE(\hat{\mathbf{y}}_f, \mathbf{y}^*), \\ &\textit{where } \mathbf{y}^* = \begin{cases} \frac{1}{|T|}\sum_{t\in T}\mathbf{y}_t & \textit{if } |T| \geq 1 \\ \textit{Uniform}(\mathcal{O}_E) & \textit{otherwise} \end{cases} \\ &\textit{and } T = \{t < |\mathcal{O}_E| \ : \ \mathbf{m}_t = \mathbf{m}_f\}. \end{aligned} \tag{6}$$

In these equations, T is the set of time steps in the protocol establishment phase in which the final message $\mathbf{m}_f$ was also sent.

For the SIC error, the implication is that the student should guess any of the observations seen in the T time steps with equal probability. As a result, this measure does not penalise the student if the teacher fails to produce a coherent protocol.

Definition 5 (Teacher-Message (TM) Error)**.** *The TM error is the categorical cross-entropy between the utterance that the teacher made and the message they should have sent, given the protocol established within the episode:*

$$\mathcal{L}_{TM} = CCE(\mathbf{u}_f, \mathbf{m}_t), \textit{ where } o_t = o_f. \tag{7}$$

Definition 6 (Protocol Diversity (PD) Error)**.** *Given a matrix P with $|\mathcal{O}_E|$ rows and $|\Sigma|$ columns, where each row i corresponds with the message sent by the teacher at $t = i$, we define the Protocol Diversity (PD) error as:*

$$\mathcal{L}_{PD} = \max_{\mathbf{c}_i} \|\mathbf{c}_i\|_1 \quad \textit{where } P = \big[\mathbf{c}_1, \dots, \mathbf{c}_{|\Sigma|}\big]. \tag{8}$$

Because of the communication channel, the rows of P are normalised, meaning that $\mathcal{L}_{PD}$ is bounded by 1 and $|\mathcal{O}_E|$. When this metric is 1, the protocol being expressed is injective. The higher the error gets the more ambiguous the protocol is, keeping in mind that each row of the P matrix is ideally a one-hot vector, but during training the distribution could be imperfect (see Section 3.2).

4.2 Evaluation Metrics

In this section, we outline five measures that we will use to evaluate agents: (1) *Self-play Performance*, (2) *Zero-shot Performance*, (3) *Student Responsiveness*, (4) *Teacher Responsiveness*, and (5) *Protocol Diversity*. The first two of these measure the performance of different agent pairings in the environment. The latter three measures will be used to provide a more in-depth analysis of the behaviours learned under different channel randomisation settings.

Definition 7 (Self-play Performance)**.** *We measure the cooperative performance of agents as the classification accuracy of the student's prediction of o_f. In other words, the mean number of times that the student made the correct prediction. The* ***self-play performance*** *is the performance when the teacher and student are instantiated with co-trained weights, i.e. weights from the same training run.*

Definition 8 (Zero-shot Performance (ZSP):)**.** *For each of our different experimental settings, we measure the cooperative performance of pairs of agents from separate training runs. For a given set of training hyperparameters, k sets of agents, $A_1, \dots, A_k$, are trained in self-play without any parameter sharing. We then take every combination of $A_i, A_j, j \neq i$ and run two sets of 170 games, one set where A_i is the teacher and A_j is the student, and vice versa. The final* ***Zero-shot Performance*** *(ZSP) score is the mean performance across these games. We refer to each novel combination of agents as a* ***'stranger encounter'****.*

When evaluating agents in self-play, high performance alone cannot indicate whether they would perform well in the zero-shot setting. In the following, we use the metrics defined in the previous section to detect whether protocols are being established within episodes. To start, we note that high performance in the presence of a high TM error, high SIC error, or high PD error implies that the agents have converged on a fixed protocol, i.e. the agents are using an *inter-episodically established protocol*.

For example, given a fixed injective protocol p, let us consider a teacher that sends the same message m, such that $m \notin \text{Image}(p)$, every timestep until the final timestep. In other words, they are not acting to establish a protocol during the episode. Then, in the final time step, the teacher observes o_f and sends the message $p(o_f)$. By construction, this results in a high TM error as there is no correct message from the protocol establishment phase.

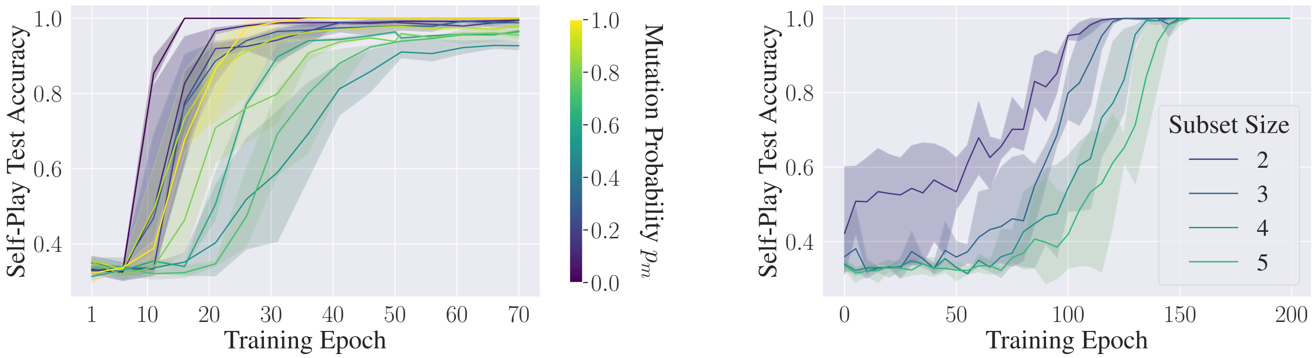
Next, suppose that after receiving this message, the student makes the correct guess, and thus the team achieves a high final performance. If the student were acting according to the messages sent in the protocol establishment phase, the student would assume uniform probabilities for each of the possible guesses. Therefore, the student receives a high SIC error. Additionally, because all of the messages sent during the protocol establishment phase are the same, the PD error is also high.

On the other hand, all these error metrics can be low, and the performance metric can be high, and yet there still to be a fixed protocol p in play. For example, if the teacher and student act in the same manner regardless of the timestep: the teacher always sends $m_t = p(o_t)$, and the student always makes the prediction $P(p^{-1}(m_t)) = 1$.

This tells us that, in the case where two agents have trained with one another, the error metrics and performance metrics are not sufficient to detect whether or not the teacher and/or the student are positively listening to the protocol being expressed within the episode, i.e. whether *intra-episodic protocol establishment* is happening. To measure this we need to look at the agents' responses to counterfactual protocols. This leads to two new *protocol responsiveness* measures, one for each of the roles in the environment:

Definition 9 (Student Responsiveness)**.** *To measure the student responsiveness R_S, we put an agent in the role of the student and have it play with a* ***synthetic teacher*** *that generates and uses a random protocol. More precisely, in each episode, the synthetic teacher samples a random injective function p from observations $\mathcal{O}_E$ to messages Σ, and always Thereby, the student's behaviour is isolated from the teacher's performance. We then measure the* ***Student-Implied Class*** *(SIC) error to test whether or not the agent correctly reacts to the random protocol. This value is then mapped to the unit interval such that high R_S implies low SIC error, and vice versa. Formally, R_S is computed:*

$$R_S = \exp\Big(-\overline{\mathcal{L}_{SIC}}^*\Big) \tag{9}$$



(a) Message mutation experiments.

(b) Channel permutation experiments.

Figure 2: Test performance during training for channel randomisation experiments. Error bands show 95% confidence intervals.

The additional notation over the error metric indicates that we are measuring the mean of $\mathcal{L}_{SIC}$ when playing with the synthetic teacher across multiple games.

Definition 10 (Teacher Responsiveness)**.** *To measure the teacher responsiveness R_T, we put an agent in the role of the teacher for only the last time step of an episode. For the prior time steps, we use the same synthetic teacher as in Definition 9, thus we are now measuring the teacher's capacity to the protocol established during the episode. We measure the* ***Teacher-Message*** *(TM) error, and this value is then mapped to the unit interval such that high R_T implies low TM error, and vice versa. Formally, R_T is computed:*

$$R_T = \exp\left(-\overline{\mathcal{L}_{TM}}^*\right) \tag{10}$$

The additional notation over the error metric indicates that we are measuring the mean of $\mathcal{L}_{TM}$ when playing with the synthetic teacher across multiple games.

To further understand what these measures tell us, recall that *positive listening* (Definition 2) measures an agent's sensitivity to the messages that they receive. For the cases of teacher/student responsiveness, instead of asking whether or not an agent is listening to a particular message, we are asking if an agent is listening to the protocol expressed within the episode.

Definition 11 (Protocol Diversity)**.** *This is a more interpretable variation on the protocol diversity error metric and is only ever measured in an environment without message mutation, as message mutation can obscure seeing whether or not an agent has learned the ability to create an injective protocol. We compute this measure $P_D = \mathcal{L}_{PD}^{-1}$. The closer P_D is to zero, the worse, the closer it is to one, the better.*

5 Experimental Results

For each of our experiments, we use an RMSprop optimiser with a learning rate of 0.01, a decay factor of 0.9, and a batch size of 32. Each agent's policy was an RNN composed of three layers: a dense layer with 128 hidden units, an LSTM layer with 64 units, and a final dense layer with $|\mathcal{O}_E| + |\Sigma|$ units. For our experiments, we set $|\mathcal{O}_E| = 3$ and $|\Sigma| = 5$. Each training epoch consists of 50 training steps. Figure 2 shows how the self-play performance improves throughout training for each of the different experimental settings that we discuss in this section.

5.1 Baseline

As a baseline, we trained 6 agents with the loss function $\mathcal{L}_{AC}$. Each agent trained to minimal loss and achieved perfect self-play performance. After 30 stranger encounters we computed a zero-shot cooperative performance of 0.39 ± 0.32. This is close to the expected value from sampling answers from a uniform distribution, so we can conclude that these agents have not learned any capacity for zero-shot communication. Additionally, as expected, we find that a fixed protocol was established by each agent in self-play.

5.2 Effects of Message Mutation

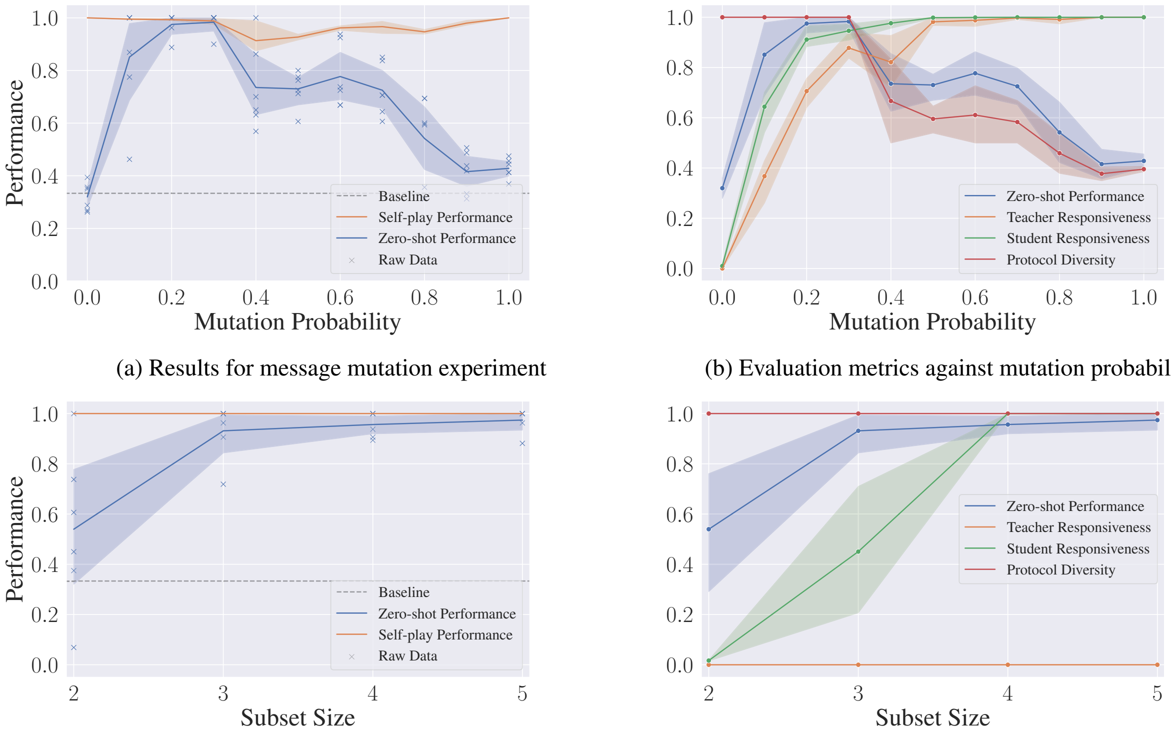
Message mutation intervenes on the protocol establishment phase, therefore to be effective it requires that the agents be sensitive to the protocol expressed in the first $|\mathcal{O}_E|$ timesteps. However, when optimising $\mathcal{L}_{AC}$, the utilisation phase of the environment is the only time in which the agents' behaviours matter. To fix this, we instead optimise a new loss function $\mathcal{L}_{MM}$ that judges each agent's behaviour according to the protocol formed in the protocol establishment phase:

$$\mathcal{L}_{MM} = \mathcal{L}_{SIC} + \mathcal{L}_{TM} + \mathcal{L}_{PD} \tag{11}$$

Thus, the teacher is required to create a consistent protocol (by $\mathcal{L}_{PD}$) and send the message in the utilisation phase corresponding to the respective observation in the establishment phase. On the other hand, the student is incentivised to make the appropriate guess given the answer implied by the protocol and the final message, which may not be the same as the ground-truth correct answer.

To investigate the effects of message mutation we trained three pairs of agents for 11 different evenly spaced values of the mutation probability p_m in the unit interval. For each set of three agents, we formed 6 stranger encounters and measured zero-shot performance. In Figure 3a we visualise the zero-shot cooperative performance (in blue). During training each agent takes both roles in the game with the mutation probability indicated on the x-axis, but during the zero-shot evaluation, the mutation probability is set to zero.

In order to make a fair comparison between the performance of agents trained with and without message mutation, we use the same evaluation environment without message mutation. In short, during self-play training, there are 11 different levels of mutation probability, but during the zero-shot evaluation, there is no randomisation in the communication channel.



(a) Results for message mutation experiment

(b) Evaluation metrics against mutation probability

(c) Results for channel permutation experiment.

(d) Evaluation metrics against permutation subset size.

Figure 3: Analysis of channel randomisation experiments. Error bands denote 95% confidence intervals of mean estimates.

We can see a clear relationship between mutation probability and zero-shot performance, where the performance peaks at $p_m = 0.3$ (ZSP $= 0.98 \pm 0.04$). When the mutation probability is 0, so when there is no randomness involved, the zero-shot performance is the poorest. This is as expected and verifies that the combination of loss functions is not enough on its own to encourage agents to learn the necessary skills for zero-shot communication, i.e. positive listening and signalling of the protocol itself.

Naively, one may assume that as the randomness in the environment increases, the performance increases, as the agents are forcefully exposed to more protocols. However, when the randomness is very large during training, the teacher may never learn to construct a protocol. This is reflected in the fact that when $p_m = 1.0$, the protocol diversity measure P_D, is low (Table 1). Recall, that when measuring this metric we evaluate the agent in an environment without any channel randomisation, so, unsurprisingly, the teacher has not learned skills for this domain. In other words, the teacher does not have to learn to create diverse protocols when mutation probability is 1, because the environment will ensure an injective protocol.

We also look at the performance the agents had during self-play training, which is visualised in orange in Figure 3a. We see that the performance during training is consistently high, in particular, for mutation probabilities 0 and 1. Lastly, we visualise a baseline, described in Section 5.1. In Figure 3b we see the zero-shot cooperative performance moving in concert with the protocol responsiveness measures and the protocol diversity measure. The P_D starts high and remains high until around $p_m = 0.3$, after which it starts to drop. The responsiveness starts low and monotonically increases with mutation probability. The ZSP peaks at the point where P_D, R_S, and R_T are all high. This supports the argument that intra-episodic protocol establishment drives zero-shot communication.

	R_T	R_S	P_D	ZSP
Baseline	0	0	1	0.39
Permutation ($k = 5$)	0.00	1.00	1.00	0.96
Mutation ($p_m = 0.3$)	0.85	0.97	1.00	0.98
Mutation ($p_m = 1.0$)	1.00	1.00	0.38	0.49

Table 1: Mean metrics from different experiments

5.3 Effects of Channel Permutation

We trained agents with channel permutation by using the $\mathcal{L}_{AC}$ loss function. To get the agents to reliably converge we found that we needed to use a temperature annealing schedule on the communication channel [18]. We used an exponential decay schedule where the temperature starts at 10, updates once an epoch, and ends at 0.1 at epoch 200. After which it stays constant at this value. After training 6 agents with permutation over all symbols, we ran 30 stranger encounters and found a mean zero-shot test performance of 0.96±0.05.

In Figure 3c we visualise the zero-shot cooperative performance (in blue) and self-play performance (in orange) across various subset sizes. We do not show results for subset size zero or one as these are functionally equivalent to no channel randomisation and are thus represented by the baseline. On the left, at subset size 2, we find that there is some improvement, but a very high variance. As we move to the right we see that the variance decreases and the performance approaches perfect play. But this does not come without any cost; we find that as the subset size grows so does the number of training steps needed for the system to converge, as shown in Figure 2b. Fortunately, this growth is not too dramatic; it takes roughly 130 epochs to converge with $k = 4$ and 150 epochs for $k = 5$. Finally, in Figure 3d we see that student responsiveness goes up as the permutation subset size increases and that the protocol diversity is high for any level of permutation. However, we also see that teacher responsiveness remains at zero, meaning that the teacher does not send the correct final message, given a random protocol.

5.4 Comparing the Methods

As we can see from comparing 3a and 3c, both channel randomisation methods can dramatically improve the zero-shot cooperative performance. While both methods can be tuned by a hyperparameter – mutation probability vs. subset size – we see in the case of message mutation that there is a balancing act that needs to be performed between exposure to new protocols and inhibiting the teacher from learning how to develop good protocols. Our finding that the optimum mutation probability is around 0.3 cannot be assumed to hold in other environments. On the other hand, increasing the channel permutation subset size does not introduce any similar trade-off. It also has the advantage that it does not require specific training signals that explicitly reference the adherence to the protocol expressed within the episode, i.e. it does not require manual identification of the protocol establishment phase.

However, channel permutation is not without its disadvantages. Firstly, Figure 2 shows that training times were significantly longer for channel permutation. Additionally, more hyperparameter tweaking was necessary to get the system to reliably converge – although we did not systematically explore different hyperparameters (learning rate, temperature annealing schedules, etc.). Finally, under channel permutation, teachers do not learn to pay attention to the protocol that they communicate, although this skill was not strictly necessary for zero-shot communication in our environment. This is shown in Table 1 where we see that the teacher protocol responsiveness, R_T, is zero for channel permutation, compared to 0.85 for message mutation ($p_m = 0.3$).

5.5 Talking About Time

We observed that when training agents with the loss function $\mathcal{L}_{MM}$ (and with no channel randomisation, i.e. $p_m = 0$) there was convergence on fixed protocols, as would be expected. However, we also found that under certain conditions we could get the agents to consistently converge on *temporally-fixed* (TF) protocols rather than *observation-fixed* (OF) protocols. By this, we mean that the teacher would always send the same ordered sequence of messages within each episode, regardless of the ordering of the observations, and they would adapt their final message appropriately. Intuitively, rather than messages corresponding to different observations, they would correspond to different time steps.

Figure 4 visualises a TF protocol as a heat map where each cell shows the proportion of the time that a particular class or timestep index coincides with each symbol, e.g. we can see that symbol 4 can be sent alongside any class but is only ever sent on the first timestep.

Whether an OF or TF protocol was learned depended on a single part of the agent's architecture. Namely, whether the first layer of the RNN, before the LSTM cell, had a ReLU activation or no activation. This seems to be in line with other work that has found that temporal referencing in emergent communication is sensitive to agent architectures [24]. Nonetheless, we found that regardless of which strategy emerged in the absence of channel randomisation, when channel randomisation is applied there is convergence on protocols with observations as the subject. Furthermore, both the OF and TF agents were unable to communicate with strangers when no channel randomisation was applied.

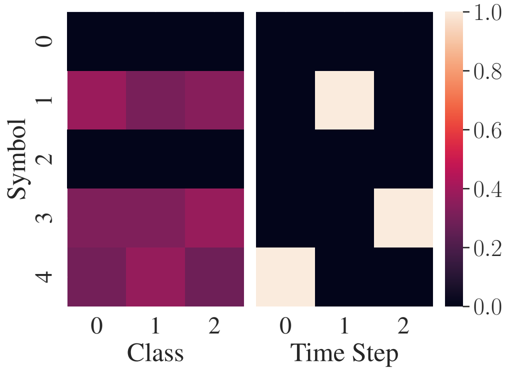


Figure 4: A visualisation of a temporally-fixed protocol

6 Discussion

This paper has explored the problem of *zero-shot communication*, where independently trained agents must cooperate via dedicated communication channels. We have presented two *channel randomisation* methods that facilitate zero-shot communication. The first of these methods, *message mutation*, is easier to train but is sensitive to the mutation probability hyperparameters, and requires intervening with the loss function and the communication channel. The second approach, *channel permutation*, is simpler to apply, but harder to optimise. We introduced a simple environment to test these methods in which agents may establish new shared communicative conventions within an episode. Furthermore, the simplicity of this environment allowed us to precisely measure and analyse the behaviours learned under different training hyperparameters.

Further work should assess the scalability of these proposals, they could be transported to more complex domains. There are several aspects of the set-up presented in this work that could be subjected to further empirical scrutiny. For example, to keep training runs short and stable, the number of possible environment observations and messages were kept relatively low ($|\Sigma| = 5$, $|\mathcal{O}_E| = 3$). Finally, in our environment, agents only interact with one another via the communication channel so situations where this is not the case should also be explored.

Acknowledgements

Dylan Cope and Nandi Schoots are supported by the UKRI Centre for Doctoral Training in Safe and Trusted AI (EPSRC Project EP/S023356/1). Corresponding Author: dylan.cope@kcl.ac.uk

References

[1] S. Barrett, N. Agmon, N. Hazon, S. Kraus, and P. Stone. Communicating with unknown teammates. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, AAMAS '14, pages 1433–1434, Richland, SC, May 2014. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-2738-1.

[2] K. Bullard, F. Meier, D. Kiela, J. Pineau, and J. Foerster. Exploring Zero-Shot Emergent Communication in Embodied Multi-Agent Populations, Dec. 2020. URL http://arxiv.org/abs/2010.15896. arXiv:2010.15896 [cs].

[3] K. Bullard, D. Kiela, F. Meier, J. Pineau, and J. Foerster. Quasi-Equivalence Discovery for Zero-Shot Emergent Communication, June 2021. URL http://arxiv.org/abs/2103.08067. arXiv:2103.08067 [cs].

[4] R. Chaabouni, E. Kharitonov, E. Dupoux, and M. Baroni. Anti-efficient encoding in emergent communication. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[5] E. Choi, A. Lazaridou, and N. de Freitas. Compositional obverter communication learning from raw visual input. In *6th International Conference on Learning Representations, Vancouver, Canada*, 2018.

[6] D. Cope and P. McBurney. Learning Translations: Emergent Communication Pretraining for Cooperative Language Acquisition. volume 1, pages 40–48, Aug. 2024. doi: 10.24963/ijcai.2024/5. URL https://www.ijcai.org/proceedings/2024/5.

[7] G. Dagan, D. Hupkes, and E. Bruni. Co-evolution of language and agents in referential games. In P. Merlo, J. Tiedemann, and R. Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2993–3004. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.260.

[8] Y. N. Dauphin, G. Tür, D. Hakkani-Tür, and L. P. Heck. Zero-shot learning and clustering for semantic utterance classification. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[9] K. Evtimova, A. Drozdov, D. Kiela, and K. Cho. Emergent communication in a multi-modal, multi-step referential game. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[10] J. Foerster, I. A. Assael, N. d. Freitas, and S. Whiteson. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In D. D. Lee and M. Sugiyama and U. V. Luxburg and I. Guyon and R. Garnett, editor, *Advances in Neural Information Processing Systems 29*, pages 2137–2145. Curran Associates, Inc., 2016.

[11] C. V. Goldman and S. Zilberstein. Decentralized control of cooperative systems: categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22(1):143–174, Nov. 2004. ISSN 1076-9757.

[12] C. V. Goldman and S. Zilberstein. Communication-Based Decomposition Mechanisms for Decentralized MDPs. *Journal of Artificial Intelligence Research*, 32:169–202, May 2008. ISSN 1076-9757. doi: 10.1613/jair.2466. URL http://arxiv.org/abs/1111.0065. arXiv:1111.0065 [cs].

[13] S. Havrylov and I. Titov. Emergence of Language with Multi-agent Games: Learning to Communicate with Sequences of Symbols. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30. Curran Associates, Inc., 2017.

[14] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[15] H. Hu, A. Lerer, A. Peysakhovich, and J. Foerster. “Other-Play” for Zero-Shot Coordination. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4399–4410. PMLR, Nov. 2020. ISSN: 2640-3498.

[16] H. Hu, A. Lerer, A. Peysakhovich, and J. N. Foerster. "other-play" for zero-shot coordination. *CoRR*, abs/2003.02979, 2020. URL https://arxiv.org/abs/2003.02979.

[17] H. Hu, A. Lerer, B. Cui, D. Wu, L. Pineda, N. Brown, and J. Foerster. Off-Belief Learning. In *Proceedings of the 38th International Conference on Machine Learning*, PMLR 139, Mar. 2021. arXiv: 2103.04000.

[18] E. Jang, S. Gu, and B. Poole. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations*, 11 2017.

[19] A. Lazaridou and M. Baroni. Emergent Multi-Agent Communication in the Deep Learning Era, July 2020. URL http://arxiv.org/abs/2006.02419. arXiv:2006.02419 [cs].

[20] A. Lazaridou, K. M. Hermann, K. Tuyls, and S. Clark. Emergence of Linguistic Communication from Referential Games with Symbolic and Pixel Input. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, BC, Canada, 2018.

[21] J. Lee, K. Cho, J. Weston, and D. Kiela. Emergent Translation in Multi-Agent Communication. In *The International Conference on Learning Representations (ICLR)*, 2018.

[22] D. K. Lewis. *Convention: A Philosophical Study*. Wiley-Blackwell, Cambridge, USA, 1969. doi: 10.2307/2218418. ISSN: 00318094.

[23] Y. Li, S. Zhang, J. Sun, Y. Du, Y. Wen, X. Wang, and W. Pan. Cooperative Open-ended Learning Framework for Zero-Shot Coordination. In *Proceedings of the 40th International Conference on Machine Learning*, pages 20470–20484. PMLR, 2023. ISSN: 2640-3498.

[24] O. Lipinski, A. J. Sobey, F. Cerutti, and T. J. Norman. On Temporal References in Emergent Communication, Oct. 2023. URL http://arxiv.org/abs/2310.06555. arXiv:2310.06555 [cs].

[25] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, page 163. Elsevier, 1994. doi: 10.1016/b978-1-55860-335-6.50027-1.

[26] R. Lowe, J. Foerster, Y.-L. Boureau, J. Pineau, and Y. Dauphin. On the Pitfalls of Measuring Emergent Communication. *IFAAMAS*, 9, 2019.

[27] W. Macke, R. Mirsky, and P. Stone. Expected Value of Communication for Planning in Ad Hoc Teamwork. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11290–11298, May 2021. ISSN 2374-3468. doi: 10.1609/aaai.v35i13.17346. URL https://ojs.aaai.org/index.php/AAAI/article/view/17346. Number: 13.

[28] C. J. Maddison, A. Mnih, and Y. W. Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *5th International Conference on Learning Representations*, 2016.

[29] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull. Active domain randomization. In L. P. Kaelbling, D. Kragic, and K. Sugiura, editors, *3rd Annual Conference on Robot Learning, Osaka, Japan*, volume 100 of *Proceedings of Machine Learning Research*, pages 1162–1176. PMLR, 2019.

[30] R. Mirsky, W. Macke, A. Wang, H. Yedidsion, and P. Stone. A Penny for Your Thoughts: The Value of Communication in Ad Hoc Teamwork. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 254–260, Yokohama, Japan, July 2020. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-6-5. doi: 10.24963/ijcai.2020/36. URL https://www.ijcai.org/proceedings/2020/36.

[31] I. Mordatch and P. Abbeel. Emergence of grounded compositional language in multi-agent populations. In S. A. McIlraith and K. Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1495–1502. AAAI Press, 2018.

[32] I. Mordatch and P. Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI’18/IAAI’18/EAAI’18, pages 1495–1502, New Orleans, Louisiana, USA, Feb. 2018. AAAI Press. ISBN 978-1-57735-800-8.

[33] F. A. Oliehoek and C. Amato. *A Concise Introduction to Decentralized POMDPs*. Springer International Publishing, Cham, 2016. ISBN 978-3-319-28927-4. doi: 10.1007/978-3-319-28929-8. Series Title: SpringerBriefs in Intelligent Systems.

[34] T. Sarratt and A. Jhala. The Role of Models and Communication in the Ad Hoc Multiagent Team Decision Problem. In *Proceedings of the Third Annual Conference on Advances in Cognitive Systems*. Cognitive Systems Foundation, 2015.

[35] N. Schoots and D. Cope. Low-Entropy Latent Variables Hurt Out-of-Distribution Performance. In *The Domain Generalization Workshop at ICLR 2023*, 2023.

[36] P. Stone, G. A. Kaminka, S. Kraus, and J. S. Rosenschein. Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, 2010.

[37] G. Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6(2):215–219, 1994.

[38] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.

[39] K. Wagner, J. A. Reggia, J. Uriagereka, and G. S. Wilkinson. Progress in the Simulation of Emergent Communication and Language. *Adaptive Behavior*, 11(1):37–69, 2003. ISSN 1059-7123. doi: 10.1177/10597123030111003.

[40] L. Yu, Y. Qiu, Q. Yao, X. Zhang, and J. Wang. Improving zero-shot coordination performance based on policy similarity. In *Proceedings of the Thirty-Third International Conference on Automated Planning and Scheduling*, volume 33 of *ICAPS ’23*, pages 438–442, Prague, Czech Republic, 2023. AAAI Press. ISBN 978-1-57735-881-7.

[41] C. Zhu, M. Dastani, and S. Wang. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, 38(1):4, Jan. 2024. ISSN 1573-7454. doi: 10.1007/s10458-023-09633-6.